

**Entwurf und Implementierung eines Administrationssystems zur
bibliothekarischen Verwaltung elektronischer Publikationen an
einer wissenschaftlichen Großforschungseinrichtung**

Diplomarbeit

zur Erlangung des Grades Diplom-Informatikerin (FH)

an der

Hochschule Bremerhaven

Fachbereich Informatik/Wirtschaftsinformatik

Studiengang mit Schwerpunkt Informatik

vorgelegt von:

Verena Graßmann

Matr.-Nr.: 021179

aus:

Bürgermeister-Smidt-Strasse 75

27568 Bremerhaven

Tel.: (0471/45332)

vgrassmann@awi-bremerhaven.de

Prüfer: Prof. Dr. Günter Matthiessen

Korreferentin: Dr. Ana Macario

Bremerhaven, den 02.10.03

Danksagung

Mit dieser Danksagung möchte ich mich bei allen Personen bedanken, die mich sowohl fachlich als auch persönlich bei der Erstellung dieser Arbeit unterstützt haben.

Ich danke Prof. Dr. Günter Matthiessen und Dr. Ana Macario für die Betreuung meiner Diplomarbeit.

Ich danke Maren Lütjen, die wahrscheinlich gar nichts davon weiß.

Ganz besonders danke ich meinen Eltern, ohne die mein Studium nicht möglich gewesen wäre.

Ein weiterer ganz besonderer Dank geht an Karen Mindermann, die mich in allen Lebenslagen unterstützte. Besonders lecker war das Tschakka-Pralinchen.

Bei Sebastian bedanke ich mich für die wirklich freundliche Unterstützung, er war niemals genervt.

Ich danke Kim.

Und zum Schluss danke ich natürlich dem großen Honk.

Inhaltsverzeichnis:

Abbildungsverzeichnis: vi

Tabellenverzeichnis: vii

Abkürzungsverzeichnis viii

Danksagung i

1 Einleitung 1

1.1 Aufgabenstellung 2

1.2 Aufbau der Arbeit 3

1.3 Ziel der Arbeit 4

2 Anforderungen an das System 5

2.1 Ist-Situation bestehendes Programm ePIC 5

2.2 Funktionalitäten 5

2.3 Berechtigte Benutzer 5

2.4 Verständlichkeit 6

2.5 Wartbarkeit und Erweiterbarkeit 6

2.6 Internationalisierung 6

3 Verwendete Systeme 7

3.1 LDAP 7

3.2 Directory 7

3.2.1 Directory Information Tree (DIT) 7

3.2.2 Distinguished Name 7

3.2.3 Objektklassen und Attribute 8

3.2.4 Schema 9

3.2.4.1 AWI-Schema 9

3.2.5 LDAP-Operationen 9

3.2.6	Interaktion zwischen LDAP-Client und -Server.....	9
3.3	Publikation.....	10
3.3.1	Publicationentry	10
3.4	Java Servlets	14
3.5	Dynamische Webseiten.....	15
4	<u>Arbeits- bzw. Entwicklungsumgebung am AWI.....</u>	16
4.1	Dokumentation der Hardware- und Softwarekonfiguration am AWI	16
4.1.1	Serverseitige Konfiguration.....	16
4.1.1.1	Webserver	16
4.1.1.2	Directoryserver	16
4.1.1.2.1	Zugriffskontrolle	17
4.1.1.2.2	Authentifizierung	17
4.1.1.3	Java-Entwicklungsumgebung	18
4.1.1.3.1	Plattform	18
4.1.1.3.2	Werkzeug	18
4.1.1.3.3	Servlet-API (Application Programming Interface).....	18
4.1.1.4	Netscape LDAP SDK	18
4.1.1.5	Javascript	20
4.1.1.6	GUI-Komponenten	20
4.1.2	Clientseitige Konfiguration.....	20
4.2	Client-Server Kommunikation.....	21
4.2.1	Intranet	21
5	<u>OOA (Objekt Orientierte Analyse).....</u>	22
5.1	Pflichtenheft.....	22
5.1.1	Zielbestimmung	22
5.1.1.1	Muss-Kriterien	22
5.1.1.2	Kann-Kriterien	22
5.1.1.3	Abgrenzungskriterien.....	22
5.1.2	Einsatz.....	23
5.1.2.1	Anwendungsbereiche.....	23
5.1.2.2	Zielgruppen	23

5.1.2.3	Betriebsbedingungen	23
5.1.3	Umgebung.....	23
5.1.3.1	Software	23
5.1.3.2	Hardware.....	23
5.1.3.3	Orgware	23
5.1.4	Funktionalität	23
5.1.5	Daten.....	24
5.1.6	Leistungen.....	24
5.1.7	Benutzungsoberfläche.....	24
5.1.8	Qualitätsziele	24
5.2	OOA-Modell.....	25
5.2.1	Beschreibung der Geschäftsprozesse.....	26
5.2.1.1	Use-Case „Publikation suchen“	26
5.2.1.2	Use-Case „Attribute hinzufügen“	28
5.2.1.3	Use-Case „Attribute korrigieren“	30
5.2.1.4	Use-Case „Attribute löschen“	32
5.2.1.5	Use-Case „Publikation sperren“	34
5.2.1.6	Use-Case „Publikation entsperren“	36
5.3	Prototyp der Benutzeroberfläche	37
5.3.1	Dialoggestaltung	37
5.3.2	Fenster.....	38
5.3.3	Interaktionselemente.....	39
5.3.4	Gestaltung von Fenstern	39
5.3.5	Layout	40
6	<u>OOD (Objekt Orientiertes Design)</u>	41
6.1	OOD-Modell.....	41
6.1.1	Klassenstruktur von ePIC Admin	41
6.2	Architekturentwurf.....	42
6.3	Implementierungsentwurf.....	42
6.3.1	Implementierung vorhandener Klassen	42
6.3.2	UI- Klassen	43
6.3.2.1	GeneralUI.....	43

6.3.2.2	LoginUI.....	44
6.3.2.3	SearchUI	44
6.3.2.4	ModifyUI	45
6.3.3	Main	46
6.3.4	Properties-Dateien	46
<u>7</u>	<u>Implementierung.....</u>	<u>48</u>
7.1	Berechtigte Personen	48
7.2	Authentifizierung und Autorisierung.....	48
7.3	Session	48
7.4	Programmablauf.....	49
7.4.1	Loginmaske.....	49
7.4.2	Suchmaske	50
7.4.3	Ergebnisanzeige und Modifikationsmaske	51
7.4.4	Detailansicht	53
7.4.5	Erfolgs- bzw. Fehlermeldungen.....	54
<u>8</u>	<u>Resümee.....</u>	<u>55</u>
<u>9</u>	<u>Glossar</u>	<u>56</u>
<u>10</u>	<u>Literaturverzeichnis.....</u>	<u>60</u>

Abbildungsverzeichnis:

Abbildung 1: Organisation der Stiftung Alfred-Wegener-Institut [AWI 02]	2
Abbildung 2: LDAP-Verzeichnisstruktur	8
Abbildung 3: Beispiel eines Publicationentries im LDAP-Explorer	14
Abbildung 4: Geschäftsprozessdiagramm der relevanten Geschäftsprozesse	25
Abbildung 5: Klassendiagramm „Publikation suchen“	26
Abbildung 6: Sequenzdiagramm „Publikation suchen“	27
Abbildung 7: Klassendiagramm „Attribute hinzufügen“	28
Abbildung 8: Sequenzdiagramm „Attribute hinzufügen“	29
Abbildung 9: Klassendiagramm „Attribute korrigieren“	30
Abbildung 10: Sequenzdiagramm „Attribute korrigieren“	31
Abbildung 11: Klassendiagramm „Attribute löschen“	32
Abbildung 12: Sequenzdiagramm „Attribute löschen“	33
Abbildung 13: Klassendiagramm „Publikationen sperren“	34
Abbildung 14: Sequenzdiagramm „Publikationen sperren“	35
Abbildung 15: Klassendiagramm „Publikationen entsperren“	36
Abbildung 16: Sequenzdiagramm „Publikationen entsperren“	37
Abbildung 17: Zustandsdiagramm Dialogstruktur der Bibliotheksanwendung	38
Abbildung 18: Suchmaske	38
Abbildung 19: Modifikationsmaske	39
Abbildung 20: Klassenstruktur ePIC Admin	41
Abbildung 21: Drei-Schichten-Architektur	42
Abbildung 22: Loginmaske	49
Abbildung 23: Suchmaske	50
Abbildung 24: Ergebnisanzeige und Modifikationsmaske	52
Abbildung 25: Detailansicht	53
Abbildung 26: Erfolgsmeldung	54
Abbildung 27: Fehlermeldung	54

Tabellenverzeichnis:

Tabelle 1: Klasse GeneralUI.....	43
Tabelle 2: Klasse LoginUI.....	44
Tabelle 3: Klasse SearchUI.....	44
Tabelle 4: Klasse ModifyUI	46
Tabelle 5: Klasse Main	46

Abkürzungsverzeichnis

10k	10000
AWI	Alfred-Wegener-Institut für Polar- und Meeresforschung
ACI	Access Control Information
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CGI	Common Gateway Interface
CN	Common name
CSS	Cascading Style Sheets
DC	Domain Component
DIB	Directory Information Base
DIT	Directory Information Tree
DN	Distinguished Name
ePIC	electronic Publication Information Center
GB	GigaByte
GUI	Graphical User Interface
HGF	Hermann von Helmholtz-Gemeinschaft Deutscher Forschungszentren
HTML	Hypertext Markup Language
J2SE	Java 2 Platform Standard Edition
JDK	Java Development Kit
LDAP	Lightweight Directory Access Protocol
MAC	Macintosh
MHZ	Megahertz
NT	New Technology
OOA	Objekt Orientierte Analyse
OOD	Objekt Orientiertes Design
OS	Operation System
OU	Organizational Unit
PC	Personal Computer
RDN	Relative Distinguished Name
SASL	Simple Authentication and Security Layer
SDK	Software Development Kit

SN	Surname
SSL	Secure Sockets Layer
SUN	Stanford University Network
UI	User Interface
UID	User Identifier
URL	Uniform Resource Locator
XP	eXPerience

1 Einleitung

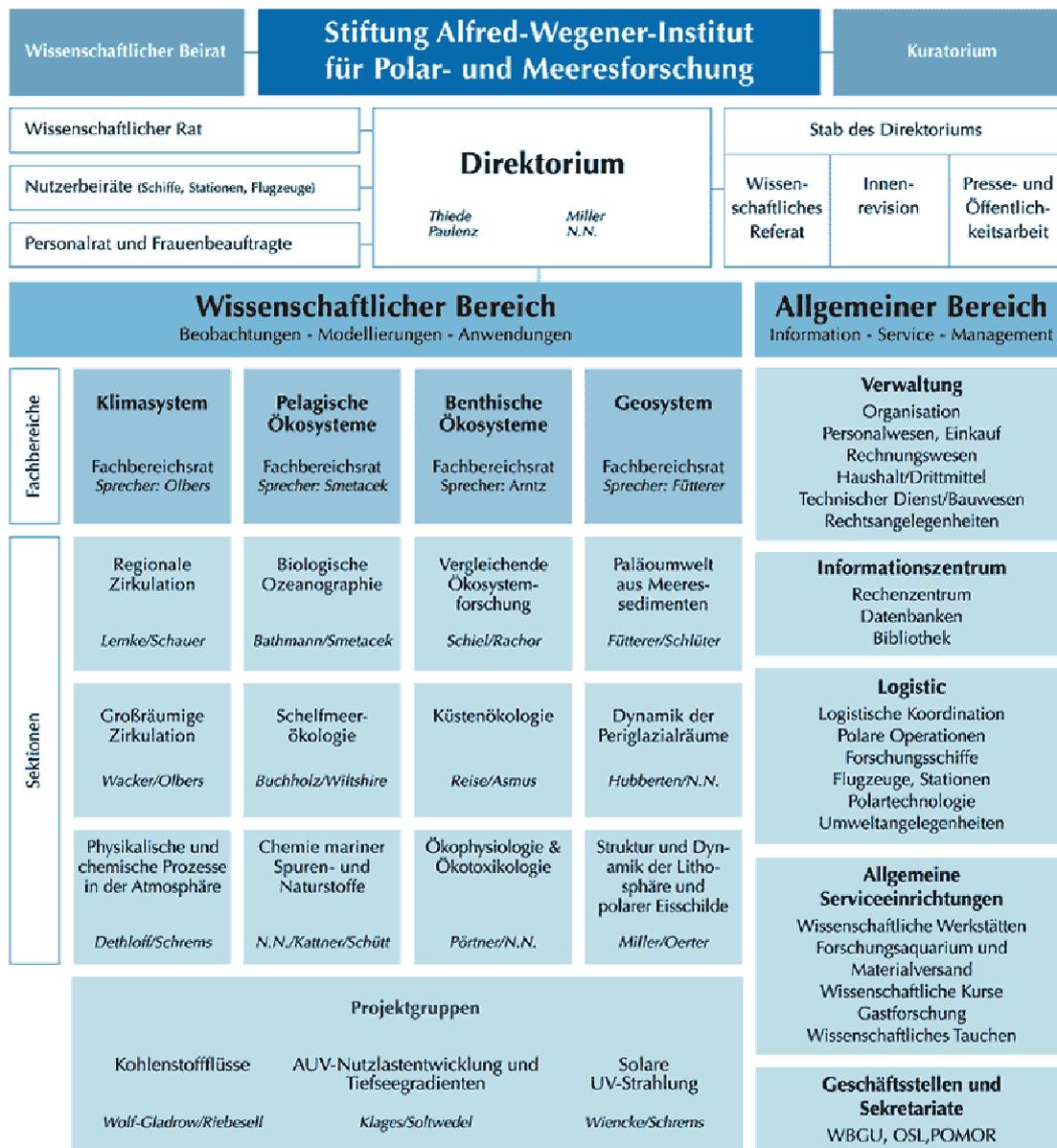
Die vorliegende Diplomarbeit wurde am Alfred-Wegener-Institut für Polar- und Meeresforschung (AWI) Bremerhaven erstellt. Der im Rahmen dieser Diplomarbeit entwickelte Programmprototyp soll nach der Implementierung in der Bibliothek des AWI eingesetzt werden.

Das Alfred-Wegener-Institut für Polar- und Meeresforschung wurde 1980 als Stiftung des öffentlichen Rechts gegründet. Die Stiftung umfasst das Alfred-Wegener-Institut für Polar- und Meeresforschung in Bremerhaven, die Wattenmeerstation Sylt, die Forschungsstelle Potsdam, welche 1992 zum Institut hinzukam und die Biologische Anstalt Helgoland, die seit 1998 zum Institut dazugehört. Die Stiftung ist Mitglied der Hermann von Helmholtz-Gemeinschaft Deutscher Forschungszentren (HGF) und umfasst die vier wissenschaftlichen Fachbereiche Klimasystem, Pelagische Ökosysteme, Benthische Ökosysteme und Geosystem, die wiederum in Sektionen untergliedert sind.

In der wissenschaftlichen Arbeit des Alfred-Wegener-Instituts werden die Beziehungen zwischen Ozean, Eis und Atmosphäre sowie der Tier- und Pflanzenwelt der Arktis und Antarktis erforscht. Zudem wird die Entwicklungsgeschichte der polaren Kontinente und Meere untersucht.

Die Bibliothek des Alfred-Wegener-Instituts ist dem allgemeinen Bereich Informationszentrum zugeordnet.

Sie ist eine Spezialbibliothek, die in erster Linie der Literaturversorgung der Mitarbeiter der Stiftung dient. In die Bibliothek integriert ist auch der Bestand der ehemaligen Bibliothek der Biologischen Anstalt Helgoland. Bei den Bibliotheksbeständen handelt es sich hauptsächlich um wissenschaftliche Publikationen aus den verschiedensten Forschungsbereichen, wie z.B. Marine Biologie und Ökologie, Meeresbotanik, Ozeanographie, Marine Zoologie, u.s.w..
[AWI02]



Stand 02/03

Abbildung 1: Organisation der Stiftung Alfred-Wegener-Institut [AWI 02]

1.1 Aufgabenstellung

Die Aufgabe der vorliegenden Diplomarbeit besteht darin, ein Administrationssystem zu programmieren, mit dessen Hilfe Publikationsdaten aufgerufen und anschließend bearbeitet werden können. Die gefilterten Daten können auf verschiedene Weise von einem Bibliothekar des AWI modifiziert werden.

Bei den elektronischen Publikationen handelt es sich um Daten, die über eine Publikation eines AWI-Mitarbeiters in elektronischer Form gespeichert werden.

Modifikationen der Publikationsattribute sind möglich sowie das Hinzufügen von Attributen zu diesen Publikationen, aber auch das Löschen eines Attributes oder einer kompletten Publikation. Mit dem System könnte nunmehr eine gewünschte Anzahl bzw. Auswahl an Publikationen geändert werden. Das System ist also nicht auf das Korrigieren einer einzelnen Publikation pro Vorgang beschränkt. Die Publikationsverwaltung wird auf diese Weise weiter optimiert. Korrekturen einzelner Publikationen werden zur Zeit mit dem Programm ePIC vorgenommen, dessen Hauptaufgabe es ist, neue Publikationen anzulegen. Klassen von ePIC sind teilweise implementiert worden, auf die im weiteren Verlauf noch näher eingegangen wird.

1.2 Aufbau der Arbeit

Nach der Einleitung, die sowohl die Aufgabenbeschreibung als auch den Aufbau und das Ziel der Arbeit enthält, folgt das Kapitel Anforderungen an das System. In diesem Kapitel wird zunächst die derzeitige Ist-Situation beschrieben sowie anschließend die Funktionalitäten des zukünftigen Systems.

In Kapitel drei werden die verwendeten Systeme für die Umsetzung des Prototypen beschrieben.

Kapitel vier beschreibt die Arbeits- bzw. Entwicklungsumgebung am AWI. Hier wird die Hardware- und Softwarekonfiguration dokumentiert, die serverseitige und clientseitige Konfiguration werden beschrieben, sowie die Client-Server Kommunikation.

In Kapitel fünf wird eine objektorientierte Analyse durchgeführt, die ein Pflichtenheft beinhaltet und ein OOA-Modell.

Im sechsten Kapitel wird das objektorientierte Design beschrieben, es wird hier auf das OOD-Modell eingegangen und auf den Implementierungsentwurf, in welchem die Programmklassen dargestellt werden.

Im Anschluss daran wird der Implementierungsentwurf vorgestellt.

Die Diplomarbeit wird mit einem Resümee abgeschlossen.

1.3 Ziel der Arbeit

Ziel ist es, die Korrektur für elektronische Publikationen zu vereinfachen.

Im Laufe der Jahre haben sich in der Publikationsverwaltung Fehler eingeschlichen durch z.B. Umbenennen von Zeitschriftennamen usw..

Dem Bibliothekar soll es ermöglicht werden, alle von ihm ausgewählten Publikationen mit einem Vorgang zu modifizieren.

2 Anforderungen an das System

2.1 Ist-Situation bestehendes Programm ePIC

Das bestehende Programm ePIC ist bisher die einzige Möglichkeit, um nachträgliche Korrekturen bzw. Änderungen an den Publikationen vorzunehmen, jedoch mit der Einschränkung, zumindest aus bibliothekarischer Sicht, dass nur eine Publikation pro Korrekturvorgang geändert werden kann. ePIC dient hauptsächlich der Neueintragung von Publikationen und ist nicht speziell für die Korrektur von bereits eingetragenen Publikationen vorgesehen.

2.2 Funktionalitäten

Das zu entwickelnde Programm, genannt ePIC Admin, ist speziell für die Korrektur der Publikationen ausgelegt, das bedeutet es kann ein Attribut modifiziert, hinzugefügt oder gelöscht werden für alle vom Benutzer gesuchten bzw. ausgewählten Publikationen. Wurde z.B. ein Zeitschriftenname durch einen Autor anders als gefordert eingegeben, so kann solch ein Eintrag kaum durch eine Suche gefunden werden. Beim Suchergebnis nämlich werden die Publikationen mit der abweichenden Zeitschriftenbezeichnung nicht aufgezeigt. Der Bibliothekar kann nun mit Hilfe von ePIC Admin den abweichenden Zeitschriftennamen für alle entsprechenden Publikationen auf einmal ändern. Neben den Funktionen: Suche von Einträgen, Modifizieren, Hinzufügen und Löschen von Publikationsattributen gibt es die Funktion der Löschung einer kompletten Publikation, also nicht nur deren Attribute.

Die o.g. Funktionalitäten werden auf später folgenden Seiten näher erläutert.

2.3 Berechtigte Benutzer

Die im Directoryserver unter dem Gruppenzweig PublicationGroup eingetragenen Personen sind die einzigen berechtigten Benutzer für ePIC Admin. Sie müssen sich authentifizieren, bevor sie Korrekturen an den Publikationseinträgen vornehmen können. Dementsprechend müssen im Programm entsprechende Rechte vergeben werden.

2.4 Verständlichkeit

Der Programmcode soll leicht verständlich sein, um mit wenig Zeitaufwand Änderungen bzw. Anpassungen vornehmen zu können.

2.5 Wartbarkeit und Erweiterbarkeit

Der Programmcode soll flexibel gestaltet werden. Programmpfadänderungen bzw. Host- oder Portänderungen sollen auf unkomplizierte Art durchgeführt werden können, ohne dass für eine Änderung der gesamte Quellcode durchgesehen werden muss. Die Klassennamen sollen denen bereits bestehender Klassen anderer Java-Anwendungen, die eine ähnliche Funktion haben, angepasst werden. Z.B. soll eine Klasse die Modifikationen durchführt ModifyUI heißen.

2.6 Internationalisierung

Die Anwendung soll die Möglichkeit einer deutschen bzw. englischen Sprachauswahl bieten.

3 Verwendete Systeme

3.1 LDAP

LDAP (Lightweight Directory Access Protocol) ist ein Kommunikationsprotokoll, das den Transport und das Format von Nachrichten, die zwischen einem Client und einem Verzeichnis (Directory) ausgetauscht werden, definiert. LDAP definiert nicht das Directory selbst.

LDAP unterstützt somit Verzeichnisdienste. Es wurde als offener Standard für globale und lokale Verzeichnisdienste in Netzwerken entwickelt.

3.2 Directory

In einem Directory (Verzeichnis) werden Informationen, bzw. Daten gespeichert. Es handelt sich bei einem Directory um eine Datenbank, die ihre Daten in Form eines hierarchischen Datenbaumes speichert.

Die gesamte Informationsmenge, die in einem Directory steht, wird als Directory Information Base (DIB) bezeichnet. Die Directory Information Base besteht aus einzelnen Verzeichniseinträgen, sogenannten Entries, welche in einer hierarchischen Beziehung zueinander stehen.

3.2.1 Directory Information Tree (DIT)

Der hierarchische Datenbaum des Verzeichnisses, in dem alle Entries (Einträge) eingetragen sind, wird als Directory Information Tree bezeichnet.

In ihm lässt sich jeder Eintrag eindeutig über seinen DN (Distinguished Name) identifizieren.

3.2.2 Distinguished Name

Jeder Entry kann eindeutig über seinen Distinguished Name identifiziert werden.

Dieser DN gibt an, wo sich ein Entry innerhalb des Verzeichnisbaumes (Directory Information Tree) befindet. Jeder DN besteht aus mehreren Teilen, der Wurzel und den Relative Distinguished Names (RDNs). Alle Namen in einem Verzeichnis stellen den Namensraum (Namespace) dar.

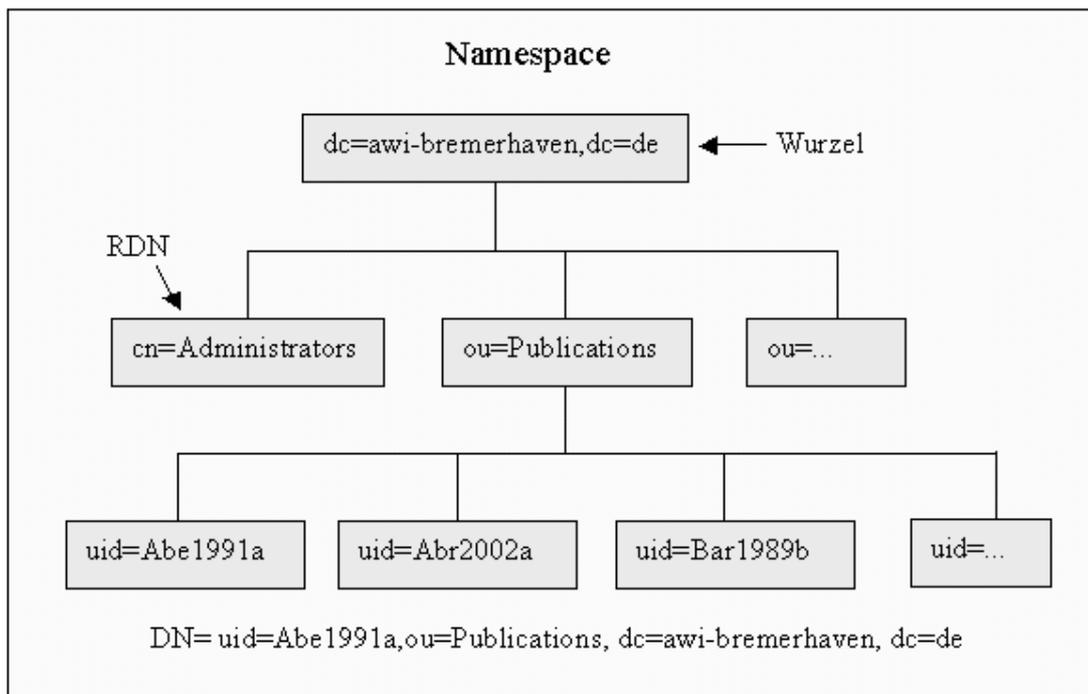


Abbildung 2: LDAP-Verzeichnisstruktur

3.2.3 Objektklassen und Attribute

Ein Entry ist eine Menge von Informationen über ein einzelnes Objekt, er ist eine konkrete Instanz einer Objektklasse bzw. eine Instanz von kombinierten Objektklassen. Als Beispiel ein Personenentry: dieser Entry enthält den Vornamen einer Person, den Nachnamen, die E-Mail-Adresse usw..

Eine Objektklasse ist eine logische Klassifizierung von individuellen Objekten mit einheitlichen Eigenschaften und Strukturen. Die Informationen über ein einzelnes Objekt werden in Attributen gespeichert. Um welche Attribute es sich handelt, ist von den Objektklassen abhängig. Die Objektklassen verfügen über eine Liste von Attributen die zwingend notwendig sind (required) und die optional zur Verfügung stehen (allowed).

Abstrakte Objektklassen dienen als Superklasse oder Template. Sie werden dazu benutzt, um aus ihnen andere Objektklassen abzuleiten. Alle abgeleiteten Klassen haben dann die Attribute der abstrakten Klasse gemeinsam. Abstrakte Objektklassen haben keine Entries. Top ist solch eine abstrakte Klasse.

3.2.4 Schema

Das Verzeichnisschema ist eine zusammenfassende Beschreibung, die definiert, wie die Daten im Verzeichnis abgelegt werden müssen. Es beschreibt, welche Objektklassen im Verzeichnis erlaubt sind, welche Attribute vorhanden sein müssen, bzw. sein dürfen und wie die Attribute definiert sein müssen. Die Daten werden in Form von Verzeichniseinträgen gespeichert. Jeder Eintrag ist ein Satz von Attributen und Werten. Außerdem muss jeder Eintrag eine Objektklasse haben. Das Standardschema kann auf eigene Bedürfnisse angepasst, bzw. erweitert werden, wie z.B. neue Objektklassen anlegen.

3.2.4.1 AWI-Schema

Das am AWI bestehende Schema ist sehr umfangreich. Zu den Standard-Objektklassen sind zusätzliche Objektklassen, auf die Bedürfnisse des AWI angepasst, angelegt worden. Das gleiche gilt für die Attribute.

3.2.5 LDAP-Operationen

LDAP bietet verschiedene Operationen, um Entries zu finden, zu lesen und zu bearbeiten. Es handelt sich hierbei um die Suche nach Entries, das Hinzufügen, Löschen, Ändern und das Vergleichen von Entries sowie das Ändern des DN oder des RDN eines Entries.

3.2.6 Interaktion zwischen LDAP-Client und -Server

Für eine Interaktion zwischen LDAP-Client und LDAP-Server wird eine Session errichtet (Binding). Für diese Verbindung muss der Hostname und die Port-Nummer angegeben werden. Mit dem Usernamen und dem entsprechenden Passwort kann sich der Client (User) authentifizieren. Anschließend kann er, bei entsprechender Berechtigung, Einträge suchen, auslesen und/oder bearbeiten. Hat der Client alle gewünschten Aktionen ausgeführt, schließt er die Sitzung (Unbinding).

3.3 Publikation

Eine Publikation ist als Entry im Directoryverzeichnis gespeichert und besteht aus Objektklassen und Attributen.

3.3.1 Publicationentry

Der Publicationentry ist eine Instanz aus kombinierten Objektklassen. Im folgenden sind dies:

objectclass top

requires

- objectClass
Objektklassendefinition

allows

- aci
Access Control Information

objectclass person

superior top

requires

- sn,
Surname
- cn
Common name

allows

- description,
- seeAlso,
Verweis auf AWI-Arbeitsbereich des Nutzers (Internetadresse)
- telephoneNumber,
- userPassword

objectclass organizationalPerson

superior person

allows

-
- destinationIndicator,
 - facsimileTelephoneNumber,
 - internationaliSDNNumber,
 - l
Ort (l = location)
 - ou,
Organisationseinheit (ou = organizational unit)
 - physicalDeliveryOfficeName,
 - postOfficeBox,
 - postalAddress,
 - postalCode,
 - preferredDeliveryMethod,
 - registeredAddress,
 - st,
 - street,
 - teletexTerminalIdentifier,
 - telexNumber,
 - title,
Titel (Prof., Dr.)
 - x121Address

objectclass publicationclass

superior top

allows

- discipline,
- platformdn,
- platformdiscipline,
- projectdn,
- sensor,
- changetype,
- createtimestamp,
Zeitpunkt der Publikationserstellung

-
- creatorsname,
Uid des Erstellers.
 - cruiseleg,
 - modifiersname,
Name der Person, welche zuletzt eine Änderung am Eintrag vorgenommen hat.
 - modifytimestamp,
Zeitpunkt der letzten Änderung
 - ownersname,
 - publicationauthor,
Erster Autor der Publikation.
 - publicationauthoraffiliation,
 - publicationauthoremail,
 - publicationawi,
Wert für AWI-Publikation ja oder nein
 - publicationawinumber,
AWI-Nummer für eine Publikation
 - publicationbegutachtet,
begutachtet ja oder nein
 - publicationcitation,
 - publicationcoauthor;alternate
Hier werden alle weiteren Autoren gespeichert.
 - publicationdataset,
 - publicationemail,
 - publicationidentifier,
ISBN oder ISSN.
 - publicationlock,
Wert wird gesetzt, wenn die Publikation vom Bibliothekar auf bibliographische Richtigkeit überprüft worden ist und danach nicht mehr geändert werden kann.
 - publicationlockedby,
Uid der Person, welche die Publikation gesperrt hat.

-
- publicationpatentroyalties,
 - publicationpsabstract,
 - publicationpscruise,
 - publicationpsleg,
 - publicationsource,
Quellenangabe zur Publikation
 - publicationsourceisi,
 - publicationstatus,
Publikationsstatus (Einreichung beantragt, eingereicht, im Druck,
veröffentlicht)
 - publicationtag,
 - publicationtitle,
Publikationstitel.
 - publicationtype,
Publikationstyp (Artikel, Buchkapitel, Buch, Konferenzbeitrag, Manual,
Patent, Dissertation, Habilitation)
 - publicationurl,
Url zum Publikationsvolltext
 - publicationyear,
Jahr der Veröffentlichung
 - thesisacceptedby,
ausstellende Hochschule
 - uid

Ein Publikationentry setzt sich also folgendermaßen zusammen:

Von der abstrakten Objektklasse top leiten sich die Objektklassen person und publicationclass ab und die Klasse OrganizationalPerson wiederum von der Klasse person.

Aus jeder dieser Klassen, abgesehen von der abstrakten Klasse, können Attribute in einem Publikationseintrag auftreten.



Abbildung 3: Beispiel eines Publicationentries im LDAP-Explorer

3.4 Java Servlets

Java Servlets sind Java-Komponenten, die auf einem Java-basierenden Web-Server laufen und das spezifische Interface `javax.servlet.Servlet` implementieren. Nach Kompilierung des Servlets wird es beim ersten Aufruf auf dem Server gestartet. Ab diesem Zeitpunkt wartet das Programm auf eingehende Requests. Ein Servlet wird von einem Client aufgerufen und nimmt POST- oder GET-Anfragen von diesem entgegen. Das Servlet besitzt zwei Interfaces, über die es mit dem Client kommuniziert, zum einen das `HttpServletRequest` und zum anderen das `HttpServletResponse`. Aufgrund der Anfragen eines Clients nimmt das Servlet auf dem Server eine entsprechende Auswertung vor. Als Output erzeugt das Servlet Zeichenketten, die an den Client zurückgesendet werden. Auf Clientseite werden dann HTML-Dokumente (HTML = Hypertext Markup Language) erzeugt, sofern die Zeichenketten `print`-Befehle enthalten. Servlets haben kein eigenes Graphisches UserInterface (GUI).

Die Servlet-API bietet Funktionen zur Unterstützung von Cookies und damit zur Session-Verwaltung. Auf diese Weise lassen sich einer Benutzer-Session Java-Objekte auf dem Server zuordnen.

3.5 Dynamische Webseiten

Die dynamischen Inhalte der Webseite werden durch Abfragen des Directoryservers generiert. Im Gegensatz zu statischen Webseiten können so kurzfristig die Inhalte aktualisiert werden, was für diese Anwendung von großer Wichtigkeit ist, da die Daten einer ständigen Änderung unterliegen können, sei es durch den Autor, die Administratoren, den Bibliothekar usw. Es soll daher immer nur mit den aktuellen Daten gearbeitet werden.

4 Arbeits- bzw. Entwicklungsumgebung am AWI

Im folgenden wird die für diese Diplomarbeit wichtige Hardware- bzw. Softwarekonfiguration beschrieben. Das AWI besitzt diverse Server, Domänen, usw., welche jedoch für diese Diplomarbeit nicht relevant sind und deshalb nicht näher beschrieben werden.

4.1 Dokumentation der Hardware- und Softwarekonfiguration am AWI

4.1.1 Serverseitige Konfiguration

Die Domäne e-bck mit dem Testwebserver iPlanet 4.1 Service Pack 7 und dem Directoryserver sind eingerichtet unter dem Enterprise 10000 Server. Die Domäne ist ausgestattet mit zwei 333 MHz Prozessoren und 2 GB Hauptspeicher.

4.1.1.1 Webserver

Der iPlanet Webserver unterstützt zusätzlich zu den Standards HTML und SSL (Secure Sockets Layer) Server Side Java (Servlets), serverbasiertes Javascript, CGI (Common Gateway Interface), Schnittstellen zu verschiedenen Datenbanken usw.. Mit Hilfe dieser Möglichkeiten wurde die vorliegende Arbeit, bzw. der Prototyp des Programms, entwickelt. [Softguide03]

4.1.1.2 Directoryserver

Der eingesetzte Directoryserver am AWI ist ebenfalls von der Firma iPlanet. Es handelt sich hierbei um den iPlanet Directory Server in der Version 5.1. Er stellt bestimmte Dienste (Directoryservices) zur Verfügung, wie z.B. ein Zugriff eingeschränkt wird, wann und wie die Daten aktualisiert werden usw.. Im Verzeichnis des Directoryservers werden diverse AWI Informationen und Daten abgelegt, auf welche in den folgenden Seiten noch näher eingegangen wird.

4.1.1.2.1 Zugriffskontrolle

Access Control Information (ACI) regeln die Zugriffsrechte auf die Daten des Directoryservers. Es können Rechte für das gesamte Verzeichnis vergeben werden, für einzelne Zweige und für einzelne Einträge. Die entsprechenden Rechte werden für Benutzergruppen angegeben. Die gesetzten Rechte können hierbei sein:

- Read, Leserecht
- Write, Schreibrecht
- Add, Recht Einträge bzw. Attribute hinzuzufügen
- Delete, Löschrecht
- Search, Suchrecht
- Compare, Vergleichrecht von Einträgen
- Selfwrite, Schreibrecht für den eigenen Eintrag
- All, alle Rechte.

Die einzelnen Rechte sind unabhängig voneinander, ein Benutzer mit Schreibrechten muss für den gleichen Eintrag nicht automatisch Löschrechte besitzen.

Die Access Control Information sind in einer Access Control List aufgelistet.

4.1.1.2.2 Authentifizierung

Um sich dem Directoryserver gegenüber zu authentifizieren, bestehen drei Möglichkeiten:

- Einfache, passwortbasierte Authentifizierung
Bei dieser Art der Authentifizierung wird der DN des Benutzers und sein Passwort unverschlüsselt an den Server weitergegeben.
- Zertifikatbasierte Authentifizierung
Mit Hilfe der zertifikatbasierten Authentifizierung werden die Verbindungen zum Directoryserver über SSL hergestellt. Der Client stellt ein Zertifikat zur Identifizierung.
- Authentifizierung unter Benutzung von SASL Mechanismen
Hier wird mit Hilfe von SASL (Simple Authentication and Security Layer) Mechanismen eine Sicherheitsschicht für die Verbindung zwischen Client und Directoryserver hergestellt.

Da die Anwendung dieser Diplomarbeit nur für das Intranet gedacht ist, also kein Zugriff von außen (Internet) stattfindet, wird die erste Methode angewandt und mit Hilfe von SSL verschlüsselt.

4.1.1.3 Java-Entwicklungsumgebung

Mit Hilfe der im folgenden genannten Entwicklungsumgebung wurde die Java-Anwendung programmiert.

4.1.1.3.1 Plattform

Als Plattform diente die Java 2 Platform Standard Edition (J2SE) mit dem Java Development Kit 1.2.2_08 (JDK).

4.1.1.3.2 Werkzeug

Als Programmierwerkzeug diente der Borland JBuilder6 Personal.

4.1.1.3.3 Servlet-API (Application Programming Interface)

Es sind APIs (Schnittstellen) nötig um Kommunikationsverbindungen bzw. Zugriffe zu ermöglichen.

Mit Hilfe der Servlet-API werden Servlets erstellt, welche eine Kommunikation zwischen Webclient und Webserver möglich machen.

4.1.1.4 Netscape LDAP SDK

Mit dem Netscape LDAP Software Development Kit (SDK) kann auf den Directoryserver zugegriffen werden.

Es stellt die Klassen bzw. Methoden zur Verfügung, mit denen Einträge im Verzeichnis gesucht, geändert und gelöscht werden können und neue Einträge angelegt werden. Für diese Aktionen muss eine Verbindung zum Directory hergestellt werden, dies geschieht mit connect unter Angabe von Hostnamen und Portnummer als Parameter. Nach der Connection erfolgt die Authentifizierung mit

dem DN der Person die sich authentifizieren möchte und dem dazugehörigen Passwort. Nach der Authentifizierung wird die Methode für eine entsprechende Aktion angegeben: für das Hinzufügen eines Attributes die Methode `add()`, für das Ändern die Methode `modify()` und für das Löschen `delete()`. Es werden den Methoden Parameter für die Einträge mitgegeben.

Beispiel für das Modifizieren eines Eintrags:

```
ldcon = new LDAPConnection();

// Verbindung zum Server
ldcon.connect (Hostname, Port);

// Authentifizieren gegenüber dem Server
String DN = „uid=Person x“;
String PW = “xyz”;
ldcon.authenticate (DN, PW);

// Modifikation des Eintrags
ldcon.modify(EntryDN, mods);

// an dieser Stelle ist eine Ausgabe ganz sinnvoll
System.out.println(„Der Eintrag wurde erfolgreich geändert.“);

// Ein Exception Handling wird durchgeführt
catch (LDAPException e)...

// Nach Abschluss der Aktion wird die Verbindung geschlossen
ldcon.disconnect();
```

4.1.1.5 Javascript

Javascript ist eine Skriptsprache. Sie kann als Ergänzung zu HTML angesehen werden. Javascriptprogramme werden nicht kompiliert, sondern als Quelltext zur Laufzeit interpretiert.

Javascript wurde benutzt für Hinweis- bzw. Meldungsfenster und zur Markierung von Checkboxes.

4.1.1.6 GUI-Komponenten

Für die graphische Oberfläche wurde die Markierungssprache HTML benutzt. HTML beschreibt den Aufbau eines Dokuments mit Absätzen, Überschriften, Verweisen zu anderen Dokumenten (Links) usw.. HTML besteht aus einer Reihe von ASCII-Zeichen (ASCII = American Standard Code for Information Interchange), die vom Webbrowser als Formatierungsbefehle für das Seitenlayout übersetzt werden. Die Befehle werden als Tags (Markierungen) in das Dokument eingesetzt. Um ein einheitliches Bild mit den bereits bestehenden AWI-Java-Anwendungen zu erhalten, wurden Cascading Style Sheets (CSS) eingesetzt.

4.1.2 Clientseitige Konfiguration

Es sind unterschiedliche Profile auf Clientseite gegeben. Bei den Rechnern handelt es sich um Sun mit den Betriebssystemen Solaris 2.8 oder 2.9, um Personal Computer mit den Betriebssystemen Windows NT, 2000 oder XP oder um MAC mit dem Betriebssystem MacOS. Als Browser kommen in Frage: Netscape, Microsoft Internet Explorer oder Mozilla.

4.2 Client-Server Kommunikation

4.2.1 Intranet

Da es sich bei ePIC Admin um eine interne Anwendung für die Bibliothekare handelt, also von außen über das Internet kein Zugriff auf das Programm bestehen soll, ist das Intranet die Verbindung zwischen Client und Server.

Das Intranet bildet die hauptsächliche Informationsquelle für die Mitarbeiter.

Es gibt schwarze Bretter, einen Terminkalender, diverse Informationen zu PC Grundlagen usw..

5 OOA (Objekt Orientierte Analyse)

Die Objektorientierte Analyse dient dazu, die Anforderungen des Auftraggebers (AWI) an das Softwaresystem zu ermitteln und zu beschreiben. Das Ergebnis aus dieser Analyse ist das OOA-Modell. Die OOA beschreibt nicht das fertige Endprodukt, da bei der Entwicklung der Anwendung immer wieder neue Ideen einfließen können. Es können also durchaus Abweichungen zum späteren Objektorientierten Design bzw. zur Implementierung bestehen.

5.1 Pflichtenheft

5.1.1 Zielbestimmung

5.1.1.1 Muss-Kriterien

- Auswählen von Publikationsattributen
- Anzeigen der Attribute
- Modifizieren der Attribute
- Hinzufügen von Attributen zu Publikationen
- Löschen von Attributen
- Sperren der Publikationen
- Entsperrern der Publikationen

5.1.1.2 Kann-Kriterien

- Internationalisierung

5.1.1.3 Abgrenzungskriterien

- Keine

5.1.2 Einsatz

5.1.2.1 Anwendungsbereiche

- Bibliothek

5.1.2.2 Zielgruppen

- „Bibliothekare“

5.1.2.3 Betriebsbedingungen

- Tägliche Nutzung durch Bibliothek

5.1.3 Umgebung

5.1.3.1 Software

- iPlanet Directory Server 4.1
- iPlanet Webserver 4.1sp7
- Java JDK 1.2.2_008

5.1.3.2 Hardware

- Server Plattform: SUN Enterprise 10k
- Client Plattform: Browser

5.1.3.3 Orgware

- LDAP-Schema
- Benutzer-, Organisations-, Publikationsdaten im Directory Server

5.1.4 Funktionalität

Die Funktionalität des Systems umfasst im Wesentlichen die Korrektur von Publikationen. Dabei stehen die Abläufe Hinzufügen, Modifizieren und

Löschen von Publikationsattributen bei Publikationen im Vordergrund. Bei allen ausgewählten Publikationen wird das gewählte Attribut gleichzeitig korrigiert.

5.1.5 Daten

Die Daten betreffen alle Publikationen der am AWI beschäftigten Mitarbeiter. Die Zahl der Publikationen ist ständig steigend.

5.1.6 Leistungen

Die zu korrigierenden Publikationen sollen innerhalb von maximal 3 Sekunden auf dem Benutzbildschirm angezeigt werden.

5.1.7 Benutzungsoberfläche

Die Benutzungsoberfläche wird als Webanwendung auf einem Browser dargestellt. Die Bedienung soll eindeutig und selbsterklärend sein. Für eventuell auftretende Fragen soll ein Hilfesystem in Form eines „Hilfe-Buttons“ integriert sein.

5.1.8 Qualitätsziele

- Ausgabe der Webanwendung auf verschiedenen Plattformen darstellbar
- Hohe Benutzungsfreundlichkeit
- Hohe Änderbarkeit des Quellcodes

5.2 OOA-Modell

Das OOA-Modell ist die fachliche Lösung des zu realisierenden Systems. Es besteht aus einem statischen und dynamischen Modell. Das statische Modell beschreibt insbesondere die Klassen. Das dynamische Modell zeigt die Funktionsabläufe unter anderem mit Hilfe von Geschäftsprozessen.

Es lassen sich 6 Geschäftsprozesse identifizieren.

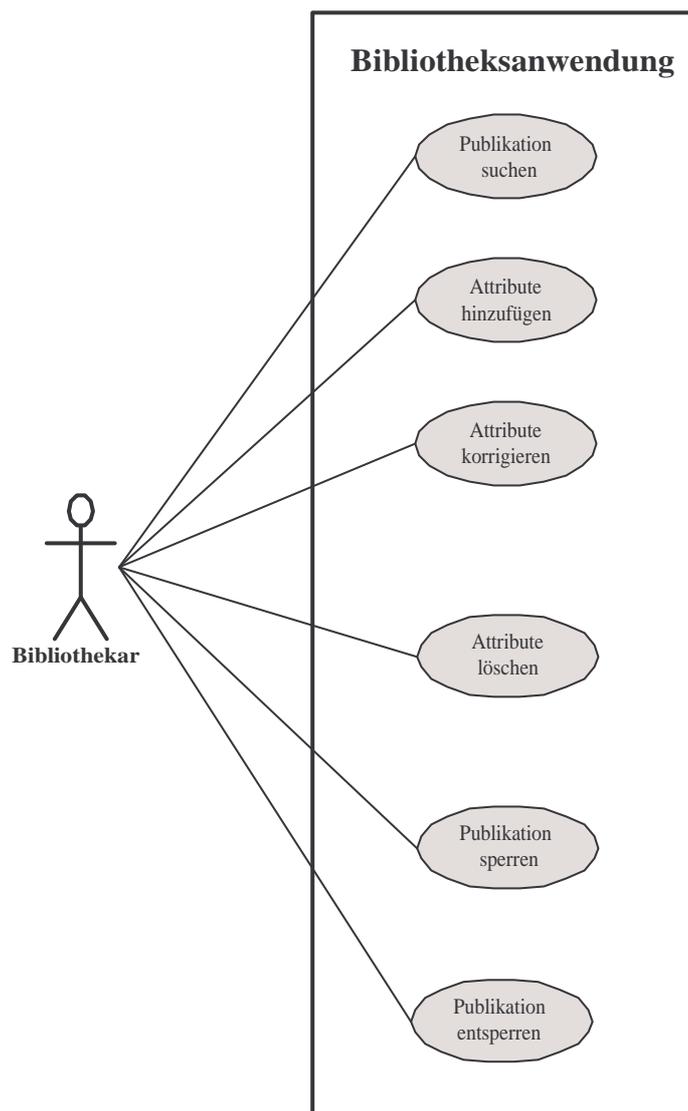


Abbildung 4: Geschäftsprozessdiagramm der relevanten Geschäftsprozesse

5.2.1 Beschreibung der Geschäftsprozesse

5.2.1.1 Use-Case „Publikation suchen“

Geschäftsprozess	Publikation suchen
Ziel	Gewünschte Publikation, die geändert werden soll, herausfinden
Kategorie	Primär
Vorbedingung	Vorhandene Publikation
Nachbedingung Erfolg	Ausgewählte Publikation anzeigen
Nachbedingung Fehlschlag	Falsch programmiert
Akteur	Bibliothekar
Auslösendes Ereignis	Bibliothekar möchte Attribute hinzufügen, korrigieren oder löschen
Beschreibung	<ol style="list-style-type: none"> 1. Auswahl der Attribute mit logischen Verknüpfungen 2. Anzeige der Publikation 3. Hinzufügen von Attributen zur Publikation, korrigieren oder löschen der Attribute
Erweiterungen	- Keine -
Alternativen	- Keine -

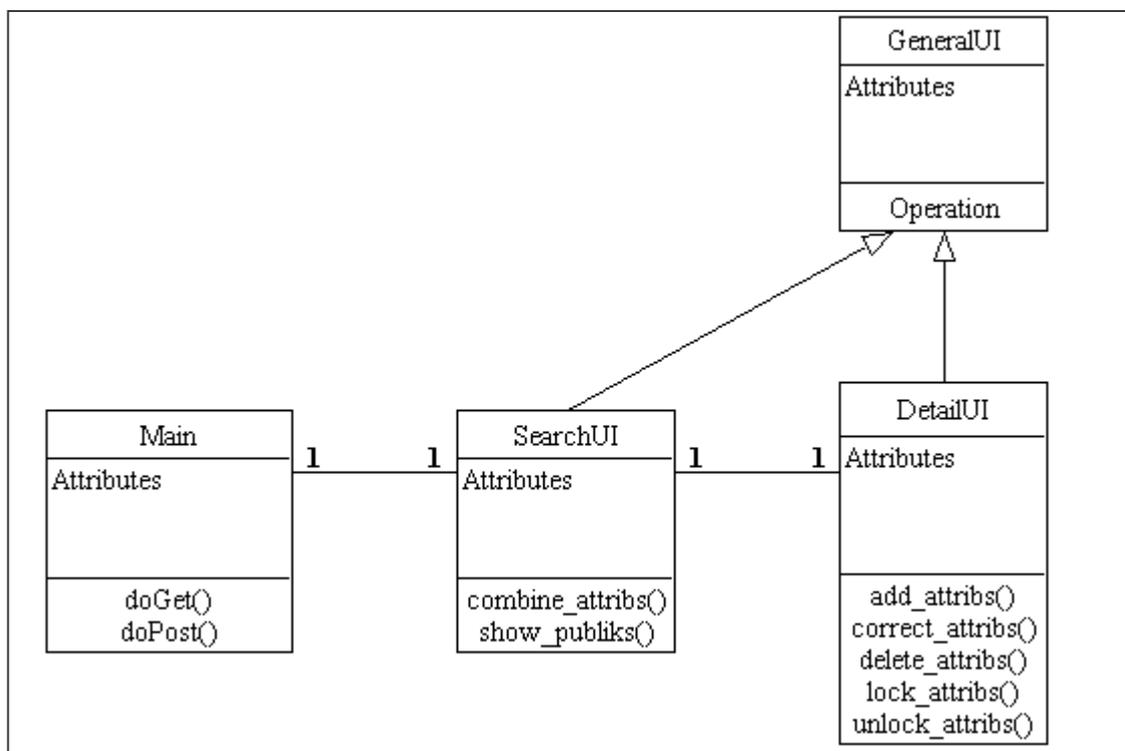


Abbildung 5: Klassendiagramm „Publikation suchen“

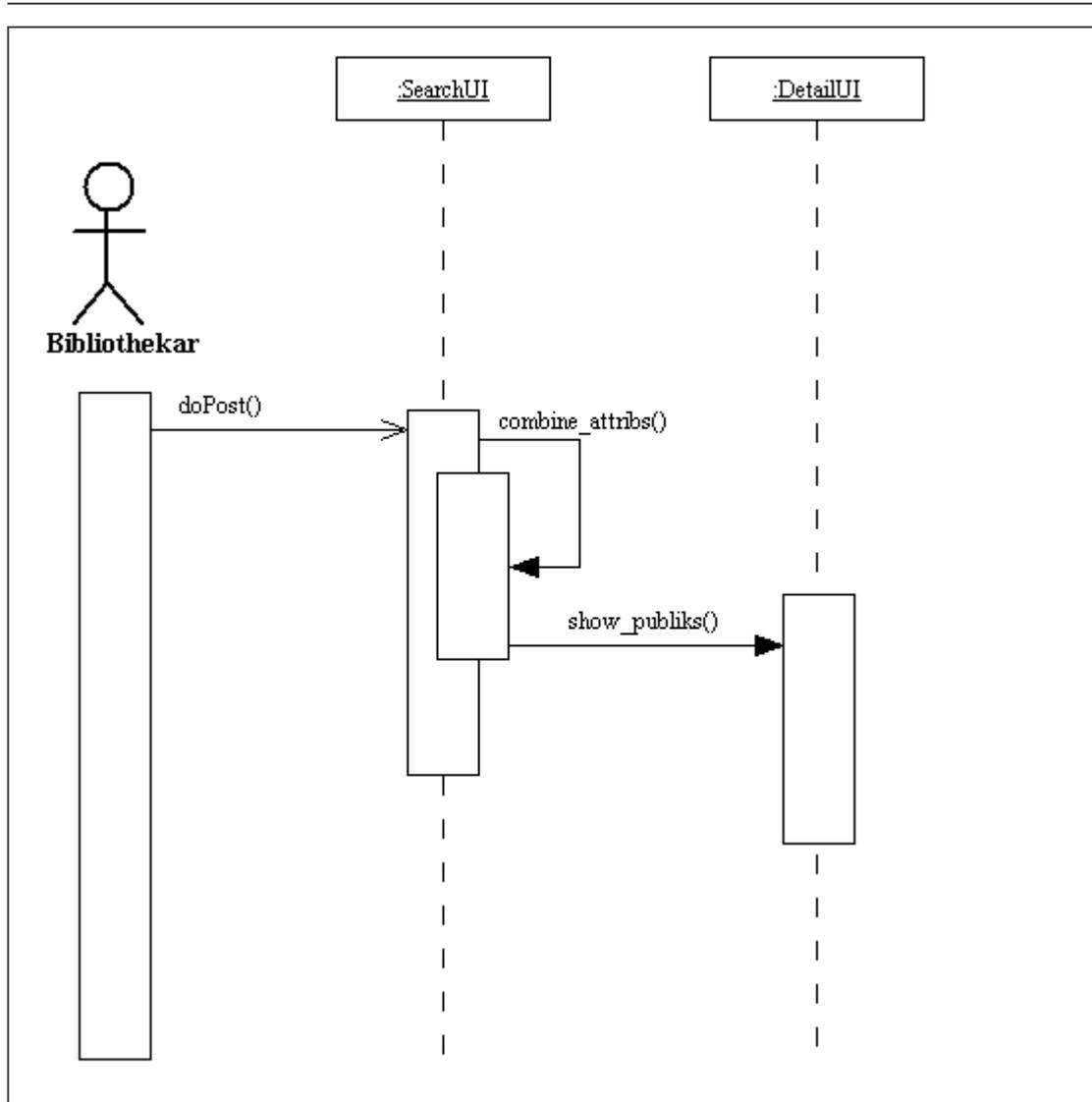


Abbildung 6: Sequenzdiagramm „Publikation suchen“

5.2.1.2 Use-Case „Attribute hinzufügen“

Geschäftsprozess	Attribute hinzufügen
Ziel	Publikation ändern, indem Attribute hinzugefügt werden
Kategorie	Primär
Vorbedingung	Auswahl der gewünschten Publikation
Nachbedingung Erfolg	Publikation wird erweitert
Nachbedingung Fehlschlag	Falsch programmiert
Akteur	Bibliothekar
Auslösendes Ereignis	Bibliothekar möchte Attribute zu einer Publikation hinzufügen
Beschreibung	<ol style="list-style-type: none"> 1. Anzeige der Publikation 2. Auswählen der Publikation 3. Auswählen des Attributes, das hinzugefügt werden soll 4. Hinzufügen des Attributes zur Publikation 5. Sperren der Publikation
Erweiterungen	- Keine -
Alternativen	5a Nicht sperren der Publikation 5b Entsperren der Publikation 5c Löschen der Publikation

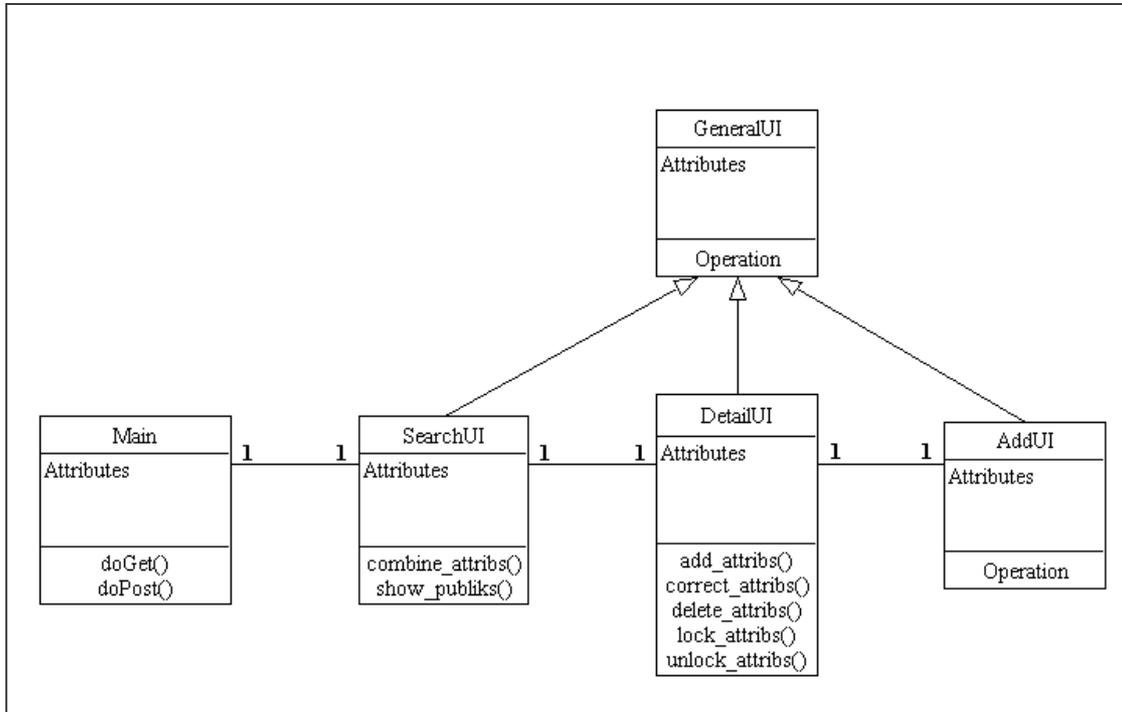


Abbildung 7: Klassendiagramm „Attribute hinzufügen“

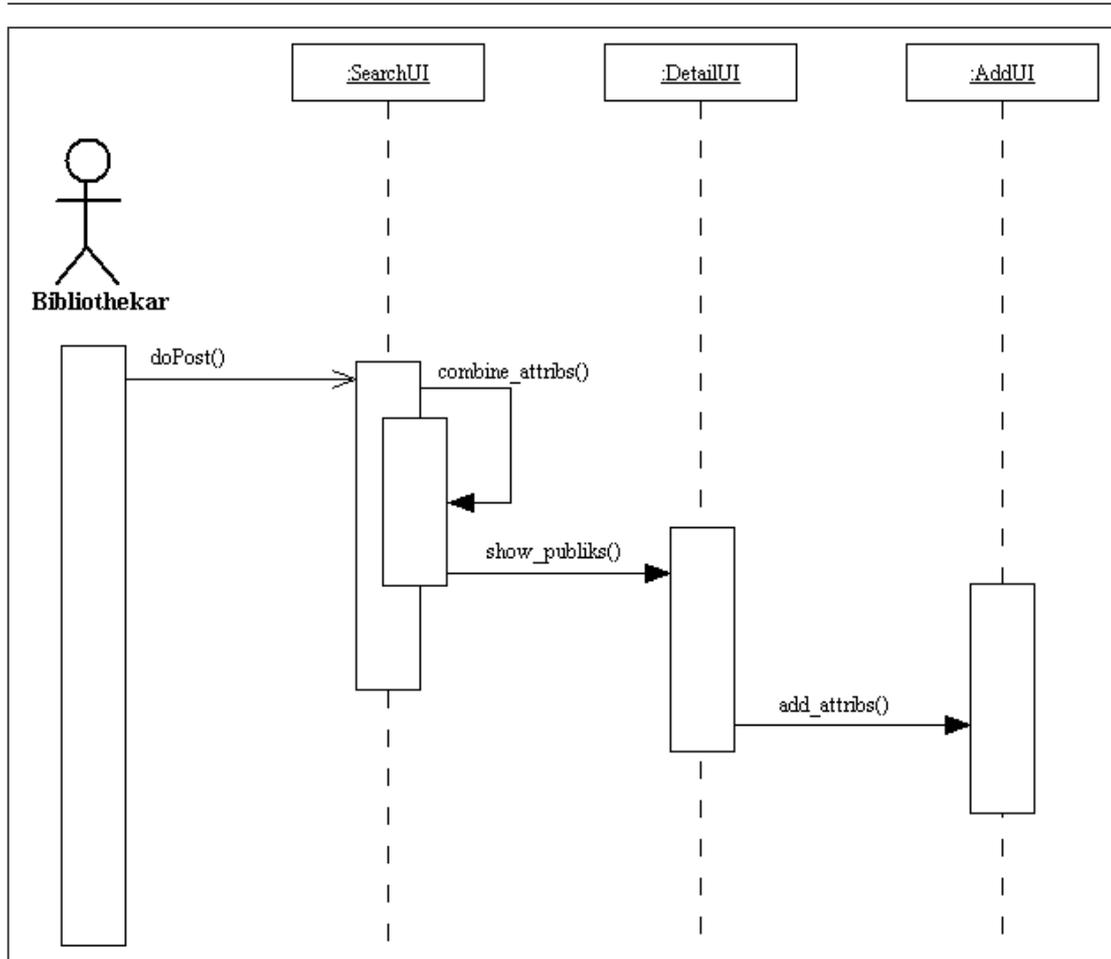


Abbildung 8: Sequenzdiagramm „Attribute hinzufügen“

5.2.1.3 Use-Case „Attribute korrigieren“

Geschäftsprozess	Attribute korrigieren
Ziel	Publikation korrigieren
Kategorie	Primär
Vorbedingung	Auswahl der gewünschten Publikation
Nachbedingung Erfolg	Publikation wird korrigiert
Nachbedingung Fehlschlag	Falsch programmiert
Akteur	Bibliothekar
Auslösendes Ereignis	Bibliothekar möchte Attribute einer Publikation ändern
Beschreibung	<ol style="list-style-type: none"> 1. Anzeige der Publikation 2. Auswählen der Publikation 3. Auswählen des Attributes, das geändert werden soll 4. Ändern des Attributes 5. Sperren der Publikation
Erweiterungen	- Keine -
Alternativen	5a Nicht sperren der Publikation 5b Entsperrn der Publikation 5c Löschen der Publikation

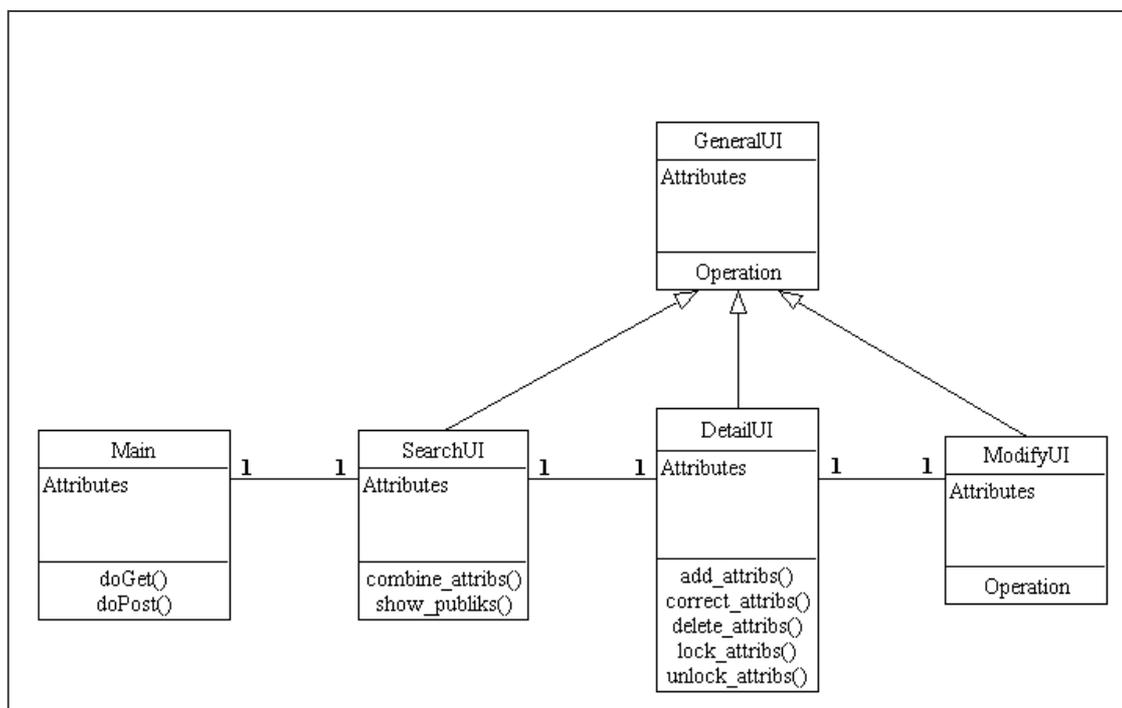


Abbildung 9: Klassendiagramm „Attribute korrigieren“

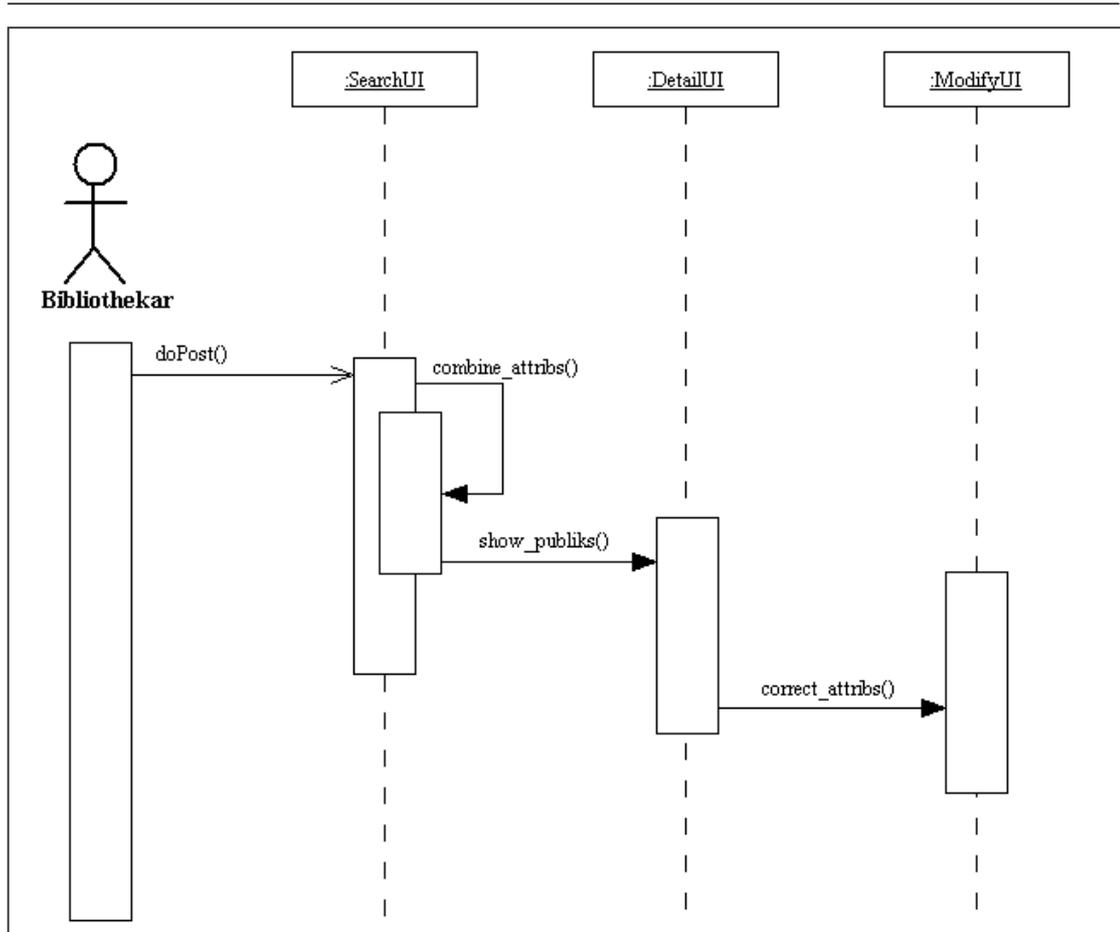


Abbildung 10: Sequenzdiagramm „Attribute korrigieren“

5.2.1.4 Use-Case „Attribute löschen“

Geschäftsprozess	Attribute löschen
Ziel	Publikation ändern durch Entfernen von Attributen
Kategorie	Primär
Vorbedingung	Auswahl der gewünschten Publikation
Nachbedingung Erfolg	Attribut wird gelöscht
Nachbedingung Fehlschlag	Falsch programmiert
Akteur	Bibliothekar
Auslösendes Ereignis	Bibliothekar möchte Attribute einer Publikation löschen
Beschreibung	<ol style="list-style-type: none"> 1. Anzeige der Publikation 2. Auswählen der Publikation 3. Auswählen des Attributes, das gelöscht werden soll 4. Löschen des Attributes 5. Sperren der Publikation
Erweiterungen	- Keine -
Alternativen	5a Nicht sperren der Publikation 5b Entsperren der Publikation 5c Löschen der Publikation

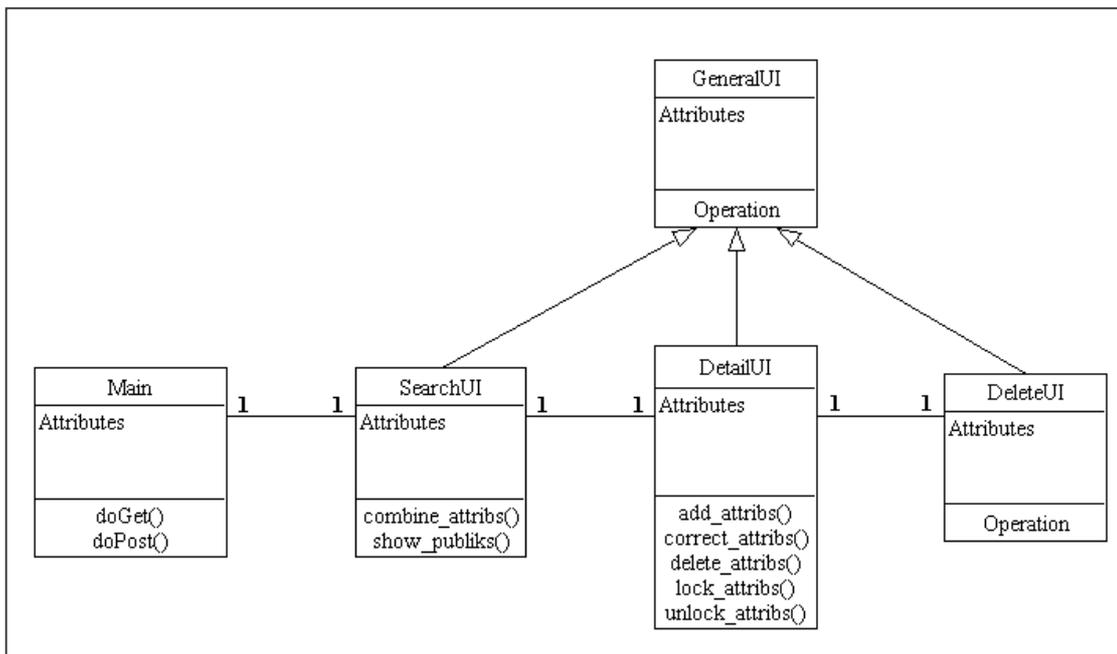


Abbildung 11: Klassendiagramm „Attribute löschen“

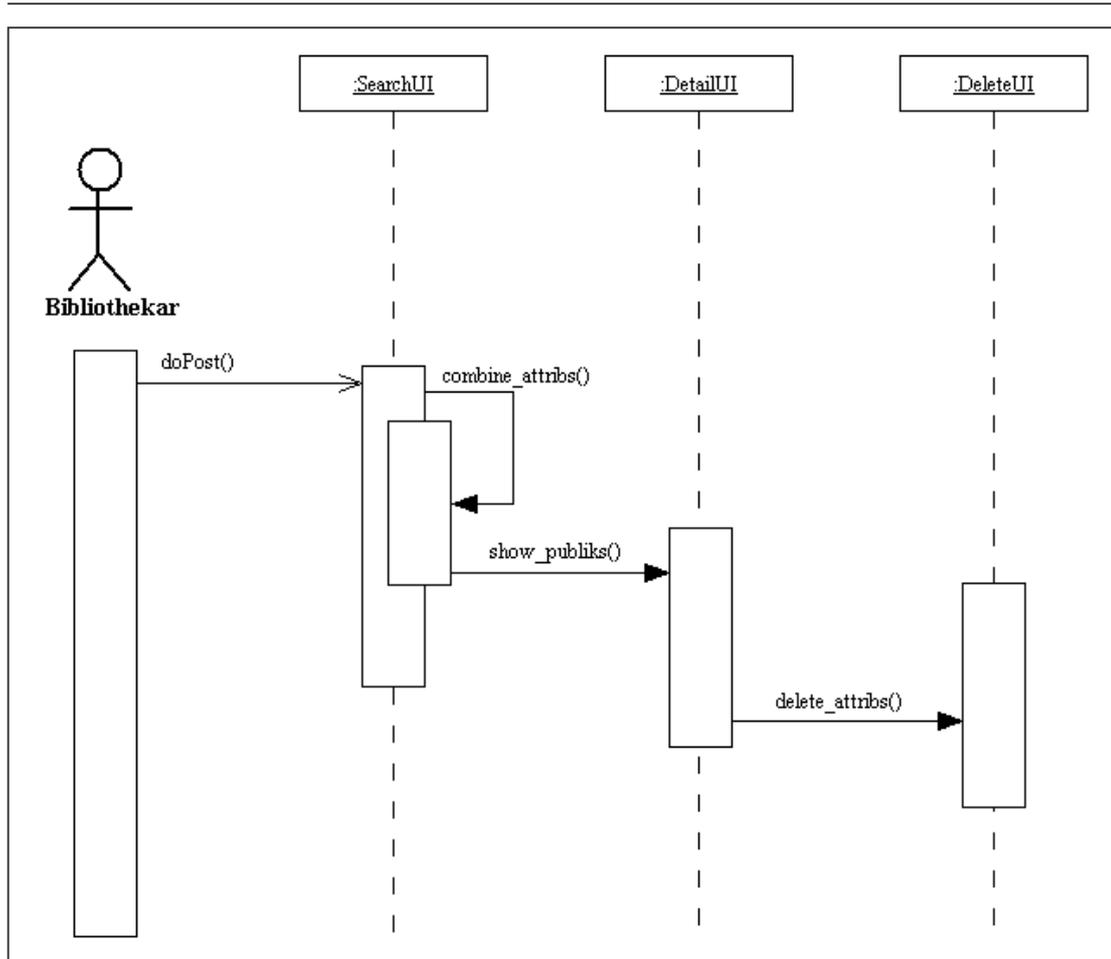


Abbildung 12: Sequenzdiagramm „Attribute löschen“

5.2.1.5 Use-Case „Publikation sperren“

Geschäftsprozess	Publikation sperren
Ziel	Publikation bei bestimmten Attributen nicht mehr zugänglich für Autoren
Kategorie	Primär
Vorbedingung	Auswahl der gewünschten Publikation
Nachbedingung Erfolg	Publikation gesperrt
Nachbedingung Fehlschlag	Falsch programmiert
Akteur	Bibliothekar
Auslösendes Ereignis	Bibliothekar möchte Publikation unzugänglich für Autor machen
Beschreibung	<ol style="list-style-type: none"> 1. Anzeige der Publikation 2. Auswählen der Publikation 3. Sperren der Publikation
Erweiterungen	- Keine -
Alternativen	- Keine -

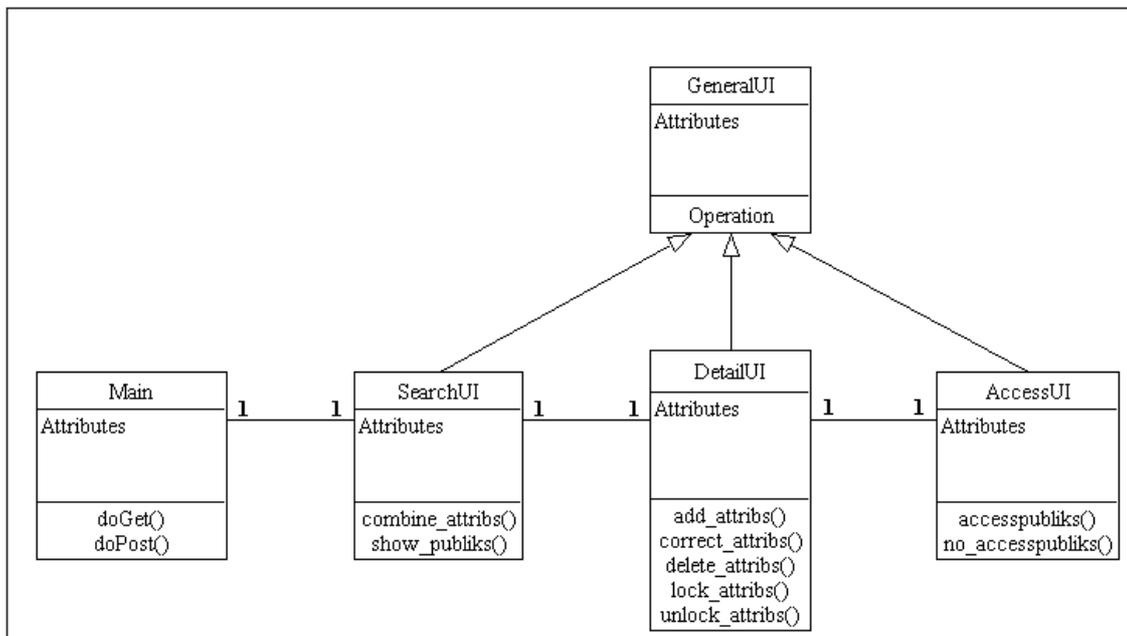


Abbildung 13: Klassendiagramm „Publikationen sperren“

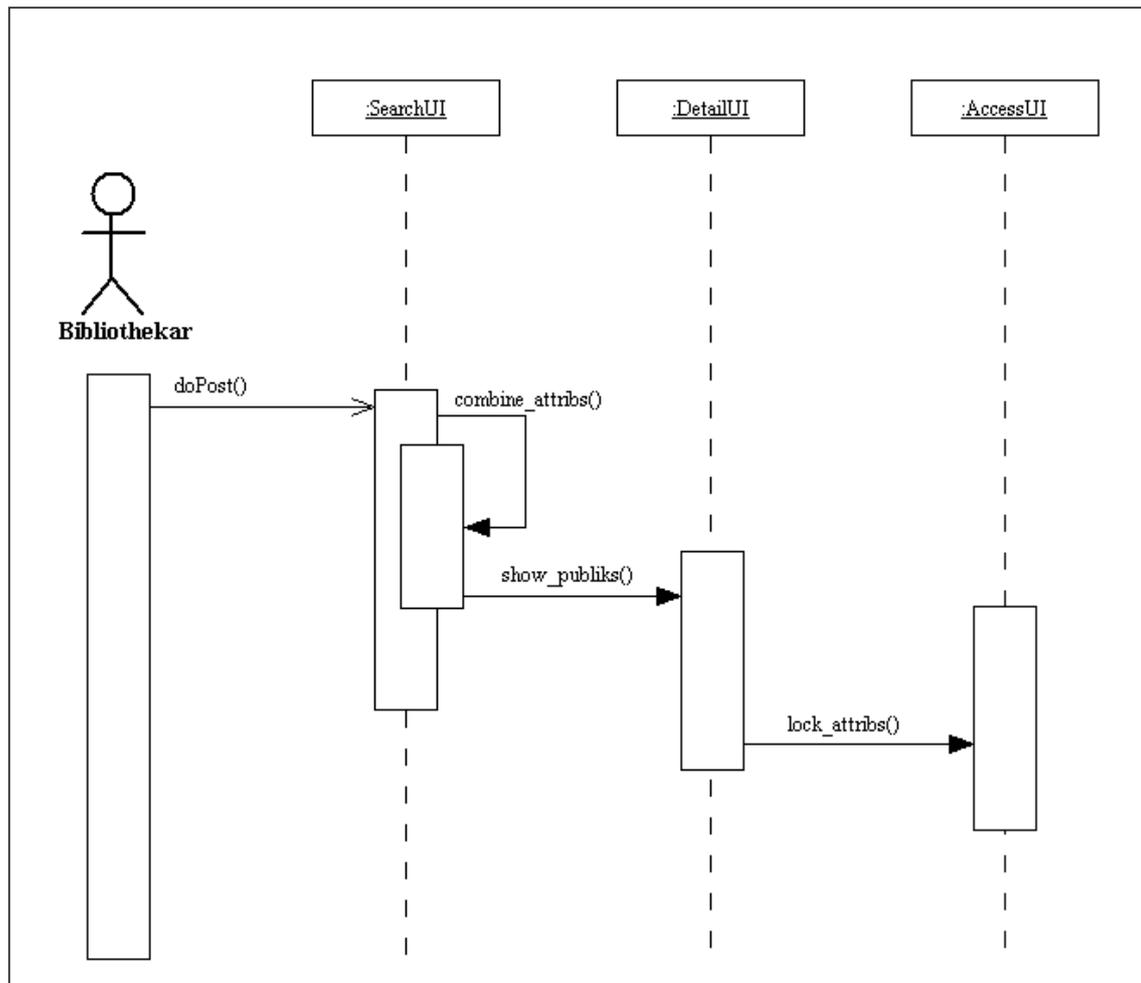


Abbildung 14: Sequenzdiagramm „Publikationen sperren“

5.2.1.6 Use-Case „Publikation entsperren“

Geschäftsprozess	Publikation entsperren
Ziel	Publikation wieder zugänglich für Autoren
Kategorie	Primär
Vorbedingung	Auswahl der gewünschten Publikation
Nachbedingung Erfolg	Publikation entsperrt
Nachbedingung Fehlschlag	Falsch programmiert
Akteur	Bibliothekar
Auslösendes Ereignis	Bibliothekar möchte Publikation wieder zugänglich für Autor machen
Beschreibung	<ol style="list-style-type: none"> 1. Anzeige der Publikation 2. Auswählen der Publikation 3. Entsperren der Publikation
Erweiterungen	- Keine -
Alternativen	- Keine -

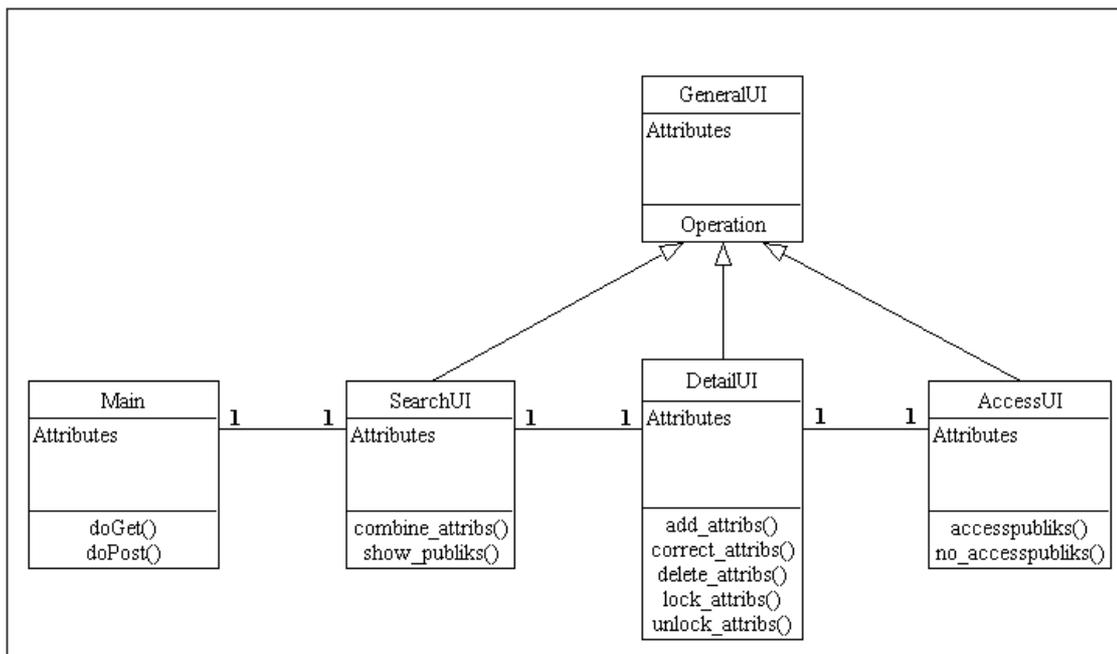


Abbildung 15: Klassendiagramm „Publikationen entsperren“

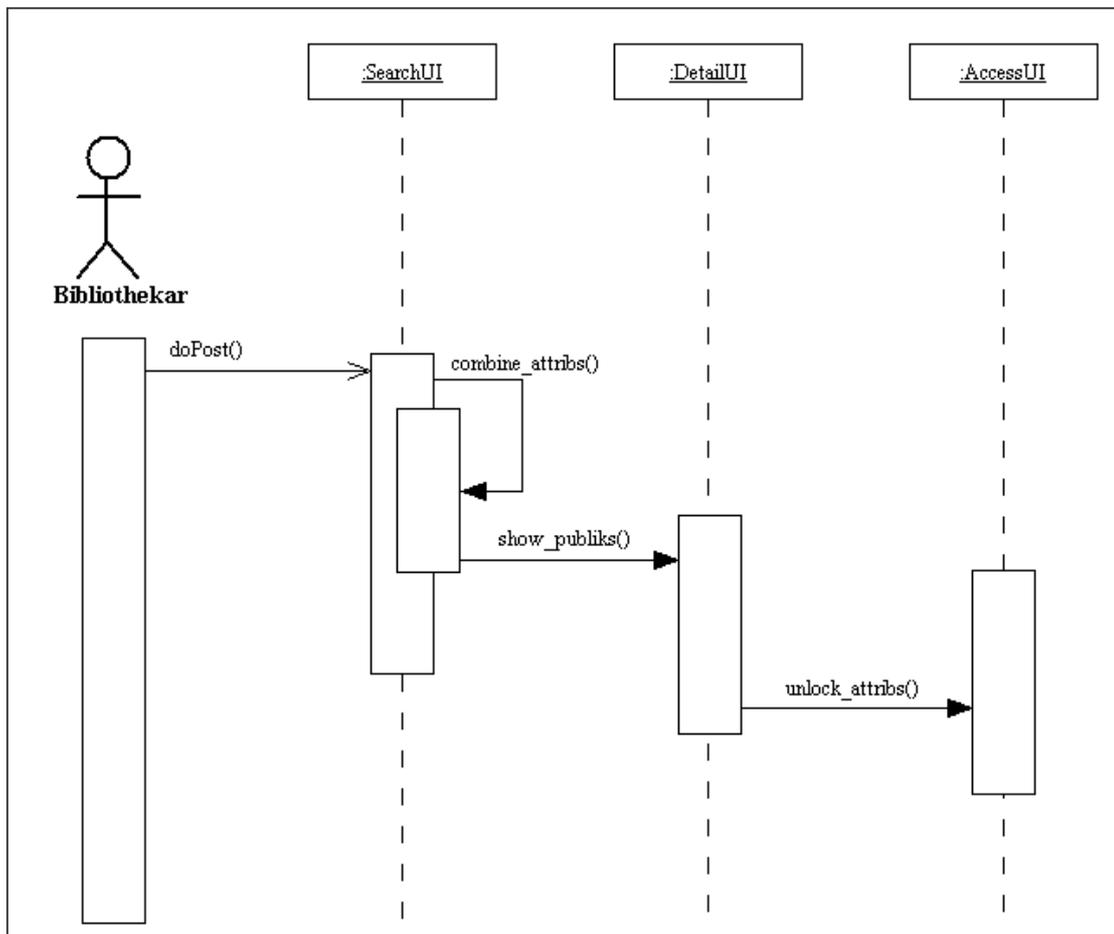


Abbildung 16: Sequenzdiagramm „Publikationen entsperren“

5.3 Prototyp der Benutzeroberfläche

5.3.1 Dialoggestaltung

Die Interaktion zwischen einem Benutzer und einem Dialogsystem ist ein Dialog. Der Benutzer arbeitet mit dem Dialogsystem. Für die Dialoggestaltung ist das GUI-System wichtig. Für die Dialoggestaltung bzw. die Benutzeroberfläche des Prototypen wurde HTML gewählt.

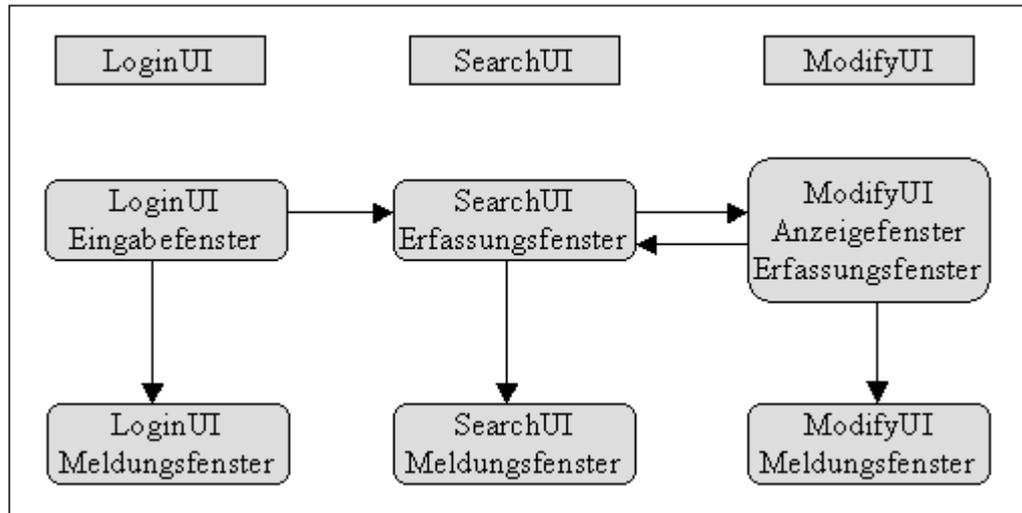


Abbildung 17: Zustandsdiagramm Dialogstruktur der Bibliotheksanwendung

5.3.2 Fenster

Das zentrale Element der Dialoggestaltung sind die Fenster, mit ihnen finden die Aktivitäten des Benutzers statt. Die Fenster der Anwendung enthalten hauptsächlich Interaktionselemente, (s. Interaktionselemente S. 37), zur Dateneingabe. Es können Daten gesucht und modifiziert bzw. gelöscht werden.

Abbildung 18: Suchmaske

The screenshot shows the 'ePIC Admin' interface with the following elements:

- Header:** 'ePIC Admin version 1.0' logo and 'Electronic Publication Information Center' text. Below it, a link for 'Komplette Liste (PDF) | Handbuch'.
- Auswahlkriterien:** A section with buttons for 'Alle Anwählen', 'Alle Abwählen', 'Löschen', 'Sperren', and 'Entsperren'.
- Attribute modifizieren:** A form with a 'Quelle' dropdown menu, 'Alt:' and 'Neu:' text labels, and two input fields. An 'OK' button is on the right.
- Attribute hinzufügen:** A form with a 'Quelle' dropdown menu and one input field. An 'OK' button is on the right.
- Attribute löschen:** A form with a 'Quelle' dropdown menu, an 'Alt:' label, and one input field. An 'OK' button is on the right.

Abbildung 19: Modifikationsmaske

5.3.3 Interaktionselemente

Als Interaktionselemente dienen Eingabefelder (edit control), Schaltflächen (command button, push button), Optionsfelder (radio button), Kontrollkästchen (check box) und Klapplisten (drop-down list box).

5.3.4 Gestaltung von Fenstern

Der Anwender muss optisch durch das Fenster geführt werden, entsprechend sollen die Interaktionselemente der Anwendung gruppiert und hervorgehoben werden. Es wird auf Balance und Symmetrie der Fenster geachtet, dass also die Informationsdichte auf beiden Seiten des Fensters, (würde man es mit einer vertikalen Linie trennen), ähnlich groß ist und die Interaktionselemente gleichartig sind. Jedes Fenster soll dabei so einfach wie möglich gestaltet sein.

5.3.5 Layout

Das Layout ist den bereits bestehenden AWI-Java-Anwendungen angepasst. Das betrifft sowohl die Farben, welche hauptsächlich in blau und grau gehalten sind, als auch das Logo und die Navigation. Für ein einheitliches Bild sorgen Cascading Style Sheets, die in HTML eingebunden sind. So ist bei allen Anwendungen die gleiche Schriftart, Schriftgröße, Hintergrundfarbe usw. eingestellt.

6 OOD (Objekt Orientiertes Design)

Im Objektorientierten Design wird die Softwarearchitektur erstellt sowie die Spezifikation der Klassen aus Sicht der Realisierung. Das Ergebnis ist das OOD-Modell, welches auf dem OOA-Modell aufbaut. Die Namen der Klassen sind den bereits bestehenden Java-Klassen anderer Anwendungen mit ähnlicher Funktion angepasst, für ein besseres, bzw. schnelleres Verständnis.

6.1 OOD-Modell

6.1.1 Klassenstruktur von ePIC Admin

Die folgende Graphik zeigt die Klassenstruktur von ePIC Admin.

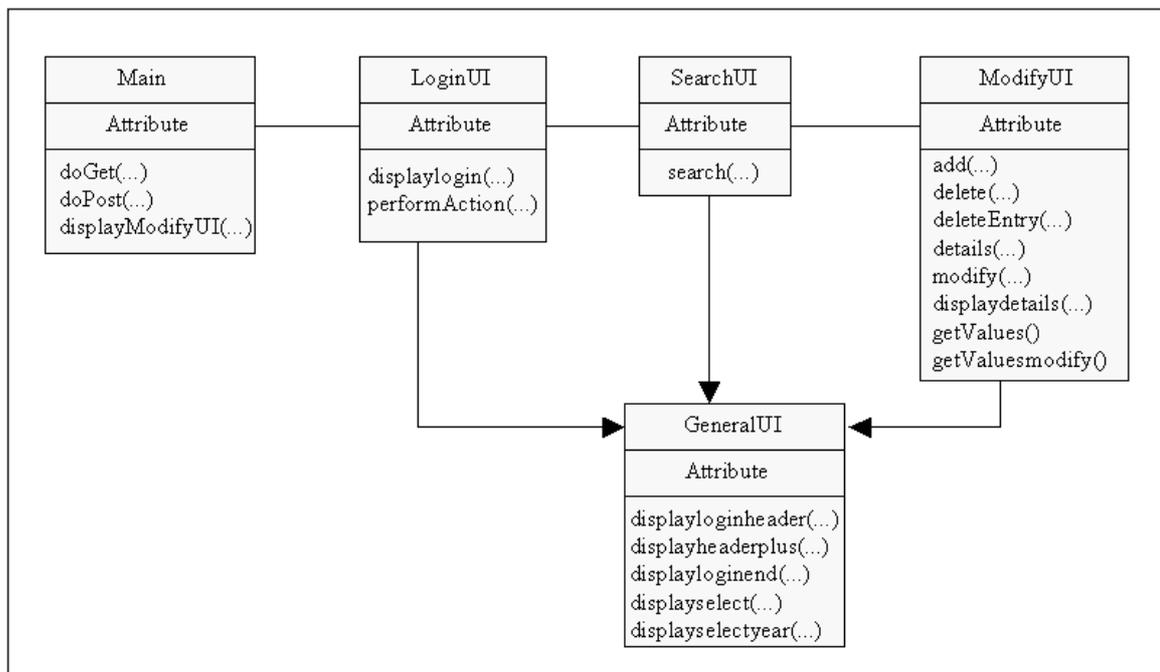


Abbildung 20: Klassenstruktur ePIC Admin

6.2 Architekturentwurf

Der Architekturentwurf stellt die grundlegenden Komponenten des Softwaresystems und ihre Abhängigkeiten dar. Die Anwendung wird hierbei in Schichten (layers) unterteilt. Für diese Anwendung ist dies die GUI-Schicht, die Fachkonzeptschicht und die Datenhaltungsschicht. Die GUI-Schicht präsentiert die Benutzeroberfläche der Anwendung. Die Fachkonzeptschicht realisiert die Klassen, das OOA-Modell bildet hierbei die erste Version der Fachkonzeptschicht. In der Datenhaltungsschicht wird die Datenspeicherung realisiert.



Abbildung 21: Drei-Schichten-Architektur

6.3 Implementierungsentwurf

Aufgabe des Implementierungsentwurfs ist es, aus den gegebenen Anforderungen an das Softwaresystem eine software-technische Lösung im Sinne einer Softwarearchitektur zu entwickeln. Im Entwurf werden die Programmklassen der Anwendung beschrieben. Die Entwurfsergebnisse bilden die Grundlage für die Implementierungsphase.

6.3.1 Implementierung vorhandener Klassen

Klassen des Programms ePIC konnten implementiert werden. Es handelt sich hierbei um die Klasse `PublicationList`, mit welcher eine Liste von Publikationseinträgen erstellt werden kann. Mit der Klasse `PublicationEntry` wird ein Publikationseintrag abgebildet d.h., auf die einzelnen Attribute des Eintrages kann zugegriffen werden. Die Klasse `PeopleEntry` ist ähnlich der `PublicationEntry`, allerdings wird hier ein Personeneintrag abgebildet. [Lorr01]

6.3.2 UI- Klassen

Die UI-Klassen (UI = User Interface) sind, wie auch in den bereits bestehenden AWI-Java-Anwendungen, die Benutzerschnittstellen. Mit diesen Klassen, bzw. deren Masken findet die Interaktion mit dem Benutzer statt.

Die entsprechenden Masken für die Datenein- und Ausgabe werden hier erstellt. Es werden Sucheingaben verarbeitet. Erfolgs- bzw. Fehlermeldungen sind in den UI-Klassen implementiert worden, teilweise durch JavaScript, aber auch durch ExceptionHandling.

6.3.2.1 GeneralUI

In der Klasse GeneralUI wird das generelle Design festgelegt, deshalb ist auch in dieser Klasse der meiste Teil des HTML-Codes für die Programmmasken gespeichert. Die UIs implementieren die entsprechende Methode der GeneralUI. Individueller HTML-Code, d.h. Code speziell für eine Maske ist in der jeweiligen UI-Klasse direkt gespeichert.

Die Methoden `displayloginheader(...)` bzw. `displayheaderplus(...)` sowie `displayloginend(...)`, enthalten das Grundgerüst der Masken, den Header mit Links, den Hintergrund sowie den Hinweistext auf zu aktivierende Cookies usw.. Die Methoden `displayselect(...)` und `displayselectyear(...)` enthalten den HTML-Code für die entsprechenden Klapplisten.

Klasse	GeneralUI
Methoden	<code>public void displayloginheader(HttpServletRequest request, HttpServletResponse response, PrintWriter out)</code>
	<code>public void displayheaderplus(HttpServletRequest request, HttpServletResponse response, PrintWriter out)</code>
	<code>public void displayloginend(HttpServletRequest request, HttpServletResponse response, PrintWriter out)</code>
	<code>public void displayselect(HttpServletRequest request, HttpServletResponse response, PrintWriter out)</code>
	<code>public void displayselectyear(HttpServletRequest request, HttpServletResponse response, PrintWriter out)</code>

Tabelle 1: Klasse GeneralUI

6.3.2.2 LoginUI

Die Klasse LoginUI steuert die Authentifizierung bzw. die Autorisierung eines Benutzers gegenüber dem Administrationsprogramm. Die Eingabe des Benutzernamens und des Passwortes erfolgt über die Loginmaske, welche durch die Methode displaylogin(...) aufgerufen wird. Nur ein berechtigter Benutzer, also ein Mitglied der Bibliothek oder ein Administrator, eingetragen im LDAP-Verzeichnis unter cn=PublicationGroup,ou=Groups,dc=awi-bremerhaven,dc=de, hat Zugang zum System. Nach erfolgreicher Authentifizierung wird die Autorisierung mit Hilfe der ACIs (Access Control Information) entschieden. Die LoginUI baut nach erfolgreicher Autorisierung eine Session auf, in welcher die Berechtigungsdaten gespeichert werden. Diese Daten sind dann für den weiteren Programmverlauf vorhanden und müssen nicht mit jeder Maske neu eingegeben werden. Die Authentifizierung gegenüber dem Server erfolgt mit der Methode performAction(...).

Klasse	LoginUI
Methoden	public void displaylogin(HttpServletRequest request, HttpServletResponse response, PrintWriter out)
	performAction(HttpServletRequest request,HttpServletResponse response, PrintWriter out) throws javax.servlet.ServletException, java.io.IOException

Tabelle 2: Klasse LoginUI

6.3.2.3 SearchUI

Die Klasse SearchUI gibt die vom Benutzer in die Searchmaske eingegebenen Daten an die ModifyUI weiter. Die Suchfilterinhalte, aus denen in der ModifyUI der Suchfilter zusammengesetzt wird, werden mit Hilfe der Search-Methode festgelegt.

Klasse	SearchUI
Methode	public void search(HttpServletRequest request, HttpServletResponse response, PrintWriter out)

Tabelle 3: Klasse SearchUI

6.3.2.4 ModifyUI

In der Klasse ModifyUI werden die eingegebenen Daten aus der SearchUI verarbeitet. Aus den gegebenen Daten wird mit der Methode details(...) ein Suchfilter erstellt, mit dessen Hilfe die entsprechenden Daten aus dem LDAP-Verzeichnis ausgelesen werden. Der Suchfilter setzt sich aus einem Operator, mindestens einem Attributnamen und dem dazugehörigen Wert zusammen. Setzt sich der Suchfilter aus mehreren Attributen zusammen, muss zwischen der logischen Attribut-Verknüpfung „UND“, bzw. „ODER“ gewählt werden. Standardmäßig ist die UND-Verknüpfung aktiviert.

Beispiel: (&(publicationyear=2000)(publicationauthor=m)).

Die Modifikationsmaske zeigt das Ergebnis des Suchfilters sowie die unterschiedlichen Möglichkeiten zum Bearbeiten der angezeigten Publikation bzw. Publikationen an. Mit Hilfe der Methode add(...) können Attribute zu den Einträgen hinzugefügt werden. Mit der Methode modify(...) können Werte von bestehenden Attributen geändert werden. Die Methode delete(...) nimmt bei Ausführung eine Löschung eines Attributes vor. Mit deleteEntry(...) wird eine komplette Publikation gelöscht. Displaydetails zeigt die Detailansicht an. Mit den Methoden getValues() und getValuesmodify() wird eine Verbindung zum LDAP-Server hergestellt, um die Daten auszulesen, die mit dem Suchfilter gesucht, bzw. angefordert werden.

Klasse	ModifyUI
Methoden	public void add(HttpServletRequest request, HttpServletResponse response, PrintWriter out)
	public void delete(HttpServletRequest request, HttpServletResponse response, PrintWriter out)
	public void deleteEntry(HttpServletRequest request, HttpServletResponse response, PrintWriter out)
	public void details (HttpServletRequest request, HttpServletResponse response, PrintWriter out) public void getValues()
	public void displaydetails(HttpServletRequest request, HttpServletResponse response, PrintWriter out)

	public void getValues()
	public void getValuesmodify()
	public void modify(HttpServletRequest request, HttpServletResponse response, PrintWriter out)

Tabelle 4: Klasse ModifyUI

6.3.3 Main

Die Klasse Main steuert die UI-Klassen-Aufrufe. Mit entsprechender Parameterübergaben werden die entsprechenden UI-Klassen aufgerufen.

Klasse	Main
Methoden	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,IOException
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,IOException
	public void displayModifyUI(HttpServletRequest request, HttpServletResponse response, PrintWriter out) throws ServletException,IOException

Tabelle 5: Klasse Main

6.3.4 Properties-Dateien

Mit Hilfe von Properties-Dateien können Werte für Zeichenkettenobjekte ausgelagert werden. Das bedeutet z.B. für die Internationalisierung, dass in den entsprechenden Properties-Dateien die Werte für die Sprache Deutsch und Englisch ausgelagert werden. Z.B.: In der Properties-Datei für die deutsche Sprache, hier UIPubResource_de_DE.properties genannt, steht:

LoginUI.description=Bitte geben Sie Ihren E-Mail-Benutzernamen und das E-Mail-Passwort ein.

In der Properties-Datei für die englische Sprache (UIPubResource_en_EN.properties) steht:

LoginUI.description=Please enter your e-mail username und the e-mail password.

Die Werte der Properties-Dateien werden an entsprechender Stelle im Programm mit dem Befehl „UIResource.getString“ aufgerufen.

7 Implementierung

In der Implementierung gilt es, die geforderten Leistungen in Form des Programms zu realisieren. Es werden hier die berechtigten Benutzer und deren Authentifizierung beschrieben, sowie der Programmablauf.

7.1 Berechtigte Personen

Die Bibliothekare sowie Administratoren sind die berechtigten Personen für ePIC Admin. Die Gruppe befindet sich im Directory Server unter dem DN `cn=PublicationGroup,ou=Groups,dc=awi-bremerhaven,dc=de`.

Sie können entscheiden, ob eine Publikation modifiziert oder gelöscht werden darf. Der Autor einer Publikation ist verantwortlich für seine Publikation; sperrt bzw. schließt (`publicationlock`) nun der Bibliothekar die Publikation eines Autors, hat dieser keinen Zugriff mehr auf die Publikation. Mit Sperrung ist nicht die Sperrung der Publikation während der Bearbeitungszeit durch den Bibliothekar gemeint, sondern die Publikation wird nur noch für den Bibliothekar zugänglich gemacht.

7.2 Authentifizierung und Autorisierung

Mit Hilfe der Daten aus dem Directoryserver wird der Zusammenhang von Benutzernamen und dem dazugehörigen Passwort überprüft. Zuerst wird festgestellt, ob es den Benutzernamen gibt. Gibt es ihn, wird überprüft ob Benutzername und Passwort zusammengehören. Nach der Authentifizierung erfolgt die Autorisierung. Hier wird ermittelt welche Rechte der erfolgreich authentifizierte Benutzer hat, indem seine Gruppenzugehörigkeit im Directory Server geprüft wird.

7.3 Session

Beim ersten Aufruf des Programms wird eine Session gesetzt, in welcher Daten gespeichert werden können. In der Session werden die Benutzerangaben gespeichert, auf diese Weise bleiben die Daten über mehrere Seiten, bzw. Anfragen und Antworten gespeichert.

Die Daten werden serverseitig abgelegt. Auf der Clientseite wird die Session-ID in einem Cookie gespeichert. Die Daten des Cookies werden auf Verlangen des Servers

an diesen übermittelt. In der Servlet API wird die Session mit HttpSession dargestellt.

In der Anwendung werden Sessions benutzt um Informationen über die ausgewählte Sprache und Benutzerdaten zu speichern.

Eine Session wird erzeugt mit der Anweisung:

```
HttpSession session = request.getSession(boolean).
```

Mit dem Wert boolean wird überprüft, ob bereits eine Session existiert oder nicht.

Die Methode invalidate() zerstört die Session wieder.

7.4 Programmablauf

7.4.1 Loginmaske

Mit Hilfe der Loginmaske soll sich der Benutzer in die Anwendung einloggen. Dazu muss er seinen Benutzernamen und sein Passwort eingeben und anschließend den OK-Button betätigen.

ePIC Admin version 1.0 Electronic Publication Information Center

Bitte geben Sie Ihren E-Mail-Benutzernamen und das E-Mail-Passwort ein

Benutzername:

Paßwort:

OK Löschen Zurück

© AWI

[Webmaster](#)

ACHTUNG: "Cookies", "Java", "JavaScript" und "Formatvorlagen (Style Sheets)" müssen aktiviert werden! Die Optionen stehen unter Bearbeiten-Einstellungen-Erweitert

Abbildung 22: Loginmaske

7.4.2 Suchmaske

In der Suchmaske bestehen mehrere Möglichkeiten gezielt nach Publikationen zu suchen. Es können einzelne Jahre angegeben werden oder Zeiträume aus denen Einträge gesucht werden sollen. Ferner besteht die Möglichkeit nach bestimmten Attributwerten zu suchen, z.B. nach Autoren. Für diese Suche wird das Attribut Autor ausgewählt und in das entsprechende Textfeld der Autorenname eingetragen. Attribut-Verknüpfungen sind ebenfalls möglich. Hierzu kann zwischen den Verknüpfungen UND bzw. ODER ausgewählt werden, standardmäßig ist UND eingestellt. Mit diesen Verknüpfungen bestehen vielfältige Möglichkeiten der Publikationssuche. Es können alle Attribute bzw. Werte miteinander verknüpft werden. Z.B. wird nach allen Publikationen aus dem Jahr 2001 gesucht, deren Autorenname mit einem „m“ beginnt, zusätzlich soll die Quelle „marine“ enthalten. Nach Bestätigung des Suchen-Buttons erfolgt die Suche.

ePIC Admin version 1.0
Electronic Publication Information Center
Komplette Liste (PDF) | Handbuch

Jahr von bis

Attribut [\[Richtlinie\]](#)

Attribute-Verknüpfung UND ODER

© AWI

[Webmaster](#)

ACHTUNG: "Cookies", "Java", "JavaScript" und "Formatvorlagen (Style Sheets)" müssen aktiviert werden! Die Optionen stehen unter Bearbeiten-Einstellungen-Erweitert

Abbildung 23: Suchmaske

7.4.3 Ergebnisanzeige und Modifikationsmaske

Das Ergebnis der Suchmaske wird zusammen mit der Modifikationsmaske dargestellt. Vor der Ergebnisanzeige werden noch einmal die Auswahlkriterien angezeigt, nach denen gesucht werden sollte. Darunter werden die gefundenen Ergebnisse aufgelistet. Zu den einzelnen Publikationen kann man sich eine Detailansicht anzeigen lassen, in der alle Attribute, die dieser Eintrag enthält, aufgeführt werden. Mit Hilfe der Modifikationsmaske soll es dem Benutzer möglich gemacht werden die Publikationen zu modifizieren oder zu löschen. Für eine entsprechende Aktion muss mindestens eine Publikation markiert werden, ansonsten erscheint eine entsprechende Fehlermeldung. Soll ein Attribut modifiziert werden, muss das entsprechende Attribut ausgewählt werden. Zur Sicherheit, damit keine falsche Publikation angesprochen wird, muss der alte Wert des Attributes eingetragen werden. Der neue Wert wird in das entsprechende Feld eingegeben. Nach Bestätigung des OK-Buttons erscheint bei erfolgreicher Änderung eine Erfolgsmeldung. Mit dem Zurück-Button gelangt man wieder zur Modifikationsmaske. Für das Hinzufügen eines Attributes zum Eintrag wählt man das entsprechende Attribut aus, gibt einen Wert hierfür ein und bestätigt ebenfalls mit OK. Will man ein Attribut löschen, muss auch hier zur Sicherheit der alte Wert angegeben werden. Um eine komplette Publikation zu löschen, muss der Löschenbutton betätigt werden. Eine Sperrung bzw. Entsperrung einer Publikation erfolgt durch den Sperren- bzw. Entsperrbutton. Vor jeder dieser Aktionen erscheint eine Hinweismeldung, ob man die entsprechende Aktion wirklich durchführen möchte.

ePIC Admin
version 1.0 **Electronic
Publication Information Center**
[Komplette Liste \(PDF\)](#) | [Handbuch](#)

Auswahlkriterien:

Attribute-Verknüpfung UND
 Jahr: 2001
 Autor: m
 Quelle: marine

Gefundene Publikationen: 7

- Laabir, M. (2001) Effect of specific dinoflagellate and diatom diets on gamete ultrastructure and fatty acid profiles of the copepod *Temora stylifera*** *Marine Biology*, 138, 1241-1250
[Detail](#)
- Meon, B. (2001) Dynamics and molecular composition of dissolved organic material during experimental phytoplankton blooms** *Marine Chemistry*, 75, 185-199
[Detail](#)
- Lass, S. (2001) On the food of northern krill *Meganyctiphanes norvegica* in relation to its vertical distribution** *Marine Ecology Progress Series*, 214, 177-200
[Detail](#)
- Bathmann, U.V. (2001) Future marine zooplankton research - a perspective** *Marine Ecology Progress Series*, 222, 297-308
[Detail](#)
- Michels, K. H. (2001) Recognition of contour-current influence in mixed contourite-turbidite sequences of the western Weddell Sea, Antarctica** *Marine Geophysical Researches*, 22, 5-6, 465-485
[Detail](#)

- Mackensen, A. (2001) Erratum to "Microhabitat preferences and stable carbon isotopes of endobenthic foraminifera: clue to quantitative reconstruction of new production?" [Mar. Micropaleontol. 40 (2000) 233-258]** *Marine Micropaleontology*, 42, 101
[Detail](#)
- Volkmann, R. (2001) Stable isotope composition (d18O, d13C) of living planktic foraminifers in the outer Laptev Sea and the Fram Strait** *Marine Micropaleontology*, 42, 163-188
[Detail](#)

Attribute modifizieren [\[Richtlinie\]](#)

	Quelle			Alt:
			Neu:	

Attribute hinzufügen [\[Richtlinie\]](#)

	Quelle			
--	--------	--	--	--

Attribute löschen [\[Richtlinie\]](#)

	Quelle			
--	--------	--	--	--

© AWI

[Webmaster](#)

ACHTUNG: "Cookies", "Java", "JavaScript" und "Formatvorlagen (Style Sheets)" müssen aktiviert werden! Die Optionen stehen unter Bearbeiten-Einstellungen-Erweitert

Abbildung 24: Ergebnisanzeige und Modifikationsmaske

7.4.4 Detailansicht

Die Detailansicht zeigt alle, für den Bibliothekar wichtigen Attribute an, die im Directory gespeichert sind und die er bearbeiten kann. In der Detailansicht werden deshalb auch alle weiteren Autoren aufgelistet. Wird z.B. nach einem Autor mit „m“ gesucht, kann die Ergebnisanzeige auch Autoren mit anderen Anfangsbuchstaben anzeigen, in der Detailansicht wird dann deutlich, dass für diese Publikation auch ein Autor mit dem Anfangsbuchstaben „m“ existiert.

ePIC Admin version 1.0 **Electronic Publication Information Center**
 Komplette Liste (PDF) | Handbuch | English

[Zurück](#)

Detailansicht

Publikationsnummer	awi-n10041
AWI- Publikation	yes
Autor	Volkman, R.
Autor	Mensch, M.
Email	jmatthiessen@awi-bremerhaven.de
Jahr	2001
Titel	Stable isotope composition (d18O, d13C) of living planktic foraminifers in the outer Laptev Sea and the Fram Strait
Quelle	Marine Micropaleontology, 42, 163-188
ISI Quelle	kein Eintrag
ISBN	none
Typ	article
Hochschulname	kein Eintrag
Begutachtet	yes
Status	published
Organisationseinheit	awi-2000-04
Organisationseinheit	awi-2000-0401

[Webmaster](#) © AWI

ACHTUNG: "Cookies", "Java", "JavaScript" und "Formatvorlagen (Style Sheets)" müssen aktiviert werden! Die Optionen stehen unter Bearbeiten-Einstellungen-Erweitert

Abbildung 25: Detailansicht

7.4.5 Erfolgs- bzw. Fehlermeldungen

Erfolgsmeldungen werden nach ausgeführten Aktionen angezeigt, um dem Benutzer zu versichern, dass die entsprechende Aufgabe erfolgreich ausgeführt wurde.



Abbildung 26: Erfolgsmeldung

Fehlermeldungen werden angezeigt, wenn z.B. etwas nicht markiert wurde oder wenn etwas falsch eingegeben wurde.



Abbildung 27: Fehlermeldung

8 Resümee

Im Rahmen dieser Diplomarbeit wurde ein bibliothekarisches Administrationssystem zur Verwaltung von elektronischen Publikationen entwickelt. Mit Hilfe dieses Systems wird es dem Bibliothekar nun möglich gemacht, systematisch die Publikationsdaten nach Fehlern durchzusuchen und diese zu korrigieren.

9 Glossar

Cascading-Style-Sheet

„Abkürzung: CSS. Eine HTML-Spezifikation, die vom World Wide Web Consortium (W3C) entwickelt wurde und es sowohl dem Gestalter von HTML-Seiten als auch dem Benutzer erlaubt, HTML-Dokumente mit Dokumentvorlagen (Stylesheets) zu verknüpfen. Die Dokumentvorlagen enthalten typografische Informationen in Bezug auf das Erscheinungsbild der Seite, z. B. hinsichtlich der Schriftart, der Zeilenabstände, der Einrückungen, usw. Die Spezifikation bestimmt außerdem die Art und Weise, mit der die Dokumentvorlage eines HTML-Dokuments mit der vom Benutzer definierten Dokumentvorlage in Einklang gebracht wird. Cascading Style Sheets sind für alle HTML-Standards und vorgeschlagenen HTML-Standards ab der Version 2.0 anwendbar. Weitere Informationen zu Cascading Style Sheets sind unter der Webadresse <http://www.w3.org/Style> abrufbar.“ [MSPress00]

CGI

„Abkürzung für »Common Gateway Interface«. Eine Spezifikation, die die Kommunikation zwischen Informationsservern (z. B. HTTP-Servern) und den Ressourcen (z. B. Datenbanken und anderen Programmen) auf den jeweiligen Hostcomputern definiert. Schicken Benutzer z. B. ausgefüllte Formulare über einen Webbrowser ab, führt der HTTP-Server ein Programm aus (das häufig als »CGI-Skript« bezeichnet wird) und übergibt die Benutzereingaben per CGI an das Programm. Das Programm verarbeitet die Daten und gibt die Ergebnisdaten per CGI an den Server zurück. Die Verwendung von CGI erlaubt es, Webseiten in höherem Maße dynamisch zu gestalten und mit interaktiven Elementen zu versehen. Weitere Informationen zu CGI sind z. B. unter den Webadressen <http://hoohoo.ncsa.uiuc.edu/cgi/> und <http://www.cgi-resources.com> abrufbar.“ [MSPress00]

Clientserverarchitektur

„In lokalen Netzwerken eingesetzte Anordnung, die von der »verteilten Intelligenz« Gebrauch macht, um sowohl den Server als auch die individuellen Arbeitsstationen als intelligente, programmierbare Geräte zu behandeln. Dadurch lässt sich die volle

Rechenleistung aller angeschlossenen Geräte nutzen. Hierfür wird die Verarbeitung einer Anwendung zwischen zwei selbständigen Komponenten aufgeteilt: dem »Front-End-Client« und dem »Back end-Server«. Die Clientkomponente stellt dabei einen vollständigen, eigenen Personal Computer dar (im Gegensatz zu einem »dummen« Terminal) und bietet den Benutzern uneingeschränkt alle Leistungen und Funktionen für den Betrieb von Anwendungen. Die Serverkomponente, bei der es sich um einen Personal Computer, einen Minicomputer oder einen Großrechner handeln kann, erweitert die Möglichkeiten der Clientkomponente durch Bereitstellung der konventionellen Leistungsmerkmale, die für Minicomputer und Großrechner in einer Timesharingumgebung typisch sind: Datenverwaltung, gemeinsame Nutzung von Informationen zwischen Clientcomputern sowie intelligente Netzwerkverwaltung und Sicherheitseinrichtungen. Der Vorteil der Clientserverarchitektur gegenüber älteren Architekturen liegt darin, dass die Verarbeitung der eingesetzten Anwendung von den Client- und Servercomputern gemeinsam realisiert wird. Dadurch erhöht sich nicht nur die verfügbare Verarbeitungsleistung, sondern diese Leistung wird auch effizienter genutzt. Der Clientteil der Anwendung ist in der Regel für die Interaktion mit den Benutzern optimiert, während der Serverteil die zentralisierte Mehrbenutzerfunktionalität zur Verfügung stellt.“ [MSPress00]

Cookie

„Im weiteren Sinn ein Block von Daten, den ein Server als Reaktion auf eine Anforderung vom Client zurückgibt.

Im World Wide Web ein Block von Daten, den ein Webserver auf dem Client speichert. Begeben sich Benutzer auf eine bereits zu einem früheren Zeitpunkt angewählte Seite, sendet der Browser eine Kopie des Cookies an den Server zurück. Cookies dienen dazu, Benutzer zu identifizieren, den Server anzuweisen, eine benutzerspezifische Version einer angeforderten Webseite zu senden, Konteninformationen für Benutzer anzufordern sowie der Durchführung weiterer Verwaltungsaufgaben.“ [MSPress00]

GUI (Graphical User Interface)

„Eine Umgebung, in der Programme, Dateien und Optionen durch Symbole, Menüs, und Dialogfelder am Bildschirm dargestellt werden. Der Benutzer kann diese Optionen mit der Maus oder über Tastaturbefehle markieren und aktivieren. Ein bestimmtes Element (z. B. eine Bildlaufleiste) funktioniert in allen Anwendungen gleich, weil die grafische Benutzeroberfläche Standardroutinen zum Behandeln dieser Elemente enthält, die die Aktionen des Benutzers weiterleiten (z. B. ein Mausklick auf ein bestimmtes Symbol oder auf eine bestimmte Textposition oder ein Tastendruck). Diese Routinen werden nicht von Anfang an neu erzeugt, sondern über bestimmte Parameter aufgerufen.“ [MSPress00]

SSL (Secure Sockets Layer)

„Ein von der Firma Netscape Communications geschaffenes Protokoll zur Schaffung einer sicheren (verschlüsselten) Kommunikation im Internet. SSL bildet eine Schicht zwischen den Applikationsprotokollen wie HTTP, SMTP, FTP, Telnet und NNTP und dem Transportprotokoll TCP/IP. Hauptsächlich wird es von HTTPS benutzt. Die durch SSL verwendete Verschlüsselung bietet Schutz gegen die meisten Datenangriffe. Man sollte aber berücksichtigen, daß trotzdem Möglichkeiten existieren, diese zu knacken. Auch bei SSL ist die Schlüssellänge künstlich verkürzt. Dafür ist die US-Regierung verantwortlich, die Verschlüsselungstechnik als Militärtechnik einstuft und den Export beschränkt. Zumindest die Geheimdienste können also immer mithören. Außerdem nutzt auch die sicherste Verbindung nichts, wenn das System, auf das die Daten übertragen werden, nicht sicher ist.“ [seicom03]

URL

„Abkürzung für Uniform Resource Locator (einheitliche Ressourcenadresse). Eine Adresse für eine Ressource im Internet. URL-Adressen werden von Webbrowsern verwendet, um Internetressourcen zu lokalisieren. Eine URL-Adresse gibt das für den Zugriff auf eine Ressource zu verwendende Protokoll an (z. B. »http:« für eine Webseite oder »ftp:« für eine FTP-Site), den Namen des Servers, auf dem sich die Ressource befindet (z. B. »//www.whitehouse.gov«) sowie - frei wählbar - den Pfad

zu einer Ressource (z. B. einem HTML-Dokument oder einer Datei auf diesem Server).“ [MSPress00]

Webbrowser

„Eine Clientanwendung, die es einem Benutzer ermöglicht, HTML-Dokumente auf dem World Wide Web bzw. einem anderen Netzwerk oder auf dem eigenen Computer zu betrachten. Auch können mit Hilfe des Browsers die Hyperlinks zwischen den Dateien verfolgt und Dateien übertragen werden. Textorientierte Webbrowser (z. B. Lynx) sind zwar nützlich für Benutzer mit Shellzugängen, zeigen aber lediglich die Textelemente eines HTML-Dokuments an. Die meisten Browser jedoch basieren auf einer Verbindung, die neben der Verarbeitung der IP-Pakete auch Grafiken im Dokument darstellen, Audio- und Videodateien wiedergeben und kleine Programme ausführen kann (z. B. Java Applets, ActiveX -Steuerelemente), die in HTML-Dokumenten eingebettet sein können. Einige Webbrowser benötigen zusätzliche Hilfsanwendungen (Plug-Ins), um solche Funktionen bewerkstelligen zu können. Darüber hinaus ermöglichen es die meisten heutigen Webbrowser ihren Benutzern, E-Mail zu senden und zu empfangen sowie an Newsgroups teilzunehmen.“ [MSPress00]

10 Literaturverzeichnis

[AWI02]

<http://www.awi-bremerhaven.de>, letzter Aufruf 28.08.2003.

[Balzert99]

Balzert, Heide: Lehrbuch der Objektmodellierung, Spektrum Akademischer Verlag, Heidelberg – Berlin 1999.

[Baier02]

Baier, Michael: Javascript für Einsteiger, 2. Ausgabe, 3. Auflage, KnowWare Verlag, 2002.

[Born00]

Born, Günter: HTML, Markt + Technik Verlag, München 2000.

[BraSch01]

Brantner, Matthias, Schmidt, Stefan, Wabnitz, Birgit, Wabnitz, Detlef: Java Servlets und JavaServer Pages, Data Becker Verlag, Düsseldorf 2001.

[Galileo03]

<http://www.galileocomputing.de>, letzter Aufruf 14.09.03.

[Lorr01]

Lorr, Sebastian: Entwurf und Implementierung eines Systems zur Verwaltung elektronischer Publikationen an einem wissenschaftlichen Institut, Bremerhaven 2001.

[MidSin99]

Middendorf, Stefan, Singer, Reiner: Java Programmierhandbuch und Referenz für die Java-2-Plattform, 2., überarbeitete und erweiterte Auflage, dpunkt Verlag, Heidelberg 1999.

[MSPress00]

Microsoft Press: Computer Fachlexikon Fachwörterbuch, 4. Auflage, Microsoft Press, Unterschleißheim 2000.

[Oreilly02]

http://www.oreilly.de/artikel/ldap_2.html, letzter Aufruf 10.09.2003.

[RosPrä01]

Rossig, Wolfram E., Prätsch, Joachim: Wissenschaftliche Arbeiten, 3. erweiterte Auflage, Wolfdruck Verlag, Bremen 2001.

[Scholz01]

Scholz, Dieter: Diplomarbeiten normgerecht verfassen, Vogel Verlag, Würzburg 2001.

[seicom03]

http://www.seicom-muc.de/db_nav/fkt/nav.frame.php4?mid=65, letzter Aufruf 16.07.2003.

[Softguide03]

http://www.softguide.de/prog_s/ps_0485.htm, letzter Aufruf 16.07.2003.

[Voss01]

Voss, Dr. Andreas: Das große PC & Internet Lexikon 2001 2002, Data Becker Verlag, Düsseldorf 2001.

Versicherung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen sind, sind als solche unter Angabe der Quellen kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Bremerhaven, den 02.10.2003