

INSTITUT FÜR INFORMATIK
DER
LUDWIG-MAXIMILIANS-UNIVERSITÄT
MÜNCHEN



DIplomarbeit

Sascha Peschke

Entwicklung eines Intranets zur wissensbasierten
Unternehmenskommunikation

Aufgabensteller:	Prof. Dr. Martin Wirsing
Betreuer:	Hubert Baumeister Christian Bering, Medialab GmbH
Abgabedatum:	16. August 1999

Erklärung

Hiermit versichere ich, daß ich diese Diplomarbeit selbstständig verfaßt und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Sascha Peschke

München, den 16. August 1999

Vielen Dank

an Herrn Professor Wirsing für die Annahme des Themas.

an meinen Betreuer Hubert Baumeister für die hervorragende Betreuung, für die Hilfestellung und Tips bei der Implementierung und dafür daß er immer Zeit für mich opferte.

an alle Mitarbeiter der Firma Medialab Informationsdesign GmbH, die sich für meine Fragen offen gezeigt haben, insbesondere die Herren von der Produktion und das Bavaria-Team.

für meine liebe Oma, Mom und Dad

*und alle,
die mich immer unterstützt haben
und an mich glauben*

Inhaltsverzeichnis

1	EINFÜHRUNG	1
2	STATE OF THE ART	6
2.1	Lotus Notes / Domino	6
2.2	Hyper-G	7
2.3	Wiki Wiki Web	7
2.4	Vergleich zu diesem System	8
3	OOSE , OOHDM UND UML	10
3.1	Überblick	10
3.2	Vorgehensmodelle	11
3.3	Methoden	14
3.4	OOSE	14
3.5	OOHDM und UML	16
4	ANALYSE	19
4.1	Beschreibung des Ist-Zustands	20
4.1.1	Unternehmensprofil	20
4.1.2	Software- und Hardwareausstattung	21
4.2	Anforderungsanalyse	22
4.2.1	Allgemeine Anforderungen	22
4.2.2	Inhaltliche und funktionelle Anforderungen	23
4.2.3	Sicherheitsaspekte	27

4.3	Anwendungsfälle	28
4.3.1	Mitarbeiter	29
4.3.2	Telefonliste	33
4.3.3	Kategorien	37
4.3.4	Information	41
4.3.5	Speisekarte	45
4.3.6	Essenbestellung	50
5	ENTWURF (DESIGN)	52
5.1	Konzeptioneller Entwurf (Conceptual Design)	52
5.2	Entwurf der Navigation (Navigational Design)	56
5.2.1	Navigationsklassenmodell	57
5.2.2	Navigationsstrukturmodell (Navigational Structure Model)	59
5.3	Entwurf der Präsentation (Presentational Design)	68
5.3.1	Statisches Präsentationsmodell	68
5.3.2	Dynamisches Präsentationsmodell	69
5.3.3	Beschreibung des entwickelten Präsentationsmodells	69
6	WEBBASIERTE INFORMATIONSSYSTEME	78
6.1	Überblick	78
6.2	Bestandteile eines webbasierten Informationssystems	81
6.2.1	Webserver und Webbrowser	81
6.2.2	Webapplikation	82
6.2.3	Datenbank	83
6.3	Vergleich möglicher Lösungen	84
6.3.1	Lotus Notes / Domino	84
6.3.2	Jasmine	85
6.3.3	Gemstone	86
6.3.4	MySQL	87
6.3.5	Zusammenfassung und Bewertung	87
7	IMPLEMENTIERUNG	89

7.1	Einführung	89
7.2	Implementierung des konzeptionellen Designs	90
7.2.1	Datenbankschema	90
7.2.2	Applikation	95
7.3	Implementierung des Navigationsdesigns	96
7.4	Implementierung des Präsentationsdesigns	97
7.5	Implementierungstechnologien	98
7.5.1	Servlets	98
7.5.2	Eingesetzte Komponenten	99
7.5.3	Aufbau der Applikation	100
7.5.4	Erweiterbarkeit	102
8	BENUTZERHANDBUCH	103
8.1	Allgemeine Bedienhinweise	103
8.2	Starten der Applikation	104
8.3	Hompage	104
8.4	Artikel anzeigen	105
8.5	Artikel anlegen und bearbeiten	106
8.6	Übersicht der Artikel in einer Kategorie	107
8.7	Kategorien anlegen und bearbeiten	107
8.8	Kategorie löschen	109
8.9	Suche	110
8.10	Verwaltung	111
8.11	Telefonliste	113
9	BEWERTUNG UND AUSBLICK	115

1 Einführung

Informationen sind in wissensintensiven Dienstleistungs-, Software- und Medienunternehmen der zentrale Produktionsfaktor. Dabei handelt es sich bei Information zum Beispiel um Know-How, das für die Erstellung neuer Produkte oder das Anbieten neuer Dienstleistungen benötigt wird, oder auch das Wissen, das alltäglich benutzt wird. Der Aufbau von Know-How ist ein langwieriger und deshalb kostenintensiver Prozeß. Darum muß dieses Wissen und dieses Know-How gehütet werden. Es reicht aber nicht allein, Informationen irgendwo aufzubewahren, sondern das Wissen muß den Mitarbeitern auch stets zur Verfügung stehen. Das bedeutet auch, daß sie davon Kenntnis erlangen, welche Informationen ihnen bereitstehen, so daß das Rad nicht zweimal erfunden wird. Damit die Mitarbeiter eines Unternehmens von den ihnen zur Verfügung stehenden Informationen profitieren können, müssen diese Informationen kommuniziert werden. Die unternehmensinterne Kommunikation des Know-Hows muß also gefördert werden. Durch die zunehmende Spezialisierung in Unternehmen leidet jedoch die Kommunikation. Gerade der interdisziplinäre Informationsaustausch kommt oft zu kurz.

Die Multimedia-Agentur Medialab Informationsdesign GmbH will dieses Problem durch den Aufbau eines Informationssystems lösen. Im Rahmen dieser Diplomarbeit wurde in Zusammenarbeit mit dem Lehrstuhl Programmierung und Softwaretechnik ein Konzept dafür erarbeitet und eine prototypische Implementierung erstellt.

Ziel der Diplomarbeit war es, eine Applikation zu erstellen, die die Informationen effektiv verwaltet und den Mitarbeitern einfachen Zugriff darauf bietet. Dabei soll die Bedienung des Systems für die Mitarbeiter möglichst einfach und intuitiv sein. Die

Basis dafür stellt eine für alle Mitarbeiter einheitliche Repräsentation der Informationen dar. Da bei Medialab die verschiedensten Rechner- und Betriebssystemplattformen eingesetzt werden, sollte eine von diesen Systemen unabhängige Möglichkeit der Repräsentation erreicht werden. Andernfalls wäre eine Anpassung auf alle vorhandenen Systeme notwendig.

Eine Lösung für das Problem der einheitlichen Repräsentation bietet das World-Wide-Web. Das World-Wide-Web (WWW) ist ein globales Hypertextsystem auf Basis des Internets. Die im WWW vorhandenen Hypertextdokumente sind untereinander mit Querverweisen, sogenannten Hyperlinks, vernetzt. Hypertextdokumente können von jeder Stelle des Dokuments auf eine beliebige Stelle eines anderen Dokuments verweisen. Es können außer Text auch andere Inhalte wie Bild oder Ton präsentiert werden. Neben der Möglichkeit, statische Dokumente anzubieten, können auch dynamische Dokumente erstellt werden, deren Erscheinungsbild und Inhalt durch Benutzeranfragen bestimmt wird. Unter anderem kann man den Benutzern auch die Möglichkeit geben, nicht nur bestimmte Inhalte abzurufen, sondern diese auch selbst zu verändern oder zu generieren. So kann die Aktualität der vorhandenen Informationen aufrecht erhalten werden. Wird die Informationspflege jedoch vernachlässigt, so wird die Qualität der Informationen auf Dauer durch die schwindende Aktualität beeinträchtigt.

Die in dieser Diplomarbeit erstellte Applikation unterstützt die Informationspflege und übernimmt das Informationsmanagement. So können alle Mitarbeiter der Unternehmung auf einfache Weise neue Informationen einpflegen und vorhandene Informationen aktualisieren oder löschen. Es sind somit die Mitarbeiter gefordert, selbst ihre Erfahrungen und ihr Wissen in das System einzubringen.

Dies geschieht in Form von Foren, in welchen die Mitarbeiter entweder selbst ihren Kollegen Tips und Tricks zu bestimmten Problemstellungen in Form von Text, Grafik, Audiodaten, Programmbeispielen usw. anbieten oder gezielt nach Informationen suchen können. Neue Mitarbeiter oder Praktikanten sollen sich mit dem System einen Überblick über das Unternehmen verschaffen und ein Basiswissen aneignen können, so

daß die Einarbeitung nicht zwingend die Einweisung durch Teamangehörige erfordert. Das System soll also die Benutzer bei ihrer Arbeit unterstützen.

Das System ist so entworfen, daß das Produkt den Anforderungen und Wünschen der Benutzer gerecht wird. Da sich diese Anforderungen im Laufe der Zeit wandeln, muß eine leichte Anpassungsfähigkeit berücksichtigt werden.

Um diesen Zielen gerecht zu werden, wurde bei der Entwicklung des Systems der Ansatz der objektorientierten Softwareentwicklung verfolgt. Bei der objektorientierten Softwareentwicklung wird unter anderem besonderer Wert darauf gelegt, daß das erstellte Produkt mit geringem Aufwand an zusätzliche Bedürfnisse angepaßt werden kann. Die Entwicklung der Applikation gliedert sich in folgende Phasen: Analyse, Design, Implementierung und Test.

In der Analysephase werden die genauen Anforderungen an das System aus Sicht des Auftraggebers ermittelt. Aus diesen Anforderungen wird eine schriftliche Anforderungsanalyse erstellt. Ausgehend von der erarbeiteten Anforderungsanalyse werden die Anwendungsfälle (Use Cases) erstellt. Anwendungsfälle identifizieren die Akteure des Systems und die benötigten Funktionsmodule. Die Anforderungen wurden bei dieser Diplomarbeit in Einzelgesprächen und Gruppendiskussionen mit den Mitarbeitern von Medialab ermittelt.

Die Ergebnisse aus der Analysephase werden in der Designphase weiter verwendet und unter Bezugnahme von Implementierungsaspekten verfeinert. Eine Anforderung war es, eine webbasierte Applikation zu entwickeln. Daher wurde beim Design eine spezielle Methodik, die OOHDM (Object Oriented Hypermedia Design Method) angewandt. Dabei wird der Designprozess in verschiedene Stadien unterteilt. Beim konzeptionellen Design (Conceptual Design) werden aus den Anwendungsfällen Klassendiagramme erstellt. Dann wird das Navigationsdesign (Navigational Design) entwickelt und schließlich eine Benutzerschnittstelle (Presentational Design) erstellt. Bei der Entwicklung der in diesen Phasen entwickelten Modelle kam die Unified Modelling Language (UML) mit speziellen UML-konformen Erweiterungen zum Einsatz.

Die drei Phasen wurden nicht nur sequentiell durchlaufen, da sich die Phasen gegenseitig sehr stark beeinflussen. Während des Navigationsdesigns und des abstrakten Schnittstellendesigns traten noch kleinere Probleme auf, so daß die erstellten Modelle wieder angepaßt werden mußten.

Auf die Designphase folgt die Implementierungsphase. Es wird das Designmodell in einer konkreten Programmiersprache implementiert. Hier war es jedoch zuerst nötig, sich für bestimmte Technologien zu entscheiden, mit denen der Entwurf umgesetzt werden soll. Für ein webbasierendes Informationssystem, wie es bei Medialab entstehen soll, mußte in puncto Datenbank, Webserver und Applikationsschicht eine Entscheidung gefällt werden.

Im allgemeinen gehören diese Aspekte zu den Anforderungen, im vorliegenden Fall mußte jedoch zwischen den Ansprüchen des Lehrstuhls und den finanziellen Vorstellungen des Unternehmens ein Kompromiß getroffen werden, so daß die Entscheidung hinausgezögert wurde. Als Datenbank wurde die frei verfügbare relationale Datenbank MySQL ausgewählt. Als Webserver dient der ebenfalls frei verfügbare Apache Webserver. Die Applikation stellt die Schnittstelle zwischen Webserver und Datenbank, sowie zwischen Webserver und Benutzer dar. Da auch bei der Implementierung objektorientierte Techniken eingesetzt werden sollten, wurde Java als Programmiersprache ausgewählt. Java eignet sich sowohl für die Erstellung von Webapplikationen wie auch für den Zugriff auf Datenbanken und ist eine objektorientierte Sprache.

Aufbau der Arbeit

Die vorliegende Diplomarbeit gliedert sich in neun Abschnitte. In Abschnitt 2 werden aktuelle Lösungen für die bestehenden Anforderungen beschrieben und zum erstellten System abgegrenzt. In Abschnitt 3 behandelt in Kürze die in dieser Diplomarbeit verwendeten Methoden und die Notation. Im nachfolgenden Abschnitt 4 wird auf die Resultate der Analyse und im Abschnitt 5 auf das Design eingegangen. Nach diesem Abschnitt folgt in Abschnitt 6 noch eine allgemeine Beschreibung webbasierter Informationssysteme. Abschnitt 7 ist der Beschreibung der Implementierung gewidmet,

gefolgt von einer kurzen Anleitung für den Benutzer der Applikation in Abschnitt 8. Zum Schluß werden in der Zusammenfassung in Abschnitt 9 die Resultate beschrieben, sowie eine persönliche Bewertung und Ausblick auf Erweiterungen und andere Einsatzgebiete der Applikation gegeben.

2 State of the Art

Die Idee, ein Intranet als Basis eines Informationssystems zu nutzen, ist per se nicht neu. Auf dem Markt gibt es dafür einige Produkte, die diesen Zweck erfüllen.. In diesem Kapitel werden drei verschiedene Konzepte zur Realisierung eines webbasierten Informationssystems vorgestellt und zu dem in dieser Diplomarbeit erstellten System abgegrenzt.

2.1 Lotus Notes / Domino

Bei Lotus Notes / Domino (Release 5) handelt es sich um eine Weiterentwicklung eines klassischen Groupware-Systems. Die Software besteht aus einem Webserver, Lotus Domino und der Groupware Lotus Notes, der eine relationale Datenbank zu Grunde liegt. Beide Produkte werden von der Firma Lotus hergestellt. Lotus Notes stellt verschiedene Möglichkeiten zur Dokumentverwaltung zur Verfügung, die in Verbindung mit dem Domino Webserver ohne großen Aufwand im World Wide Web zur Verfügung gestellt werden können. Notes besitzt noch weitere Fähigkeiten, beispielsweise Mailedienste oder eine Terminverwaltung. Üblicherweise wird Lotus Notes deswegen auch als Groupware benutzt, um alle Fähigkeiten des Software nutzen zu können. In diesem Fall kommt am Arbeitsplatz des Benutzers ein Client zum Einsatz. Auch ohne des Clients ist die Nutzung von Zusatzdiensten wie Mail oder Terminverwaltung möglich. In diesem Fall können aber nicht alle Fähigkeiten genutzt werden, die Notes eigentlich zur Verfügung stellt.

2.2 Hyper-G

Bei Hyper-G handelt es sich um ein webbasiertes Informationssystem. Hyper-G wurde an der Universität Graz in Österreich entwickelt. Es verfügt über eine integrierte Datenbank und eingebettete Multimedia-Unterstützung. Außerdem bietet es die Möglichkeit Benutzer zu verwalten und Rechte zu vergeben.

Die in das System eingebettete Datenbank ist ein zentraler Bestandteil. Sämtliche Dokumente werden in dieser Datenbank gehalten. Das System erlaubt die Speicherung von Dokumenttyp, Autor, Erstellungsdatum, Titel usw. und unterstützt Volltextsuche. Zudem besteht die Möglichkeit, die Dokumente nach eigenen Kriterien zu strukturieren.

Negativ ist zu bemerken, daß die Webstandards nicht vollständig eingehalten werden. So basieren die Hypertext-Fähigkeiten des Systems auf dem Hyper-G Text Format (HTF), das nur Ähnlichkeiten mit HTML aufweist. Aus diesem Grund ist auch ein eigener Hyper-G-Browser erforderlich, der allerdings HTML-Dokumente darstellen kann. Jedoch ist es nicht möglich, mit einem herkömmlichen Browser auf dieses Informationssystem zuzugreifen.[SchEi97]

2.3 Wiki Wiki Web

Das Wiki Wiki Web ist der Prototyp webbasierter öffentlicher Diskussionsforen. Die Diskussionsforen sind jedermann zugänglich und jedermann kann die Inhalte, die als HTML-Seiten repräsentiert werden, beliebig verändern. Mit speziellen SteuerCodes können sogar bestimmte HTML-Formatierungen vorgenommen werden, um die Form und Gestaltung des Textes zu beeinflussen. Es können auch jederzeit neue Artikel angelegt. Alle Artikel sind untereinander durch Hyperlinks vernetzt, wobei der Titel des Artikels gleichzeitig auch die Bezeichnung des Links zu dieser Seite ist. Dies entspricht dem Prinzip aus dem das WWW ursprünglich einmal entstanden ist, nämlich dem Verketteten von Informationen über Hyperlinks.

Alle Daten werden in einer einzigen Textdatei gespeichert, auf die über eine Datenbank (UNIX DBM) zugegriffen wird. Die Seiten werden aus diesem Textfile anhand von

Schlüssel-Wert-Paaren erzeugt. Es gibt Schlüssel für den Inhalt der Seite, das Erstellungsdatum, die Versionsnummer und noch einige mehr. Die Bezeichnung „wiki wiki“ stammt eigentlich aus dem hawaiianischen und bedeutet „schnell“. Dies soll zum Ausdruck bringen, daß der zugriff auf die Daten sehr zügig von Statten geht, da für einen Zugriff maximal zwei Lesevorgänge benötigt werden. Das Erzeugen, Editieren oder Darstellen von Informationen wird über CGI-Skripte gesteuert. [Wiki99]

Schön am WikiWikiWeb ist, das die Informationen allesamt untereinander vernetzt sind. Allerdings gibt es keine Hierarchien, was die Übersichtlichkeit des Gesamtangebots an Informationen erheblich einschränkt. Zudem ist das System nur für den Einsatz als Diskussionsforum konzipiert. Eine Erweiterung der Funktionalität ist mit dem zu Grunde liegenden Design sehr aufwendig.

2.4 Vergleich zu diesem System

Das in dieser Diplomarbeit entwickelte System stellt den Austausch von Informationen und deren Strukturierung in den Mittelpunkt. Produkte wie Lotus Notes/Domino legen das Hauptaugenmerk auf den Groupwareaspekt. Auch mit Notes ist es sicherlich möglich ein Informationssystem einzurichten. Zusätzlich werden sogar noch weitere Fähigkeiten zur Verfügung gestellt. Allerdings besteht für diese Fähigkeiten gar kein Bedarf, so daß sie brach liegen. Unter diesem Aspekt ist Notes zu überdimensioniert. Dazu kommt, daß die Einführung von Notes neben den Kosten für das Produkt selbst auch noch erhebliche Kosten im Bereich Schulung nach sich zieht. Außerdem gibt es beim Erstellen einer webbasierten Applikation mit Notes Einschränkungen, was die Verwendung neuester Techniken für Webapplikationen. So gibt es bei der Einbindung von JavaScript erhebliche Probleme. JavaScript bietet jedoch Möglichkeiten, die die Dynamik der Applikation erhöhen und deren Bedienbarkeit erleichtern. Eine einfache Bedienung der Applikation ist für die Akzeptanz der Applikation aber von höchster Priorität.

Das WikiWikiWeb zielt als Diskussionsforum ebenfalls auf den Austausch von Informationen ab. Dazu werden alle Informationen untereinander vernetzt. Jedoch gibt

es hier starke Einschränkungen, was die Strukturierbarkeit der Informationen angeht. Mit der Möglichkeit Informationen bereitzustellen und untereinander zu vernetzen ist das Funktionsspektrum schon ausgeschöpft. Das Hinzufügen neuer Anwendungsgebiete ist nur mit erheblichem Aufwand zu erreichen. Gerade dies ist aber notwendig, um das System auch an bestehende und zukünftige Anforderungen anzupassen.

Hyper-G bietet besonders auf dem Gebiet der Dokumentverwaltung Vorteile. Ebenso wie das WikiWikiWeb ist jedoch sind Funktionen für andere Anwendungsgebiete nicht vorhanden. Außerdem verwendet Hyper-G keine Webstandards. Daher schafft man sich mit Hyper-G eine Insellösung. Die Möglichkeit des Zugriffs ist auf Hyper-G-Webbrowser eingeschränkt.

Die erstellte Applikation ist für das Unternehmen eine maßgeschneiderte Lösung, die sich an den Bedürfnissen des Auftraggebers orientiert. Durch offenes Design ist die Möglichkeit es zu erweitern gegeben. Die Applikation setzt genau da an, wo bei den anderen vorgestellten Lösungen Defizite bestehen. Daher ist die Applikation für die Summe der Aufgabengebiete, für den sie geschaffen wurde, am besten geeignet.

3 OOSE , OOHDM und UML

3.1 Überblick

Software-Engineering ist ein Fachgebiet der Informatik, das sich mit der Bereitstellung und systematischen Verwendung von Methoden und Werkzeugen für die Herstellung und Anwendung von Software beschäftigt.

Während ein Vorgehensmodell die Abfolge der einzelnen Entwicklungsphasen (Aktivitäten) bei der Entwicklung von Softwaresystemen beschreibt, ist eine Methodologie eine Vorschrift zur Durchführung dieser Entwicklungsphasen und zur Repräsentation entsprechender Ergebnisse unter Verwendung verschiedener vordefinierter Techniken und Notationskonventionen.

Das in dieser Diplomarbeit entwickelte Softwaresystem wurde nach den Grundsätzen der objektorientierten Softwareentwicklung erstellt. OOSE (Object Oriented Software Engineering) [Jac92] ist eine Methodologie zur Durchführung der in Vorgehensmodellen definierten Aktivitäten, wobei in allen Phasen der Entwicklung objektorientierte Techniken zum Einsatz kommen. In der Entwurfsphase kam die OOHDM-Methode zum Einsatz, um den speziellen Anforderungen einer Hypermedia-Anwendung gerecht zu werden. Die OOHDM (Object Oriented Hypermedia Design Method) stellt eine Methode zum Design von komplexen Hypermedia-Anwendungen dar und wurde von D. Schwabe und G. Rossi [RS95] entwickelt. Um während der gesamten Entwicklungsphase eine einheitliche Notation zu verwenden, wurden nicht die in OOSE bzw. OOHDM spezifizierten Notationen verwendet, sondern die Unified Modeling Language, der de facto Standard beim Entwurf objekt-orientierter Systeme.

Die Unified Modeling Language (UML) [UML97] wurde von Booch, Rumbaugh und Jacobsen bei der Firma Rational entwickelt und ist eine Notation zur Beschreibung von Softwaresystemen. Zum Einsatz kamen zudem UML-konforme Erweiterungen der Notation. Diese Erweiterungen von UML zur Verwendung als Notation für OOHDH wurde von Hubert Baumeister, Nora Koch und Luis Mandel [BKM99] entwickelt.

3.2 Vorgehensmodelle

Seit den 70er Jahren gibt es Vorgehensmodelle, die die Abfolge der Entwicklungsphasen bei der Softwareentwicklung beschreiben. Folgende Phasen werden unterschieden:

Anforderungsspezifikation und Analyse: In dieser Phase werden die spezifischen Produkt- und Entwicklungsanforderungen ermittelt. Ein Produkt der Analysephase ist der Anforderungskatalog. In älteren Modellen ist diese Phase noch in zwei Phasen unterteilt.

Entwurf (Design): Auf der Grundlage des Anforderungskatalogs wird in dieser Phase die innere Struktur des Softwaresystems, d.h. die Architektur und die Schnittstellen zwischen den Komponenten der Architektur, festgelegt.

Implementierung: Dies ist die Phase, in der die Komponenten des Entwurfs in einer konkreten Programmiersprache und auf einem konkretem System implementiert werden.

Test: Die in der Implementation entstandenen Programmteile werden in dieser Phase auf Vollständigkeit und Korrektheit überprüft.

Einsatz und Wartung: Zusammenfassung aller Aktivitäten, die im Praxiseinsatz eines Programms ausgeführt werden, wie Korrektur von Fehlern, Portierung, Änderungen und Erweiterungen der Funktionalität.

Man kann diese Aktivitäten in einem Wasserfall-Modell [siehe Abbildung 3-1] darstellen. Das Modell wird auch als sequentielles Phasenmodell bezeichnet. Die Idee dabei ist,

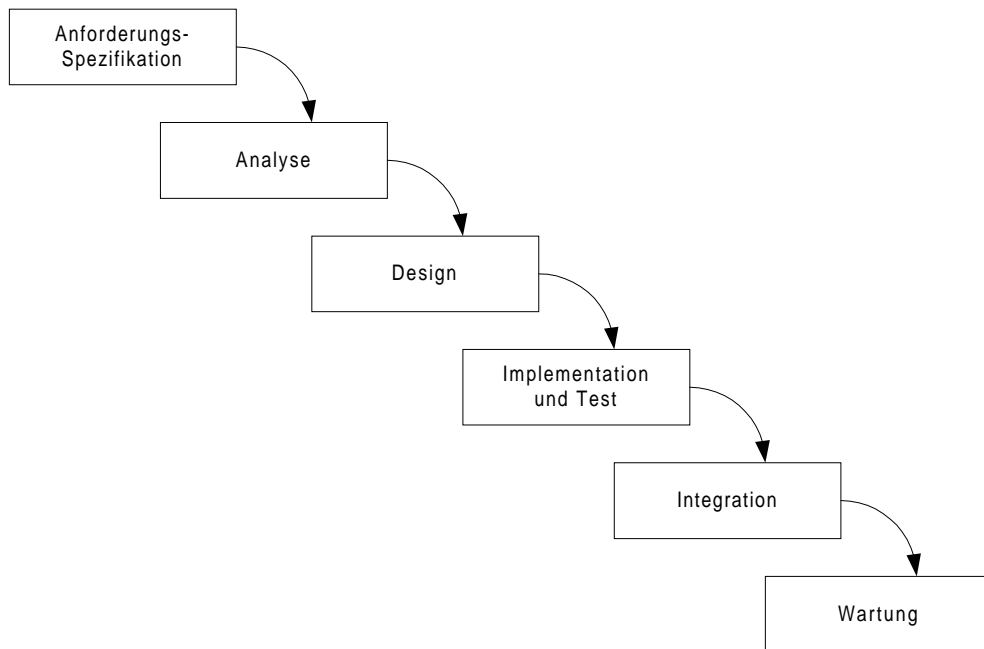


Abbildung 3-1 Wasserfallmodell

daß jede einzelne Phase abgeschlossen ist, bevor die darauffolgende Phase begonnen wird. Mit Abschluß der Testphase ist das Softwareprodukt fertig und voll einsatzfähig.

Ein anderer Ansatz ist das iterierte Phasenmodell, bei dem Rückkopplungen die Rückkehr von einer späten Phase in eine frühere Phase ermöglichen. Den Gegensatz zum sequentiellen Phasenmodell stellt die evolutionäre Softwareentwicklung (Prototyping) dar. Dieses Vorgehensmodell sieht Softwareentwicklung nicht mehr als linearen Prozeß, sondern als eine Folge von Entwicklungszyklen, an deren Ende jeweils eine verbesserte Version des Software-Produktes steht. Daraus ergibt sich ein Spiralmodell [siehe Abbildung 3-2]: Die durchzuführenden Aktivitäten werden spiralförmig durchlaufen, am Ende des Zyklus bewertet und neue Pläne für die nächste zu durchlaufende Windung verabschiedet.

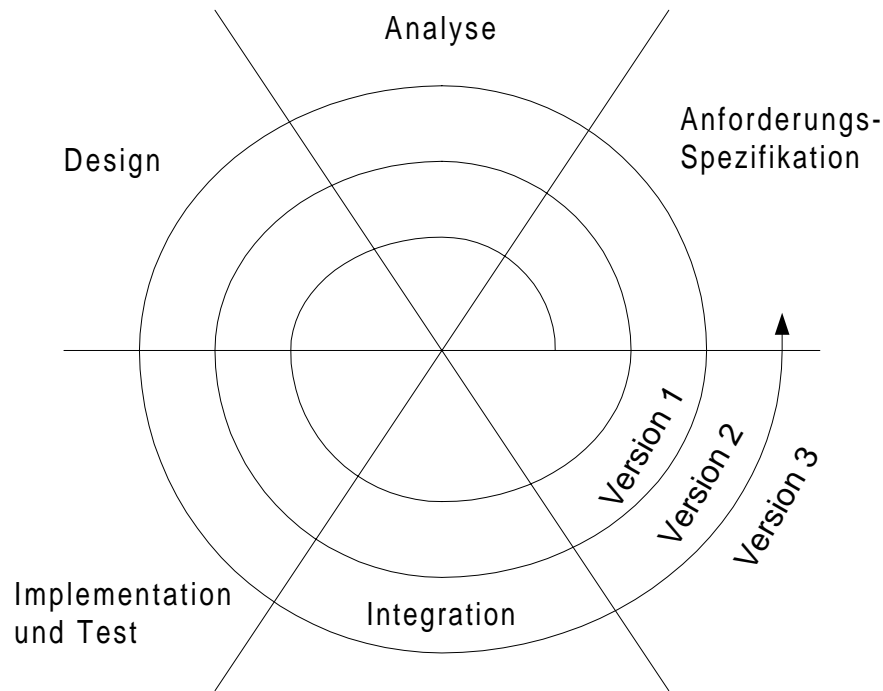


Abbildung 3-2

Spiralmodell

Als nachteilig beim Wasserfallmodell erweist sich das strikte Durchlaufen der einzelnen Entwicklungsphasen. In der Praxis stellt man oft fest, daß Funktionalitäten nicht exakt in der Form realisiert werden können, wie ursprünglich geplant. Dies zieht Änderungen der vorhergehenden Phasen nach sich. Beim Wasserfallmodell geht man davon aus, daß bei durchdachter Planung die Phasen sequentiell durchlaufen werden können und nachträgliche Änderungen nicht nötig sind. Das iterative Phasenmodell ist in dieser Hinsicht flexibler und berücksichtigt eventuell notwendige Änderungen und somit den Rücksprung in vorhergehende Phasen. Beim Spiralmodell geht man sogar davon aus, daß man vorausschauend nicht alles so planen kann, daß die so erstellte Software genau den Bedürfnissen des Auftraggebers und der Benutzer entspricht.

3.3 Methoden

Eine Methode ist eine Vorschrift zur Durchführung organisierter Produktion von Software unter Verwendung verschiedener vordefinierter Techniken und Notationskonventionen für bestimmte Aktivitäten. Objektorientierte Methoden gibt es seit etwa Mitte der 80er Jahre. Schon vorher gab es zahlreiche Methoden für die strukturierte Programmierung, die mit Erfolg in der Industrie eingesetzt wurden. Die Überlegung hinter der Entwicklung objektorientierter Methoden war, die Vorteile der objektorientierten Programmierung von der Implementierungsphase auf den gesamten Entwicklungsprozeß zu erweitern, um so die Entwicklung von Softwaresystemen noch effizienter und qualitativ hochwertiger zu machen.

Objektorientierten Methoden ist gemein, daß der Problembereich nach Dingen durchsucht wird, denen gewisse Eigenschaften gemeinsam sind. Diese Dinge bezeichnet man als Objekte. Gibt es mehrere Objekte mit ähnlichem Verhalten und ähnlicher Struktur, so kann man diese auf eine gemeinsame Klasse zurückführen. Ein Objekt ist eine Instanz der Klasse. Ist eine Klasse von einer anderen Klasse abgeleitet, so erbt sie die in der Oberklasse definierten Eigenschaften und deren Verhalten.

3.4 OOSE

Eine der wichtigsten Methoden ist das von Ivar Jacobsen entwickelte OOSE (Object Oriented Software Engineering) [Jac92]. OOSE beschreibt die Durchführung der Phasen Analyse, Design (Entwurf), Implementierung und Test sowie die Repräsentation der in diesen Phasen erstellten Modelle als Ergebnisse.

Die Analyse im Rahmen der OOSE beschränkt sich darauf, Darstellungen und Abstraktionen des Problembereiches zu schaffen, und so diesen ausreichend zu modellieren, während der Entwurf die entwickelten Modelle um diejenigen Abstraktionen erweitert, die notwendig werden, um das Gesamte in einer Implementierung abbilden zu können. Dazu gehört der Entwurf von Klassen zur Benutzerführung, Ein-/Ausgabe der Daten, Speicherung und ähnlichen Aufgaben.

Um mit OO7SE ein System zu erstellen, werden fünf Modelle erstellt. Mit dem ersten Modell, dem Anforderungsmodell, werden alle funktionellen Anforderungen an das System aus Benutzersicht definiert. Es enthält die Akteure und die Anwendungsfälle (Use Cases). Akteure sind Benutzer, aber auch Ereignisse, Dialoge oder externe Systeme, die mit dem zu erstellenden System interagieren. Ein Anwendungsfall beschreibt eine typische Interaktion eines Akteurs mit dem System. Dadurch werden die funktionalen Anforderungen, die das System erfüllen soll, modelliert. Dabei beschreibt ein Anwendungsfall keine einzelnen Operationen wie "Textfeld ausfüllen", sondern relativ große Abläufe wie "Neuen Mitarbeiter anlegen". Aus den Anwendungsfällen werden Diagramme [siehe Abbildung 3-3] erstellt, welche die Beziehungen zwischen den Akteuren und dem System zeigen. Werden einzelne Operationen eines Anwendungsfalles in mehreren Use Cases benötigt, so kann man diese herausfiltern und im Diagramm darstellen. [Jac92]

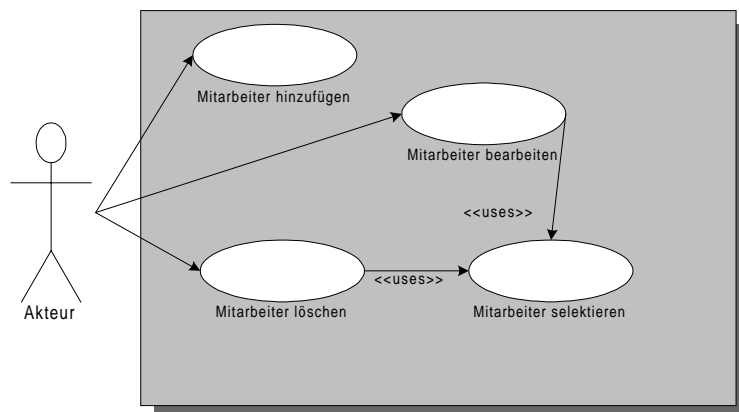


Abbildung 3-3

Anwendungsfalldiagramm in UML-Notation

Aus dem Anforderungsmodell wird das Analysemodell entwickelt. Bei der Entwicklung dieses Modells werden Implementierungsaspekte nicht berücksichtigt. Dies hat zwei große Vorteile. Zum einen konzentriert man sich durch die Vernachlässigung von Faktoren, die später die Implementierung beeinflussen wie beispielsweise der ausgewählten Hardware oder die Programmiersprache, auf das Wesentliche: eine robuste, objektorientierte Struktur zu schaffen, die während des gesamten

Entwicklungsprozesses des Systems erhalten bleibt. Zum anderen können sich während des Entwicklungsprozesses die zur Verfügung stehenden Technologien verändern und verbessern. Wenn schon in einer so frühen Phase speziell auf konkrete Technologien eingegangen wird, ist man darauf festgelegt und der Ansatz somit wenig flexibel. Verbesserte Möglichkeiten können unter Umständen nicht genutzt werden oder es ist eventuell eine aufwendige Anpassung notwendig. In der dazu benötigten Zeit können sich die Technologien natürlich wieder weiterentwickeln.

Im Designmodell wird das Analysemodell verfeinert. Die zentrale Systemstruktur, das heißt die Domänenobjekte, soll so weit als möglich unverändert erhalten bleiben. Allerdings werden jetzt genau die Implementierungsaspekte in das Modell mit einbezogen, die im Analysemodell noch unbeachtet geblieben sind. Es werden Entscheidungen bezüglich einzusetzender Hardware getroffen, oder beispielsweise welche Datenbank verwendet werden soll und wie sie in das System mit eingebunden wird, wenn nicht schon aufgrund bestehender Infrastruktur oder anderen nicht funktionellen Anforderungen diesbezüglich Vorgaben bestehen. Das Implementierungsmodell besteht im Prinzip aus dem Quellcode, mit dem das Designmodell in einer konkreten Programmiersprache umgesetzt wird. OOSE schreibt keine Programmiersprache vor. Um aber möglichst effizient die erstellten Konzepte umzusetzen, sollte eine objektorientierte Programmiersprache verwendet werden.

Für das Testmodell werden anhand der Spezifikationen Tests entwickelt, mit denen die in den Anwendungsfällen vorgegebene Funktionalität überprüft wird. Die Testergebnisse werden aufgezeichnet und dienen als Grundlage für die eventuell notwendige Fehlerbehebung.

3.5 OOHDM und UML

Während klassische Softwaresysteme schon seit längerem mit objektorientierten Methoden entwickelt werden, wurden Webanwendungen bis vor kurzer Zeit nahezu ohne jegliche Methodik entworfen. Dabei ist es erwiesen, daß eine strukturierte Vorgehensweise beim Entwurf von Applikationen die Qualität der Software erhöht und

den Entwicklungsprozess verkürzt. Deshalb sollten auch für die Entwicklung von Webapplikationen Methoden angewendet werden. Die Standardmethoden des objektorientierten Software-Engineering sind dafür jedoch nicht ausreichend. Entscheidend bei Hypermedia-Anwendungen ist die Entwicklung einer gut funktionierenden, durchdachten Navigation. Aufgrund der Eigenarten von Hypertext ergibt sich ein nicht linear strukturierter, komplexer Informationsraum. Eine durchdachte Navigation ermöglicht es dem Benutzer, sich möglichst intuitiv durch die Webapplikation zu bewegen.

Vor diesem Hintergrund entwickelten Daniel Schwabe und Gustavo Rossi [RS95] die OOHDM (Object Oriented Hypermedia Design Method). Die Notation von OOHDM basiert auf der Object Modelling Technique (OMT), die James Rumbaugh und andere entwickelt haben [Rum+91]. OOHDM gliedert sich in vier verschiedene Phasen:

- **Conceptual Design:**

Das Hauptziel in dieser Phase ist es, das gesamte Gebiet der späteren Anwendung zu erfassen und zu beschreiben, jedoch so allgemein wie möglich, unabhängig vom Bildschirmlayout und Implementierung. Hierbei werden die Vorgehensweise und Notation von OMT verwendet. Im Prinzip entspricht das Conceptual Design der Analyse im klassischen objektorientierten Software-Engineering.

- **Navigational Design:**

Nachdem die Klassen und ihre Beziehungen untereinander festgelegt wurden, müssen die Navigationsbeziehungen näher spezifiziert werden. Dazu dienen drei Grundobjekte:

- ◆ **Nodes** (Knoten) präsentieren Informationseinheiten, die von verschiedenen Medientypen (Text, Grafik, Video, etc.) repräsentiert werden können. Knoten enthalten zudem noch Anker, die auf andere Knoten hinweisen. Node-Klassen können im allgemeinen von den bestehenden Conceptual-Klassen abgeleitet werden.
- ◆ **Links** (Querverweise) stellen Beziehungen von Knoten untereinander dar und bilden die möglichen Navigationspfade. Links haben als Anfangspunkt einen Anker in einem Knoten und können mehrere Zielpunkte haben.
- ◆ **Access Structures** (Zugriffsstrukturen) sind weitere Möglichkeiten, um zwischen den Knoten zu navigieren. Zu den am meisten verwendeten Strukturen gehören Indices, Menüs und Kataloge.

- **Presentational Design:**

In dieser Phase wird festgelegt, in welcher Weise der Benutzer interagieren kann, welche Informationen und Navigationsobjekte erforderlich sind und welche Auswirkungen die Aktionen des Benutzers auf den Bildschirmaufbau und die Darstellung der Informationen haben. Zu jedem Knoten wird ein Presentational Object entworfen, das alle Elemente des späteren Bildschirms enthält (Informationseinheiten, Navigations-/Interaktionsobjekte). Die Definition dieser Presentational Objects ist unabhängig von näheren Angaben zum Layout wie Farbe, Bildschirmposition oder genaue Größe der Elemente.

- **Implementierung und Test**

Die entwickelten Modelle werden mit einer konkreten Programmiersprache implementiert und die geforderten Funktionalitäten überprüft.

OOHDH gibt keine feste Notation vor, es wird jedoch die Verwendung der in OMT spezifizierten Notation empfohlen. In der Zwischenzeit wurde allerdings eine allseits anerkannte und mächtige Notation entwickelt. Diese Notation heißt Unified Modelling Language (UML) und dient zur Spezifikation, Konstruktion, Visualisierung und Dokumentation von Modellen für Softwaresysteme. Da bereits anstelle der OOSE-eigenen Notation die UML verwendet werden kann, bietet sich auch der Einsatz in Verbindung mit OOHDH an, um während des gesamten Softwareentwicklungsprozesses eine einheitliche Notation zu benutzen. Die klassischen UML-Diagramme reichen jedoch nicht aus, um alle Aspekte einer Hypermedia-Anwendung, beispielsweise den Navigationsraum und dessen graphische Repräsentation, zu modellieren. Aus diesem Grund entwickelten Baumeister, Koch und Mandel [BKM99] zusätzliche, UML-konforme Notationselemente für den Einsatz der UML im Webdesign.

4 Analyse

Die Anforderungsanalyse ist im Rahmen der strukturierten, objektorientierten Softwareentwicklung die erste Phase, die durchlaufen wird. Dabei werden die Wünsche und Anforderungen des Kunden an die zu erstellende Software festgehalten und mit objektorientierten Modellierungstechniken auf eine höhere Abstraktionsebene gebracht. Desweiteren wird analysiert, ob und wie die Aufgaben, die vom Softwaresystem übernommen werden sollen, bisher erledigt werden. Zusätzlich werden die bestehende Infrastruktur sowie die Arbeitsabläufe analysiert.

Es werden funktionale und nichtfunktionale Anforderungen unterschieden. Die funktionalen Anforderungen definieren, welche Aufgaben das System auf welche Art und Weise erfüllen soll. Nichtfunktionale Anforderungen beschreiben Vorgaben wie beispielsweise Art und Umfang der Dokumentation oder die Einhaltung eines bestimmten Budgets.

Es gibt mehrere Ansätze, wie man als Softwareentwickler an diese Anforderungen herankommt. Aus der Belegschaft werden die an dem Projekt beteiligten Mitarbeiter, die späteren Benutzer und die Auftraggeber, beispielsweise die Geschäftsleitung oder die Leitung einer Abteilung zu ihren Anforderungen und Wünschen befragt. Im Dialog mit den Mitarbeitern wird so ein erstes Anforderungsprofil gewonnen. Auf Grund dieser Kenntnisse werden Bewertungsfragebögen erstellt und an die späteren Benutzer ausgegeben. Mit diesen Fragebögen können die zukünftigen Anwender überprüfen, ob alle wichtigen Funktionalitäten erfaßt wurden und diese nochmals nach Wichtigkeit bewerten.

Da bei Medialab ein firmenweites Kommunikationsinstrument geschaffen werden sollte, waren alle Mitarbeiter betroffen. Die Ergebnisse der Befragungen und Diskussionen werden ausgewertet und zu einem ersten Anforderungsprofil verdichtet. Dieses wird mit den Mitarbeitern noch einmal besprochen, um Detailfragen zu klären und die gewonnenen Erkenntnisse zu verifizieren.

Sind die Anforderungen spezifiziert, so werden daraus sogenannte Anwendungsfälle modelliert. Diese dienen dazu, die am System beteiligten Akteure zu identifizieren und die Funktionsmodule des Systems zu entwickeln. Mit diesen Erkenntnissen kann dann das Design angegangen werden.

4.1 Beschreibung des Ist-Zustands

4.1.1 Unternehmensprofil

Medialab ist eine im Münchener Süden ansässige Multimedia-Agentur und besteht seit 1989. Hier werden für Kunden aus Handel und Industrie Internetauftritte konzipiert und umgesetzt. Die Kunden werden meist über einen längeren Zeitraum betreut. Zu den weiteren Dienstleistungsangeboten zählen die Erstellung von Multimedia-Präsentationen oder Multimedia-CD-ROMs. Außerdem werden Kunden bei ihren Messeauftritten mit dem Präsentations-Know-How von Medialab unterstützt. Dafür wurden auch schon spezielle Geräte konzipiert.

In der Agentur sind etwa 40 Mitarbeiter in den verschiedenen Geschäftsbereichen beschäftigt. Folgende Geschäftsbereiche kann man unterscheiden:

Geschäftsleitung, Produktion, Grafik, Projektleitung, Administration, Konzeption, Vertrieb, Marketing, PR und Buchhaltung / Controlling.

Geschäftsleitung und Vertrieb kümmern sich um die Beschaffung neuer Aufträge. Die Projektleitung setzt diese Aufträge mit Hilfe der Grafik- und Produktionsabteilung um. Von der Konzeption werden sie dabei unterstützt. Marketing und PR (Public Relations)

befassen sich mit dem Auftreten der Agentur nach außen, die Administration kümmert sich um technische und die Buchhaltung / Controlling um finanzielle interne Belange.

Zwischen den einzelnen Hierarchien und Abteilungen gibt es keine Berührungängste, was schon in der offenen Gestaltung der Firmenräume wurzelt. Bei projektbezogenen Fragen tauschen sich die Mitarbeiter verbal oder per E-Mail aus, für strategische Besprechungen werden interne Meetings angesetzt, bei denen alle Interessierten mitdiskutieren können. Feste Dienstwege wie sie beispielsweise bei Behörden gepflegt werden, gibt es bei Medialab kaum. Lediglich für Urlaub und größere Anschaffungen müssen schriftliche Anträge gestellt werden. Es gibt zwar allgemeine Verhaltensregeln, aber diese schränken die Mitarbeiter nicht in ihrer Freiheit und Eigeninitiative ein. Eigenverantwortliches Handeln wird gefördert und auch praktiziert. Das Unternehmen lebt von der Kreativität seiner Mitarbeiter und gibt diesen auch die Freiheit, ihre Ideen gemeinsam umzusetzen. Dadurch entsteht eine globale Teamatmosphäre. Die Offenheit und Möglichkeit zur Selbstverwirklichung sollte sich im Rahmen des Vertretbaren auch im Intranet widerspiegeln, so daß die Informationsinhalte von allen Mitarbeitern beigesteuert und genutzt werden können.

4.1.2 Software- und Hardwareausstattung

Bei Medialab sind etwa siebzig Rechner in den verschiedensten Einsatzgebieten im Gebrauch. Dabei handelt es sich um unterschiedliche Modelle und Hersteller. In der Grafikabteilung, bei den Projektleitern, der Geschäftsleitung und in den Abteilungen Marketing, Vertrieb und Konzeption werden hauptsächlich Apple Computer eingesetzt. In der Produktionsabteilung (Programmierung) kommen dagegen vor allem IBM kompatible Personal Computer zum Einsatz.

Neben den Programmierer-Rechnern, die mit Microsoft Windows 95 / 98 oder NT 4.0 als Betriebssystem laufen, gibt es noch andere Rechner, die mit Linux konfiguriert sind. Diese dienen als Webserver und Datenbankserver für Tests. Ein weiteres Unix-System wird zur Erstellung von High-End-Grafikanimationen benutzt, eine SGI (Silicon

Graphics) O2 mit Irix 6.3 als Betriebssystem. Auf allen vorhandenen Computern ist ein Webbrowser von Microsoft und/oder Netscape , Version 4 oder höher, installiert.

Im Unternehmen kommt als zentrales Softwareprodukt eine Datenbankanwendung namens Ophis zum Einsatz, über die die Zeiterfassung für Projekte sowie andere geschäftsinterne Vorgänge abgewickelt werden.

4.2 Anforderungsanalyse

Die folgenden Anforderungen wurden aus zahlreichen Gruppendiskussionen und Einzelgesprächen mit Mitarbeitern des Unternehmens gewonnen. Die daraus erhaltenen Erkenntnisse wurden allen Mitarbeitern nochmals schriftlich vorgelegt. Damit sollte allen die Möglichkeit gegeben werden, zu kontrollieren, ob auch alle wichtigen inhaltlichen Punkte erfaßt wurden.

4.2.1 Allgemeine Anforderungen

Wie bereits dargelegt, gibt es bei der Firma Medialab im Moment kein System, mit dem das Firmen-Know-How gespeichert und abgerufen werden kann. Die Firma lebt aber von diesem Know-How. Techniken zur Erstellung ausgefallener Grafikeffekte oder Programmierkniffe zur Realisierung spezieller Navigationen gehen eventuell verloren, wenn ein Mitarbeiter die Firma verläßt. Da ständig einige Studenten ihre Praxissemester absolvieren und sie stark in Projekte mit eingebunden werden, wird dieses Problem noch verstärkt. Dieses Know-How muß dann, wenn es wieder gefordert ist, neu erworben werden. Manchmal ist das aber nicht mehr möglich, wenn beispielsweise die Dokumente, die eine bestimmte Technik beschrieben haben, nicht mehr verfügbar sind.

Damit dies also nicht passiert, will die Firma ein System, in das solches Know-How eingegeben werden und bei Bedarf abgerufen werden kann. Eine weitere Forderung ist, das System auf der Basis des WWW zu realisieren. Denn das System soll als Intranet dienen, in dem sich die Mitarbeiter selbst informieren und anderen Mitarbeitern Informationen zur Verfügung stellen. Außerdem soll bei Bedarf auch Kunden der

Zugriff auf das Intranet gegeben werden, um ihnen eine Vorstellung davon zu geben, was alles webbasiert zu realisieren ist.

Von der Geschäftsleitung kam die Vorgabe, das System mit einem möglichst geringen Kostenaufwand zu realisieren.

4.2.2 Inhaltliche und funktionelle Anforderungen

Folgenden Inhalte sollten im Informationssystem realisiert werden:

Telefonliste

Es soll eine Telefonliste entstehen mit der die Mitarbeiter bei Bedarf nach der Durchwahl von anderen Mitarbeitern suchen können. Wenn ein Mitarbeiter ausscheidet, muß der entsprechende Eintrag gelöscht werden. Kommt ein neuer Mitarbeiter hinzu, wird für ihn ein neuer Eintrag erstellt. Bei Wechsel an einen anderen Arbeitsplatz innerhalb der Firma wird der entsprechende Eintrag verändert. Da die Systemadministratoren auch die Telefonanlage pflegen, sollen Veränderungen nur von ihnen durchgeführt werden können. Bisher werden die Mitarbeiter bei Veränderungen informiert. Diese Vorgehensweise sollte allerdings durch die Suchfunktion überflüssig werden.

Know-How Datenbank

Die Know-How-Datenbank ist die zentrale und wichtigste Anwendung im Intranet vom Medialab. Die Inhalte der Know-How-Datenbank sollten nach den Unternehmensabteilungen geordnet sein. Das heißt, es soll eine Hierarchie eingeführt werden, durch die die Inhalte schon einmal vorsortiert sind. So haben die Mitarbeiter die Möglichkeit, gezielt nach bestimmten Themen zu suchen und sich dadurch alle passenden Informationen auflisten zu lassen. Daneben soll aber auch eine Stichwortsuche realisiert werden, die eine Liste von Verweisen auf vorhandene Beiträge zu dem Thema erzeugt. Dazu sollte von jedem Beitrag eine Art Zusammenfassung angezeigt werden, damit sich der Benutzer auf einen Blick einen detaillierten Eindruck verschaffen kann. Die Ergebnisliste sollte nach Kriterien sortiert werden können,

beispielsweise nach Programmiersprache, Autor oder Erstellungsdatum der Information. Damit die Liste nicht zu lang wird oder um schon eine Vorauswahl treffen zu können, sollten Filter gesetzt werden können. Wird dann tatsächlich eine Information ausgewählt, sollte mit Navigationsschaltflächen zur nächsten oder vorherigen Information in der Liste gewechselt werden können. Für den Fall, daß einem Mitarbeiter beim Lesen einer Information noch etwas Wichtiges einfällt, sollte ein Kommentarfeld vorhanden sein, in das Text eingegeben werden kann. Dieser sollte dann in die Datenbank übernommen werden (entweder an die aktive Information angehängt oder als eigenständige Information).

Neben den Menüpunkten Produktion (Programmierung), Grafik, Projektleitung, Vertrieb/Konzept und Marketing soll noch ein Menüpunkt Allgemeines entstehen.

Dieser Menüpunkt soll die Firmenleitlinien, eine Art Schwarzes Brett (Kleinanzeigen), einen Menüpunkt News und noch weitere Menüpunkte wie beispielsweise ein Presseecho oder Hinweise zum Rechnerbetrieb enthalten. Hinter den Firmenleitlinien verbergen sich allgemeine Hinweise zu Medialab, grundlegende Verhaltensregeln und so weiter. Unter dem Menüpunkt Kleinanzeigen haben die Mitarbeiter die Möglichkeit, Anzeigen in Form von Texten - wenn erforderlich mit Bild - aufzugeben. Der Menüpunkt News dient dazu, den Kollegen interessante Neuerungen aus allen Bereichen des Lebens zukommen zulassen, primär natürlich bezogen auf Dinge, die mit dem allgemeinen Geschäftsbetrieb zu tun haben. Unter dem Menüpunkt Presseecho sollen durch die Marketing und PR-Abteilung Reaktionen der Presse auf die Arbeit von Medialab und auch der Konkurrenz plaziert werden. Um den Systemadministratoren die Arbeit zu erleichtern, sollten Bedienungstips zu allgemeinen Benutzung der Rechner vorhanden sein, die alle Mitarbeiter nutzen können. Hierbei handelt es sich beispielsweise um Informationen, wie man auf Netzwerklaufwerke zugreifen kann, wie man den Rechner herunterfährt und ähnliches. Alle Mitarbeiter können diese Informationen nutzen, aber auch verändern.

Der zweite Menüpunkt bezieht sich auf die Produktion, das heißt auf alle Themen, die mit der Programmierung von Webseiten oder Multimedia-CD-ROMs für Präsentationen

zu tun haben. Hier finden sich Hinweise zu häufig gestellten Fragen ebenso wie Dokumentationen zu im Rahmen von Projekten gelösten Problemen oder Tips und Tricks zur Programmierung. Jeder Beitrag soll eine kurze Beschreibung des Problems und seiner Lösung sowie Beispielcode enthalten.

Zusätzlich sollte für alle Mitarbeiter die Möglichkeit bestehen, neue Informationen über Problemlösungen oder Programmertips und -tricks anzulegen. Nur so ist gewährleistet, daß vom vorhandenen und sich weiterentwickelnden Firmenwissen auch ein Großteil nachvollziehbar bleibt und nachzuschlagen ist. Damit es nicht zu Wildwuchs kommt, sollten die Mitarbeiter überholte Informationen löschen können. Es liegt somit an den Benutzern selbst, daß das System auch benutzbar bleibt. Sollte sich diese Lösung nicht bewähren, kann die Pflege auf dazu angehaltene Mitarbeiter verteilt werden.

Unter dem Menüpunkt Grafik soll es eine Abschnitt Corporate Design geben, über die sich neue Mitarbeiter oder Praktikanten über Richtlinien im Zusammenhang mit grafischer Gestaltung bei Medialab-Projekten informieren können. Dazu gehören Hinweise zur Dateiablage und den Dateiformaten von Grafiken oder Animationen, Hinweise zur Farbgestaltung, Schriftgrößen und -stilen und Aussagen darüber, wie Entwürfe und fertige Grafiken zu dokumentieren sind. Daneben sollen sich in dieser Datenbank Dokumentationen darüber befinden, wie bei abgeschlossenen Projekten die Grafiken erzeugt wurden, mit welchen Mitteln und Programmen bestimmte optische Effekte erzielt werden können. Dabei können ein oder mehrere Beispielbilder oder auch Animationen einem Beitrag zugeordnet werden. Des weiteren soll ein Schriftenarchiv, das heißt eine Sammlung von Schriftarten entstehen, die nach Schriftstilen sortiert sind. Bei Auswahl einer Schriftart soll ein Schriftmuster angezeigt werden und ausgedruckt werden können. Zusätzlich soll die Möglichkeit bestehen, diese Schriftarten bei gefallen auch auf den lokalen Rechner zu laden. Neue Schriftarten sollten auf den Server geladen werden können und dann automatisch auch in die Liste der verfügbaren Schriften mit aufgenommen werden. Beim Löschen einer Schriftart vom Server durch Berechtigte wird der Schrifteintrag aus der Liste entfernt.

Die Projektleitung will im System To-Do-Listen festhalten. To-Do-Listen sind strukturierte Auflistungen von Menüpunkten, die bei der Abwicklung eines Projekts beachtet werden müssen. Es ist beschrieben wie und in welcher Reihenfolge ein Projekt Schritt für Schritt abgewickelt wird. Zu jedem abgeschlossenen Projekt wird von den Projektleitern ein Protokoll erstellt, das im System zur Dokumentation abgelegt wird. Neben Protokollen gibt es zu vielen Projekten auch Präsentationen, welche ebenfalls im System verwaltet werden sollen.

Für viele Projekte werden sowohl bei der Akquise als auch während der Durchführung Marktforschungsdaten benötigt, beispielsweise demographische Statistiken oder Statistiken über die Benutzung von Browsern. Damit diese Daten nicht ständig neu recherchiert oder gesucht werden müssen, werden auch diese Daten im Informationssystem gehalten. Es soll die Möglichkeit bestehen, die gesammelten Daten nach bestimmten Kriterien zu sortieren. Diese Daten können eventuell auch für die Programmierung von Interesse sein, wenn es darum geht, ob und welche Features beispielsweise in einer Webseite verwirklicht werden.

Alle Mitarbeiter sollen auf die Know-How-Datenbank zugreifen und neue Beiträge erstellen können. Auch das Verbessern von Beiträgen und das Löschen überflüssiger Beiträge wird allen Mitarbeitern gestattet. Wenn eine restriktivere Regelung nötig sein sollte, sollte diese realisierbar sein.

Essenbestellung

Bei Medialab gibt es keine eigene Kantine und außer dem angrenzenden Gutshof Menterschwaige befinden sich keine Gaststätten oder Supermärkte in der unmittelbaren Nähe.

Da sich nicht alle Mitarbeiter selbst etwas mitbringen oder die vorhandene Küche zur Zubereitung ihres Essens benutzen, können sich die Mitarbeiter bei diversen Lieferdiensten Essen bestellen. Die Bestellung muß bis spätestens 12.00 Uhr getätigt werden, um noch berücksichtigt zu werden.

Die Speisekarten der verschiedenen Bringdienste sollten hier zur Verfügung stehen, wobei das Anklicken eines Gerichts dieses in die Bestellliste übernimmt. Wenn die Bestellung abgeschlossen ist, wird das Sekretariat benachrichtigt. Von dort wird die Sammelbestellung telefonisch vorgenommen. Denkbar wäre auch, die Bestellungen zu sammeln und nach Ablauf der Bestellzeit automatisch ein Fax an die entsprechende Lieferdienste zu senden. Wenn sich an den Speisekarten etwas ändert, sollte das Sekretariat die Möglichkeit haben, die Intranet-Speisekarte zu aktualisieren.

4.2.3 Sicherheitsaspekte

Da sich im geplanten Medialab-Intranet Informationen befinden, die nicht für beliebige Personen zugänglich sein sollen, muß eine sichere Methode gefunden werden, um nichtberechtigte Nutzer fernzuhalten. Innerhalb des Unternehmens können die Rechte der einzelnen Mitarbeiter eventuell anhand ihrer bereits vorhandenen Netzwerkkennungen vergeben werden. Wenn die Inhalte alle über einen Webbrowser nutzbar sein sollen, wie erwünscht, ist dies besonders wichtig. Rechte sind in diesem Zusammenhang die Option, bestimmte Inhalte des Intranets anzulegen, zu verändern oder zu löschen. Hier muß in Abstimmung mit den Mitarbeitern noch eine vernünftige Regelung getroffen werden. Da die Zusammenarbeit der Mitarbeiter sehr ausgeprägt ist, können alle von einer freizügigen Gestaltung dieser Regelung profitieren. Sollte sich eine tolerante Regelung der Rechte nicht bewähren, werden die Rechte vom Benutzerkennwort abhängig gemacht.

Wenn auch Kunden auf Teile des Intranets Zugriff gewährt wird, muß eine sichere Lösung gefunden werden, damit der Kunde nur auf eigene Projektdaten zugreifen kann. Das Verändern von Informationen im Intranet durch den Kunden ist nicht erwünscht und muß unterbunden werden.

Da für das Intranet bereits eine eigene WWW-Domain besteht, kann darauf auch von außerhalb des Unternehmens zugegriffen werden. Es ist also besondere Vorsicht geboten was unternehmenskritische Daten im Intranet angeht. Solange hier keine sichere Lösung bereitsteht, werden diese Inhalte noch nicht verwirklicht.

4.3 Anwendungsfälle

Ein Anwendungsfall beschreibt eine typische Interaktion eines Akteurs mit dem System. Akteure sind Benutzer aber auch Ereignisse, Dialoge oder externe Systeme, die mit dem zu erstellendem System interagieren. Dadurch werden die funktionalen Anforderungen, die das System erfüllen soll, modelliert. Dabei beschreibt ein Anwendungsfall keine einzelnen Operationen, wie "Textfeld ausfüllen", sondern relativ große Abläufe wie "Neuen Mitarbeiter anlegen". Aus den Anwendungsfällen werden Diagramme erstellt, welche die Beziehungen zwischen den Akteuren und dem System zeigen. Werden einzelne Operationen eines Anwendungsfalles in mehreren Use Cases benötigt, so kann man diese herausfiltern und im Diagramm darstellen. [Jac92]

Im folgenden werden die für das zu erstellende System erarbeiteten Anwendungsfälle beschrieben. Für jeden Anwendungsfall wurde auch ein Anwendungsfalldiagramm erstellt.

4.3.1 Mitarbeiter

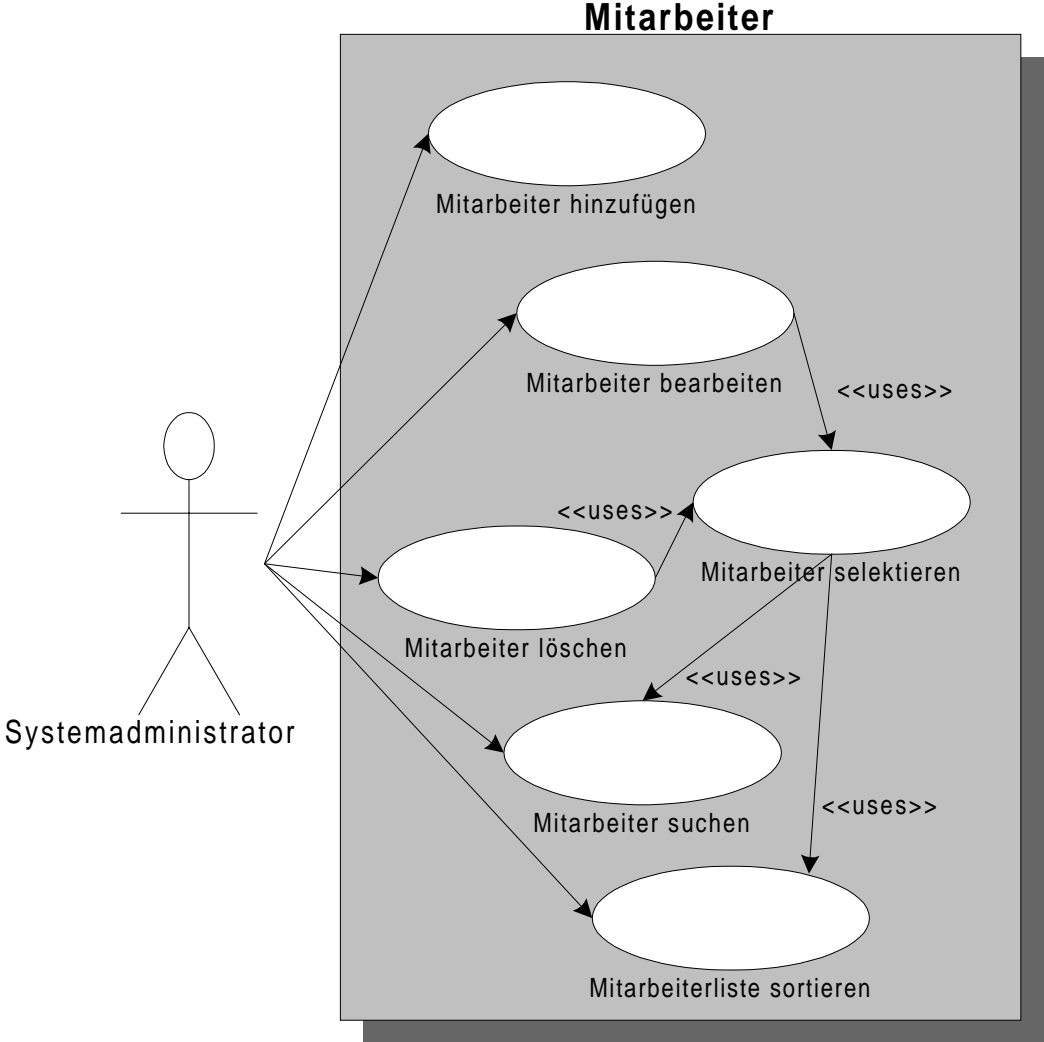


Abbildung 4-1 Anwendungsfalldiagramm Mitarbeiter

<i>Use Case</i>	Mitarbeiter hinzufügen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • Mitarbeitermaske wird geöffnet • Mitarbeiterdaten werden in leeres Formular eingetragen • Übernahme der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • Daten werden automatisch in Datenbank übernommen • neuer Mitarbeitereintrag ist in Mitarbeiterliste aufgenommen
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Mitarbeitermaske • Liste aller angelegten Mitarbeiter

<i>Use Case</i>	Mitarbeiter bearbeiten
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • nach Mitarbeiter wird gesucht oder Mitarbeiter wird aus Mitarbeiterliste selektiert • Mitarbeiterdaten werden in vorbesetztem Formular verändert • Übernahme der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • veränderte Daten werden automatisch in Datenbank übernommen
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Mitarbeitermaske • Suchmaske • Liste aller angelegten Mitarbeiter

<i>Use Case</i>	Mitarbeiter löschen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • nach Mitarbeiter wird gesucht oder Mitarbeiter wird aus Mitarbeiterliste selktiert • Löschen der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • Daten werden automatisch aus Datenbank entfernt • Mitarbeitereintrag ist aus Mitarbeiterliste entfernt
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Suchmaske • Liste aller angelegten Mitarbeiter

<i>Use Case</i>	Mitarbeiter selektieren
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • in Suchmaske Namen, Vorname, Login oder Durchwahl des gesuchten Mitarbeiters eingeben und Suche starten • Mitarbeitereintrag in Mitarbeiterliste markieren
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • selektierte Daten werden für weitere Verwendung zwischengespeichert
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Suchmaske • Liste aller angelegten Mitarbeiter

<i>Use Case</i>	Mitarbeiter suchen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • in Suchmaske Namen, Vorname, Login oder Durchwahl des Gesuchten Mitarbeiters eingeben und Suche starten oder Mitarbeitereintrag in Mitarbeiterliste suchen
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • bei Suche über Suchmaske werden alle Treffer als Liste dargestellt
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Suchmaske • Liste aller angelegten Mitarbeiter • Liste der Suchergebnisse

<i>Use Case</i>	Mitarbeiterliste sortieren
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • durch Anklicken angeben, nach welchem Kriterium (Nachname, Vorname, Login, Durchwahl) sortiert werden soll
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • neu sortierte Liste wird angezeigt
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Liste aller angelegten Mitarbeiter

4.3.2 Telefonliste

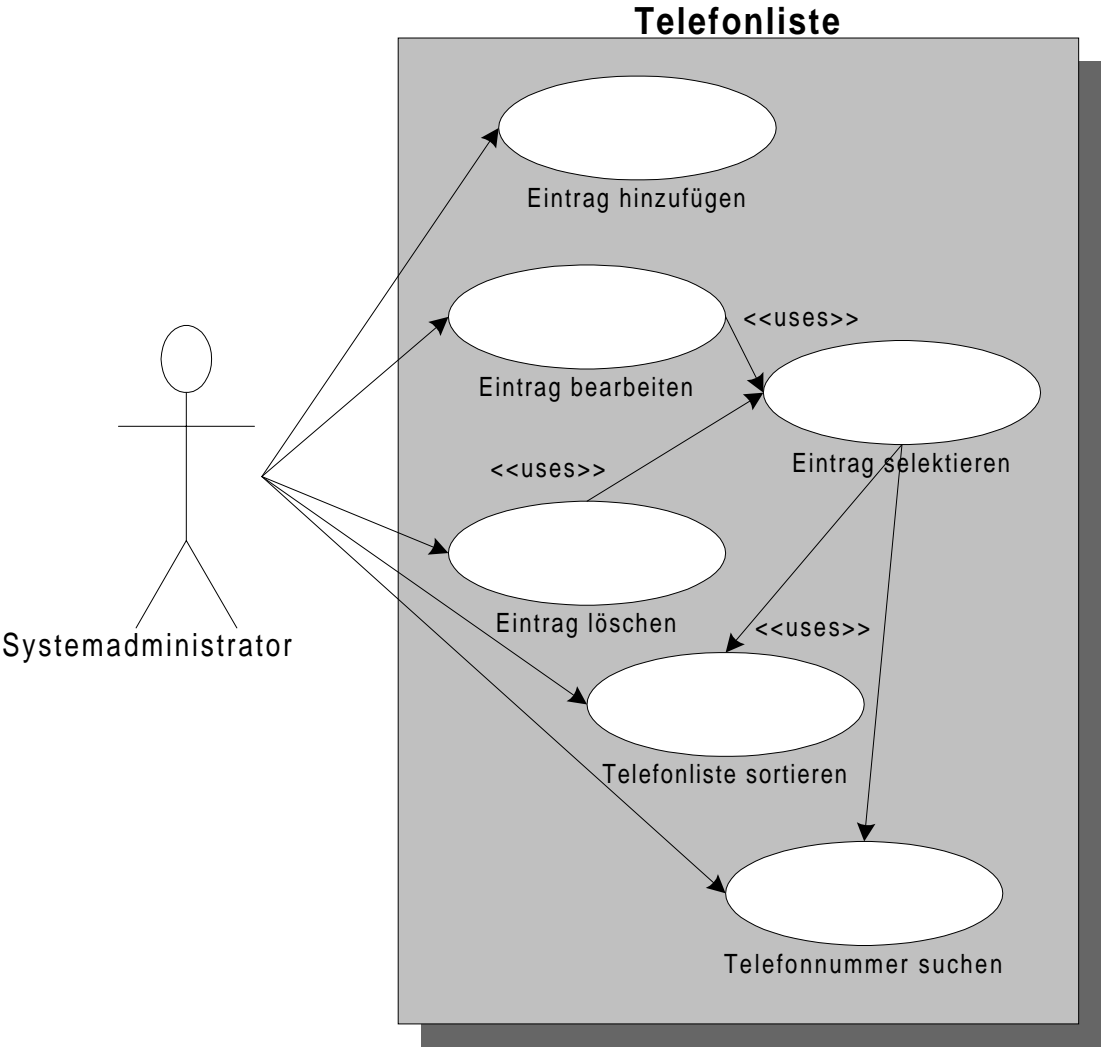


Abbildung 4-2 Anwendungsfalldiagramm Telefonliste

<i>Use Case</i>	Telefonlisteneintrag hinzufügen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • Telefonlistenmaske wird geöffnet • Daten werden in leeres Formular eingetragen • Übernahme der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • Daten werden automatisch in Datenbank übernommen • neuer Telefonlisteneintrag ist in Telefonliste aufgenommen
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Telefonlistenmaske • Telefonliste

<i>Use Case</i>	Telefonlisteneintrag bearbeiten
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • nach Telefonlisteneintrag wird gesucht oder Telefonlisteneintrag wird aus Telefonliste selektiert • Daten werden in vorbesetztem Formular verändert • Übernahme der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • veränderte Daten werden automatisch in Datenbank übernommen
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Telefonlistenmaske • Suchmaske • Telefonliste

<i>Use Case</i>	Telefonlisteneintrag löschen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • nach Telefonlisteneintrag wird gesucht oder Telefonlisteneintrag wird aus Telefonliste selektiert • Löschen der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • Daten werden automatisch aus Datenbank entfernt • Telefonlisteneintrag ist aus Telefonliste entfernt
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Suchmaske • Telefonliste

<i>Use Case</i>	Telefonlisteneintrag selektieren
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • in Suchmaske Namen, Vorname, Login oder Durchwahl des gesuchten Mitarbeiters eingeben und Suche starten • Telefonlisteneintrag in Telefonliste markieren
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • selektierte Daten werden für weitere Verwendung zwischengespeichert
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Suchmaske • Telefonliste

<i>Use Case</i>	Telefonlisteneintrag suchen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • in Suchmaske Namen, Vorname, Login oder Durchwahl des Gesuchten Mitarbeiters eingeben und Suche starten • Telefonlisteneintrag in Telefonlisteliste suchen
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • bei Suche über Suchmaske werden alle Treffer als Liste dargestellt
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Suchmaske • Telefonliste • Liste der Suchergebnisse

<i>Use Case</i>	Telefonliste sortieren
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • durch Anklicken angeben, nach welchem Kriterium (Nachname, Vorname, Login, Durchwahl) sortiert werden soll
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • neu sortierte Liste wird angezeigt
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Telefonliste

4.3.3 Kategorien

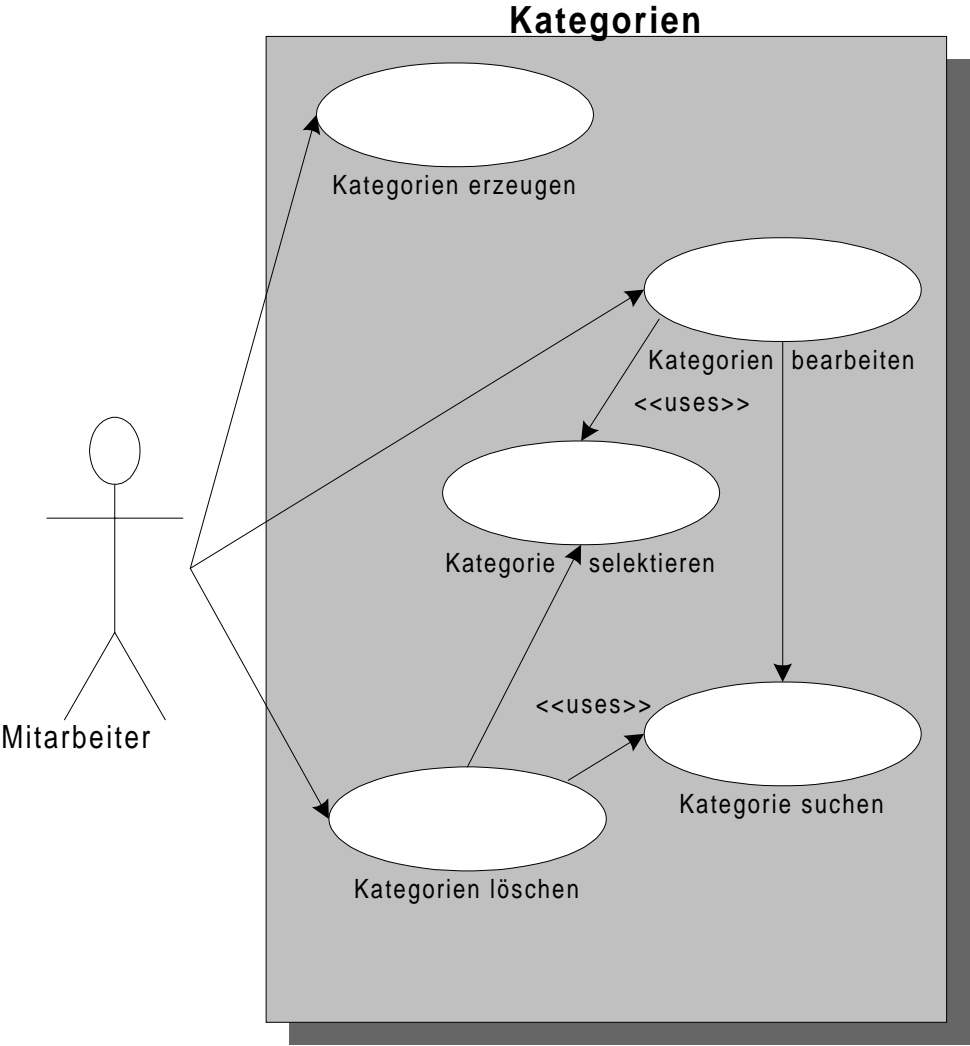


Abbildung 4-3 Anwendungsfalldiagramm Kategorien

<i>Use Case</i>	Kategorie hinzufügen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • Neue Kategorie anlegen wird angewählt • Kategoriemaske wird geöffnet • Daten werden in leeres Formular eingetragen • Kategorie, in welcher diese Kategorie untergeordnet werden soll, wird ausgewählt • aktuelle Kategorie wird als Oberkategorie verwendet • Übernahme der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • Daten werden automatisch in Datenbank übernommen • neue Kategorie ist in Kategoriebaum aufgenommen
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Kategoriebaum • Kategorie-Eingabemaske

<i>Use Case</i>	Kategorie bearbeiten
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • Kategorie wird über Kategoriebaum selektiert • oder aus Suchergebnisliste selektiert • Daten werden in vorbesetztem Formular verändert • Übernahme der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • veränderte Daten werden automatisch in Datenbank übernommen
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Kategoriebaum • Kategorie-Eingabemaske • Suchmaske • Kategorieliste

<i>Use Case</i>	Kategorie löschen
<i>Vorbedingung</i>	<ul style="list-style-type: none"> • Kategorie enthält keine Unterkategorien oder Informationen
<i>Beschreibung</i>	<ul style="list-style-type: none"> • Kategorie wird über Kategoriebaum selektiert oder aus Suchergebnisliste selektiert • Löschen der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • Daten werden automatisch aus Datenbank entfernt und Kategorieeintrag aus Kategoriebaum entfernt
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Kategoriebaum • Suchmaske • Kategorieliste

<i>Use Case</i>	Kategorie selektieren
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • Kategorie in Suchergebnisliste markieren oder Kategorie über Kategoriebaum auswählen
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • selektierte Daten werden für weitere Verwendung zwischengespeichert
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Kategoriebaum • Suchmaske • Suchergebnisliste

<i>Use Case</i>	Kategorie suchen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none">• in Suchmaske Stichwort(e) zu gesuchter Information eingeben und Suche starten
<i>Nachbedingung</i>	<ul style="list-style-type: none">• bei Suche über Suchmaske werden alle Treffer als Liste dargestellt
<i>Schnittstellen</i>	<ul style="list-style-type: none">• Suchmaske• Suchergebnisliste

4.3.4 Information

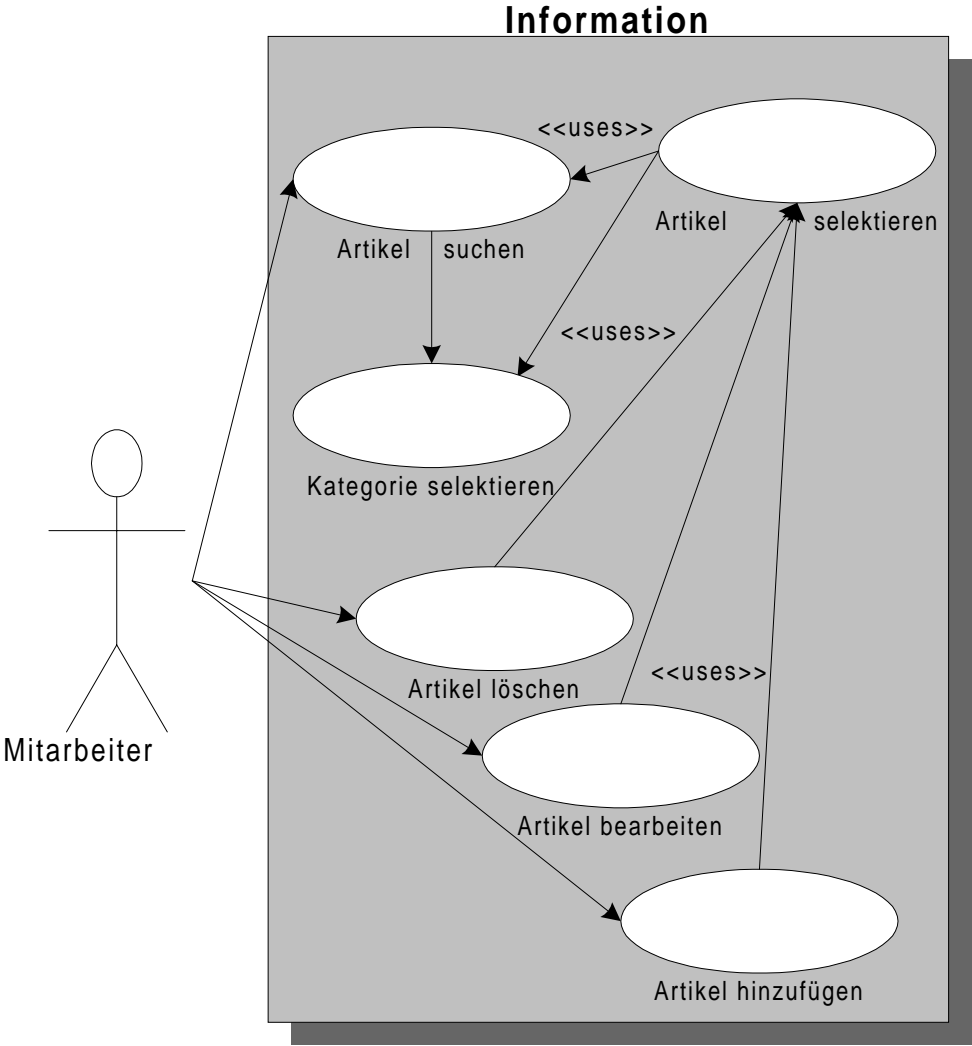


Abbildung 4-4 Anwendungsfalldiagramm Information

<i>Use Case</i>	Information hinzufügen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none">• “neue Information anlegen” wird angewählt• Informationsmaske wird geöffnet• Daten werden in leeres Formular eingetragen• Kategorie, in welcher die Information erscheinen soll, wird ausgewählt oder aktuelle Kategorie wird verwendet• Übernahme der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none">• Daten werden automatisch in Datenbank übernommen• neue Information ist in Informationsliste der entsprechenden Kategorie aufgenommen
<i>Schnittstellen</i>	<ul style="list-style-type: none">• Informationsmaske• Kategoriebaum• Suchmaske• Informationsliste

<i>Use Case</i>	Information bearbeiten
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • nach Information wird gesucht oder über Kategorie wird in der zugehörigen Informationsliste eine Information selektiert • Daten werden in vorbesetztem Formular verändert • Übernahme der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • veränderte Daten werden automatisch in Datenbank übernommen
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Informationsmaske • Kategoriebaum • Suchmaske • Informationsliste

<i>Use Case</i>	Information löschen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • nach Information wird gesucht oder über Kategorie wird in der zugehörigen Informationsliste ein Information selektiert • Löschen der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • Daten werden automatisch aus Datenbank entfernt
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Kategoriebaum • Suchmaske • Informationsliste

<i>Use Case</i>	Information selektieren
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • in Suchmaske Stichwort(e) zu gesuchter Information eingeben und Suche starten • Kategorie selektieren und in der Informationsliste dieser Kategorie eine Information markieren
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • selektierte Daten werden für weitere Verwendung zwischengespeichert
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Kategoriebaum • Suchmaske • Informationsliste

<i>Use Case</i>	Information suchen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • in Suchmaske Stichwort(e) zu gesuchter Information eingeben und Suche starten oder Kategorie selektieren und die Informationsliste dieser Kategorie selbst nach der Information durchsuchen
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • bei Suche über Suchmaske werden alle Treffer als Liste dargestellt
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Suchmaske • Informationsliste

4.3.5 Speisekarte

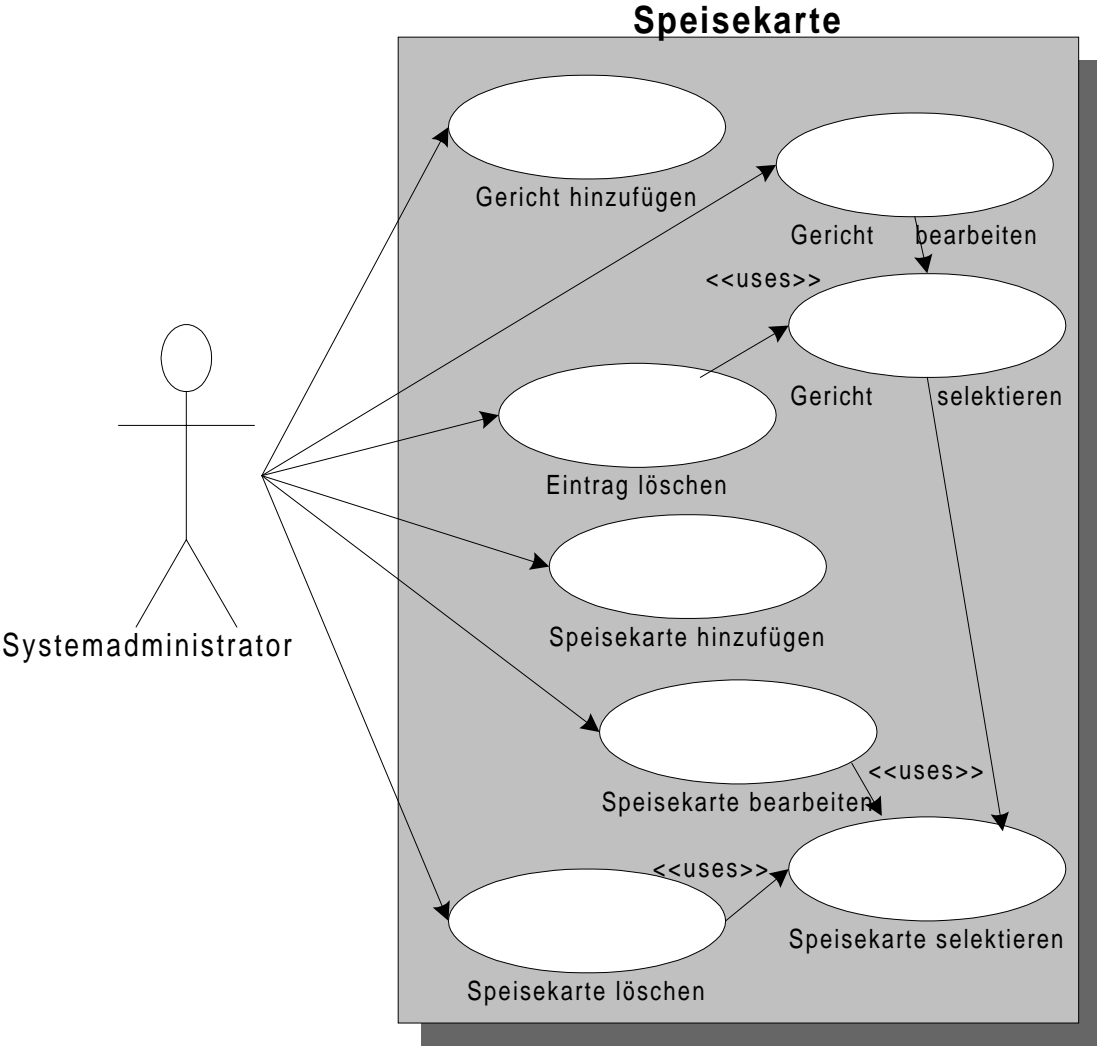


Abbildung 4-5 Anwendungsfalldiagramme Speisekarte

<i>Use Case</i>	Speisekarte hinzufügen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • “neuen Speisekarte anlegen” wird angewählt • Speisekartenmaske wird geöffnet • Daten werden in leeres Formular eingetragen • Übernahme der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • Daten werden automatisch in Datenbank übernommen • neue Speisekarte ist in Speisekartenliste aufgenommen
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Speisekartenliste • Speisekartenmaske

<i>Use Case</i>	Speisekarte bearbeiten
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • Speisekarte wird in Speisekartenliste selektiert • Daten werden in vorbesetztem Formular verändert • Übernahme der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • veränderte Daten werden automatisch in Datenbank übernommen
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Speisekartenliste • Speisekartenmaske

<i>Use Case</i>	Speisekarte löschen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • Speisekarte wird in Speisekartenliste selektiert • Löschen der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • Speisekarte und die zugehörigen Gerichte werden automatisch aus Datenbank entfernt
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Speisekartenliste

<i>Use Case</i>	Speisekarte selektieren
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • Eintrag in Speisekartenliste markieren
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • selektierte Daten werden für weitere Verwendung zwischengespeichert
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Speisekartenliste

<i>Use Case</i>	Gericht anlegen
<i>Vorbedingung</i>	<ul style="list-style-type: none"> • Speisekarte für dieses Gericht muß bestehen
<i>Beschreibung</i>	<ul style="list-style-type: none"> • Speisekartenverwaltung wird gestartet • Speisekarte wird in Speisekartenliste selektiert • Daten werden in leerer Eingabemaske eingegeben • Übernahme der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • Daten werden automatisch in Datenbank übernommen und mit der entsprechenden Speisekarte verlinkt
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Speisekartenliste • Gerichte-Eingabemaske

<i>Use Case</i>	Gericht bearbeiten
<i>Vorbedingung</i>	<ul style="list-style-type: none"> • Speisekarte für dieses Gericht muß bestehen
<i>Beschreibung</i>	<ul style="list-style-type: none"> • Speisekartenverwaltung wird gestartet • Speisekarte wird in Speisekartenliste selektiert • Gericht wird in Speisekarte selektiert • Daten werden in vorbesetzter Eingabemaske verändert • Übernahme der Daten wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • veränderte Daten werden automatisch in Datenbank übernommen
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Speisekartenliste • Speisekarte • Gerichte-Eingabemaske
	<ul style="list-style-type: none"> •

<i>Use Case</i>	Gericht löschen
<i>Vorbedingung</i>	<ul style="list-style-type: none"> • Speisekarte für dieses Gericht muß bestehen
<i>Beschreibung</i>	<ul style="list-style-type: none"> • Speisekartenverwaltung wird gestartet • Speisekarte wird in Speisekartenliste selektiert • Gericht wird in Speisekarte selektiert • Löschen des Gerichts wird bestätigt
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • Link mit Speisekarte wird entfernt und Daten aus Datenbank gelöscht
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Speisekartenliste • Speisekarte

<i>Use Case</i>	Gericht selektieren
<i>Vorbedingung</i>	<ul style="list-style-type: none">• Speisekarte für dieses Gericht muß bestehen
<i>Beschreibung</i>	<ul style="list-style-type: none">• Speisekartenverwaltung wird gestartet• Speisekarte wird in Speisekartenliste selektiert• Gericht wird in Speisekarte markiert
<i>Nachbedingung</i>	<ul style="list-style-type: none">• selektierte Daten werden für weitere Verwendung zwischengespeichert
<i>Schnittstellen</i>	<ul style="list-style-type: none">• Speisekartenliste• Speisekarte

4.3.6 Essenbestellung

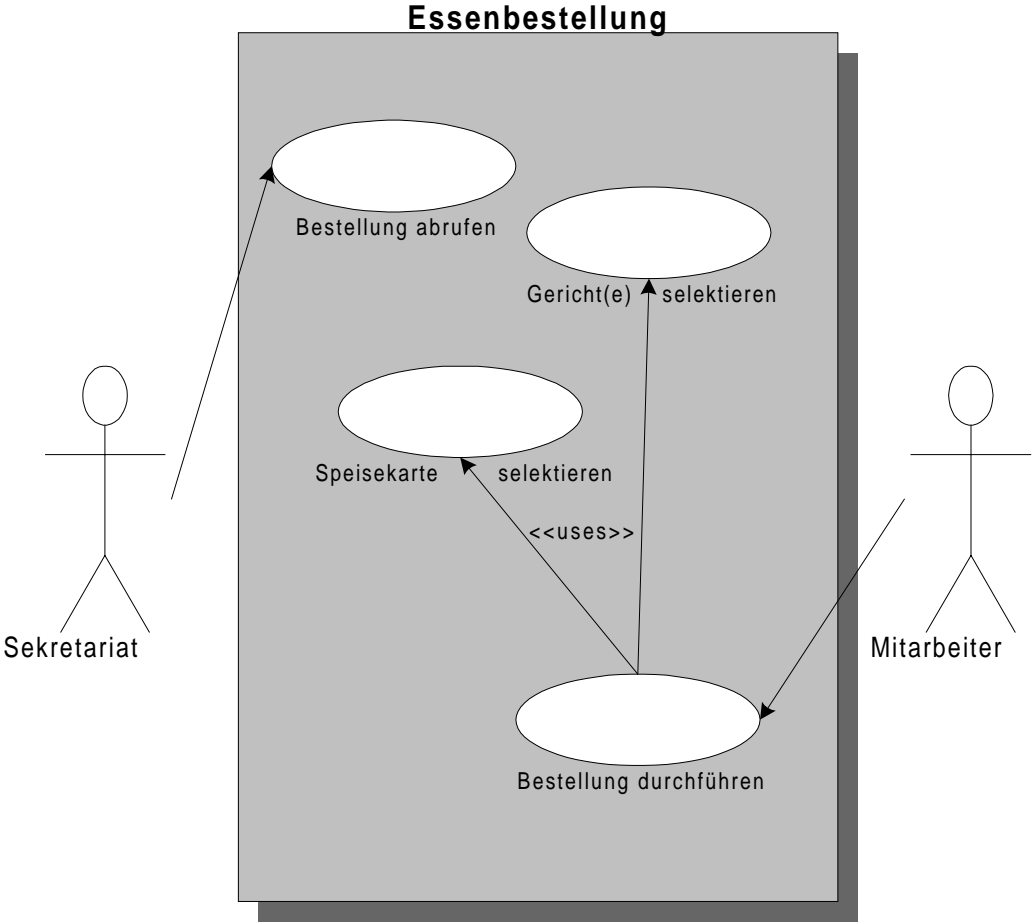


Abbildung 4-6 Essenbestellung

<i>Use Case</i>	Gericht bestellen
<i>Vorbedingung</i>	<ul style="list-style-type: none"> • Speisekarte für dieses Gericht muß bestehen
<i>Beschreibung</i>	<ul style="list-style-type: none"> • Speisekarte wird in Speisekartenliste selektiert • ein oder mehrere Gerichte werden in Speisekarte markiert • Bestellung bestätigen
<i>Nachbedingung</i>	<ul style="list-style-type: none"> • selektierte Daten werden mit Identifizierungsmerkmal des Bestellenden (z.B. Login) in Bestellliste mit übernommen
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Speisekartenliste • Speisekarte

<i>Use Case</i>	Bestellungen abrufen
<i>Vorbedingung</i>	
<i>Beschreibung</i>	<ul style="list-style-type: none"> • “Abrufen der Bestellungen” wird angewählt • Liste der bestellten Gerichte wird erstellt
<i>Nachbedingung</i>	
<i>Schnittstellen</i>	<ul style="list-style-type: none"> • Bestellverwaltung

5 Entwurf (Design)

5.1 Konzeptioneller Entwurf (Conceptual Design)

Der konzeptionelle Entwurf dient dazu, ein statisches Modell der Anwendungsdomäne zu erstellen. Im Rahmen von OOHDM ist dies die erste Stufe bei der Entwicklung des Designs einer Webapplikation. Dieser Schritt entspricht der Erstellung eines Analysemodells im klassischen objektorientierten Software-Engineering. Dieses konzeptionelle Modell wird in UML-Notation als UML-Klassendiagramm dargestellt. Bei der Modellierung werden objektorientierte Konzepte wie Vererbung, Aggregation und Assoziation verwendet.

Die Anwendung besteht aus mehreren Klassen [siehe *Abbildung 5-1*]. Die zentrale Klasse ist die Information. Informationen können neu erzeugt, verändert oder auch wieder gelöscht werden. Daneben kann noch nach Informationen gesucht werden. Informationen lassen sich des weiteren auswählen, um sie anzuzeigen. Eine Information enthält einen Titel, der als eine Art Überschrift dient. Ein Titel besteht aus einem String. Informationen verfügen des weiteren über eine Zusammenfassung, die den Inhalt der Information kurz umschreibt. Die Zusammenfassung ist ebenfalls als String realisiert. Der eigentliche Textinhalt einer Information wird im Attribut Text gehalten. Dieser Text kann beliebig lang sein, und ist wie auch die Attribute Titel und Zusammenfassung ein String. Einer Information kann eine beliebige Anzahl von zusätzlichen Dokumenten, also auch Null, als Anlage beigefügt werden. Daneben kann einer Information auch ein Projekt zugeordnet werden, im Rahmen dessen diese Information entwickelt wurde. Eine Information muß aber nicht zwingend einem Projekt zugeordnet werden. Da Informationen von den Mitarbeitern erstellt werden, ist jeder Information ein

Mitarbeiter als Autor zugeordnet. Um später feststellen zu können, wann eine Information erstellt wurde, wird beim Erzeugen der Information das aktuelle Datum und die aktuelle Uhrzeit dem Attribut Erstellungsdatum zugewiesen.

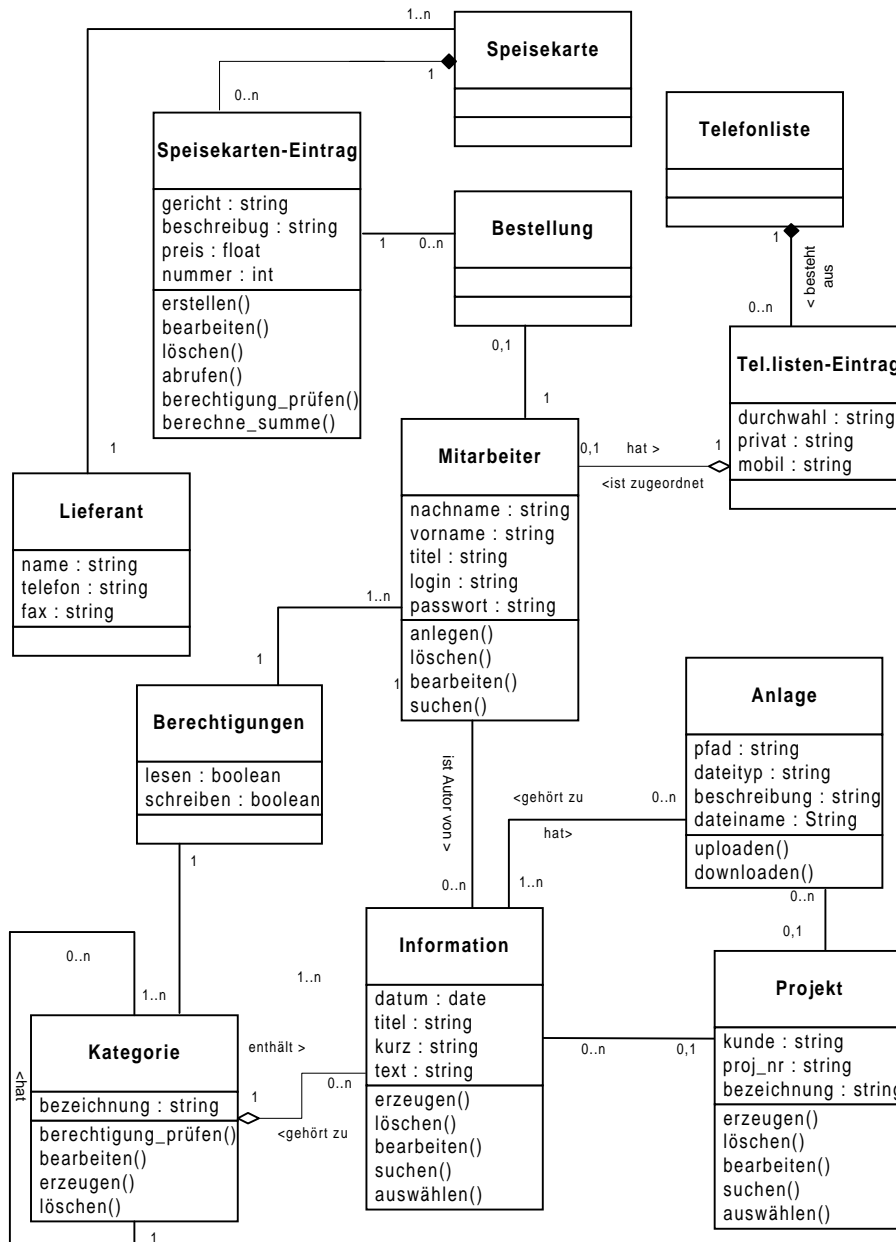


Abbildung 5-1

Konzeptionelles Modell in UML-Notation

Die Informationen sind in verschiedenen Kategorien geordnet. Kategorien dienen als Themenbereiche, in denen Informationen zum gleichen Thema zusammengefaßt werden. Eine Kategorie kann beliebig viele Informationen und Unterkategorien enthalten. Eine Information ist immer einer bestimmten Kategorie zugeordnet. Die Unterkategorien können natürlich ebenfalls weitere Unterkategorien und Informationen enthalten. Es entsteht also eine Baumstruktur, wie sie von Dateisystemen mit Ordnern, Unterordnern und Dateien bekannt ist. Kategorien verfügen über einen String Bezeichnung, mit dem ein Name für diese Kategorie festgelegt wird. Kategorien können neu erzeugt, oder gelöscht werden. Die Bezeichnung der Kategorie kann geändert werden und eine Kategorie kann inklusive aller zugehörigen Unterkategorien und Informationen verschoben werden. Das heißt, eine Kategorie kann innerhalb der Baumstruktur in einen anderen Ast verschoben werden. In diesem Fall ändert sich also die Zuordnung der Oberkategorie, nicht aber die Unterkategorie- und Informations-Zuweisungen.

Die Klasse Anlage, die mit der Klasse Information in Relation steht, enthält als Attribute die Strings Beschreibung, Dateiname, Pfad und Dateityp. Die Beschreibung soll kurz umreißen, um was für eine Anlage es sich handelt, zum Beispiel "Bild mit bestimmtem Effekt". Mit Dateiname wird der genaue Dateiname der Anlagedatei angegeben. Das Attribut Dateityp kategorisiert die Anlage in Kategorien wie Bild, Textdokument, Audio, Animation und ähnliches. Mit Pfad wird der Ort angegeben, an dem sich die Anlagedatei im Netzwerk befindet. Wie auch eine Information kann eine Anlage einem Projekt in zugeordnet werden, in dessen Rahmen sie erstellt wurde. Eine Anlage kann dabei genau einem oder keinem Projekt zugeordnet werden. Anlagen können auf den Server geladen werden und von dort wieder heruntergeladen werden. Außerdem können Anlagen wieder gelöscht werden, falls sie nicht mehr aktuell sein sollten oder nicht mehr gebraucht werden sollten.

Ein Projekt kann wie oben bereits erwähnt, Informationen und Projekten zugeordnet werden. Dabei kann ein und das gleiche Projekt mehreren Informationen oder Anlagen zugewiesen werden, eine Information oder Anlage jedoch nur keinem oder einem Projekt. Ein Projekt ist aufgebaut aus einem String Kunde, einem String Bezeichnung

und einem String Projnr. Das Attribut Kunde nimmt den Kundennamen auf, Bezeichnung dient zur genaueren Beschreibung des Projekts. Dies ist nötig, da der Kundename allein nicht zur eindeutigen Identifizierung des Projekts ausreicht, da für einen Kunden mehrere Projekte abgewickelt werden können. Der String Projnr dient zur Angabe einer internen Projektnummer. Hier kann kein Integerwert verwendet werden, da die intern genutzten Projektnummern alphanumerisch sind. Projekte können neu angelegt, verändert und auch wieder gelöscht werden. Daneben kann man noch nach Projekten suchen und diese auswählen.

Eine zweite zentrale Klasse ist die Klasse Mitarbeiter. Sie dient zum Bearbeiten von Mitarbeiterdaten. Es können Mitarbeiterdaten neu angelegt, verändert und auch wieder gelöscht werden. Zu jedem Mitarbeiterbeitrag gibt es genau einen Eintrag in der Telefonliste. Es werden folgende Mitarbeiterattribute angelegt: der String Nachname nimmt den Nachnamen auf, der String Vorname den Vornamen des Mitarbeiters. Zum Speichern der internen Berufsbezeichnung dient der String Titel. Mit den Strings Login und Paßwort werden Zugangsdaten angelegt, die den Mitarbeiter eindeutig identifizieren. Dies ist nötig, um über Berechtigungen Inhalte des Intranets für bestimmte Mitarbeiter sperren oder freigeben zu können. Außerdem dient das eindeutige Login auch als Identifikation bei Essensbestellungen und beim Anlegen von Informationen.

Die Telefonliste besteht aus null oder mehr Telefonlisteneinträgen und kann nach verschiedenen Kriterien sortiert werden. Es gibt für jeden Mitarbeiter genau einen Eintrag, es gibt jedoch auch Einträge, die sich nicht auf Mitarbeiter beziehen, sondern beispielsweise den Ort angeben, an dem ein Anschluß zu finden ist. Ein Telefoneintrag besteht aus einem String Durchwahl, mit dem die Durchwahlnummer eines Apparats angegeben wird, sowie einem String Mobil, und einem String Privat. Der String Mobil dient zur Aufnahme einer eventuell vorhandenen Mobilfunk-Telefonnummer eines Mitarbeiters, Privat zur Angabe der Privattelefonnummer. Telefonlisteneinträgen können neu erzeugt, verändert und auch wieder gelöscht werden. Jeder neu angelegte Telefonlisteneintrag wird automatisch in die Telefonliste aufgenommen, gelöschte Einträge werden automatisch wieder entfernt.

Die Klasse `Bestellungen` dient zur Aufnahme von Essensbestellungen von Mitarbeitern. Jeder Mitarbeiter kann jeweils nur eine aktive Bestellung anlegen. Bestellungen bestehen dabei aus einem oder mehreren Gerichten. Der Gesamtpreis aller in der Bestellung aufgenommenen Gerichte eines Benutzers wird berechnet. Ein Gericht ist Bestandteil einer Speisekarte und ist aufgebaut aus einem `String` Gericht, der den Namen der Speise beschreibt, einem `String` Beschreibung, mit dem zusätzliche Angaben zur Speise gemacht werden können, einer `Fließkommazahl` Preis, mit dem der Preis des Gerichtes angegeben wird und einer ganzen Zahl Nummer, mit der die Bestellnummer des Gerichts angegeben wird. Gerichte können neu angelegt, bearbeitet und wieder gelöscht werden. Sie sind immer eindeutig einer Speisekarte zugeordnet. Eine Speisekarte wiederum ist eindeutig einem Lieferanten zugeordnet, wobei es von einem Lieferanten jedoch mehrere Speisekarten, beispielsweise Mittagmenü und Abendkarte geben kann. Auch Speisekarten können neu erzeugt und gelöscht werden. Das Bearbeiten von Speisekarten entspricht dem Hinzufügen, Löschen oder Bearbeiten von Gerichten dieser Speisekarte.

Die Klasse `Lieferant` ist einer oder mehreren Speisekarten zugeordnet. `Lieferant` besteht aus einem `String` Name für den Firmennamen des Lieferanten, sowie einem `String` Fax und einem `String` Telefon für die Telefonnummer und die Telefaxnummer dieses Lieferanten. Lieferanten können neu angelegt und auch wieder gelöscht werden, die Lieferantendaten können verändert werden. Es können beliebig viele Lieferanten angelegt werden.

5.2 Entwurf der Navigation (Navigational Design)

Eine Webapplikation besteht aus Knoten und Querverweisen, die Verbindungen zwischen verschiedenen Knoten herstellen. Knoten, die aus dem konzeptionellen Modell abgeleitet sind, nennt man Navigationsobjekte. Der Entwurf der Navigation ist in jeder Webapplikation ein kritischer Schritt. Sogar Webapplikationen mit wenigen Knoten und einer flachen Hierarchie können über eine komplexe Navigationsstruktur verfügen. Querverweise bieten zwar einerseits eine Möglichkeit zur Navigation, andererseits erhöhen sie das Risiko, daß der Benutzer die Orientierung verliert. Der

Entwurf eines Navigationsmodells dient nicht nur zur Dokumentation der Struktur der Webapplikation, sondern trägt zu einer besseren Strukturierung der Navigation bei. Für den Entwurf der Navigation werden zwei Modelle entwickelt, die die Struktur der Webapplikation definieren: ein Navigationsklassenmodell und ein Modell der Navigationsstruktur.

5.2.1 Navigationsklassenmodell

Das Navigationsklassenmodell stellt eine Ansicht auf das Modell dar und zeigt, welche Klassen des konzeptionellen Modells in der Applikation durch Navigation erreicht werden können. Das Modell besteht aus einer Menge von Navigationsklassen und Verknüpfungen zwischen diesen Klassen. Jede Klasse und Verknüpfung des Navigationsklassenmodells wird auf eine Klasse des konzeptionellen Modells abgebildet. Im Navigationsklassenmodell wird für die Verknüpfungen durch einen Pfeil am Ende der Verknüpfungslinie eine Richtung definiert, welche der Richtung der Navigation entspricht. Man kann auch an beiden Enden der Linie einen Pfeil anhängen. Das bedeutet dann, daß in beide Richtungen der Verknüpfung navigiert werden kann. Für jeden Querverweis wird ein Navigationsziel und eine Navigationsquelle unterschieden.

Navigationsklassen werden mit dem Stereotyp **<<navigational class>>** ausgezeichnet. Der Name der Klasse gleicht dem der entsprechenden Klasse im konzeptionellen Modell. Navigationsobjekte sind Instanzen dieser Navigationsklassen. Sie sind untereinander mit Querverweisen verbunden. Klassen, die nicht für die Navigation benötigt werden, werden weggelassen, oder in Attribute einer anderen Klasse umgewandelt. In diesem Fall kann der Wert der Attribute aus den konzeptionellen Objekten berechnet werden. Die Formel zur Berechnung des Werts eines abgeleiteten Attributs wird durch einen OCL-Ausdruck (Object Constraint Language) [OCL97] angegeben. Um anzuzeigen, daß ein Attribut abgeleitet ist, wird bei Verwendung der UML ein Slash (/) vor dem Namen angegeben. [BKM99]

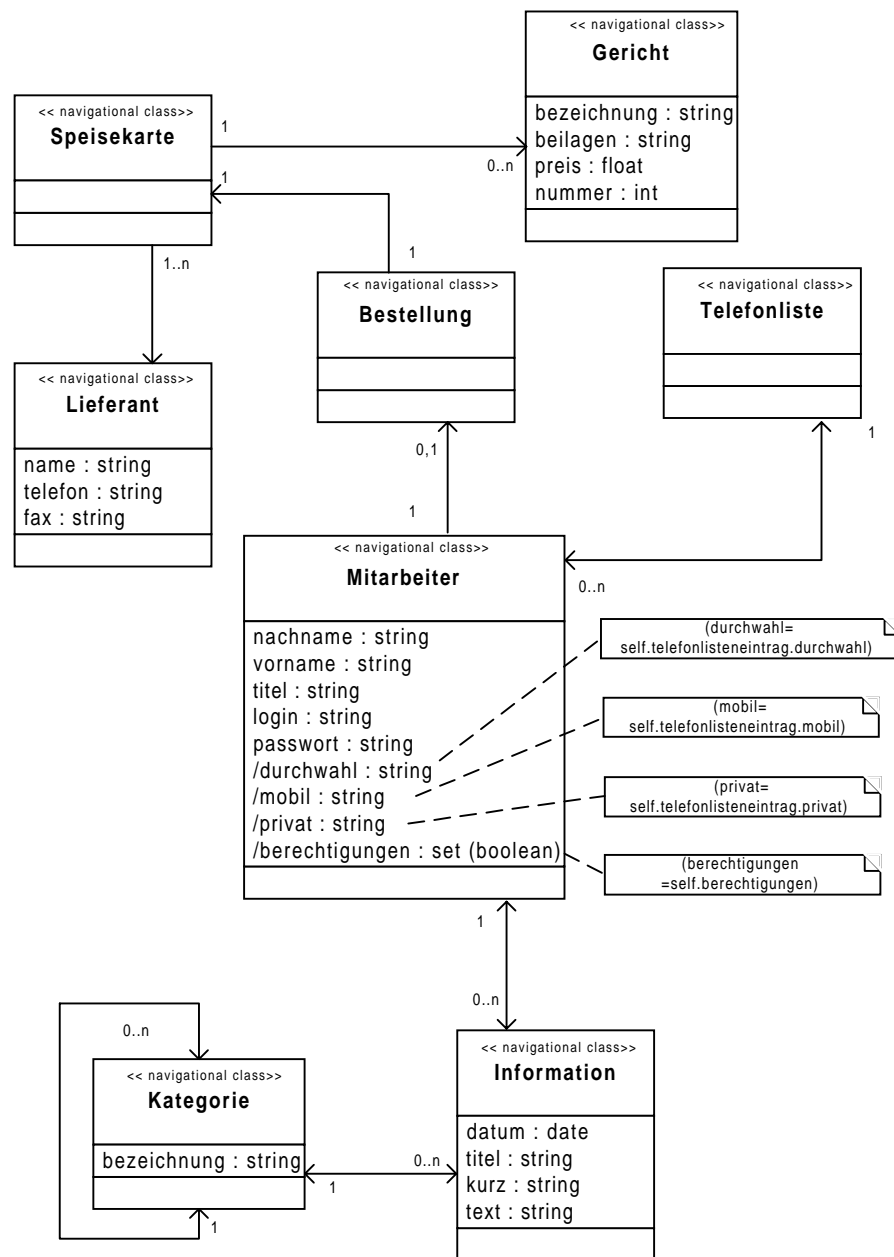


Abbildung 5-1 Navigationsklassendiagramm in UML-Notation

Ausgehend vom konzeptionellen Modell muß entschieden werden, welche Klassen für die Webapplikation entscheidend sind. Bei der in dieser Diplomarbeit entwickelten Applikation enthält das Navigationsklassenmodell alle Klassen des konzeptionellen Modells mit Ausnahme von Berechtigungen und Telefonlisteneintrag. Die Attribute von

Telefonlisteneintrag sind als abgeleitete Attribute in der Klasse Mitarbeiter enthalten. Ebenso ist eine Liste von Berechtigungen als abgeleitetes Attribut in Mitarbeiter eingeordnet. Dies dient dazu, die Verwaltung von Mitarbeiterdaten einfacher zu gestalten. Als Notation für das Navigationsklassendiagramm wie auch für andere Diagramme wurde die UML gewählt.

Es kann zwischen den Navigationsobjekten vom Typ Kategorie und Information hin und zurück navigiert werden. Das gleiche gilt für die Navigation zwischen Information und Mitarbeiter, sowie zwischen Mitarbeiter und Telefonliste. Von Mitarbeiter aus kann außerdem zu Bestellung navigiert werden. Die Navigationsklasse Bestellung ist mit der Navigationsklasse Speisekarte verknüpft. Von Speisekarte aus kann zu Lieferant sowie zu Speisekarten-Eintrag navigiert werden. Das Diagramm ist nachstehend dargestellt [siehe Abbildung 5-1].

5.2.2 Navigationsstrukturmodell (Navigational Structure Model)

Das Navigationsstrukturmodell beruht auf dem Navigationsklassenmodell. Es definiert die Navigationsstruktur der Webapplikation, das heißt auf welche Weise die Navigationsobjekte besucht werden. Um die Navigationsstruktur zu beschreiben, sind zusätzliche Modellelemente erforderlich: Menüs, Indices, externe Knoten und Navigationskontexte. [siehe Abbildung 5-3].

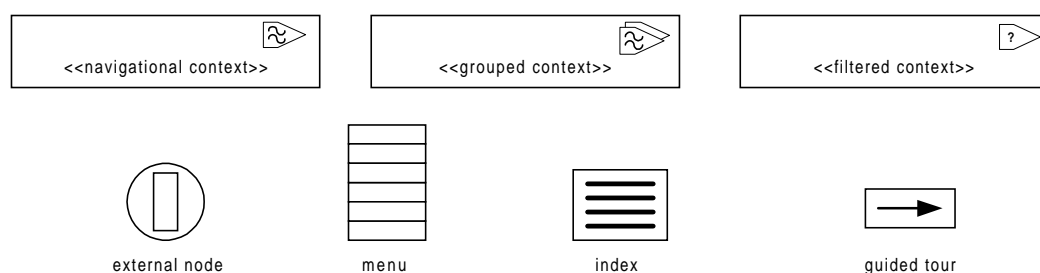


Abbildung 5-3

Stereotypen für das Navigationsstrukturmodell

Ein Navigationskontext besteht aus einer Folge von Navigationsknoten. Er enthält die Definition von Querverweisen, die jeden Navigationsknoten innerhalb des gleichen

Navigationskontextes mit dem nächsten oder vorhergehenden Navigationsknoten verbinden. Ein Navigationsobjekt kann in mehreren Navigationskontexten auftauchen. Ein Navigationskontext wird als Objekt mit dem Stereotyp **<<navigational context>>** dargestellt. Dieses Objekt ist mit einem OCL-Ausdruck gekoppelt, der die Abfolge der Navigationsknoten angibt. Die Navigation findet innerhalb eines Navigationskontextes statt, aber es ist auch der Wechsel des Navigationskontextes möglich.

Es werden drei Arten von Navigationskontexten unterschieden: einfache Navigationskontexte (navigational contexts), gruppierte Kontexte (grouped contexts) und gefilterte Kontexte (filtered contexts).

Ein einfacher Navigationskontext ist beispielsweise „alle Mitarbeiter“. Dieser Kontext bezeichnet die Menge aller Instanzen der Klasse Mitarbeiter, sortiert nach einem bestimmten Kriterium, beispielsweise dem Nachnamen. Das Kriterium ist dabei ein Attribut der Navigationsklasse.

Ein gruppierter Kontext ist eine zwei- oder mehrstufige Sequenz von Navigationsknoten. Ein Beispiel dafür ist „Information by Kategorie“. Dieser Kontext wird aus zwei Navigationsklassen gebildet. Zum einen gibt es die Menge aller Kategorien, zum anderen die Menge aller Informationen. Eine Information soll dabei immer einer Kategorie zugeordnet sein, eine Kategorie kann beliebig viele Informationen enthalten. Die erste Sequenz sind also die Informationen in einer bestimmten Kategorie, die zweite Sequenz sind alle Kategorien. Auch gruppierte Kontexte sind nach einem Kriterium sortiert, allerdings ist das Kriterium ein Attribut einer verknüpften Klasse. Aus relationalen Datenbanken sind solche Gruppierungen als „join“ bekannt. Für gruppierte Kontexte wird das Stereotyp **<<grouped context>>** als Notation verwendet. Außerdem wird im Kasten noch ein Ausdruck der Art **Klasse by Attribut / Rollenname / Klassenname** angegeben.

Unter einem gefilterten Kontext versteht man eine dynamische Auswahl von Elementen aus einem Navigationskontext, die ein bestimmtes Merkmal gemeinsam haben. Üblicherweise wird dieses Merkmal vom Benutzer im Rahmen einer Suche angegeben.

Die Suche ist Teil des gefilterten Kontextes. Ein Beispiel für einen gefilterten Kontext ist das Ergebnis der Suche „alle Mitarbeiter, die mit Vornamen Hubert heißen“. Verschiedene Navigationskontexte ein und der selben Navigationsklasse können in einem UML-Paket zusammengefaßt werden [siehe Abbildung 5-4]. Navigationskontexte sind durch spezielle Verbindungen verknüpft, die einen Wechsel des Navigationskontextes erlauben, wenn ein Objekt Teil mehrerer Navigationskontexte ist. Um die Möglichkeit des Navigationskontextwechsels anzuzeigen, wird das Stereotyp **<<change>>** benutzt. Das Stereotyp muß allerdings nur angegeben werden, wenn innerhalb eines Paketes nicht alle, sondern nur bestimmte Kontextwechsel möglich sind [siehe Abbildung 5-4].

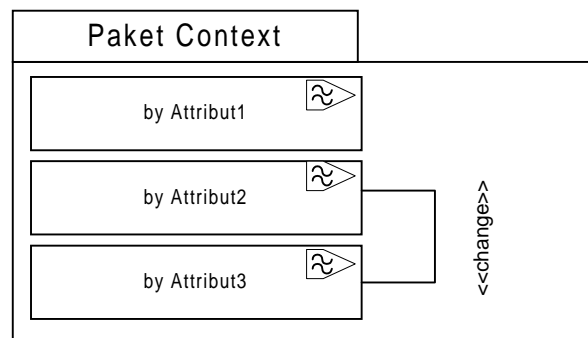


Abbildung 5-4 UML-Paket Navigationskontext

Im Navigationsmodell ist es wichtig zu zeigen, wie auf Navigationskontexte zugegriffen werden kann. Dafür gibt es Indices, Menüs und sogenannte „guided tours“ mit den entsprechenden Stereotypen **<<index>>**, **<<menu>>** und **<<guided tour>>** [siehe Abbildung 5-3].

Ein Index erlaubt direkten Zugriff auf jedes Element eines Navigationskontextes. Normalerweise enthält ein Index eine Liste von Bezeichnungen für die Navigationsknoten des Navigationskontextes, von denen der Benutzer einen Eintrag auswählt. Beispielsweise ist eine Liste von Links ein Index. Die Bezeichnungen der Links dienen dabei als Beschreibung der Navigationsknoten. Eine „guided tour“ erlaubt

den Zugriff auf das erste Objekt eines Navigationskontextes. Die Navigationsobjekte können nur in einer vorgegebenen Reihenfolge durchlaufen werden. Ein Menü ist ein Auswahl von verschiedenen Navigationskontexten. Jede Webapplikation hat zumindest ein Menü in diesem Sinne, das in Form einer Startseite (Hauptmenü) den Ausgangspunkt der Applikation darstellt.

Externe Knoten sind Navigationsknoten außerhalb der Webapplikation, das heißt diese Knoten sind nicht Teil der zu modellierenden Applikation, sondern gehören einer anderen Applikation im Internet an [BKM99].

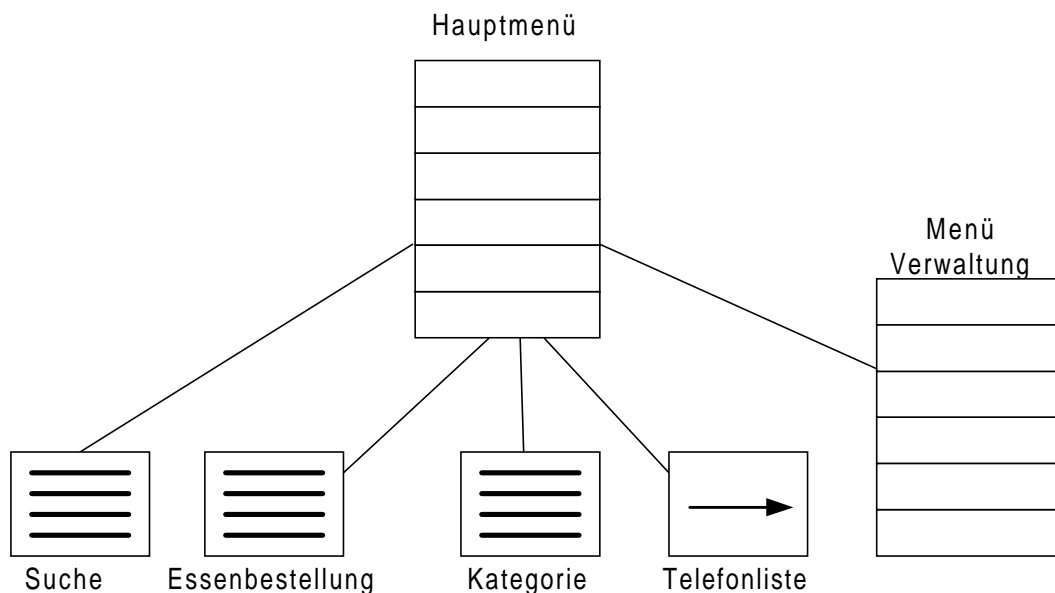


Abbildung 5-5

UML-Objektdiagramm Navigationsstruktur Hauptmenü

Zur Darstellung des Navigationsstrukturdiagramms wird ein UML-Objektdiagramm verwendet. Das Diagramm für die erstellte Anwendung ist auf mehrere Teile aufgeteilt, da es sich nicht leserlich auf einer Seite darstellen läßt. Der erste Teil des Diagramms zeigt das Hauptmenü als Ausgangspunkt der Navigation und die darüber erreichbaren Zugriffsstrukturen [siehe Abbildung 5-5]. Dabei handelt es sich um den Index „Suche“, den Index „Essenbestellung“, den Index „Kategorien“, die Guided Tour „Telefonliste“ und das Untermenü „Verwaltung“. Diese fünf Unterpunkte des Hauptmenüs spiegeln die

logische Struktur der Navigationsklassen wieder. Die weiteren Navigationsmöglichkeiten werden nachstehend beschrieben.

Von der Zugriffsstruktur „Menü Verwaltung“ [siehe Abbildung 5-6] wird auf die Guided Tour „Mitarbeiter / Telefonliste Guided Tour“ verwiesen. Über diese Guided Tour gelangt man zu einem Paket von Navigationstexten „Mitarbeiter / Telefonliste Context“, die alle die Navigationsklassen Mitarbeiter als Basis haben. Es handelt sich dabei um den Navigational Kontext „Mitarbeiter by nachname“, eine Auflistung aller Elemente der Klasse Mitarbeiter sortiert nach Nachname. Es kann jederzeit von „Mitarbeiter by Nachname“ in den gefilterten Kontext „Suche Mitarbeiter“ gewechselt werden. Auch der Wechsel zurück ist jederzeit möglich. Im Gegensatz zum Index „Suche“ im Hauptmenü, wird über diesen gefilterten Kontext nur nach Instanzen der Navigationsklasse Mitarbeiter gesucht.

Über das Menü Verwaltung kann man außerdem mittels einer Guided Tour zum Paket „Kategorie Context“ navigieren. Dieses Paket enthält den Navigationskontext „Kategorie by bezeichnung“, eine Liste aller Instanzen der Navigationsklasse Kategorie, ist nach der Bezeichnung der Kategorie sortiert. Daneben enthält das Paket noch den gefilterten Kontext „Suche Kategorie“. Über die Suche gelangt man zu all den Instanzen der Navigationsklasse Kategorie, die einem vom Benutzer definiertem Kriterium genügen. Ein Wechsel zwischen den beiden Kontexten des Paketes ist jederzeit möglich.

Vom Menü Verwaltung ist die Zugriffsstruktur „Projekt guidedTour“ erreichbar. Über diese kann man auf das Paket „Projekt Context“ zugreifen. Das Paket enthält den Navigationskontext „Projekt by bezeichnung“, eine Liste aller Navigationsknoten der Klasse Projekt nach deren Bezeichnung sortiert. Desweiteren existiert der gefilterte Kontext „Suche Projekt“ analog zu den bereits beschriebenen gefilterten Kontexten „Suche Mitarbeiter“ bzw. „Suche Kategorie“. Zwischen den beiden Kontexten kann nach Belieben hin und her gewechselt werden.

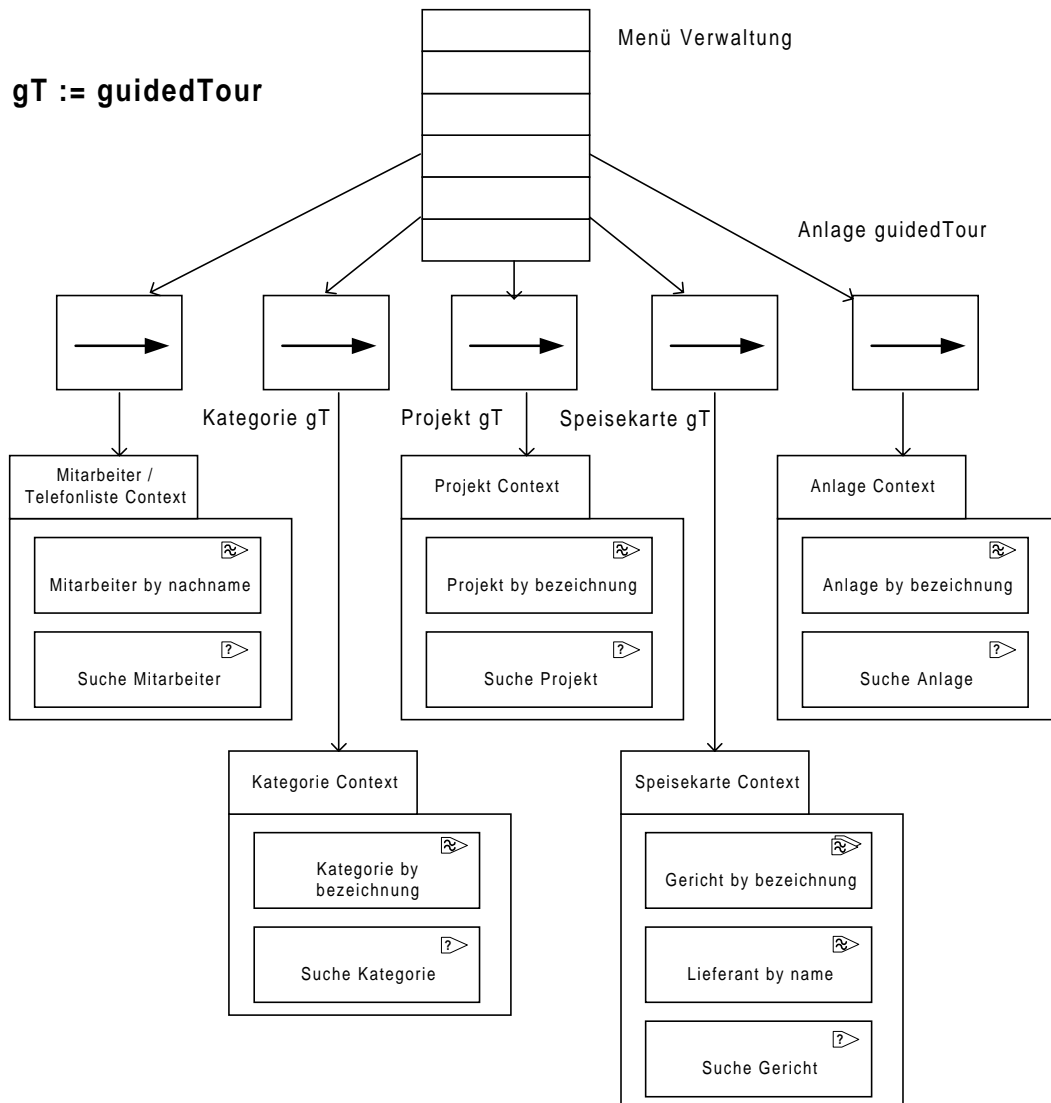


Abbildung 5-6

UML-Objektdiagramm Navigationsstruktur Menü Verwaltung

Eine weitere Zugriffsstruktur, die über das Menü Verwaltung erreichbar ist, ist die Guided Tour „Speisekarte guidedTour“, von welcher zum Paket „Speisekarte Context“ navigiert werden kann. Dieses Paket enthält den gruppierten Kontext „Gericht by bezeichnung“ und den einfachen Navigationskontext „Lieferant by Speisekarte“, sowie den gefilterten Kontext „Suche Gericht“. „Gericht by bezeichnung“ beschreibt eine Sequenz von Navigationskontexten. Jeder dieser Kontexte enthält die Gerichte einer

Speisekarte. Zu jeder Speisekarte gibt es auch einen Lieferanten. Dem trägt der Navigationskontext „Lieferant by Name“ Rechnung. „Suche Gericht“ liefert einen Index auf alle Gerichte, die das Suchkriterium erfüllen. Wie bei allen vorher beschriebenen Paketen ist auch hier keine Einschränkung des Navigationskontextwechsels definiert.

Zusätzlich kann vom Menü über die Guided Tour „Anlage guidedTour“ zum Navigationskontextpaket „Anlage Context“ navigiert werden. „Anlage guidedTour“ enthält den Navigationskontext „Anlage by Bezeichnung“ und den gefilterten Kontext „Suche Anlage“. Ein Wechsel von einem Navigationskontext zum anderen ist jederzeit möglich.

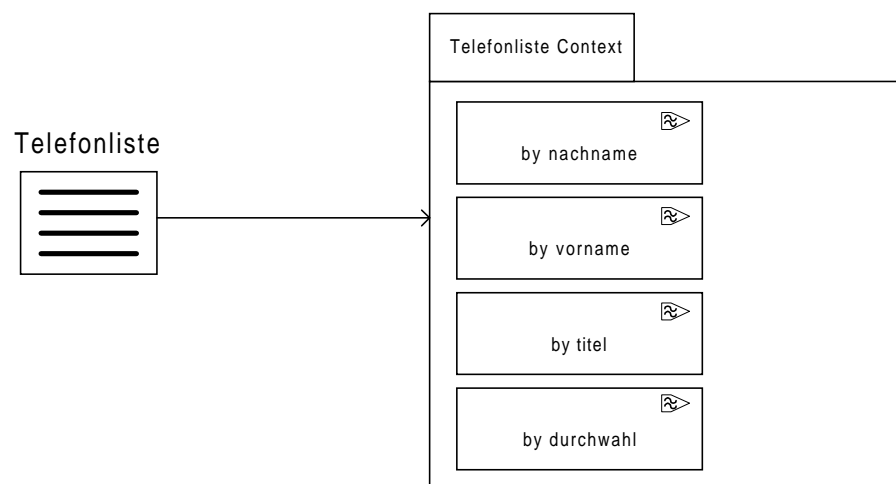


Abbildung 5-7

UML-Objektdiagramm Navigationsstruktur Index Telefonliste

Über den Index „Telefonliste“ [Abbildung 5-7] kann man zum Navigationskontextpaket „Telefonliste Context“ navigieren. Das Paket enthält die Navigationskontexte „by nachname“, „by vorname“, „by titel“ und „by durchwahl“. Jeder dieser Kontexte enthält alle Instanzen der Navigationsklasse Telefonliste, sortiert nach dem angegebenen Attribut. Der Navigationskontext kann innerhalb des Paketes ohne Einschränkung gewechselt werden.

Über den Index „Essenbestellung“ [Abbildung 5-8] wird zum Kontext „Bestellung“ navigiert. Der Navigationskontext „Bestellung“ besteht aus allen Instanzen der Navigationsklasse Bestellung.

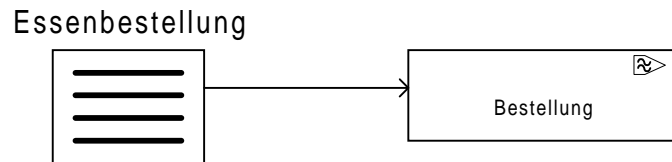


Abbildung 5-8 UML-Objektdiagramm Navigationsstruktur Index Essenbestellung

Der Index „Kategorie“ [Abbildung 5-9] dient zum Zugriff auf die Instanzen der Navigationsklasse Kategorie. Als Beschreibung der Navigationsknoten wird die Kategoriebezeichnung verwendet. Vom Navigationskontext „Kategorie by bezeichnung“ wird über einen Index Information zum gruppierten Navigationskontext „Information by Kategorie“ navigiert. Dieser gruppierte Navigationskontext enthält alle einer Kategorie zugeordneten Informationen.

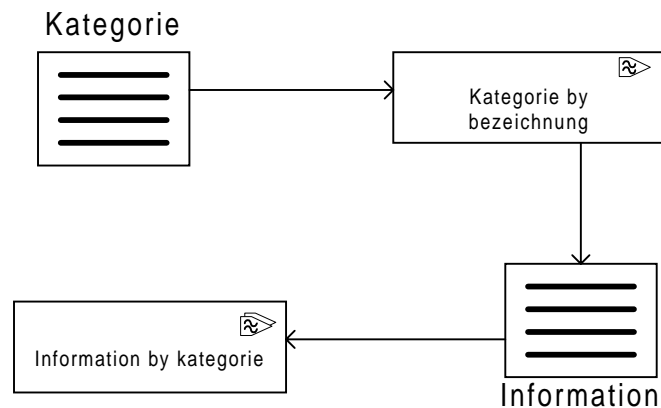


Abbildung 5-9 UML-Objektdiagramm Navigationsstruktur Index Kategorie

bezeichnung“ kann über einen Index Information zum gruppierten Navigationskontext „Information by Kategorie“ navigiert werden. Dieser gruppierte Navigationskontext enthält alle einer Kategorie zugeordneten Informationen.

Für den über das Hauptmenü erreichbaren Index kann zum Navigationskontext „Suchergebnis“ [Abbildung 5-11] navigiert werden. Für „Suchergebnis“ definiere ich hiermit ein zusätzliches Stereotyp, um den Kontext genau zu beschreiben. Dieses

Stereotyp wird als **<<grouped filtered context>>** bezeichnet [Abbildung 5-10]. Ein gruppierter gefilterter Kontext beschreibt eine Sequenz von gefilterten Kontexten. Im vorliegenden Fall besteht dieser Kontext aus der Zusammenfassung der Menge aller Suchergebnisse, die beim Suchen in jeder einzelnen Navigationsklasse entstehen

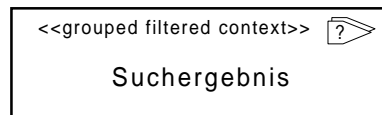


Abbildung 5-10 neues Stereotyp **<<grouped filtered context>>** für das Navigationsstrukturmodell

würden. Konkret sind dies die gefilterten Kontexte „Suche Kategorie“, „Suche Mitarbeiter“, „Suche Anlage“ und „Suche Projekt“, die bereits eingehender beschrieben wurden. Zusätzlich ist noch der gefilterte Kontext „Suche Information“ implizit enthalte. „Suche Information“ liefert einen Index auf alle Navigationsknoten der Klasse Information, die das angegebenen Suchkriterium erfüllen. Der Benutzer hat die Auswahl zwischen all diesen gefilterten Kontexten.

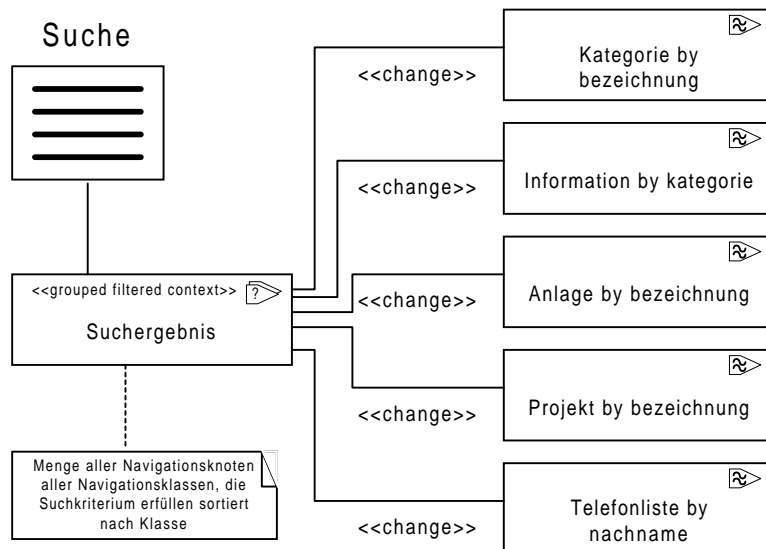


Abbildung 5-11 UML-Objektdiagramm Navigationsstruktur Index Suche

5.3 Entwurf der Präsentation (Presentational Design)

Wenn der Entwurf der Navigationsstruktur steht, gilt es die Benutzerschnittstelle der Webapplikation zu entwerfen. Beim Design der Präsentation geht es darum, zu definieren, wie dem Benutzer die Navigationsstruktur präsentiert wird. Es wird dazu ein Präsentationsmodell erstellt, welches die Bausteine der Benutzerschnittstelle enthält. Bausteine sind dabei die möglichen Inhalte einer Hypertextseite wie Text, Querverweise, Bilder, Audio, Video oder Formulare. Das Präsentationsmodell stellt lediglich ein Schema dar, das beschreibt, welche dieser Elemente auf der Seite vorhanden sein sollen, wie sie angeordnet sind und wie die Größenverhältnisse der Elemente untereinander aussehen. Eine detaillierte Definition, wie die Seite dann letztendlich exakt aussieht, wird dadurch jedoch nicht vorgenommen. Das exakte Erscheinungsbild einer Webseite ist ohnehin vom verwendeten Webbrowser abhängig.

Der Präsentationsentwurf besteht aus einem statischen und einem dynamischen Präsentationsmodell. Für die Darstellung des statischen Präsentationsmodells werden UML-Kompositionsdiagramme benutzt. Diese beschreiben, aus welchen Bausteinen die Benutzerschnittstelle besteht. Das dynamische Präsentationsmodell wird mit UML-Statusdiagrammen dargestellt. Statusdiagramme können verwendet werden, um das Verhalten der Elemente zu beschreiben, beispielsweise, wodurch eine Navigation ausgelöst wird oder welche Veränderungen sich dadurch ergeben.

5.3.1 Statisches Präsentationsmodell

Das statische Präsentationsmodell definiert, wie die Navigationsknoten des Navigationsmodells dem Benutzer angezeigt werden. Es enthält eine Menge von Schnittstellenobjekten, die durch UML-Kompositionsobjekte präsentiert werden. Dadurch kann man die Komposition von Schnittstellenobjekten aus anderen Schnittstellenobjekten darstellen.

Ein Schnittstellenobjekt kann entweder ein einfaches Schnittstellenobjekt wie Text, ein Bild oder ähnliches oder eine Komposition dieser Objekte sein. Abbildung 5-12 zeigt die Stereotypen für die gängigsten Schnittstellenobjekte. Dies sind Text, Bild, Anker

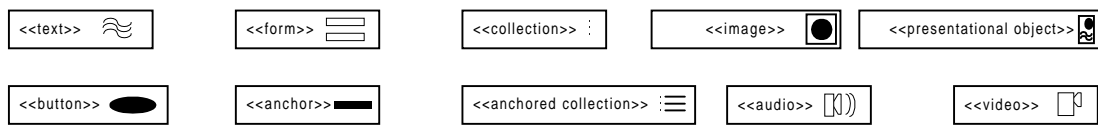


Abbildung 5-12 gängige Schnittstellenobjekte

(anchor), Audio, Button, Video, Kollektion, Kollektion von Anker (anchored collection) und Formular.

Ein Querverweis kann nicht nur ein Textlink sein, sondern auch andere anklickbare Fläche, die den Ausgangspunkt einer Navigation darstellen. Ein Button ist ebenfalls eine anklickbare Fläche, die eine Aktion auslöst. Diese Aktion wird im dynamischen Präsentationsmodell definiert. Formulare dienen dazu, um Eingaben vom Benutzer entgegenzunehmen. Ein Präsentationsobjekt (presentational object) ist aus anderen Schnittstellenobjekten - darunter können auch andere Präsentationsobjekte - zusammengesetzt und ist vom Status des Navigationsknotens abhängig.

5.3.2 Dynamisches Präsentationsmodell

Im dynamischen Präsentationsmodell werden UML-Zusatandsdiagramme benutzt, um die Auswirkungen von Ereignissen, (Events) d.h. Benutzerinteraktionen wie Mausklicks oder Tastatureingaben, auf die Schnittstellenobjekte zu zeigen. Wenn ein Schnittstellenobjekt mit einem Ereignis konfrontiert wird, kann es selbst Ereignisse erzeugen und an andere Schnittstellenobjekte Nachrichten schicken. Die meisten Schnittstellenobjekte haben bereits ein vordefiniertes Verhalten, so daß keine Spezifikationen notwendig sind. Die Ausnahme sind Buttons. Im dynamischen Präsentationsmodell wird festgelegt, was passiert, wenn ein Button geklickt wird.

5.3.3 Beschreibung des entwickelten Präsentationsmodells

Nachfolgend werden die Presentational Objects für die in dieser Diplomarbeit entwickelte Applikation beschrieben. Die Beschreibungen enthalten auch Aussagen

zum dynamischen Verhalten, beispielsweise, welche Aktion das Klicken eines Buttons auslöst.

5.3.3.1 *Hauptmenü und Index Kategorien*

Um eine möglichst einfache und flexible Navigation zu ermöglichen, wird die zur Verfügung stehende Präsentationsfläche in drei Teile aufgeteilt [Abbildung 5-13]. Diese Teile werden Frames genannt. Es gibt einen Frame, in dem das Hauptmenü [Abbildung 5-14] dargestellt wird, welches immer unverändert präsent bleibt. Dieser Frame ist am oberen Rand der Seite angeordnet und erstreckt sich über die ganze zur Verfügung stehende Breite. Das Präsentationsobjekt Hauptmenü enthält das Präsentationsobjekt Suche und die Anker Home, Verwaltung, Essensbestellung und Telefonliste. Außerdem wird mit einer „anchored Collection“ die oberste Hierarchieebene der Kategorien präsentiert.

Am linken Seitenrand existiert ein weiteres Frame. Dieses dient ebenfalls als Ausgangspunkt einer Navigation. Dort wird ein Index Kategorien dargestellt [Abbildung 5-15]. Welche Anker dieser Index enthält, wird über einen im Hauptmenü ausgewählten Anker der „anchored Collection“ Kategorie bestimmt. Der restliche zur Verfügung stehende Platz wird je nach Navigationsknoten vom entsprechenden Präsentationsobjekt ausgefüllt.

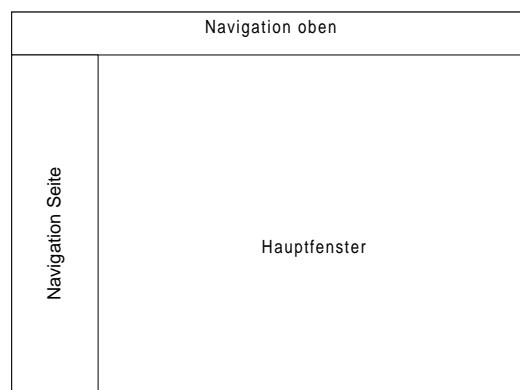


Abbildung 5-13 Layout der Präsentationsfläche

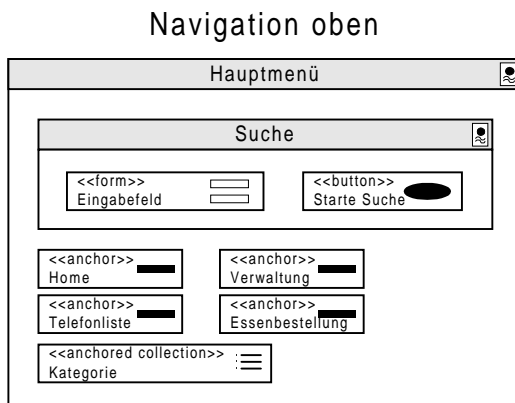


Abbildung 5-14 Präsentationsobjekt
Hauptmenü

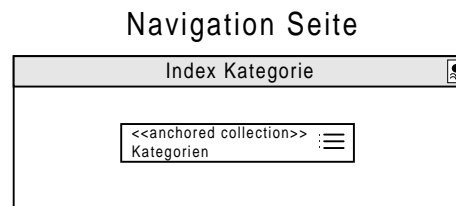


Abbildung 5-15 Präsentationsobjekt
Index Kategorie

5.3.3.2 Verwaltung

Das Präsentationsobjekt Verwaltung repräsentiert die Seite, zu der über das „Menü Verwaltung“ navigiert wird [siehe Abbildung 5-16]. Es besteht aus einem Hinweistext, und den Ankern „Mitarbeiter“, „Projekt“, „Kategorie“ und „Speisekarte“.

Über den Anker Mitarbeiter gelangt man zum Präsentationsobjekt „Mitarbeiter-Verwaltung“. Es besteht aus zwei separaten Präsentationsobjekten „Mitarbeiterauswahl“ und „Suche Mitarbeiter“ sowie dem Anker „neu“. „Mitarbeiterauswahl“ repräsentiert den Navigationskontext „Mitarbeiter by name“, da von allen Mitarbeitern eine Namensliste erstellt wird, die zur Auswahl eines Mitarbeiters dient. besteht aus einem Formular Mitarbeiterauswahl sowie einem Button „bearbeiten“ und einem Button „löschen“. Wird der Button „bearbeiten“ geklickt, kommt das Präsentationsobjekt „Mitarbeiter Formular“ zum Einsatz. Dieses wird auch über den Anker „neu“ in „Mitarbeiter-Verwaltung“ aktiviert. Das Präsentationsobjekt „Mitarbeiter-Formular“ dient zum neu Anlegen oder Bearbeiten von Mitarbeiterdaten. Es enthält Eingabefelder für Vorname, Nachname, Login, Passwort, Titel, Durchwahl, Mobil- und Privattelefonnummer. „Suche Mitarbeiter“ besteht aus einem Eingabefeld und einem Button „Suche starten“.

Dieses Präsentationsobjekt besteht aus den Präsentationsobjekten „Projektauswahl“ und „Suche Projekt“. „Projektauswahl“ repräsentiert den Navigationskontext „Projekt by bezeichnung“ als Liste mit der Möglichkeit zur Auswahl eines Elements. Es besteht aus einem Formular Projektauswahl, einem Button „bearbeiten“, einem Button „löschen“ und einem Anker „neu“. Über „bearbeiten“ und „neu“ wird das Präsentationsobjekt „Projekt-Formular“ aufgerufen. Es enthält Eingabefelder für Projektbezeichnung, Kunde und Projektnummer. Das Präsentationsobjekt „Suche Projekt“ besteht aus einem Eingabefeld und einem Button „Suche starten“.

Mit dem Anker Kategorie erreicht man das Präsentationsobjekt „Kategorieverwaltung“. Es besteht aus den Präsentationsobjekten „Kategorieauswahl“ und „Suche Kategorie“. „Kategorieauswahl“ repräsentiert den Navigationskontext „Kategorie by bezeichnung“ und stellt eine Selektiermöglichkeit zur Auswahl eines Elements zur Verfügung. Es enthält ein Formular Kategorieauswahl, einem Button „bearbeiten“, einem Button „löschen“ und einem Anker „neu“. Das Präsentationsobjekt „Formular Kategorie“ wird über den Button „bearbeiten“ oder den Anker „neu“ aktiviert. Es enthält ein Eingabefeld für die Kategorieverwaltung und Formularelemente zur alternativen Einordnung der Kategorie im Kategoriebaum. Das Präsentationsobjekt „Suche Kategorie“ besteht aus einem Eingabefeld und einem Button „Start Suche“.

Der Anker Projekt dient zur Navigation zum Präsentationsobjekt „Projektverwaltung“. Das Präsentationsobjekt „Speisekarten-Verwaltung“ wird über den Anker Speisekarte erreicht. Es enthält ein Formular Speisekartenauswahl und Button „bearbeiten“, einen Button „löschen“ und einen Anker „neu“, sowie ein Formular Lieferantenauswahl und ebenfalls die Buttons „bearbeiten“ und „löschen“ sowie einen Anker „neu“. Über den Anker „neu“ wird ein Präsentationsobjekt „Formular Speisekarte“ bzw. „Formular Lieferant“ aufgerufen, daß je nach Auswahl Eingabefelder für die Lieferantendaten oder die Speisekartendaten enthält. Das Präsentationsobjekt „Formular Speisekarte“ enthält zusätzlich das Präsentationsobjekt „Gerichtsauswahl“ mit einem Button „neues Gericht“ sowie einer Auswahl Gericht und Buttons „Gericht bearbeiten“ und „Gericht löschen“. Für die Bearbeitung von Gerichten dient das Präsentationsobjekt „Formular Gericht“.

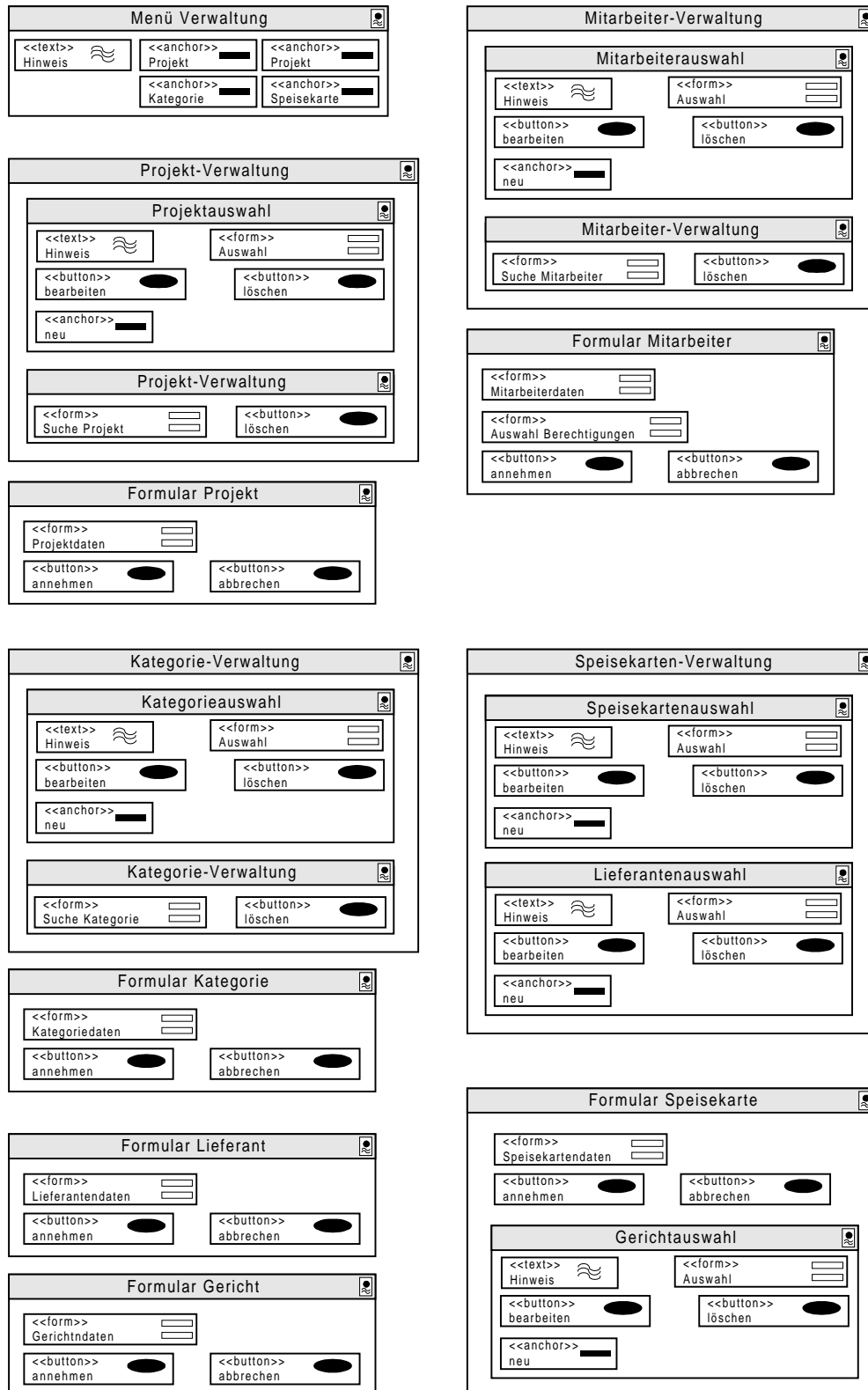


Abbildung 5-16

Präsentationsobjekte aus dem Bereich Verwaltung

Die beschriebenen Präsentationsobjekte sind auf der vorhergehenden Seite abgebildet [Abbildung 5-16].

5.3.3.3 Telefonliste

Das Präsentationsobjekt „Telefonliste“ [Abbildung 5-17] enthält neben einem Text „Hinweis“ die Präsentationsobjekte „Telefonliste by nachname“, „Telefonliste by vorname“, „Telefonliste by titel“, „Telefonliste by durchwahl“, sowie die Buttons „Vorname“, „Nachname“, „Titel“, „Durchwahl“. Es wird nur immer eines der in „Telefonliste“ enthaltenen Präsentationsobjekte angezeigt. Mit den Buttons wird das angezeigte Präsentationsobjekt ausgewählt. Standardmäßig wird das Präsentationsobjekt

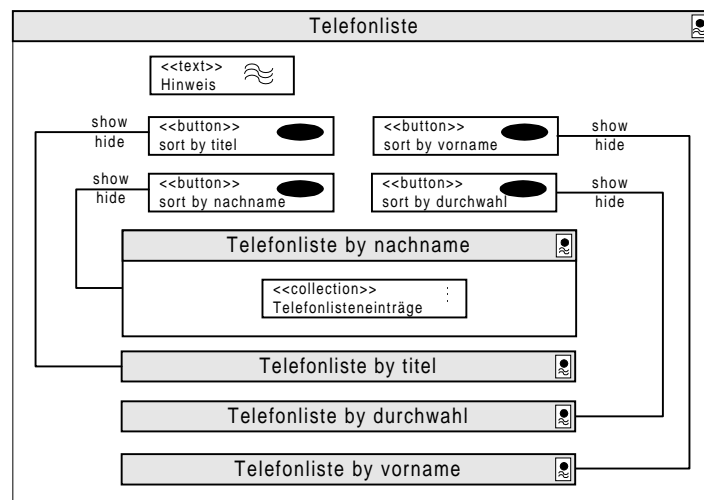


Abbildung 5-17 Präsentationsobjekt Telefonliste

„Telefonliste by nachname“ angezeigt. Die in „Telefonliste“ enthaltenen Präsentationsobjekte haben alle den gleichen Aufbau, nämlich eine Kollektion von Telefonlisteneinträgen. Deshalb wird in der folgenden Abbildung nur das Präsentationsobjekt „Telefonliste by nachname“ exakt abgebildet.

5.3.3.4 Bestellung

Über die Guided Tour „Essenbestellung“ kommt man zum Präsentationsobjekt „Bestellung“ [Abbildung 5-18]. Es enthält ein Formular zur Auswahl einer Speisekarte und eine Liste von Gerichten, von denen eines oder mehrere ausgewählt werden kann. Weiterhin gibt es noch einen Button „abschicken“.

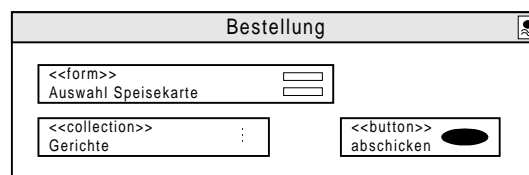


Abbildung 5-18 Präsentationsobjekt Bestellung

5.3.3.5 Kategorie

Wird über die Navigationsmöglichkeit auf der linken Seite der Webseite eine Kategorie angewählt, so erscheint im Hauptfenster das Präsentationsobjekt „Kategorie“ [Abbildung 5-19]. Es enthält den Namen der Kategorie als Text und eine anchored Collection aller in dieser Kategorie enthaltenen Informationen. Daneben gibt es noch die Buttons „bearbeiten“, „löschen“ und „unterkategorie anlegen“, sowie „neue Information“ zum Anlegen einer neuen Information in dieser Kategorie.

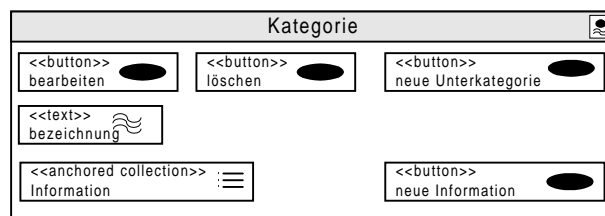


Abbildung 5-19 Präsentationsobjekt Kategorie

5.3.3.6 Information

Wird über die anchored Collection in „Kategorie“ eine Information ausgewählt, so wird diese mit dem Präsentationsobjekt „Information“ [Abbildung 5-20] dargestellt. Neben Textblöcken für Titel, Zusammenfassung, Text, Autor, Erstellungsdatum und Kategorie gibt es Buttons zum „löschen“, „bearbeiten“ oder „neu anlegen“ von Informationen.

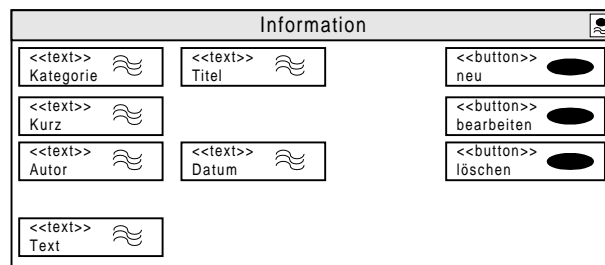


Abbildung 5-20 Präsentationsobjekt Information

5.3.3.7 Suche

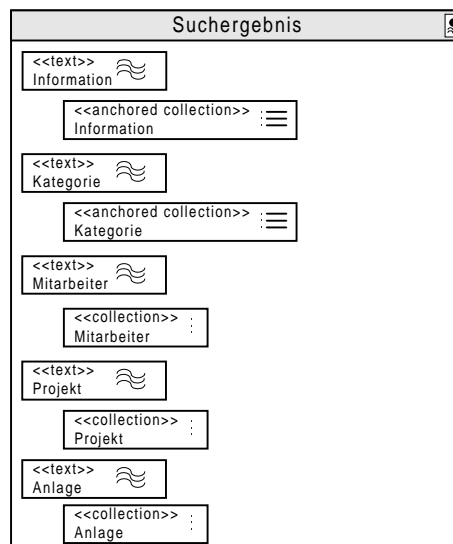


Abbildung 5-21 Präsentationsobjekt Suchergebnis

Das Suchergebnis der im Hauptnavigationfenster gestarteten Suche wird mit dem Präsentationsobjekt „Suchergebnis“ [Abbildung 5-21] repräsentiert. Es enthält Textbausteine für jede der durchsuchten Klassen Information, Kategorie, Telefonliste, Anlage und Projekt sowie jeweils eine Collection, die die Suchergebnisse auflistet. Bei Information und Kategorie handelt es sich um eine anchored Collection.

6 Webbasierte Informationssysteme

6.1 Überblick

Während früher eher der Informationsmangel als ein Problem galt, ist es heute das Filtern der relevanten Daten aus der Menge der zur Verfügung stehenden Informationen. In jedem Fall ist es mühsam, zu einem bestimmten Thema Informationen zu beschaffen. Dazu muß man beispielsweise in Bibliotheken die vorhandene Literatur zu dem Thema sichten. Das nimmt je nach Umfang der Literatur beziehungsweise des Themengebiets einige Zeit in Anspruch. Möglicherweise ist das entscheidende Buch oder der Information gerade ausgeliehen und somit erst nach Ablauf der Leihfrist wieder verfügbar. Die Entwicklung des Internets vom Informationskanal amerikanischer Militär- und Forschungseinrichtungen zum allgemein zugänglichen Medium hat dies grundlegend verändert. Über Suchmaschinen bekommt man innerhalb kürzester Zeit Verweise auf Quellen zu einem Thema. Eventuell ist jedoch ein Großteil davon unbrauchbar oder beschäftigt sich mit dem Thema nur am Rande. Ein weiteres Problem ist, daß die Webseiten, auf die verwiesen wird, zum Teil nicht mehr bestehen oder auf Grund technischer Probleme nicht darauf zugegriffen werden kann.

Weil also die Beschaffung für die eigenen Zwecke brauchbarer Informationen zeitaufwendig ist, ist man bemüht, Wissen, das einmal erarbeitet wurde, auch weiterhin nutzen zu können. Deshalb muß dieses Wissen gesammelt und adäquat verwaltet und präsentiert werden. Nur dann können die Benutzer der Informationssammlung die gesammelten Informationen auch effizient nutzen, das heißt, sie können auf die gesuchte Information schnell und einfach zugreifen.

In Firmen werden deshalb schon seit längerer Zeit sogenannte Groupware-Systeme eingesetzt. (Die im folgenden getroffenen Aussagen gelten für klassische Groupware-

Systeme. Da neuere Groupware-Systeme, bzw. neuere Versionen immer weborientierter werden, gelten einige Einschränkungen nicht mehr oder nur noch zum Teil.) Klassische Groupware-Systeme ermöglichen es den Mitarbeitern, auf einen Stamm von Firmen-Know-How zuzugreifen. Diese Informationen werden von den Benutzern oder eigens mit dieser Aufgabe betrauten Personen aktualisiert. Dies kann das Hinzufügen von neuen Informationen, das Ändern oder auch Löschen von vorhandenen Informationen bedeuten. Groupware-Systeme haben im allgemeinen noch andere Funktionalitäten, wie zum Beispiel Terminkalender oder E-Mail.

Solche Systeme haben üblicherweise folgenden zweischichtigen Aufbau:

Es gibt einen Server, auf dem eine Datenbankapplikation läuft, welche die Informationen speichert, verwaltet und die Zugriffe darauf steuert. Am EDV-Arbeitsplatz des Benutzers, dem Client, muß außerdem ein Programm installiert sein, das mit dieser Applikation zusammenarbeitet.

Grundlage für die Nutzung von Groupware-Systemen ist die Vernetzung der EDV-Arbeitsplätze innerhalb des Unternehmens. Das Groupware-System kann von den Mitarbeitern genutzt werden, solange sie sich innerhalb des Unternehmens an ihrem Arbeitsplatzrechner befinden. Von zu Hause oder beispielsweise bei einem Kundentermin kann man nicht auf die Informationen zugreifen. Die Software setzt bestimmte Betriebssysteme voraus, beispielsweise Sun Solaris, Novell Netware oder Microsoft Windows NT als Serverbetriebssystem oder neben den genannten auch die gängigen Betriebssysteme Microsoft Windows 3.1/95/98 oder Apple Mac OS auf der Clientseite. Das Unternehmen muß sich also auf bestimmte Betriebssysteme festlegen. Ein Wechsel des Betriebssystems ist danach mit sehr hohem Kosten- und Administrationsaufwand verbunden.

Die Bindung an bestimmte Betriebssysteme wird in vielen Unternehmen heute als Manko empfunden. Außerdem erzeugt die Administration von Clientprogrammen auf allen Arbeitsplatzrechnern Administrationsaufwand und somit Kosten. Hinzu kommen Probleme mit Versionsverwaltung und mehr oder minder guter Unterstützung vorhandener Rechnerinfrastruktur.

Deshalb wird nach Lösungen gesucht, die den Mitarbeitern einerseits den Zugriff auf die für ihn relevanten Informationen ermöglicht, andererseits aber nicht an eine bestimmten Betriebssystem- oder Rechnerplattformen gebunden ist, sondern von beliebigen Plattformen genutzt werden kann. Des weiteren ist auch der Zugriff auf die Informationen von außerhalb des Betriebes gefordert, denn es gibt immer mehr Arbeitsverhältnisse, die nicht in der Betriebsstätte, sondern beispielsweise zu Hause ausgeübt werden.

Informationssysteme auf Basis des World Wide Web, im folgenden webbasiert genannt, erfüllen diese Anforderungen. Auf die Informationen im World Wide Web kann von nahezu überall auf der Welt rund um die Uhr zugegriffen werden. Allerdings kann der Anbieter der Informationen den Benutzerkreis auch einschränken, indem sich Benutzer beispielsweise identifizieren müssen und nicht identifizierten Personen so der Zugang verwehrt wird. Voraussetzungen beim Anwender für die Benutzung eines webbasierten Informationssystems sind lediglich ein Telefonanschluß sowie ein Computer, auf dem ein Webbrowser installiert ist. Auch ein Webbrowser ist ein Clientprogramm. Der Vorteil von Groupware-Systemen gegenüber Clients ist, daß der Browser auch für andere Zwecke als den Einsatz in Verbindung mit einem webbasierten Informationssystem genutzt werden kann und nicht ein bestimmtes Produkt eines bestimmten Herstellers verwendet werden muß. Für Webbrowser gibt es Plug-Ins, die den Standard-Funktionsumfang erweitern. Plug-Ins können nötig sein, um bestimmte Inhalte, beispielsweise Shockwave Flash™ oder RealAudio™ im Webbrowser abspielen zu können.

Auch webbasierte Informationssysteme benutzen zum Speichern der Informationen Datenbanken. Datenbank stellen Schnittstellen bereit, über die Daten mit anderen Programmen ausgetauscht werden können. Diese Programme bereiten die Informationen aus der Datenbank so auf, daß sie von einem Webbrowser interpretiert werden können. Die Informationen werden dann durch einen Webserver öffentlich bereitgestellt. Öffentlich kann in diesem Zusammenhang bedeuten, daß die Informationen wirklich der Allgemeinheit zur Verfügung stehen, oder aber nur ein kleiner Personenkreis Nutzungsberechtigungen hat. Von einem Intranet spricht man,

wenn ein Unternehmen nur seinen Mitarbeitern bestimmte Informationsdienste zur Verfügung stellt. Dabei kann ein Intranet auf mehrere Betriebsstandorte verteilt sein und Internet-Techniken benutzen oder auch völlig von öffentlichen Netzen entkoppelt sein.

Für die Firma Medialab ist eine weitgehende Unabhängigkeit von Server- und Clientbetriebssystemen von Interesse. So kann man zukünftige Hardware-Anschaffungen allein anhand der Bedürfnisse planen und muß nicht auf die vorhandene Infrastruktur Rücksicht nehmen, um mit dem neuen Gerät auch das Informationssystem nutzen zu können. Außerdem wird ein ort- und zeitunabhängiger Zugriff auf das Firmenwissen benötigt. Eine webbasierte Lösung erfüllt diese Kriterien.

6.2 Bestandteile eines webbasierten Informationssystems

Webbasierte Informationssysteme bestehen aus vier Schichten. Zum ersten ein Webbrowser, dem Clientprogramm des Benutzers. Zum zweiten ein Webserver, der öffentlichen Zugriff auf die Informationen ermöglicht. Drittens eine Applikation, die Benutzeranfragen vom Webserver entgegennimmt, diese verarbeitet und ein Ergebnis zurücksendet. Und viertens eine Datenbank, in der die Informationen dauerhaft gespeichert werden, bis sie explizit wieder gelöscht oder durch eine Aktualisierung überschrieben werden. Wird der benötigte Webbrowser als eigenstän

6.2.1 Webserver und Webbrowser

Das World Wide Web (WWW) ist nur einer von vielen Informationsdiensten im Internet. Es ermöglicht den Zugriff auf Dokumente im Internet auf leichte und verständliche Weise. WWW-Dokumente werden in der Sprache HTML (HyperText Markup Language) geschrieben und bestehen im wesentlichen aus ASCII-Text mit Formatierungsanweisungen (Tags). Über sogenannte Hyperlinks kann auf andere HTML-Dokumente (auch: Webseiten) oder andere Internetdienste wie das File Transfer Protocol verwiesen werden. Webbrowser nehmen HTML-Dokumente entgegen,

interpretieren die logischen Formatierungen (HTML-Tags) und stellen die Dokumente grafisch dar. Webbrowser gibt es von verschiedenen Herstellern für nahezu jedes Betriebssystem.

Ein Webserver ist ein Programm, das auf Anfragen wartet, die an ihn gerichtet und gemäß des HTTP-Protokolls (HyperText Transfer Protocol) codiert sind. Kommt eine Anfrage an, so prüft der Server, ob der anfordernde Browser dazu berechtigt ist, Dokumente von diesem Server zu beziehen und sendet eine ebenfalls im HTTP-Format codierte Antwort an den Browser zurück. Im Regelfall werden auf Anforderung hin HTML-Dokumente an Webbrowser übermittelt, es kann sich aber auch um andere Text-, Bild-, Audiodaten oder sonstige Binärdateien handeln. Webserver haben außerdem die Fähigkeit, spezielle Programme auszuführen, um so eine Verbindung zu anderen Informationsquellen auf einem lokalen System oder im Internet herzustellen. Für die meisten Betriebssysteme sind Webserver verfügbar.

6.2.2 Webapplikation

Bei einer Webapplikation handelt es sich um ein meist aus mehreren Teilkomponenten bestehendes Programm, das vom Webserver ausgeführt werden kann. Um ein webbasiertes Informationssystem zu realisieren, wird eine Applikation benötigt, die Anfragen vom Webserver entgegennimmt, die vom Webbrowser gesandte Anfrage verarbeitet, gegebenenfalls Informationen aus einer Datenbank anfordert oder Informationen in dieser Datenbank verändert und das Ergebnis an den Webserver zurückgibt, der es wiederum an den Webbrowser weitergibt.

Eine solche Applikation kann im Prinzip in jeder beliebigen Programmiersprache geschrieben werden. Allerdings eignen sich die Sprachen am besten, die wie Java oder Perl bereits vorgefertigte Module für die Verarbeitung von Webapplikationen bereitstellen.

6.2.3 Datenbank

Eine Datenbank ist ein Programm, mit dem große Mengen von Informationen effizient gespeichert werden. Datenbanken stellen Funktionen zur Verwaltung dieser Informationen bereit und verwenden dazu meist eine eigene Programmiersprache (Anfragesprache). Es gibt mehrere Datenbankkonzepte. Von diesen werden zwei Datenbanktypen betrachtet: Relationale und objektorientierte Datenbanken.

Relationale Datenbanken wurden entwickelt, um einfache, elementare Daten wie Text und Zahlen zu speichern. Diese Informationen werden in Tabellen abgelegt. Beziehungen (Relationen) zwischen verschiedenen Tabellen werden über zusätzliche Hilfstabellen modelliert. Um beispielsweise ein Bankkonto zu speichern, würde man eine Tabelle anlegen und darin für Vorname, Nachname und die weiteren spezifischen Eigenschaften des Kontoinhabers Spalten anlegen. Um Kontobewegungen dieses Kontos zu speichern, benötigt man eine weitere Tabelle, in denen die Beträge und das Ausführungsdatum und Empfänger- bzw. Absenderdaten festgehalten sind. Die einzelnen Datensätze, also die einzelnen Transaktionen werden über eine dritte Tabelle dem jeweiligen Bankkonto zugeordnet. Für relationale Datenbanken gibt es eine standardisierte Anfragesprache namens SQL (Standard Query Language).

In objektorientierten Datenbanken werden die zu erfassenden Informationen als Objekte abgelegt. Sie eignen sich besonders für komplexe Datenstrukturen. Beziehungen zwischen Objekten werden in diesen Objekten selbst angelegt. Man würde also ein Objekt Bankkonto anlegen, das Vornamen, Nachnamen usw. des Kontoinhabers enthält. Neben diesen Bestandteilen würde es auch noch eine Liste der Kontobewegungen enthalten. Der Umweg über Hilfsobjekte oder ähnliches ist also nicht nötig. Leider gibt es für die am Markt befindlichen Produkte noch keine standardisierte Anfragesprache.

Datenbanken bieten Schnittstellen zu Programmiersprachen, sogenannte API's (application programming interface), damit Applikationen entwickelt werden können, die mit der Datenbank zusammenarbeiten und direkt auf die gespeicherten Daten zugreifen können. Diese Schnittstellen sind entweder schon fest in der Datenbank verankert oder als Zusatzmodul erhältlich, das erst eingebunden werden muß. Bekannte

Schnittstellen für sich nach außen hin als relational präsentierende Datenbanken sind JDBC (Java Database Connection) für Java, DBI (Database Interface) für Perl oder ODBC (Open Database Connectivity) .

6.3 Vergleich möglicher Lösungen

Für jede der drei Schichten stehen diverse Produkte zur Auswahl, so daß sich eine Vielzahl von Möglichkeiten zur Realisierung eines webbasierten Informationssystems ergibt. Von diesen Möglichkeiten werden im folgenden vier vorgestellt.

6.3.1 Lotus Notes / Domino

Bei Lotus Notes / Domino (Release 5) handelt es sich um eine Weiterentwicklung eines klassischen Groupware-Systems. Die Software besteht aus dem Webserver, Lotus Domino und der Groupware Lotus Notes, der eine relationale Datenbank zugrunde liegt. Lotus Notes stellt verschiedene Möglichkeiten zur Dokumentverwaltung zur Verfügung, die in Verbindung mit dem Domino Webserver auf einfache Weise im World Wide Web zur Verfügung gestellt werden können. Für Standardanforderungen besteht bereits eine Auswahl an vorgefertigten Lösungen. Hat man speziellere Wünsche in Bezug auf visuelle Repräsentation oder Aufbereitung der Informationen, so müssen diese selbst entwickelt, das heißt programmiert werden. Da Lotus Notes intern relational, also mit Tabellen arbeitet, muß eine Zwischenschicht programmiert werden, welche die Datenbanklogik auf die Objekte der Applikation überträgt.

Lotus Notes stellt Daten über eine JDBC-API externen Anwendungen zur Verfügung. Applikationen können außerdem mit der Lotus eigenen Skriptsprache erstellt werden. Für die Entwicklung von Applikationen kann die mitgelieferte Entwicklungsumgebung Lotus Domino Developer verwendet werden.

Der Lotus Notes / Domino Server ist für die Betriebssysteme Microsoft Windows NT 4.0 und Sun Solaris 2.5.1 erhältlich, eine Linux-Version ist geplant. Das Programmpaket kostet im Moment 3.860 DM. Zusätzliche Funktionen wie E-Mail oder Terminkalender stehen bei Verwendung eines Lotus Notes Clients zur Verfügung. Für

jeden Arbeitsplatz muß dann eine Lizenz erworben werden, allerdings werden im Moment nur die Betriebssysteme Windows NT 4.0/ 95 / 98 und IBM OS2/Warp unterstützt.

6.3.2 Jasmine

Jasmine ist ein objektorientiertes Datenbank-Management-System (OODBMS), das Computer Associates und Fujitsu zusammen entwickelt haben und vertreiben. Es verwendet als interne Anfragesprache die Sprache ODQL (Object Database Query Language). Daneben gibt es aber auch Klassenbibliotheken, die SQL unterstützen und somit den Zugriff über Standard-Schnittstellen wie JDBC oder ODBC ermöglichen.

Die mitgelieferte Entwicklungsumgebung JADS (Jasmine Application Development System) unterstützt ODQL. Applikationen, die in JADS entwickelt wurden setzen beim Client ein Browser-Plugin (Modul, das die Funktionen des Browsers erweitert) voraus, das im Paket enthalten ist. Die für die Verwendung in der Datenbank mit ODQL entwickelten Objekte müssen für die Verwendung in einer Webapplikation in der entsprechenden Programmiersprache nachmodelliert werden. Da bei der Verwendung von JDBC oder ODBC SQL-Anfragen zum Einsatz kommen, können die Stärken der internen Objektorientierung nicht besonders nutzbringend eingesetzt werden. Gegenüber einer relationalen Datenbank ist der Programmieraufwand jedoch etwas geringer.

Zum Lieferumfang gehört auch das Programm WebLink, welches die Präsentation der Daten im WWW unterstützt. WebLink ist kein Webserver, sondern ermöglicht lediglich die Repräsentation von Standard-Datenbankabfragen im HTML-Format. Als Webserver kann jeder für Windows NT 4.0 verfügbare Webserver verwendet werden.

Jasmine setzt als Betriebssystem Microsoft Windows NT 4.0 voraus, des weiteren muß auf dem Server Microsoft Visual C++ 4.2 oder eine neuere Version installiert sein. Das Paket kostet ca. 10.000 DM, dazu kommen die Kosten für Windows NT Server und Visual C++ in Höhe von ca. 2.500 DM.

6.3.3 Gemstone

Auch Gemstone ist ein objektorientiertes Datenbanksystem, das von Gemstone Systems angeboten wird. Es gibt eine auf Java und eine auf Smalltalk basierende Version. Je nach Version wird als Anfragesprache Java oder Smalltalk verwendet. Objekte können also direkt objektorientiert modelliert und implementiert werden und ohne Umweg oder zusätzlichen Aufwand in den Applikationen verwendet werden.

Die Entwicklungsumgebung GemBuilder für Smalltalk muß extra erworben werden, ebenso die Programmiersprache VisualWorks Smalltalk selbst. Diese Kombination wird ergänzt durch den VisualWave Developer, eine Entwicklungsumgebung für Webapplikationen in Smalltalk und den dazu passenden Webserver VisualWave. Der Vorteil dieser Lösung ist, daß die Objekte der Datenbank in der Webapplikation weiterverwendet werden können. Der Programmieraufwand gegenüber den beiden bisher vorgestellten Lösungen ist deswegen wesentlich geringer.

Gemstone und Gembuilder sind für die Betriebssysteme Sun Solaris, Windows NT, HP-UX, AIX und neuerdings auch für Linux erhältlich, VisualWorks und Visual Wave zusätzlich noch für Windows 95/98 und Mac OS. Die Preise der einzelnen Produkte sind für alle Plattformen gleich. Die Datenbank selbst wird für 6.500 US \$ angeboten. Außerdem müssen noch Benutzerlizenzen erworben werden. Dabei gibt es zwei Lizenzmodelle. Entweder man kauft für eine feste Anzahl von Benutzern je eine Lizenz (Named User License) oder man kauft Lizenzen für eine Anzahl von gleichzeitig eingeloggten Benutzern (Concurrent User License). Gemstone empfiehlt eine Concurrent User License für zwei bis vier potentielle Benutzer. Der Preis für eine Named User License beträgt 1.560 US \$, eine Concurrent User License schlägt mit 2.340 US \$ zu Buche. Die Entwicklungsumgebung Visual Works Smalltalk beläuft sich auf 3.420 EURO, Visual Wave und Visual Wave Developer kosten 5.670 und 2.250 EURO.

6.3.4 MySQL

MySQL ist ein relationales Datenbanksystem von der schwedischen Softwarefirma T.c.X Consult und steht stellvertretend für alle alternativen relationalen Datenbanksysteme. Es stehen die Plattformen Sun Solaris, Linux und Windows 95 / 98 / NT 4.0 zur Verfügung. Für den Datenbankserver gibt es im Gegensatz zu den meisten anderen Datenbanksystemen keine grafische Bedienungsoberfläche. Diese ist jedoch entbehrlich, da Datenbanktransaktionen im laufenden Betrieb von der Webapplikation durchgeführt werden. Es gibt eine JDBC-, eine ODBC- und eine DBI-Schnittstelle.

Wie bei Lotus Notes muß die Datenbanklogik durch eine zusätzliche Schicht in der Webapplikation auf die Objekte der Webapplikation übertragen werden, was gegenüber objektorientierten Datenbanken einen höheren Programmieraufwand erfordert.

Als Webserver empfiehlt sich der Apache Web Server der Apache Group. Für ihn gibt es spezielle Java- und Perl-Erweiterungen, die die Ausführungsgeschwindigkeit von in Java oder Perl entwickelten Webapplikationen erhöhen.

Wählt man als Betriebssystem Linux, so bekommt man für alle bei dieser Lösung zum Einsatz kommenden Komponenten gratis zum Download aus dem WWW oder im Lieferumfang einer Linux-Distribution für etwa 100 DM.

6.3.5 Zusammenfassung und Bewertung

Von den vorgestellten Lösungen bietet Gemstone die besten Voraussetzungen für die Realisierung eines objektorientierten webbasierten Informationssystems. Der objektorientierte Entwurf kann durchgängig in allen Schichten der Applikation verwendet werden und die Objekte und Methoden müssen nur einmal implementiert werden. Die Gesamtkosten für diese Lösung sind jedoch sehr hoch, insbesondere wenn man die Kosten der benötigten Lizenzen für ein Unternehmen mit ca. 40 Mitarbeitern wie Medialab mit einbezieht. Der Kostenaufwand liegt dann bei etwa 40.000 bis 60.000 DM, je nach verwendetem Lizenzmodell.

Jasmine bietet ebenfalls gute Objektorientierung, es müssen jedoch für die Datenbank und für die Webapplikation separate Objekte und deren Methoden entwickelt werden. Die Verwendung der JDBC-Schnittstelle ermöglicht zwar einerseits den Einsatz von Standardklassen, erzwingt aber andererseits die Verwendung von SQL-Statements. So werden die Vorteile der internen Objektorientierung teilweise eingebüßt, da SQL nicht objektorientiert ist sondern der Anwendung eine relationale Datenbank vorspiegelt.

Damit ist man vom Programmieraufwand her schon in Reichweite der Lösung mit MySQL, die aber einen erheblichen Preisvorteil aufweist. Die Objektorientierung kann hier zwar nur in der Applikation umgesetzt werden, andererseits kommen maßgeblich die gleichen Klassen zum Einsatz, wie es auch bei der Jasmine-Lösung der Fall wäre.

Ähnlicher Programmieraufwand besteht beim Einsatz von Lotus Notes / Domino, auch wenn bereits Unterstützung für die Publikation von Datenbankinhalten im WWW gegeben ist. Die zusätzlichen nützlichen Funktionen von Lotus Notes können nur bei Einsatz von Lotus Notes Clients benutzt werden, nicht aber über einen Webbrowser. Dies soll sich aber in der neuen Version 5 ändern. Die Systemanforderungen sind gegenüber MySQL erheblich höher, wodurch sich Performancenachteile ergeben.

7 Implementierung

7.1 Einführung

In der Implementierungsphase werden die in der Analyse- und Designphase erarbeiteten Konzepte und Modelle mit konkreten Komponenten umgesetzt. In dieser Diplomarbeit sollte ein Prototyp eines Intranets geschaffen werden, das der Firma Medialab als Informationssystem dient.

Vor Beginn der Implementierung mußte deshalb über die zu verwendenden Komponenten entschieden werden. Folgende Komponenten wurden für die Realisierung des Projekts benötigt:

- ◆ Serverbetriebssystem
- ◆ Webserver
- ◆ Datenbankmanagementsystem
- ◆ Programmiersprache

Die Entscheidungen, welche Produkte und Techniken man wählt, hätte bereits während der Designphase gefällt werden können. Es mußte jedoch ein Kompromiß zwischen den Ansprüchen des Lehrstuhls und den finanziellen Interessen von Medialab getroffen werden. Aus diesem Grund wurde die Entscheidung bis unmittelbar vor den Beginn der Implementierungsphase hinausgezögert. Um den finanziellen Interessen von Medialab zu entsprechen, wurden kostenlose oder preiswerte Produkte als Datenbank, Webserver und Serverbetriebssystem gewählt. Die Entscheidung fiel auf Linux als Betriebssystem, das relationale Datenbankmanagementsystem MySQL und den Apache Webserver. Als Programmiersprache wurde Java gewählt, um den Ansprüchen des Lehrstuhls zu

genügen. Die Applikation läuft als Servlet auf dem Web- und Datenbankserver. Das Servlet befindet sich permanent im Hauptspeicher des Webservers, solange dieser nicht heruntergefahren wird.

7.2 Implementierung des konzeptionellen Designs

Das konzeptionelle Design definiert die einzelnen Klassen und ihre Relationen zueinander. Diese Klassen und Relationen dienen als Basis der Implementierung. Die Klassen mußten auf zwei Ebenen, einmal in der relationalen Datenbank und einmal als Bestandteil der Applikation in Java nachmodelliert werden.

7.2.1 Datenbankschema

Um die Klassen des konzeptionellen Entwurfs auf eine relationale Datenbank abzubilden, werden die Klassen in Tabellen umgewandelt. Für jedes Attribut einer Klasse wird eine Spalte definiert. Die Daten eines Objekts werden in eine Zeile einer Tabelle übertragen. Man nennt eine Zeile in einer Datenbanktabelle Datensatz. Zusätzlich zu den in den konzeptionellen Klassen definierten Attributen benötigt man in einer relationalen Datenbank sogenannte primäre Schlüssel und Fremdschlüssel. Jede Datenbanktabelle hat einen eigenen primären Schlüssel. Ein primärer Schlüssel dient dazu, einen Datensatz eindeutig zu identifizieren. Der primäre Schlüssel kann sich dabei auf eine oder mehrere Spalten einer Tabelle erstrecken. Ein Fremdschlüssel repräsentiert eine Beziehung zwischen zwei Tabellen und ist von einem primären Schlüssel einer anderen Tabelle abgeleitet. Die Menge der erstellten Tabellen stellt die Datenbank dar.

Ziel beim Entwurf einer Datenbank ist es, möglichst keine redundanten Daten in der Datenbank zu haben. Unter Redundanz versteht man das zwei- oder mehrmalige Vorkommen eines Nichtschlüssels in der Datenbank. Beispielsweise der Name eines Mitarbeiters, der einmal in der Mitarbeitertabelle und einmal in der Telefontabelle auftauchen würde. In diesem Fall wäre man gezwungen, beim Verändern, Löschen oder Erzeugen eines Namens die Daten in mehreren Tabellen zu verändern und zu

gewährleisten, daß der Wert überall in der Datenbank gleich ist. Wird bei einer Transaktion, das heißt beim Verändern von Datenbankinhalten, eine Tabelle vergessen, in der ein redundantes Attribut vorhanden ist, oder schlägt eine solche Transaktion fehl, kann es passieren, daß in verschiedenen Tabellen der Datenbank verschiedene Werte für ein und dasselbe Attribut vorhanden sind. In diesem Fall sind die Daten inkonsistent, d.h. widersprüchlich.

Aus diesem Grund versucht man, die Tabellen zu normalisieren. Unter Normalisierung werden Maßnahmen verstanden, um die Redundanz von Daten einzuschränken, bzw. völlig auszuschalten. Es gibt fünf Stufen der Normalisierung, die aufeinander aufbauen. Ist eine Datenbank bezüglich den Normalisierungsrichtlinien einer dieser Stufen angepaßt, so sagt man, die Datenbank ist in n-ter Normalform, wobei n die Stufe bezeichnet.

Unter erster Normalform versteht man, daß in jeder Spalte einer Tabellenzeile genau ein atomarer Wert vorhanden ist, das heißt, daß jedes Feld nur einen Wert aus dem Wertebereich des Attributs aufweist und die Attribute innerhalb einer Tabelle nur ein einziges mal vorkommen. Die zweite Normalisierungsregel schreibt vor, daß jede Spalte, die kein Schlüssel ist, vom ganzen primären Schlüssel abhängen muß. Das bedeutet, daß eine Tabelle keine Spalten enthalten darf, die kein Schlüssel sind und nur von einem Teil des primären Schlüssels abhängen. Die dritte Normalform generalisiert diese Regel. Um der dritten Normalform zu genügen, darf eine Tabelle keine Spalten enthalten, die kein Schlüssel sind und von einer anderen Spalte, die ebenfalls kein Schlüssel ist, abhängen. Wird eine Tabelle in vierter Normalform angelegt, so darf es keine 1:n-Beziehungen zwischen dem primären Schlüssel und einer Spalte, die kein Schlüssel ist, geben. Die fünfte Normalform schreibt vor, daß die Tabellen in die kleinstmöglichen Teile aufgespalten werden, um Redundanzen innerhalb einer Tabelle auszuschließen.

Für diese Diplomarbeit wurde die Datenbank in dritter Normalform entworfen.

Man kann in relationalen Datenbanken 1:1-, 1:n- und m:n-Relationen modellieren. Eins-zu-eins-Beziehungen werden üblicherweise mit einer einzigen Tabelle realisiert. Bei

eins-zu-viele-Beziehungen wird auf der Viele-Seite ein Fremdschlüssel, das heißt der Primärschlüssel der anderen Tabelle, hinzugefügt. Um m:n-Beziehungen zu zwischen zwei Tabellen modellieren, die der dritten Normalform genügen, muß eine Hilfstabelle angelegt werden, die nur die beiden primären Schlüssel der beteiligten Tabellen enthält.

Im folgenden ist das Datenbankschema als Entity-Relationship-Modell [siehe *Abbildung 7-1, Abbildung 7-2*] dargestellt. Die einzelnen Klassen des konzeptionellen Modells wurden in Tabellen der Datenbank abgebildet. Zusätzlich wurden die Tabelle „info_anl“ und „ma_berecht“ eingeführt, mit deren Hilfe die n:m-Relation zwischen der Klasse Information und der Klasse Anlage bzw. der Klasse Mitarbeiter und Berechtigungen implementiert wird. Neben den im konzeptionellen Modell angegebenen Attributen der Klassen, enthält jede Tabelle zusätzlich einen Primärschlüssel. Der Schlüssel ist in den aus den Klassen modellierten Tabellen stets eine Spalte vom Typ Integer, deren Wert sich mit jedem neu eingefügtem Datensatz automatisch um eins erhöht.

Um die eins-zu-viele-Beziehungen zu modellieren, wurde in den entsprechenden Tabellen ein Fremdschlüssel eingefügt. Die Tabelle Kategorie, welche eine rekursive Datenstruktur repräsentiert, wurde deshalb um eine Spalte „okat_id“ erweitert. Diese repräsentiert den primären Schlüssel der Oberkategorie der Kategorie. Eine Kategorie kann beliebig viele Unterkategorien besitzen. In die Tabelle Information wurden die Fremdschlüssel „kat_id“, „proj_id“ und „ma_id“ erweitert. Mit „kat_id“ wird die Zuordnung zur Kategorie realisiert, „proj_id“ ist der Schlüssel des Projekts, in dessen Rahmen eine Information erstellt wurde und „ma_id“ ist der Schlüssel des Mitarbeiters, der die Information verfaßt hat. In die Tabelle Mitarbeiter wurde der Fremdschlüssel „t_id“ eingefügt. Die Tabelle Gericht enthält für die Zuordnung zu einer Speisekarte den Fremdschlüssel „sp_id“, die Tabelle Speisekarte seinerseits den Fremdschlüssel „lief_id“ zum Bestimmen des Lieferanten, da es von einem Lieferanten mehrere Speisekarten geben kann.

Da das Schema auf einer Seite nicht gut darstellbar ist, habe ich es in zwei Entity-Relationship-Diagramme aufgeteilt. Die Tabelle Mitarbeiter ist in beiden Diagrammen vorhanden. Die Primärschlüssel sind durch ein Schlüsselssymbol in der letzten Spalte der

Tabelle gekennzeichnet. Die Spalte Null gibt an, ob für das Attribut ein Wert zwingend erforderlich ist, oder nicht.

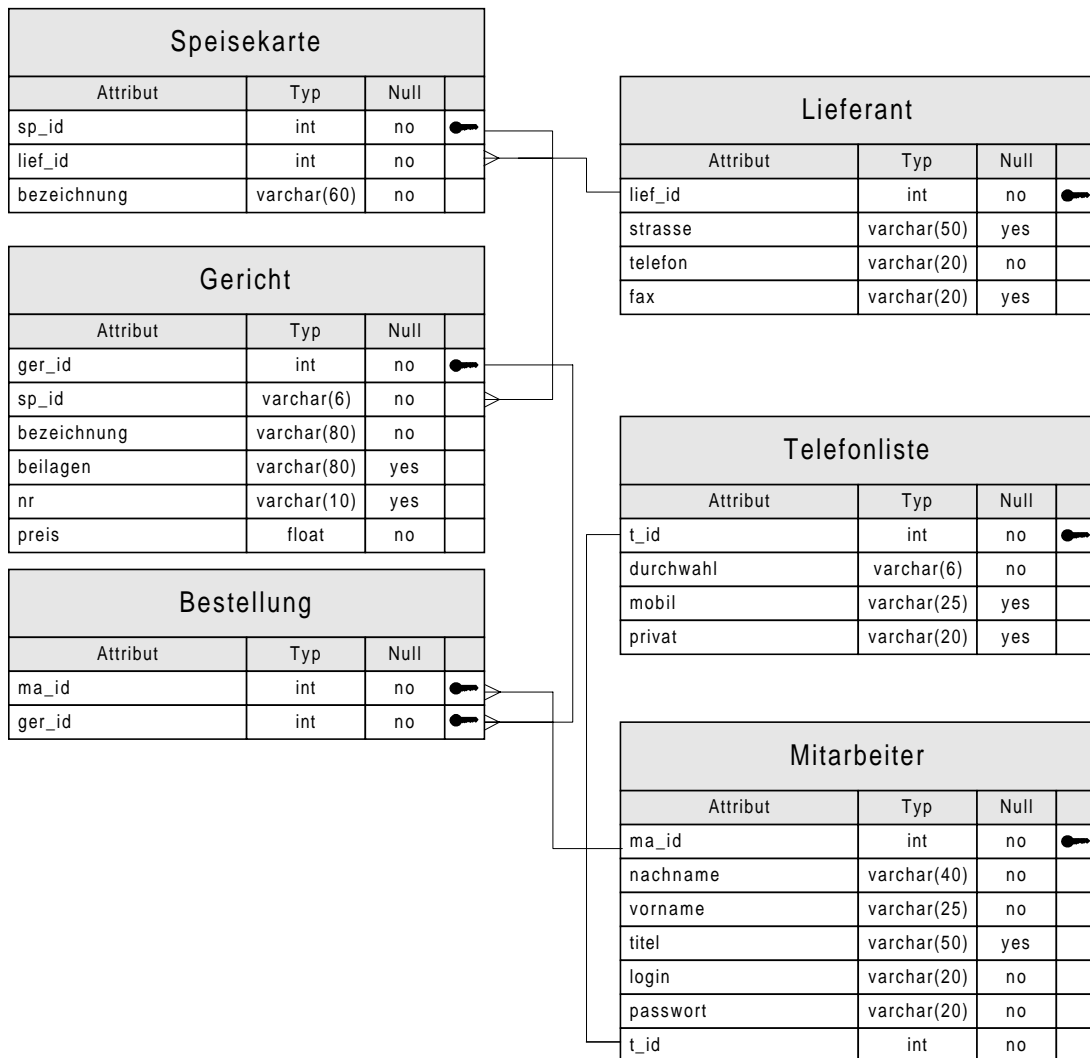


Abbildung 7-2

ER-Diagramm (Teil 1)

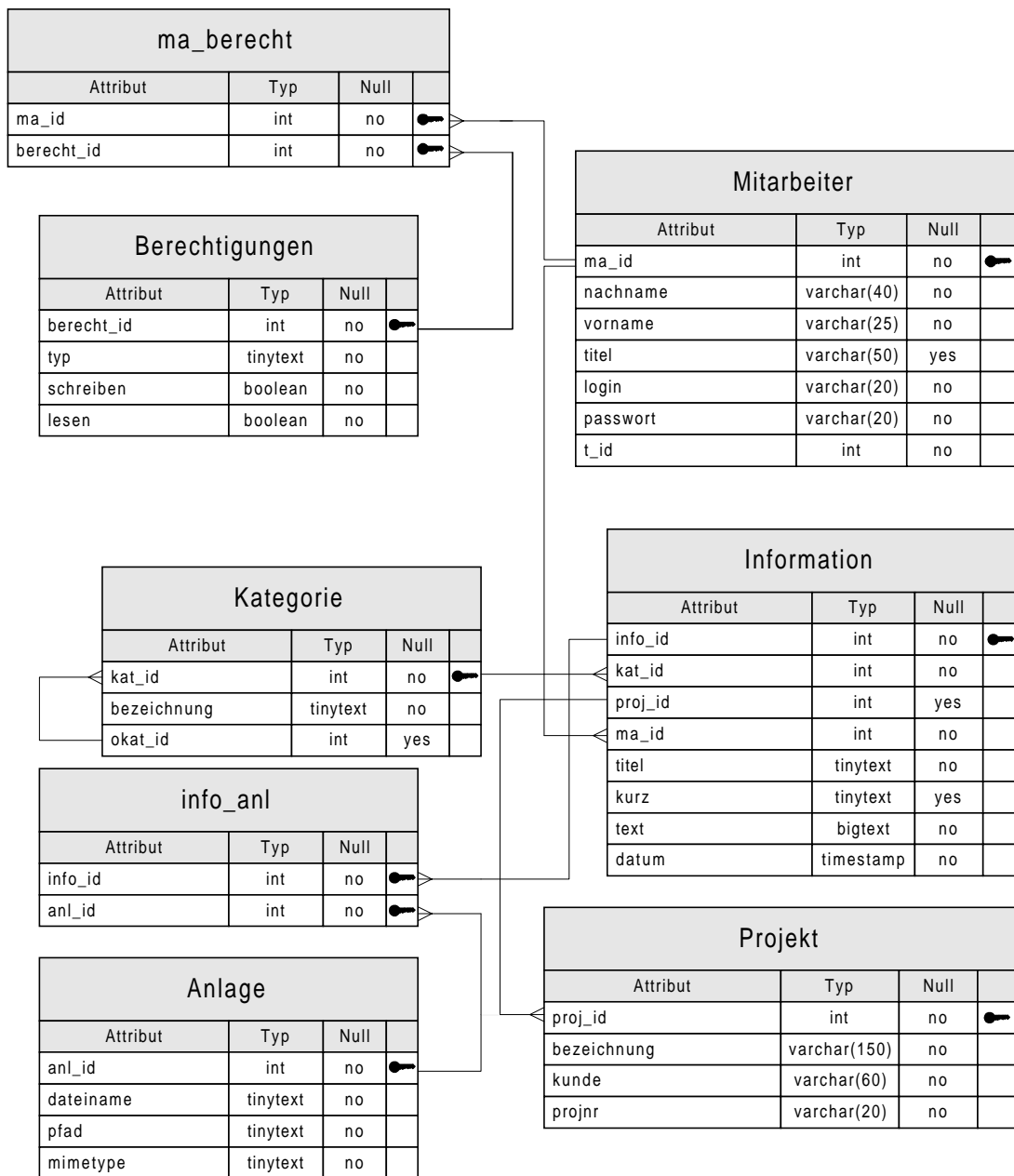


Abbildung 7-3

ER-Diagramm (Teil2)

7.2.2 Applikation

Die Applikation selbst wurde in Java realisiert. Java ist eine objektorientierte Programmiersprache, die ab 1990 bei der Firma Sun [SUN99] entwickelt wurde. Die im konzeptionellen Design entwickelten Klassen werden in einer Schicht der Applikation abgebildet. Die Applikation hat einen mehrschichtigen Aufbau. Die unterste Schicht bilden die Domänenklassen aus dem konzeptionellen Design. Sie sind für den kompletten Datenaustausch mit der Datenbank zuständig. Auf die Datenbank kann nur über die Domänenklassen zugegriffen werden. Der Zugriff wird über einen JDBC-Treiber (Java Database Connection) für das MySQL-DBMS (Datenbank-Managementsystem) realisiert. Die Applikation ist so angelegt, daß sie von der eingesetzten Datenbank weitgehend unabhängig ist. Sollte in Zukunft der Einsatz einer anderen Datenbank notwendig sein, so muß lediglich darauf geachtet werden, daß diese über eine JDBC-Schnittstelle verfügt und dann der Datenbanktreiber getauscht werden. Der Rest der Applikation bleibt vor Änderungen bewahrt.

Neben den eigentlichen Domänenklassen wurde die unterste Applikationsschicht noch um Kollektionen der Domänenklassen erweitert. Für jede Domänenklasse gibt es eine Kollektionsklasse. Mit diesen Kollektionen können Objekte permanent im Speicher gehalten werden, solange die Applikation nicht durch Herunterfahren des Webservers beendet wird. So verringert sich die Anzahl der notwendigen Zugriffe auf die Datenbank drastisch, was die Performance verbessert. Wird ein Objekt angefordert, das sich noch nicht in der Kollektion befindet, so erzeugt der Klassenkonstruktor eine Instanz der Klasse und besetzt sie mit den entsprechenden Werten aus der Datenbank. Das Objekt wird in die Kollektion aufgenommen und steht bei einer erneuten Anfrage bereits im Speicher bereit. Ebenso sorgen die Domänenklassen dafür, daß neu erzeugte Objekte in die Datenbank geschrieben werden, wenn der Datensatz dort noch nicht existiert, daß die Datenbank aktualisiert wird, wenn sich die Werte des Objekts ändern oder daß der Datensatz aus der Datenbank entfernt wird, wenn das Objekt gelöscht wird.

Neben der Fähigkeit, mit der Datenbank Daten auszutauschen und diese auf einem aktuellen Stand zu halten, verfügen die Domänenklassen noch über eine eigene Logik. Das heißt, sie erkennen selbständig, wenn sie mit Werten besetzt werden, die nicht in ihrem Wertebereich liegen. Die Applikation kann sofort auf diesen Umstand reagieren, ohne erst eine Datenbanktransaktion auslösen zu müssen, die fehlschlagen würde, weil nicht alle Attribute richtig gesetzt sind.

7.3 Implementierung des Navigationsdesigns

Beim Navigationsdesign wurden die Navigationskontexte und ihre Beziehungen untereinander aufgezeigt. Unter einem Navigationskontext versteht man eine Reihe von Navigationsknoten. Diese Navigationsknoten sind nach einem oder mehreren bestimmten Merkmalen geordnet. Über den Navigationskontext hat man die Möglichkeit, von einem Element dieser Menge zum vorherigen oder folgenden Element zu navigieren. In der Applikation sind zahlreiche Navigationskontexte definiert. Für einen Benutzer des Systems ist allerdings immer genau ein Kontext gültig. Zwar können sich mehrere Benutzer gleichzeitig innerhalb des gleichen Navigationskontextes befinden, jedoch muß der Kontext für jeden Benutzer einzeln gespeichert werden, um so eindeutig zugeordnet werden zu können.

Das HTTP-Protokoll bietet hierfür jedoch keine Lösung, da es zustandslos ist. Deswegen muß die Applikation selbst dafür sorgen, daß jedem Client ein eindeutiges Identifizierungsmerkmal zugeordnet wird. Dies geschieht mit Hilfe der im JSDK 2.0 definierten Klasse `HttpSession`. JSDK (Java Servlet Development Kit) ist eine Erweiterung von Java, die für die Erstellung von Servlets benötigt wird. Die Klasse `HttpSession` erlaubt es, jedem Client eine eindeutige Identität zuzuweisen, so daß der Server exakt feststellen kann, von wo die Anfrage stammt.

Ein Kontext stellt immer eine nach bestimmten Merkmalen definierte Menge von Objekten dar. Eine Zugriffsmöglichkeit auf diese Menge muß nun solange für einen Benutzer gespeichert werden, bis dieser den momentanen Navigationskontext verläßt.

Nach Wechsel in einen neuen Navigationskontext muß dieser neue Kontext gespeichert werden.

Da es sich beim Kontext um eine Menge gleichartiger Objekte handelt, bietet es sich an, diese auch in einer Menge zwischenspeichern. In der erstellten Applikation wird dies mit Hilfe einer Navigationskontextklasse erreicht. Diese Klasse wird in der HttpSession gespeichert und ermöglicht so das Navigieren zum vorherigen oder nächsten Element des Kontextes.

7.4 Implementierung des Präsentationsdesigns

Bereits beim Entwurf der Präsentation wurde die Aufteilung der zur Verfügung stehenden Präsentationsfläche in drei Teile festgelegt. Dies wurde durch den Einsatz von HTML-Frames umgesetzt.

Das obere Frame enthält das Hauptmenü und ist immer präsent. Neben einigen Menüpunkten ist darüber auch die oberste Ebene der Kategorien des Informationssystems erreichbar. Bei Anwählen einer dieser Kategorien erscheinen auf der linken Seite die entsprechenden Unterkategorien. Beide Menüleisten sind vom restlichen Bereich farblich abgesetzt. Im Hauptfenster wird die aktuelle Seite angezeigt.

Als Darstellung des Seitenmenüs wurden Hyperlinks gewählt. Zwar gibt es fortgeschrittene Techniken der Darstellung von Menüs im WWW mittels DHTML und JavaScript. Es wurde jedoch darauf verzichtet, da es primär galt, die Funktionalität bereitzustellen und nicht darum, eine möglichst elegante Darstellung der Hierarchien des Informationssystems zu finden. Allerdings kann nachträglich jederzeit eine solche fortgeschrittene Darstellung eingebaut werden, wenn dies von den Benutzern als erforderlich angesehen wird.

Die angezeigten Seiten werden mit Hilfe von HTML-Templates erstellt. Diese HTML-Templates enthalten Platzhalter, welche von der Applikation dynamisch mit entsprechenden Inhalten aufgefüllt werden. Dadurch kann ein und das selbe Template für mehrere Aufgaben verwendet werden.

Auf die Abbildung von Benutztermasken wurde hier verzichtet, da sie später im Rahmen des Benutzerhandbuchs noch vorgestellt werden.

7.5 Implementierungstechnologien

7.5.1 Servlets

Die Applikation wurde als Servlet realisiert. Unter einem Servlet versteht man eine Java-Applikation, die auf einem Webserver läuft und dynamische Webseiten erzeugt. Dynamische Webseiten können auch durch den Einsatz von CGI-Programmen (Common Gateway Interface) erzeugt werden. Allerdings bietet die Servlettechnologie einige Vorteile.

Während bei traditionellen Webapplikationen, die mit der CGI-Technologie realisiert wurden, der Webserver ein externes Programm aufruft, kann ein Servlet als eine Funktionserweiterung des Webserver angesehen werden. CGI wurde ursprünglich entwickelt, um für Informationssysteme eine Standardmethode zum Aufruf externer Applikationen bereitzustellen. Die Fähigkeit von CGI-Programmen, dynamische Webseiten zu erzeugen, ist so gesehen nur ein Seiteneffekt. Erhält ein Server eine

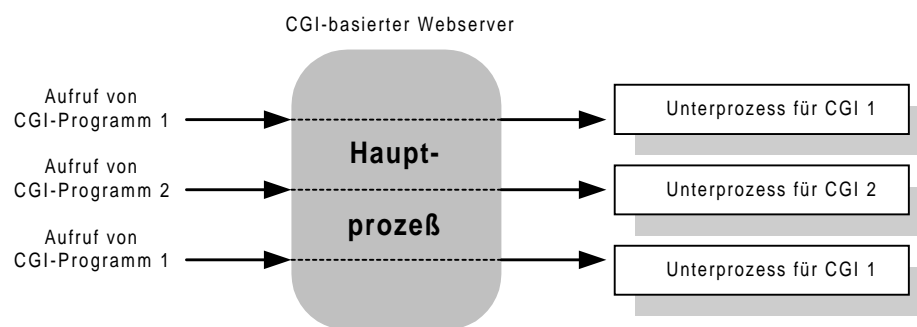


Abbildung 7-3

CGI-Lebenszyklus

Anfrage, für die ein CGI-Programm ausgeführt werden muß, so wird vom Server ein neuer Prozeß gestartet, um das Programm aufzurufen und die Anfrage an das Programm weiterzuleiten [Abbildung 7-3]. Das Erzeugen eines neuen Prozesses kostet Zeit und

verbraucht Systemressourcen, so daß die Anzahl der Anfragen, die gleichzeitig bearbeitet werden können, limitiert ist. Ein weiterer Nachteil ist, daß CGI-Programme nicht mit dem Webserver interagieren oder dessen Fähigkeiten nutzen können, da sie in einem separaten Prozeß ausgeführt werden.

Ein Servlet ist eine Java-Klasse, die vom Server dynamisch geladen wird und dessen Fähigkeiten erweitern. Servlets sind mit proprietären Servererweiterungen vergleichbar, laufen jedoch im Unterschied dazu in der Java Virtual Machine (JVM) des Servers. Während bei CGI-Programmen für jede Anfrage und jedes Programm ein eigener Prozeß erzeugt werden muß, werden Servlets über verschiedene Threads innerhalb des Webserver angesprochen. Vom Servlet selbst wird nur eine Instanz geladen, die sich ab dann persistent im Speicher des Servers befindet [Abbildung 7-1].

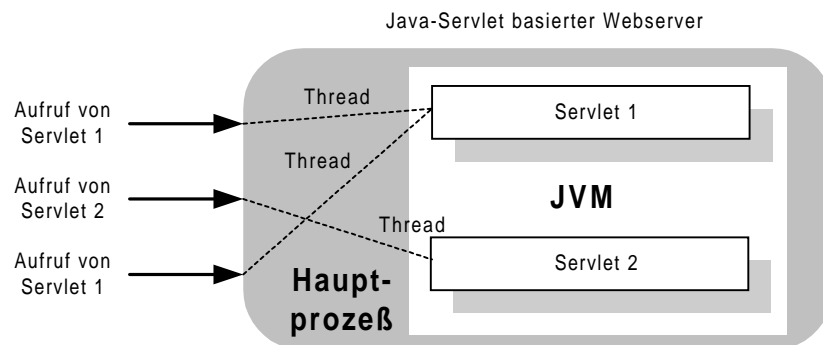


Abbildung 7-4 Servlet-Lebenszyklus

Um Servlets erstellen zu können, benötigt man eine zusätzliche Klassenbibliothek. Diese Klassenbibliothek heißt Java Servlet Development Kit (JSDK).

7.5.2 Eingesetzte Komponenten

Für das Intranet wurde ein eigenständiger neuer Server installiert. Es wird dazu ein normaler PC mit 200Mhz Prozessortaktrate eingesetzt. Als Betriebssystem wird Linux verwendet. Es handelt sich dabei um die Distribution 6.0 von SuSE [SuSE99]. Diese Distribution verwendet den Linux-Kernel 2.2.3.

Auf dem Rechner ist als Webserver ein Apache [Apa99] Version 1.3.6. installiert. Um Servlets ausführen zu können, muß das Apache-Zusatzmodul JServ [Jap99] den Webserver eincompiliert werden. Dazu wird die Version 1.0 verwendet. JServ unterstützt leider lediglich die Version 2.0 des JSDK [JSDK99] und nicht die aktuelle Version.

Die Applikation selbst wurde auf Basis des Java Development Kit 1.1.7 [JDK99] realisiert. Als Datenbank kommt MySQL [TcX99] in der Version 3.22.19 zum Einsatz. Die Anbindung der Applikation an die Datenbank erfolgt über einen JDBC-Treiber von M. Mathew, [MM99] Version 1.1.

7.5.3 Aufbau der Applikation

Die für die Applikation erstellten Java-Klassen können wie folgt unterteilt werden:

- **Servlet-Klassen:**
In diese Kategorie fallen zwei Klassen, die Klasse Intranet, welche als Servlet fungiert und die Klasse InitVars, die zum Halten von Session-Informationen und ähnlichem dient.
- **Hilfsklassen:**
Unter diesen Begriff sind die Klassen DatabaseCon
- **UseCase-Klassen**
Klassen, die die in den Anforderungen erarbeiteten UseCases nachmodellieren.
- **Domain-Klassen**
Die im konzeptionellen Design entwickelten Klassen zuzüglich Kollektions-Klassen und Oberklassen.

Im folgenden wird der Ablauf einer Anfrage an das Servlet kurz skizziert. Das Servlet nimmt eine HTTP-Anfrage entgegen. In diese Anfrage ist ein Kommando durch Anhängen an die URL mit encodiert. Dieses Kommando bestimmt, von welcher UseCase-Klasse eine Instanz erzeugt wird. Ist kein Kommando angegeben oder kann es nicht interpretiert werden, so wird eine Instanz der Standard-UseCase-Klasse geladen, die die Startseite ausgibt. Es wird festgestellt, ob die Anfrage von einem Client kommt, mit dem eine aktive Session aufrecht erhalten wird. Falls ja, werden die in der Session

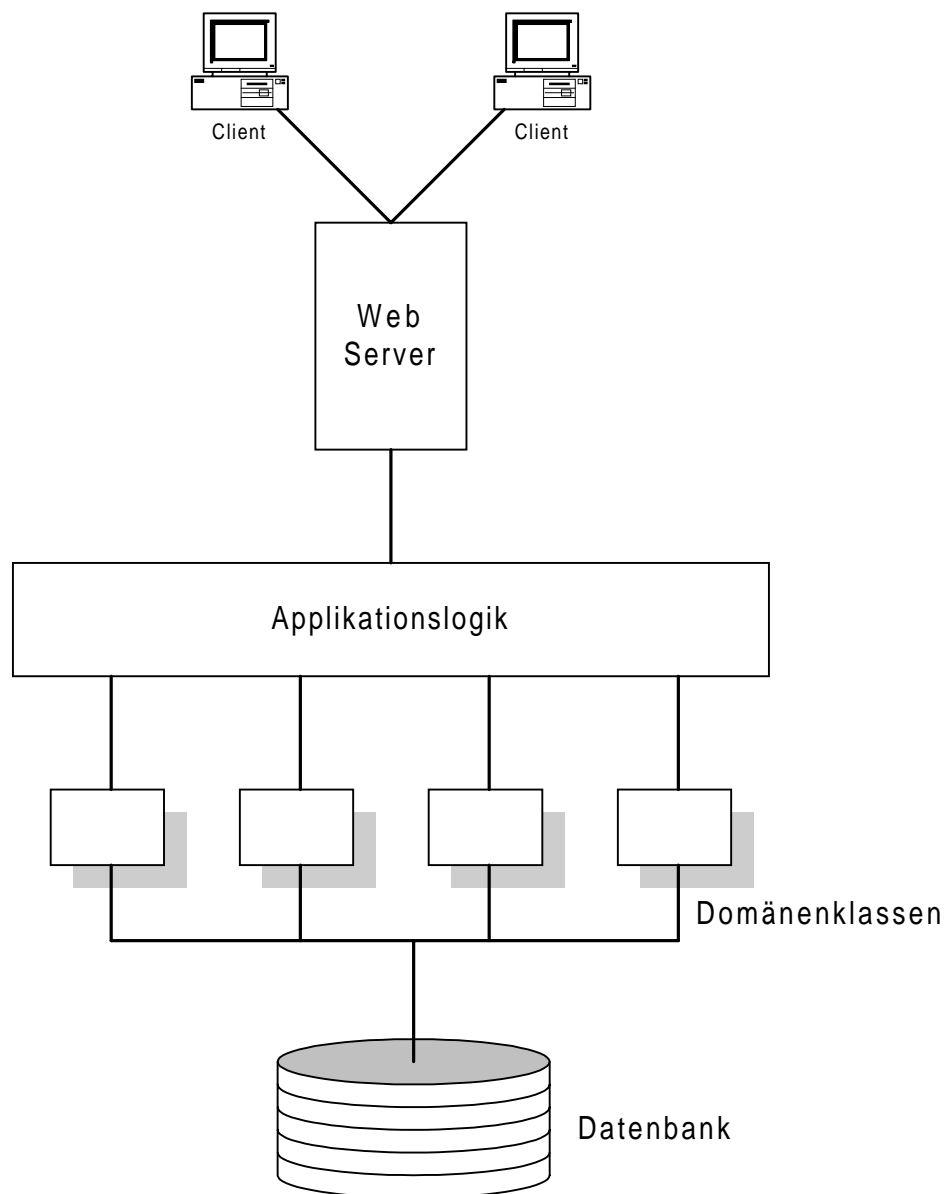


Abbildung 7-5

Architektur des Systems

gespeicherten Werte ausgelesen und für die Weiterverarbeitung aufbereitet. In den UseCases werden über die Domain-Objekte die Daten für die Platzhalter berechnet, ein HTML-Template geladen und die Platzhalter durch die aktuellen Werte ersetzt. Dann wird der entstandene HTML-Code an den Server übergeben. Abbildung 7-5 stellt dieses Prinzip noch einmal grafisch dar.

7.5.4 Erweiterbarkeit

Das Konzept gewährleistet eine leichte Erweiterbarkeit, da für zusätzliche Funktionalitäten lediglich neue UseCase-Klassen und gegebenenfalls neue HTML-Templates erstellt werden müssen. In diesen wird dann lediglich entschieden, welche Platzhalter mit aus Domain-Objekten erhaltenen Werten ersetzt werden und welches Template geladen werden soll. Die Entgegennahme und Verarbeitung der Anfrage ist bereits automatisiert, ebenso Datenaustausch der Domain-Objekte mit der Datenbank. Auch das Verarbeiten der HTML-Templates und Ausgeben der Antwort ist automatisiert.

8 Benutzerhandbuch

8.1 Allgemeine Bedienungshinweise

Das Medialab Intranet ist für den Einsatz von Netscape Navigator 4.0 und höher sowie Microsoft Internet Explorer 4.0 und höher optimiert worden. Als Bildschirmauflösung werden 1024 x 768 Pixel empfohlen. Diese Systemanforderungen können von allen bei Medialab installierten Systemen erfüllt werden. Bei der Benutzung anderer Browser kann es zu leichten Fehlanzeigen kommen, was jedoch die Funktionalität nicht weiter einschränkt.

Da das System zustandsorientiert arbeitet, muß die Verwendung von Cookies aktiviert sein.

Das Fenster ist in drei Frames unterteilt:

- Hauptmenü und Suche am oberen Fensterrand
- Kategorien (aktueller Zweig) am linken Fensterrand
- Arbeitsbereich

Im Frame am linken Seitenrand werden nur dann Hyperlinks zu Kategorien angezeigt, wenn im Arbeitsbereich eine Kategorie- oder Informationsansicht aktiviert ist. Sonst bleibt das Frame leer. Wenn Kategorien angezeigt werden, so handelt es sich dabei um alle vorhandenen Kategorien des aktuellen Zweigs.

Beispiel: Es ist eine beliebige Informations- oder Kategorieansicht aus dem Bereich „Grafik“ aktiviert. Im linken Frame werden alle Unterkategorien von „Grafik“ angezeigt.

8.2 Starten der Applikation

Die Applikation ist unter der URL <http://intra.medialab.de/intranet/Intranet> erreichbar. Um das System benutzen zu können, ist eine Identifizierung notwendig [Abbildung 8-1]. Login und Paßwort entsprechen dem Benutzernamen und Paßwort, der auch für den Zugriff auf das Firmennetz angegeben werden muß.

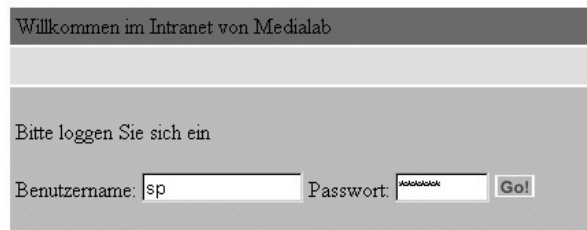


Abbildung 8-1 Login

8.3 Homepage

Nach dem erfolgreichen Anmelden landet man auf der Intranet Homepage [Abbildung 8-2]. Ab diesem Zeitpunkt sind auch das Hauptmenü und das Navigationsframe am linken Seitenrand aktiv. Die Homepage gewährt dem Benutzer einen Überblick über neue Informationen im Intranet. Dabei werden die Neuheiten in den Kategorien News und Buchtips sowie die restlichen neuen Informationen in eigenen Listen angezeigt. Der

Benutzer hat nun die Möglichkeit auszuwählen, wie alt die angezeigten Informationen maximal sein sollen. Es kann von jeder Liste die an diesem Tag neu hinzugekommen

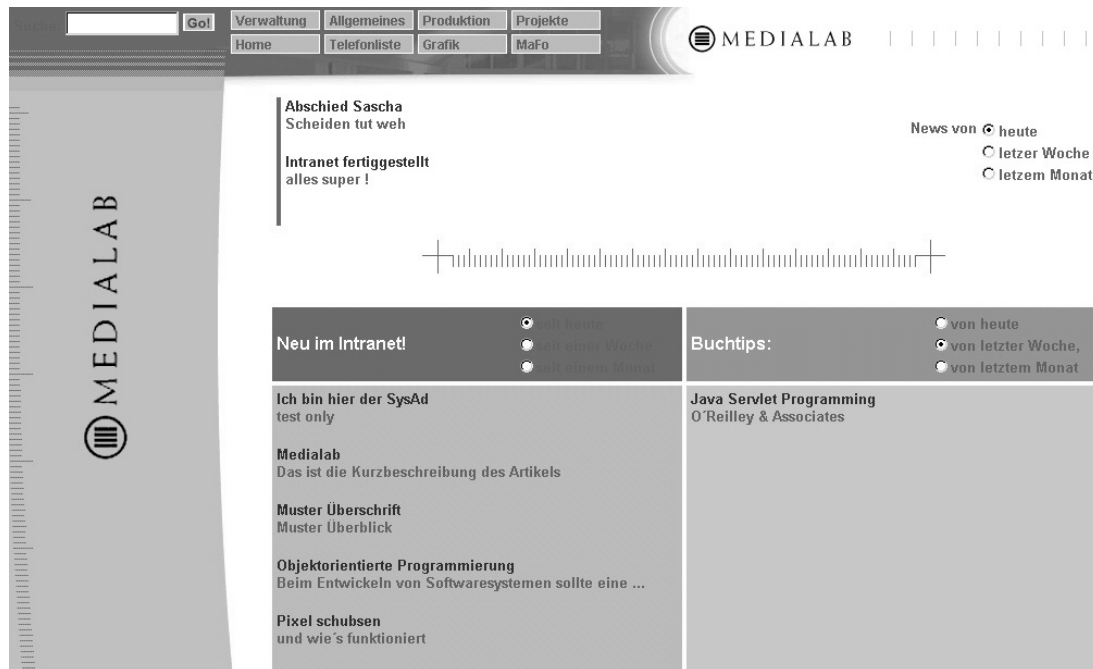


Abbildung 8-1 Intranet Homepage

Informationen, die maximal eine Woche alt sind und eine Monatsübersicht gewählt werden. Standardmäßig werden die News und Neuheiten vom Tage sowie die Buchtips der letzten sieben Tage angezeigt.

Der Menüpunkt „Essenbestellung“ taucht im Hauptmenü noch nicht auf, da er noch nicht vollständig implementiert ist.

8.4 Artikel anzeigen

Wird auf eine der in den Listen angezeigten Artikelüberschriften geklickt, so wird dieser Artikel angezeigt [Abbildung 8-3]. Gleichzeitig erscheint im linken Frame eine Liste mit den Kategorien aus dem Bereich aus dem der Artikel stammt. Mit den Buttons

„Previous“ und „Next“ können nacheinander alle Artikel der Liste angezeigt werden, aus welcher der ausgewählte Artikel stammt.

In dieser Ansicht werden auch alle Daten angezeigt, die zu dem Artikel gehören: Die Kategorie, in die er eingeordnet ist, Erstellungsdatum, Autor(en) und Projektnummer.

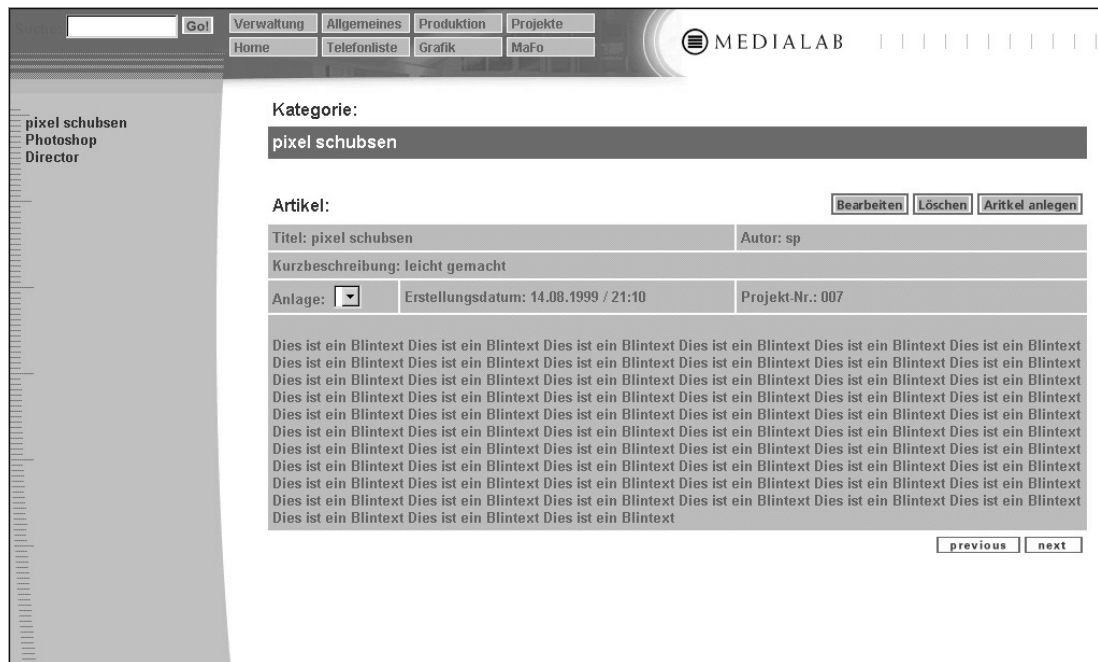


Abbildung 8-3

Ansicht eines ausgewählten Artikels

8.5 Artikel anlegen und bearbeiten

Der Benutzer kann nun auch den gerade angezeigten Artikel editieren, ihn löschen oder auch einen neuen Artikel anlegen. Dies geschieht über die Buttons oben rechts. Wird „Bearbeiten“ oder „Artikel anlegen“, so wird ein Formular angezeigt, in das die entsprechenden Daten eingegeben werden können [Abbildung 8-4].

The screenshot shows a web interface for editing an article. At the top, there is a navigation bar with a search box and menu items: Verwaltung, Allgemeines, Produktion, Projekte, Home, Telefonliste, Grafik, MaFo. The MEDIALAB logo is on the right. On the left, a sidebar lists 'pixel schubsen', 'Photoshop', and 'Director'. The main content area is titled 'Artikel bearbeiten:' and contains the following fields:

- Titel:** pixel schubsen
- Kurzbeschreibung:** leicht gemacht
- Text:** A text area containing multiple lines of 'Blintext Dies ist ein'.
- Projekt:** Intranet

Below the form is the 'Artikel einordnen:' section with three radio button options:

- Einordnen in Kategorie über Kategorie: pixel schubsen
- Einordnen in Kategorie: pixel schubsen (Default)
- Einordnen in Kategorie:

At the bottom of the form are two buttons: 'Eingaben übernehmen' and 'Eingaben löschen'.

Abbildung 8-4

Formular Artikel

8.6 Übersicht der Artikel in einer Kategorie

Aus dem Formular heraus gelangt man in eine Übersicht aller in der Kategorie des bearbeiteten Artikels befindlichen Artikel. [Abbildung 8-5] Hier kann wieder ein Artikel ausgewählt werden und dann die Liste mit „Previous“ oder „Next“ durchlaufen werden. Es besteht natürlich immer auch die Möglichkeit, sich über das Kategoriemenü eine andere Kategorie anzeigen zu lassen oder über das Hauptmenü etwas auszuwählen.

8.7 Kategorien anlegen und bearbeiten

In der Artikelübersicht einer Kategorie kann man über die Buttons „Bearbeiten“, „Löschen“ oder „Unterkategorie anlegen“ noch weitere Aktionen auslösen. Auch ein weiterer Artikel kann mit „Artikel anlegen“ neu angelegt werden.



Abbildung 8-5 Übersicht Artikel einer Kategorie

Entscheidet man sich dafür, die aktuelle Kategorie zu bearbeiten oder eine neue Kategorie anzulegen, so wird das Bearbeitungsformular für Kategorien aufgerufen [Abbildung 8-6]. Nach dem Bearbeiten oder neu Anlegen einer Kategorie wird wieder die Übersicht der aktuellen Kategorie angezeigt. Beim Anlegen einer neuen Kategorie ist die natürlich die gerade erzeugte Kategorie.

Neben der Eingabe oder Veränderung der Artikelbezeichnung kann eine Kategorie auch noch in eine andere Ebene, nämlich eine Ebene höher, oder, falls die Oberkategorie der aktuellen Kategorie noch weitere Unterkategorien besitzt, auch eine Ebene tiefer verschoben werden.

The screenshot displays the 'Formular Kategorie anlegen und bearbeiten' interface. At the top, there is a navigation bar with tabs for 'Verwaltung', 'Allgemeines', 'Produktion', 'Projekte', 'Home', 'Telefonliste', 'Grafik', and 'MaFo', along with the 'MEDIALAB' logo. On the left, a sidebar lists 'pixel schubsen', 'Photoshop', and 'Director'. The main content area is divided into sections: 'Ebene:' with 'pixel schubsen' selected; 'Kategorie neu anlegen:' with a 'Bezeichnung' input field; and 'Kategorie einordnen:' with three radio button options: 'Eine Ebene höher einordnen, falls nicht schon auf oberster Ebene', 'Einordnen in pixel schubsen (Default)', and 'Einordnen in Ebene' with a dropdown menu set to 'Ebene auswählen'. At the bottom, there are two buttons: 'Eingaben übernehmen' and 'Eingaben löschen'.

Abbildung 8-6

Formular Kategorie anlegen und bearbeiten

8.8 Kategorie löschen

Wählt man in der Artikelübersicht einer Kategorie „Kategorie löschen“ aus, so erscheint eine Sicherheitsabfrage [Abbildung 8-7]. Mit der zum Löschen ausgewählten Kategorie können auch alle Artikel dieser Kategorie und alle ihre Unterkategorien inklusive deren Artikel gelöscht werden. Es können alternativ aber auch nur alle Artikel der aktuellen Kategorie gelöscht werden.

Standardmäßig ist jedoch der Punkt „nicht löschen“ ausgewählt, um einem versehentlichen Löschen einer Kategorie vorzubeugen. Wird einer der Punkte „alle Artikel löschen“ oder „Komplett inklusive Unterkategorien löschen“ ausgewählt, so wird entsprechend eine weitere Sicherheitsabfrage angezeigt.

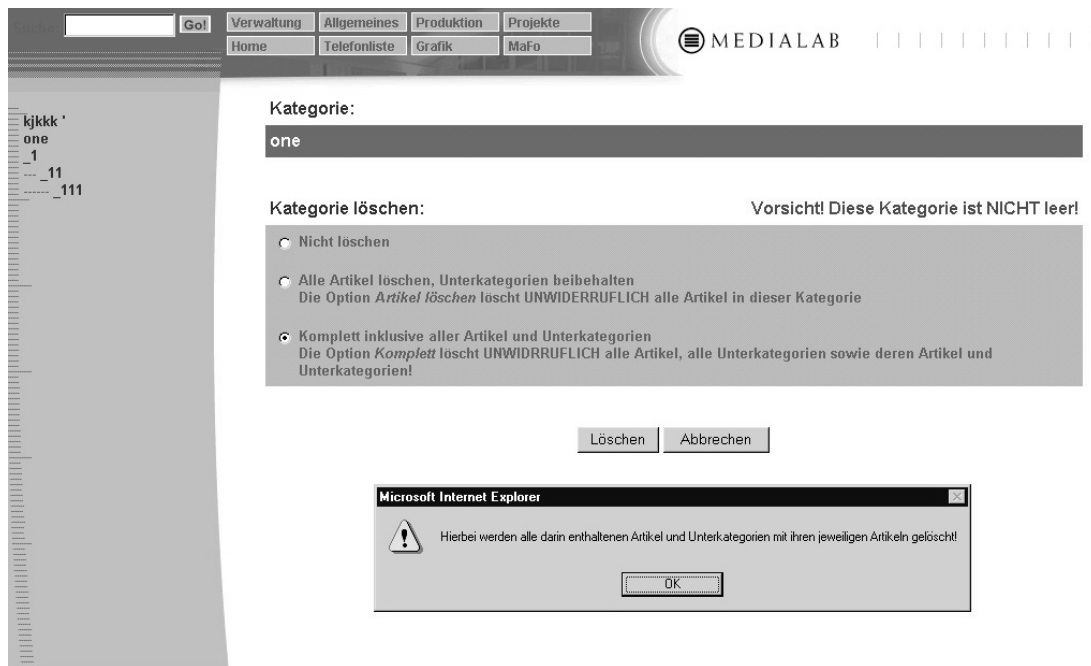


Abbildung 8-7

Sicherheitsabfrage Kategorie löschen

Wird die Kategorie letztendlich gelöscht, so wird die Artikelübersicht der Oberkategorie der gerade gelöschten Kategorie angezeigt. Ansonsten gelangt man wieder zur Übersicht der gerade aktuellen Kategorie.

8.9 Suche

In der Hauptmenüleiste befindet sich ein Textfeld, in das ein Suchbegriff eingegeben werden kann. Es wird dann in allen vorhandenen Bereichen des Intranets gesucht. Das Suchergebnis wird als in verschiedene Bereiche unterteilte Liste dargestellt [Abbildung 8-8]. Dabei werden gefundene Informationen und Kategorien mit Links für einen direkten Zugriff dargestellt, Telefonlisteneinträge, Anlagen und Projekte jedoch nur als Text. Es soll damit verhindert werden, daß die Daten von Projekten, Mitarbeitern und Anlagen von jedermann verändert einfach bearbeitet werden können. Dies ist nur dazu Berechtigten Mitarbeitern gestattet und wird über den Punkt „Verwaltung“ im Hauptmenü ermöglicht.

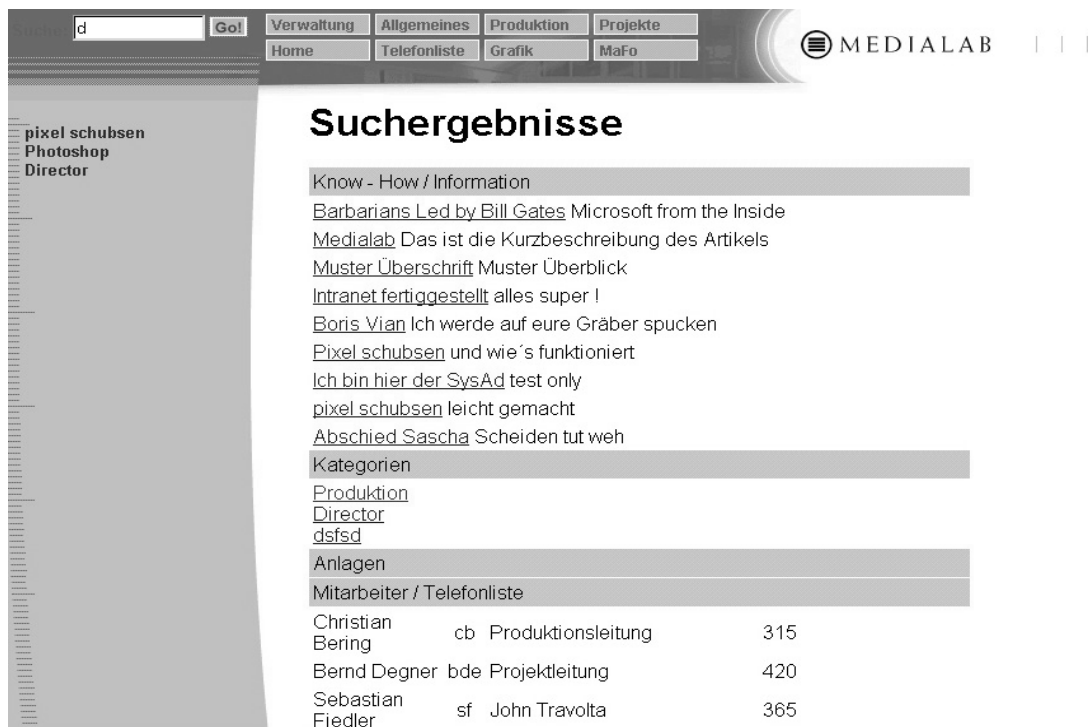


Abbildung 8-8 Suchergebnisse

8.10 Verwaltung

Über den Punkt Verwaltung können verschiedene Bereiche verwaltet werden. Mitarbeiter, Anlagen, Projekte, Speisekarten und Kategorien. Da der Ablauf für all diese Bereiche gleich ist, wird exemplarisch nur die Mitarbeiter- / Telefonlistenverwaltung beschrieben. Wird der Menüpunkt Mitarbeiter- / Telefonlistenverwaltung angewählt, so wird eine Seite angezeigt, über die neue Benutzer angelegt werden können oder vorhandene Einträge bearbeitet oder gelöscht werden können [Abbildung 8-9]. Außerdem kann nach Mitarbeitern und Telefonlisteneinträgen gesucht werden. Das Ergebnis dieser Suche wird im Gegensatz zur allgemeinen Suche mit Links dargestellt, um den entsprechenden Eintrag durch anklicken bearbeiten zu können [Abbildung 8-10].

Wählt man über die Auswahl in der Mitarbeiterverwaltung oder das Anwählen eines Links aus dem Suchergebnis Mitarbeitersuche „Bearbeiten“ aus, so wird ein

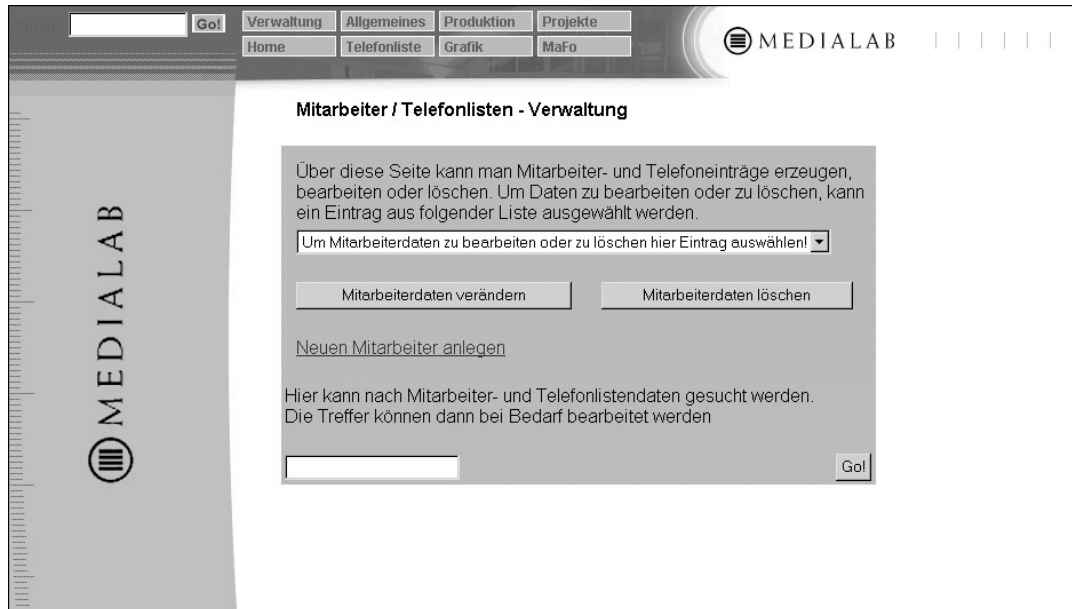


Abbildung 8-10

Suchergebnis Mitarbeiter / Telefonliste

Mitarbeiterformular angezeigt [Abbildung 8-11]. Auch beim Anlegen eines neuen Mitarbeitereintages wird dieses Formular benutzt. Nach dem Eingeben oder Bearbeiten von Daten gelangt man wieder zur Auswahl Mitarbeiterverwaltung.

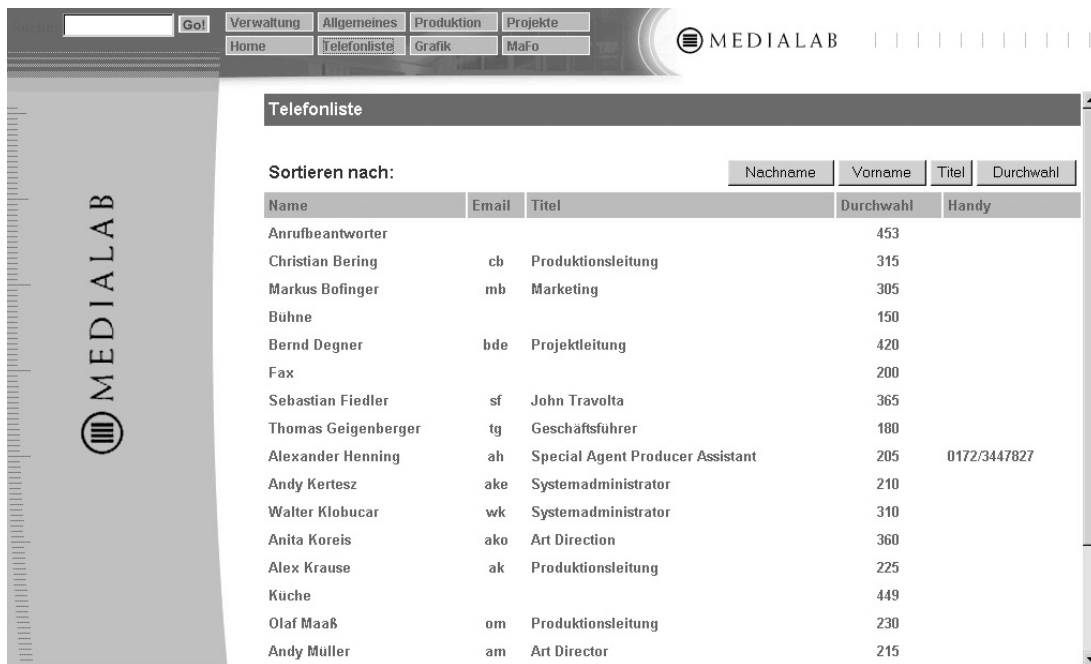
The screenshot shows a web application interface for 'Mitarbeiter - Verwaltung'. The top navigation bar includes a search box with a 'Go!' button and menu items: 'Verwaltung', 'Allgemeines', 'Produktion', 'Projekte', 'Home', 'Telefonliste', 'Grafik', and 'MaFo'. The 'MEDIALAB' logo is visible in the top right and on a vertical sidebar on the left. The main heading is 'Mitarbeiter - Verwaltung'. Below this, there are two columns: 'Mitarbeiter neu anlegen' and 'Berechtigungen:'. The first column contains input fields for 'Vorname', 'Nachname', 'Titel', 'Login', 'Passwort', 'Durchwahl', 'Mobil', and 'Privat'. The second column contains a list of permissions with checkboxes: 'Alle (Standard)' (checked), 'Empfang', 'Administration', 'Geschäftsleitung', 'Grafik', 'Produktion', 'Projektleitung', 'Vertrieb', and 'Marketing'. At the bottom of the form are two buttons: 'Eingaben übernehmen' and 'Eingaben löschen'.

Abbildung 8-11

Formular Mitarbeiter

8.11 Telefonliste

Über den Menüpunkt Telefonliste im Hauptmenü kann man sich die Telefonliste anzeigen lassen [Abbildung 8-12]. Die Liste kann über die zur Verfügung stehenden Buttons nach verschiedenen Nachname (Standard), Vorname, Titel und Durchwahl sortiert werden.



Telefonliste

Sortieren nach: Nachname Vorname Titel Durchwahl

Name	Email	Titel	Durchwahl	Handy
Anrufbeantworter			453	
Christian Bering	cb	Produktionsleitung	315	
Markus Bofinger	mb	Marketing	305	
Bühne			150	
Bernd Degner	bde	Projektleitung	420	
Fax			200	
Sebastian Fiedler	sf	John Travolta	365	
Thomas Geigenberger	tg	Geschäftsführer	180	
Alexander Henning	ah	Special Agent Producer Assistant	205	0172/3447827
Andy Kertesz	ake	Systemadministrator	210	
Walter Klobucar	wk	Systemadministrator	310	
Anita Koreis	ako	Art Direction	360	
Alex Krause	ak	Produktionsleitung	225	
Küche			449	
Olaf Maaß	om	Produktionsleitung	230	
Andy Müller	am	Art Director	215	

Abbildung 8-12

Telefonliste

9 Bewertung und Ausblick

In dieser Diplomarbeit wurde ein webbasiertes Informationssystem geschaffen, daß die Firma Medialab als Kommunikationsplattform in Gestalt ihres Intranets nutzt. Das System wurde deswegen an die Bedürfnisse des Unternehmens angepaßt. Die Mitarbeiter können das Informationssystem selbst strukturieren, indem sie selbst die Kategorien anlegen und so einordnen können, wie es ihren Bedürfnissen entspricht. In diesen Kategorien können dann Informationen angelegt werden. Wird nachträglich festgestellt, daß die Einordnung einer Kategorie oder einer Information nicht ideal ist, so kann entweder die Information oder auch eine ganze Kategorie inklusive Unterkategorien und den dazugehörigen Informationen verschoben werden. Neben dem Anlegen und Bearbeiten von Inhalten des Informationssystems besteht auch die Möglichkeit gezielt nach Informationen zu suchen. Dabei gibt es eine Suche über alle bestehenden Bereiche und für jeden einzelnen Bereich eine eigene Suche. Die einzelnen Informationen sind untereinander nicht mit Hyperlinks vernetzt. Das heißt eine klassische Hypertextstruktur wie sie beispielsweise beim Wiki Wiki Web [Wiki99] existiert, besteht nicht. Statt dessen sind die Informationen über Navigationskontexte und Menüstrukturen vernetzt. Alle Seiten werden dynamisch aus den Werten der Objekte oder Datenbankabfragen generiert.

Das ursprünglich vorgesehene Modul Essenbestellung wurde auf Grund von Zeitdruck in Absprache mit Medialab nicht mehr komplett implementiert. Die Basis für dieses Modul ist jedoch bereits geschaffen. Eine vollständige Implementierung und Test würde noch in etwa drei bis vier Tage in Anspruch nehmen. Ebenso wurde das ganz zu Anfang geplante Modul Schriftarchiv verworfen, da für die dazu notwendige Konvertierung der vorhandenen Schriften kein Werkzeug zur Verfügung stand oder auf dem Markt zu

finden war, mit dem dieser Vorgang automatisiert werden könnte. Auch sind Urheberrechtsfragen nicht geklärt.

Obwohl die Anwendung an die Bedürfnisse eines speziellen Unternehmens angepaßt wurde, gerade was das Erscheinungsbild betrifft, kann sie in der Form auch in anderen Unternehmen genutzt werden. Denn der Bedarf, Know-How zu bewahren, dieses zu sortieren und auf möglichst einfache Art und Weise dem Personal zur Verfügung zu stellen, findet sich in vielen Unternehmen wieder. Aber auch für andere Zwecke, ist die Applikation denkbar, beispielsweise als ein über das Internet zugängliches Supportwerkzeug oder Diskussionsforum. Der Kern der Applikation, das Informationssystem, ist für all diese Zwecke einsetzbar. Lediglich die Inhalte müssen angepaßt werden.

Schwierigkeiten gab es vor allen während der Design- und Implementierungsphase. Während der Designphase gab es hauptsächlich Probleme mit den UML-konformen Erweiterungen für OOADM. Zwar sollte als Notation während des gesamten Softwareentwicklungsprozesses UML verwendet werden, doch während der Designphase war die Entwicklung der Erweiterungen noch nicht abgeschlossen, so daß mehrfach Änderungen an der Notation vorgenommen werden mußten. Beispielsweise gab es während der Entwicklung des Präsentationsdesigns noch kein Stereotyp für Image. Später wurde die Notation <<filtered context>> neu eingeführt. Deswegen mußte noch bei der Erstellung der Ausarbeitung alle vorhandenen Navigationskontextdiagramme angepaßt werden. Da für die Erstellung der Diagramme kein Werkzeug zur Verfügung stand, war dies zeitaufwendig. Bis heute ist die Semantik der Notation noch nicht ohne jeden Zweifel geklärt.

Bei der Implementierung bestanden die Probleme darin, daß die Wahl der zu Grunde liegenden Technologien erst sehr spät getroffen werden konnte. Von Seiten Medialab bestand die Vorgabe, das Kostenniveau so niedrig wie möglich zu halten. Die Ermittlung der Kosten für die vom Institut bevorzugte Lösung mit Gemstone und Smalltalk zog sich auf Grund der Informationspolitik von Gemstone in die Länge. Dann stellte sich heraus, daß die Kostensituation einen Einsatz dieser Lösung vereitelte.

Erschwerend kam dazu, daß ich keine Vorkenntnisse in Java hatte und zudem zu sehr funktional dachte. Dadurch entstand eine lange Einarbeitungsphase. Zur Entwicklung des Systems standen auch keine Hilfs- oder Entwicklungstools wie beispielsweise eine integrierte Entwicklungsumgebung zur Verfügung.

Die verwendete Methodik zur Softwareentwicklung von der Erstellung der Anwendungsfälle anhand von Interviews, des nachfolgenden Entwurfs mit OOHDM und die Implementierung hat maßgeblich zum Gelingen der Diplomarbeit beigetragen. Durch die definierte ein klar umrissenes Einsatzgebiet der Applikation. Anhand dieser Anwendungsfälle konnte das Klassenmodell im konzeptionellen Design entworfen werden, welches später als Grundlage für das Datenbankdesign verwendet wurde. Das Navigationsdesign erleichterte es, sinnvollen Navigationsmöglichkeiten zu erkennen. Im Präsentationsdesign konnten daraufhin sofort die notwendigen Inhalte definiert werden. Durch die objektorientierte Implementierung mit Java konnte der Entwurf in eine gut funktionierende und anpassungsfähige Applikation umgesetzt werden. Allerdings entspricht die Navigation in der Applikation nicht mehr ganz der im Navigationsdesign ursprünglich entworfenen Navigation. Kurz vor Fertigstellung der Arbeit äußerten die Mitarbeiter noch kleinere Änderungswünsche in Bezug auf die Navigation und der damit verbundenen Bedienbarkeit. Diese Änderungen wurden noch durchgeführt, allerdings erlaubte es der Zeitdruck nicht mehr, nochmals alle Navigationsmodelle und deren Beschreibung anzupassen.

Insgesamt läßt sich sagen, daß sich die verwendeten Methoden und Techniken sehr gut für die Erstellung einer Webapplikation eignen. Besonders bei der Implementierung zählen sich die Vorteile der strukturierten Vorgehensweise aus.

Ein Punkt, der sich im Laufe der Diplomarbeit noch als kritisch herausstellte, war die Gratwanderung zwischen dem wissenschaftlichen Anspruch, den die Universität an eine Diplomarbeit stellt und den Vorgaben des Unternehmens. Limitierender Faktor war bei der Erstellung der Diplomarbeit vor allen die finanziellen Vorstellungen des Unternehmens für die Realisierung. Was die Betreuung von Seiten Medialab angeht, wurde sich viel Mühe gegeben. Allerdings gab es für das interne Projekt nie eine

wirkliche Projektleitung, so daß ich zwangsläufig von in Auftrag geben der Hardware für den Server über die komplette Installation und Administration aller benötigten Komponenten bis hin zum Test der Applikation alles selber in die Hand genommen habe. Teilweise stand auf Unternehmensseite auch kein entsprechendes Know-How zur Verfügung. Einerseits war die dadurch entstehende Abwechslung ganz angenehm, andererseits mußte ich besonders für Installation und Administration viel Zeit aufwenden und wurde von meinen eigentlichen Aufgaben abgehalten.

Eine Weiterentwicklung des Systems ist vorerst nicht geplant, da alle benötigten Anforderungen erfüllt sind und momentan im Unternehmen auch keine entsprechenden Ressourcen zur Verfügung stehen. Ob und wie das System später weiterentwickelt wird, hängt von der sich gerade im Umbruch befindenden IT-Infrastruktur ab.

ANHANG

Literaturverzeichnis

- [Apa99] **Apache Webserver**, The Apache Software Foundation
<http://www.apache.org>
- [BED97] **The Practical SQL Handbook**
Bowman, Emerson, Darnovsky – Reading, Mass.: Addison Wesley
Longman 1998 ISBN 0-201-44787-8
- [BKM99] **Towards a UML Extension for Hypermedia Design**
Hubert Baumeister, Nora Koch, Luis Mandel – Institut für Informatik,
Ludwig-Maximilians-Universität München 1999
- [DG98] **Dynamic HTML: The Definitive Reference**
Danny Goodman – Sebastopol: O'Reilly & Associates 1998
ISBN 1-56592-494-0
- [Eil98] **Apache Web-Server**
Lars Eilebrecht – Bonn, International Thompson Publishing 1998
ISBN 3-8266-0438-5
- [Gem99] **Gemstone**, Gemstone Systems
<http://www.gemstone.com>
- [HC98] **Java Servlet Programming**
Jason Hunter, William Crawford – Sebastopol: O'Reilly & Associates
1998 ISBN 1-56592-391-X
- [Hei98] **Linux Systemadministration**
Jochen Hein – Bonn, Reading, Mass.: Addison Wesley Longman 1998
ISBN 3-8273-1206-X
- [Jac+92] **Object Oriented Software Engineering: A Use Case Driven Approach**,
Ivar Jacobsen, M. Christerson, P. Jonsson, G. Overgaard – New York:
ACM Press 1992
- [JAp99] **Apache JServ**, The Apache Software Foundation
<http://java.apache.org>
- [JDK99] **Java Development Kit**, Sun Microsystems Inc.
<http://java.sun.com>
- [JSDK99] **Java Servlet Development Kit**, Sun Microsystems Inc.
<http://java.sun.com/products/servlet>
- [Lot99] **Lotus Notes / Domino**, Lotus Development Corporation
<http://www.lotus.com>
- [LP96] **Internet-Server**
Cricket Liu, Jerry Peek – Sebastopol: O'Reilly & Associates 1996

-
- [Min97] *ISBN 3-930673-17-7*
JavaScript 1.2
Stefan Mintert – Bonn, Reading, Mass.: Addison Wesley Longman 1997
ISBN 3-8273-1285-x
- [MM99] **JDBC Driver for MySQL**, M. Mathews
<http://www.worldserver.com/mm.mysql/>
- [NP97] **Java – Expedition ins Programmierreich**
Niemeyer, Peck – Bonn: O'Reilly & Associates 1997
ISBN 3-930673-52-5
- [OCL97] **The Object Constraint Language: The Expression Language for UML**
IBM Corporation 1997, <http://www.software.com/ad/satandards/ocl.html>
- [Paw98] **Wissensmanagement: Erfahrungen und Perspektiven**
Peter Pawlowski – Wiesbaden: Gabler 1998 ISBN 3-409-18974-2
- [Ree97] **Database Programming with JDBC and Java**
George Reese – Sebastopol: O'Reilly & Associates 1997
ISBN 1-56592-270-0
- [RS95] **The Object Oriented Hypermedia Design Model**,
Daniel Schwabe, Gustavo Rossi – *Comm. Of the ACM*, Vol. 38, Aug. 1995
- [RS98] **Developing Hypermedia Applications using OOHD**,
Daniel Schwabe, Gustavo Rossi – 1998
- [Rule98] **Dynamic HTML: The HTML Developers Guide**
Jeff Rule – Reading, Mass.: Addison Wesley Longman 1998
ISBN 0-201-37961-9
- [Rum+91] **Object Oriented Modeling and Design**
Rumbaugh, Blaha, Premerlani, Eddy, Lorensen – Prentice Hall 1991
- [SchEi97] **Intranets: Firmeninterne Informationssysteme mit Internet-Technologie**
Daniel Schätzler, Fabian Eilingsfeld – Heidelberg: dPunkt-Verlag 1997
ISBN: 3-920993-70-5
- [SM99] **Java: Programmierhandbuch und Referenz**
Reiner Singer, Stefan Middendorf – Heidelberg: dPunkt-Verlag 1999
ISBN: 3-920993-82-9
- [SuSE99] **SuSE Linux Distribution**, SuSE GmbH
<http://www.suse.de>
- [TcX99] **MySQL Database**, T.c.X DataKonsultAB
<http://www.mysql.com>
- [UML97] **Unified Modeling Language**, Rational Software Corporation
Rumbaugh, Booch, Jacobsen
<http://www.rational.com/uml/resources/documentation/index.jtмл>
- [Wiki99] **Wiki Wiki Web**, Cunningham & Cunningham Inc.
<http://www.c2.com/cgi/wiki?WelcomeVisitors>
-

Installationsanleitung

Um das entwickelte System zu installieren müssen neben den Java-Klassen noch andere Komponenten installiert werden. Diese Komponenten sind, sofern nicht anders beschrieben, als TAR-Archiv mit auf der CD enthalten. Für die Installation der einzelnen Komponenten ist den entsprechenden Installationsanleitungen Folge zu leisten. Die Dateien befinden sich im Verzeichnis `tarfiles`

Komponenten:

- **Java Development Kit 1.1.7**
Datei: `jdk_1_1_7-v3-glibc-x86.tar`
- **Java Servlet Development Kit 2.0**
Datei: `jsdk20-solaris2-sparc.tar`
- **MySQL Database 3.22.25**
Die Datenbank liegt als gezipptes TAR-Archiv vor (gzip !)
Datei: `mysql-3.22.25-pc-linux-gnu-i686.tar.gz`
- **MySQL Database JDBC Driver 1.2**
Datei: `mm.mysql.jdbc-1.2b.tar`
- **Apache Web Server 1.3.6**
Datei: `apache_1_3_6.tar`
- **Apache JServ 1.0b4**
Datei: `Apache-JServ-1.0b4.tar`

Neben diesen Komponenten sind auf der CD noch die Quelldateien der Applikation vorhanden. Sie befinden sich im Verzeichnis `source`.

Die Datenbankdefinition ist ebenfalls mit auf der CD. Die Dateien befinden sich im Verzeichnis `medialab`. Diese Daten (das ganze Verzeichnis) muß in das MySQL `data`-Verzeichnis kopiert werden.