

Multimediale, datenbankgestützte Lehr- und Lernplattformen

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Bayerischen Julius-Maximilians-Universität Würzburg

vorgelegt von

Holger Höhn

aus

Würzburg

Würzburg, 2002

Eingereicht am: 27. Juni 2002

bei der Fakultät für Mathematik und Informatik

1. Gutachter: Prof. Dr. Jürgen Albert
2. Gutachter: Prof. Dr. Dietmar Seipel

Tag der mündlichen Prüfung:

Herrn Karl Wagenhäuser

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Informatik II der Universität Würzburg. In diesem Zusammenhang möchte ich Herrn Professor Dr. J. Albert danken. Die regelmäßigen Gespräche mit ihm und seine Anregungen gaben mir die Möglichkeit, die Dissertation in dieser Form vorzulegen. Weiterhin gilt mein Dank Herrn Wolfram Eßer für seine Anregungen und die Durchsicht der Ausarbeitung sowie allen wissenschaftlichen Hilfskräften, die an den Projekten HagerROM, SENTIMED und DEJAVU beteiligt waren. Desweiteren wäre die Umsetzung nicht ohne die Unterstützung der Mitarbeiter der Universitätshautkliniken Jena, Mannheim und Würzburg und des Instituts für medizinische Psychologie der Universität Jena möglich gewesen.

Mein Dank gilt auch Herrn Prof. Dr. D. Seipel für die Erstellung des Zweitgutachtens.

Würzburg, im Juni 2002

Holger Höhn

Inhaltsverzeichnis

1 Einführung.....	11
2 Lernobjekte.....	17
2.1 Einführung und Definition.....	17
2.2 Metadaten für Lernobjekte.....	19
2.2.1 Dublin Core Metadata Element Set (DCMES).....	22
2.2.2 Der IEEE LOM – Standard.....	23
2.2.2.1 Einführung.....	23
2.2.2.2 Das konzeptuelle Datenschema des LOM – Standards.....	24
2.2.2.3 Ziele.....	25
2.2.2.4 Modifikationen und Spezifikation des IMS.....	26
2.2.3 Learning Material Markup Language - LMML.....	27
2.3 IMS - Spezifikationen.....	29
2.3.1 IMS Content Packaging Information Model.....	29
2.3.2 IMS Question & Test Interoperability (QTI) Specification.....	31
2.4 Zusammenfassung und Bewertung.....	33
3 Lehr- und Lernumgebungen.....	35
3.1 Einführung und Definition.....	35
3.2 Microsoft Learning Resource iNterchange (LRN) -Toolkit.....	37
3.3 e-Learn.....	38
3.4 Blackboard 5: Learning System.....	40
3.5 Authoring on the Fly.....	41
3.6 Medizinische Lernumgebungen.....	42
3.7 Zusammenfassung und Bewertung.....	44
4 Medizinische Klassifikationen.....	47
4.1 ICD – Codes der WHO und Modifikationen.....	47
4.1.1 Überblick.....	47
4.1.2 ICD-10-SGBV	48
4.1.3 Dermatologischer Diagnosenkatalog - DDK.....	49
4.2 Operationenschlüssel nach Paragraph 301 SGB V.....	50

4.3 Medical Subject Headings – MeSH.....	50
4.4 Unified Medical Language System - UMLS.....	51
4.4.1 UMLS Metathesaurus.....	51
4.4.2 SPECIALIST Lexikon.....	53
4.4.3 UMLS Semantic Network.....	53
5 Toolbox zur Erstellung von Vorlesungen.....	55
5.1 Architektur, Installationsvarianten und Kopierschutz.....	56
5.2 Bildarchiv.....	57
5.2.1 Auswahl geeigneter Bilder.....	58
5.2.2 Reinigen und Einscannen der Dias.....	59
5.2.3 Erfassen der Grunddaten.....	60
5.2.4 Sicherung der Daten.....	62
5.2.5 Aufbereitung der Bildobjekte.....	63
5.2.6 Einfache Suche im Bildarchiv.....	63
5.2.7 Eingabe der sonstigen Daten.....	63
5.3 Semantisches Netz.....	65
5.3.1 Generierung.....	67
5.3.2 Navigation.....	69
5.3.3 Präsentation des Bildmaterials.....	70
5.3.4 Interne Zwischenablage von Bildobjekten.....	72
5.3.5 Verlinkung mit externen WWW-Datenquellen.....	72
5.4 Vorlesungserstellung.....	74
5.4.1 Suche im Bildarchiv und Auswahl der Bilder.....	76
5.4.2 Import von Powerpoint-Präsentationen.....	77
5.4.3 Vergrößerungen von Bildern bzw. Bildausschnitten.....	77
5.4.4 Import von eigenen Bildern.....	78
5.4.5 Import aus dem semantischen Netz.....	78
5.4.6 Recherche im semantischen Netz.....	78
5.4.7 Wiederverwendung von Vorlesungen.....	78
5.4.8 Vorschau von Vorlesungen.....	79
5.4.9 Export von Vorlesungen.....	79
5.5 Vorlesungspräsentation.....	79

6 Datenbankgestützte Lehr- und Lernumgebung.....	81
6.1 Technische Voraussetzungen und Akzeptanz.....	83
6.2 Konzept, Module und Aufbau.....	84
6.3 Lernobjekte in der Dermatologie.....	86
6.3.1 Bildobjekte.....	88
6.3.2 Moulagen.....	90
6.3.3 Videoobjekte.....	90
6.3.4 Vorlesungen.....	91
6.3.5 Lerntexte und Skripte.....	92
6.3.6 Fallbeispiele.....	92
6.3.7 Quizelemente.....	93
6.3.8 Beziehungen der Lernobjekte untereinander.....	93
6.3.9 Matching mit dem LOM-Standard.....	95
6.4 Lernobjekt – Datenbank.....	95
6.4.1 Anforderungen.....	95
6.4.2 Umsetzung.....	96
6.4.3 Auswahl des Datenbanksystems.....	98
6.4.4 Konvertierung des textbasierten Bildarchivs.....	100
6.4.5 Mergen der lokalen Datenbanken.....	100
6.4.6 Suche in der Datenbank.....	101
6.5 Autorenwerkzeuge zur Erstellung von Lernobjekten.....	101
6.5.1 DEJAVUCapt – Einfügen von Lernobjekten in die Datenbank.....	102
6.5.2 Nachverschlagnwortung von Attributen.....	106
6.5.3 Bild- und Videoobjekte.....	107
6.5.4 Moulagen.....	107
6.5.5 Präsenzvorlesungen.....	108
6.5.6 Lerntexte.....	111
6.5.7 Fallbeispiele.....	114
6.5.8 Quizelemente.....	115
6.5.8.1 Bilderquiz.....	115
6.5.8.2 Übungsfragen.....	115
6.5.8.3 Fallquiz.....	116

6.6 Lernumgebung DEJAVULearn.....	119
6.6.1 Einleitung.....	119
6.6.2 Systemvoraussetzungen.....	119
6.6.3 Funktionen und Aufbau.....	120
6.6.4 Bild- und Video-Objekte.....	123
6.6.5 Moulagen.....	123
6.6.6 Vorlesungen zur Nachbereitung.....	124
6.6.7 Lerntexte und Fallbeispiele.....	126
6.6.8 Quizelemente.....	126
6.7 Konformität zu bestehenden Standards.....	129
7 Zusammenfassung und Ausblick.....	133
7.1 Ergebnisse.....	133
7.2 Ausblick.....	136
Anhang A - LOM-Standard.....	139
Anhang B - Textbasiertes Bildarchiv mit Toolbox.....	147
B.1 Installations CD „Textbasiertes Bildarchiv mit Toolbox“.....	147
B.2 Verzeichnisstruktur.....	148
B.3 Grundverschlagnwortung.....	148
B.4 Erweiterte Verschlagnwortung.....	153
B.5 Semantisches Netz.....	154
B.6 Vorlesungserstellung.....	158
B.7 Vorlesungspräsentation.....	159
Anhang C - Studentenbefragungen.....	161
C.1 Technische Voraussetzungen und Akzeptanz.....	161
C.2 Befragung zu Lernobjekt-Typen.....	165
Anhang D - Lernobjekt – Datenbank.....	169
D.1 Allgemeine Tabellen.....	170
D.2 Einzelnen Lernobjekt-Typen zugeordnete Tabellen.....	175
D.3 WWW-Suche in der Datenbank.....	179
D.4 Mergen mehrerer Datenbanken.....	180
Anhang E - Lehr- und Lernumgebung.....	181
E.1 Installations – CD „Lehr- und Lernumgebung“.....	181

E.2 Autorenwerkzeuge	182
E.2.1 DEJAVUCapt	182
E.2.2 Nachverschlagwortung der Attribute.....	184
E.2.3 Powerpoint AddIn.....	185
E.2.4 HTML2XML Konvertierung Lerntexte / Fallbeispiele.....	186
E.2.5 Übungsfragen.....	187
E.2.6 Fallquiz.....	188
E.3 Lernumgebung.....	190
E.3.1 DEJAVULearn.....	190
E.3.2 Bilderquiz.....	193
E.3.3 Fallquiz	193
E.3.4 Übungsfragen.....	194
Anhang F - Sonstige Skripte.....	197
Literaturverzeichnis.....	199
Abbildungsverzeichnis.....	205

1 Einführung

Distance-Learning, Web based Training, Internet based Training, Online-Learning, Computer Based Training, Edutainment und E-Learning sind Schlagworte, die immer häufiger im Bereich der Lehre und des Lernens auftauchen. Die Entwicklung des *Distance-Learning*, also des ortsunabhängigen Lernens über Entfernungen, beginnt im Grunde bereits mit der Erfindung der Schrift und führt über erste Akademien, den Buchdruck, Fernlehrgänge, Lernen über Radio (um 1900), Lernen mittels TV (seit 1945) schließlich bis zum Lernen am heimischen Rechner mit Internetanbindung. Erst in der neueren Entwicklung wird das Lernen durch multimedialen Inhalte interaktiv und lebensecht und es besteht die Möglichkeit sowohl des synchronen als auch asynchronen Lernens. *Web Based Training (WBT)*, oft auch als *Internet Based Training* oder *Online-Learning* bezeichnet, ist das Lernen über das Internet. Manchmal wird auch zwischen *Supported Distance Learning*, d.h. mit fachlicher Unterstützung des Lernenden durch einen Tutor und dem reinen *Computer Based Training (CBT)*, d.h. selbstgesteuertes Lernen ohne Tutor, unterschieden. Meist sind aktuelle Lösungsansätze von virtuellen Klassenzimmern bis zu unter dem Begriff *Edutainment* verkauften Lernprogrammen bereits Hybridlösungen und lassen sich

nicht in eine der definierten Schubladen einordnen, sondern allenfalls unter dem Oberbegriff *E-Learning* („technisch basiertes Lernen“) zusammenfassen.

Das Bundesministerium für Bildung und Forschung der Bundesrepublik Deutschland bmb+f [32] fördert in diesem Zusammenhang Projekte zum Beispiel im Programm *Neue Medien in der Bildung - Lehr- und Lernsoftware* mit nicht unerheblichen Geldmitteln. In der Fachinformation zum Förderprogramm findet man beispielsweise den folgenden Abschnitt:

- „Die breite Nutzung von Informations- und Kommunikationstechniken im Bildungsbereich und der Einsatz multimedialer Wissensvermittlung bieten die Chance,
- junge Menschen bereits in der Schule besser auf private und berufliche Anforderungen der Wissensgesellschaft vorzubereiten und sie mit multimedialem Lernen frühzeitig vertraut zu machen,
 - das bewährte und international anerkannte Duale System der beruflichen Bildung weiterzuentwickeln,
 - den Strukturwandel im Hochschulbereich zu unterstützen und die internationale Wettbewerbsfähigkeit der deutschen Hochschullehre zu erhöhen,
 - ...
 - für ein selbstbestimmtes, ortsunabhängiges Lernen, von dem vor allem Frauen und Männer profitieren können, die sich aufgrund von Kindererziehung, Krankheit, Alter oder anderen Lebensumständen zu Hause bilden wollen,
 - zu neuen kooperativen Lehr- und Lernformen.“

Und auch das bereits 1999 von Bund und Ländern eingerichtete *Forum Bildung* [42] hat als einen Themenschwerpunkt die *Neue Lern- und Lehrkultur* und schreibt in seinem Bericht der Expertengruppe [41]:

- „Der große Vorteil (der neuen Medien) besteht gerade für ... die Hochschulen und den Weiterbildungsbereich darin, daß multimediale Module zu individuellen Inhaltspaketen geschnürt werden, einzelne Lernpfade zielbezogen angelegt und eigenverantwortlich beschrritten werden können.“

Desweiteren „vermitteln fallbasierte Computerlernprogramme etwa im naturwissenschaftlichen Unterricht oder in der Mediziner Ausbildung ein hohes Maß an Authentizität“ [41]. Der Arbeitsstab des Forum Bildung zieht unter anderem die folgenden Schlußfolgerungen:

- „Eine neue Lern- und Lehrkultur muss die Individualisierung der Lernprozesse ermöglichen.,,
- „Bei offenen Lehr- und Lernformen - z.B. Freiarbeit, Projektarbeit, Stationenlernen - werden die Lehrenden zum Lernmoderator und Impulsgeber.“

- „Lernsoftware sollte das selbständige und kreative Arbeiten mit Computern ermöglichen. Möglichkeiten des Internets zu Recherche und zum Kontakt mit Partnern außerhalb der Bildungseinrichtung sollten gezielt genutzt werden.“

Raum zum Handeln findet sich demnach zum Beispiel in der Weiterentwicklung von Selbstlernmaterialien und Lernsoftware und in der Verpflichtung zur Evaluierung von Lehr- und Lernprozessen. Lehrende müssen sich ein neues Selbstverständnis in der Lehre aneignen und Lernende müssen den Umgang mit multimedialen Lehr- und Lernumgebungen ihrerseits erst erlernen. E-Learning mit seinen Möglichkeiten zum Selbstlernen soll und kann aber Präsenzveranstaltungen nicht ersetzen, sondern bietet vielmehr die Chance, ihre Inhalte von reiner Wissensvermittlung hin zu wissenschaftlichem Diskurs und zur Beratung und Unterstützung zu verschieben.

Auch die vorliegende Arbeit beschäftigt sich mit E-Learning, also dem Einsatz neuer Medien in der Lehre. Die Werkzeuge zur Erstellung von multimedialen Lehrinhalten auf der einen Seite und die entwickelte Lernplattform, also Tools zur Präsentation von Lerninhalten (siehe Kapitel 3), auf der anderen Seite versuchen, möglichst viele der in den vorigen Abschnitten genannten Folgerungen und Empfehlungen umzusetzen. Zusätzlich zu den genannten Aspekten, die vor allem das Selbststudium, also den Studenten, fördern, erleichtern die Werkzeuge aber auch die Arbeit der Dozenten. Zum Beispiel ist es nicht sinnvoll, daß an den Universitäten X und Y zwar die gleiche Vorlesung gehalten wird, ein Dozent aus Y aber nicht bereits in X geleistete Arbeit zumindest als Grundlage für seine Vorbereitung verwenden kann. Deswegen unterstützt die vorgestellte Lehr- und Lernumgebung auch die Zusammenarbeit und den Austausch von Informationen zwischen Institutionen.

Kapitel 2 gibt zunächst einen kurzen Überblick zum zentralen Thema Lernobjekte mit Definition und einer kurzen Diskussion verschiedener internationaler Standards. Starkes Augenmerk wird dabei auf die Verwendung von Metadaten zur Beschreibung von Lernobjekten gelegt. Besonders der in Abschnitt 2.2.2 beschriebene LOM-Standard dient später als Grundlage für die Modellierung der Lernobjekt-Datenbank in Abschnitt 6.4.

Kapitel 3 definiert die Begriffe Lehrplattform und Lernplattform und stellt einige existierende Lösungen vor. Dabei werden neben allgemeinen, fachübergreifenden Plattformen auch speziell für den medizinischen Bereich entwickelte Umgebungen betrachtet.

Kapitel 4 beschreibt einige wichtige medizinische Klassifikationen, die für das Verständnis der Anwendung in den nachfolgenden Abschnitten wichtig sind. So werden die vor-

gestellten ICD-Klassifikationen zur Beschreibung von Lernobjekten in Abschnitt 6.3 und das Unified Medical Language System UMLS zur Generierung eines semantischen Netzes in Abschnitt 5.3 verwendet.

Kapitel 5 geht auf die im Rahmen des Projekts SENTIMED [50] zur Erstellung einer Lehr-/Lernplattform geleisteten Vorarbeiten ein. SENTIMED unterstützt zunächst nur Dozenten, indem ihnen eine Toolbox zur Erstellung von Vorlesungen zur Verfügung gestellt wird. Studenten ziehen aus den Entwicklungen keinen direkten Nutzen.

Die Integration dieser Vorarbeiten sowie den Aufbau einer Lehr-/Lernplattform am konkreten Anwendungsbeispiel Dermatologie beschreibt Kapitel 6. Wenn es möglich ist, wird aber darauf geachtet, daß die verwendeten Konzepte auch auf andere Disziplinen anwendbar sind. Nach der Diskussion der technischen Voraussetzungen für den Einsatz einer multimedialen Lehr- und Lernumgebung (Abschnitt 6.1) erklärt Abschnitt 6.2 den Aufbau der Lehr- und Lernumgebung. In Abschnitt 6.3 werden die verschiedenen Typen von Lernobjekten in der Dermatologie beschrieben. Dabei werden zum Beispiel folgende Aspekte betrachtet:

- Wie wird bisher gelehrt/gelernt? Welche Formen der Lehre/des Lernens gibt es? Dazu zählen Vorlesungen (auch die Art, wie diese Vorlesungen stattfinden), Lehrbücher, Diskussionsgruppen, (praktische) Übungen usw.
- Welche dieser bisherigen Lehrformen kann/soll man, eventuell modifiziert, auf den Einsatz mit neuen Medien anpassen? Bestehende Lehrformen können durchaus so ausgereift sein, daß sie im Prinzip nicht verbessert werden müssen. Trotzdem kann es sinnvoll sein, multimediale Formen zu entwickeln, um zum Beispiel Dozenten Vorbereitungen zu erleichtern oder um Studenten eine einfachere Nachbereitung des Lernstoffs zu ermöglichen.
- Wie kann man Dozenten (die Lehrenden) bzw. Studenten (die Lernenden) durch den Einsatz neuer Medien, zum Beispiel durch die Entwicklung neuer Lehrformen, unterstützen? Dazu gehört sowohl die Analyse bisheriger Vorgehensweisen des Lehrenden/Lernenden als auch eventuell die Entwicklung neuer Ansatzpunkte für ein effizientes Lehren/Lernen.

In den folgenden Abschnitten werden die Lernobjekt-Datenbank (Abschnitt 6.4), Autorenwerkzeuge zur Erstellung von Lernobjekten (Abschnitt 6.5) sowie eine Lernumgebung zur Präsentation der Lernobjekte (Abschnitt 6.6) vorgestellt.

Kapitel 7 faßt dann die Ergebnisse noch einmal zusammen und stellt Erweiterungsmöglichkeiten vor.

Grundlage der vorgestellten Implementierungen waren verschiedene Projekte am Lehrstuhl für Informatik II der Universität Würzburg. Insbesondere erwähnt seien SENTIMED [50], DEJAVU [48] und HagerROM [2], [3]. Details zu den Implementierungen werden jeweils in den Anhängen ausgeführt. Die folgende Tabelle gibt noch einmal eine kurze Übersicht über die Kapitel und die zugehörigen Anhänge.

<i>Kap.</i>	<i>Inhaltsübersicht</i>	<i>siehe auch</i>
2	Lernobjekte <ul style="list-style-type: none"> ● Definition und Beschreibung ● internationale Standards zur Beschreibung/zum Austausch 	Anhang A
3	Lehr- und Lernumgebungen <ul style="list-style-type: none"> ● Definition ● Vorstellung und Bewertung vorhandener Plattformen 	
4	Medizinische Klassifikationen <ul style="list-style-type: none"> ● internationale Standards zu Beschreibung der Attribute von Lernobjekten in der Medizin 	
5	Toolbox zur Erstellung von Vorlesungen <ul style="list-style-type: none"> ● Zusammenfassung von Vorarbeiten 	Anhang B
6	Aufbau von Lehr- / Lernplattformen in der Dermatologie <ul style="list-style-type: none"> ● technische Voraussetzungen für den Einsatz und Akzeptanz ● Beschreibung der betrachteten Lernobjekte ● Aufbau der Lernobjekt-Datenbank ● Autorenwerkzeuge zur Erstellung von Lernobjekten und multimediale Lernplattform 	Anhang C Anhang D Anhang E
7	Zusammenfassung und Ausblick	

Tabelle 1: Inhaltsübersicht der Kapitel

2 Lernobjekte

2.1 Einführung und Definition

Zentraler Gegenstand der folgenden Ausführungen sind Lernobjekte. Wenn man sich einfach an der Bedeutung des Wortes orientiert, ist ein Lehr- oder Lernobjekt jeder Gegenstand, der zur Lehre / zum Lernen dienen kann. Auf den ersten Blick ist die Unterscheidung zwischen Lehr- und Lernobjekt nur eine Betrachtung desselben Objekts aus einer anderen Sicht: aus Sicht des Dozenten ist eine Vorlesung ein Lehrobjekt, aus Sicht des Studenten ist die gleiche Vorlesung ein Lernobjekt. Aber schon, wenn man nicht nur Lehr-/Lerninhalte betrachtet, sondern zum Beispiel auch die zugehörige Software als Objekt sieht, ist eine Software zur Erstellung von Vorlesungen in diesem Sinne ein Lehr-, nicht aber ein Lernobjekt.

In anderen Arbeiten werden Lernobjekte in verschiedenen Anwendungen auf unterschiedliche Weise definiert. Im IEEE-Learning Object Metadata Standard [44] (siehe Abschnitt 2.2.2) wird unter einem Lernobjekt *„jede (digitale oder nicht-digitale) Einheit, die für Lernen, Lehre oder Übung verwendet werden kann“*, verstanden. Als Beispiele für Lernobjekte werden hier multimedialer Inhalt, Lehrinhalte, Lernsoftware, aber auch Personen, Organisationen oder Ereignisse, auf die während des Lernens verwiesen wird, genannt. Es wird also

alles, was in irgendeiner Weise mit Lernen zu tun hat, als Lernobjekt bezeichnet. Dazu gehören also zum Beispiel auch Autorentools zur Erstellung von Lehreinheiten, die zwar für Dozenten/Autoren, nicht aber für Studenten relevant sind. Es wird also nicht zwischen einem Lehr- und einem Lernobjekt unterschieden. Dieser sehr umfassenden, allgemeinen Sichtweise steht in [49] eine relativ spezielle Definition gegenüber, in der „*Lernobjekte kurze, multimediale und interaktive Sequenzen sind, welche ein definiertes Lernziel mit einem bestimmten didaktischen Zugang verfolgen*“. Die Vielfalt der Objekte unterstützt dabei den Lernerfolg für verschiedene Lerntypen. Mit dieser Definition werden aber keine (nicht interaktive oder multimediale) Lerntexte oder Skripte erfaßt.

In der vorliegenden Arbeit werden im Gegensatz zur ersten Definition nur Lehr- und Lerninhalte als Lernobjekte verstanden. Lernobjekte werden aber auch nicht so stark eingeschränkt wie in der zweiten Definition:

Definition:

Ein **Lernobjekt** ist jede inhaltlich zusammengehörende Lerneinheit.

Es wird also nicht zwischen einem Lehr- und einem Lernobjekt bzw. Lehr- und Lerninhalten unterschieden. Jedes Lernobjekt ist gleichzeitig Lehrobjekt, es kann aber Lehrenden, Dozenten oder Autoren, und Lernenden in unterschiedlicher Weise präsentiert werden. So kann ein Dozent ein Lernobjekt „Powerpoint-Präsentation“ erstellen, zu dem er sich Notizen oder Bemerkungen für den Vortrag speichert. Diese sind zwar nun Teil des Lernobjekts, sind aber später nicht für den Studenten sichtbar.

Lernobjekte sind oft hierarchisch anordenbar: Das Lernobjekt *Vorlesung* besteht aus einzelnen *Folien*-Lernobjekten, auf denen wiederum *Bild*-Lernobjekte enthalten sein können usw. Beispiele von Lernobjekt-Typen in der Dermatologie sind in Abschnitt 6.3 zu finden.

Abschnitt 2.2 gibt zunächst eine kurze Einführung zum Thema Metadaten und diskutiert dann, wie Lernobjekte mit Hilfe von Metadaten einheitlich beschrieben und dadurch auch zwischen verschiedenen Umgebungen ausgetauscht werden können. Abschnitt 2.3 stellt zwei weitere interessante Spezifikationen zum Austausch von Lerninhalten bzw. zur Beschreibung von Quizelementen, zum Beispiel Prüfungsfragen, vor. Abschnitt 2.4 faßt die Ansätze mit einer kurzen Bewertung zusammen.

Die in diesem Kapitel vorgestellten Definitionen und Spezifikationen bilden eine wesentliche Grundlage für den Aufbau der Lernobjekt-Datenbank in Abschnitt 6.4.

2.2 Metadaten für Lernobjekte

Metadaten sind strukturierte Daten, die die Eigenschaften eines Objekts (oder allgemeiner einer Ressource) beschreiben und somit die Möglichkeit schaffen, selbstbeschreibende Ressourcen zu generieren, die dann, auch inhaltlich, rechnergestützt verarbeitet werden können. Erst die Beschreibung von Lernobjekten durch Metadaten ermöglicht den Austausch von Lernobjekten zwischen Institutionen und ihre gemeinsame Verwendung in unterschiedlichen Lernumgebungen und Software-Anwendungen. Ohne den Einsatz von Metadaten sind die Objekte meist nur über speziell für das jeweilige Objekt vorhandene Tools verwendbar, da entweder die Objekte kompatibel zu einer Anwendung erstellt wurden oder die Tools umgekehrt so implementiert wurden, daß sie genau diese Objekte verarbeiten können. Lernobjekt-Metadaten bilden also gewissermaßen eine Schnittstelle zum Austausch von Lernobjekten.

Metadaten im allgemeinen sind im Grunde nichts anderes als *Daten über Daten*. T. D. Wason liefert in seiner Erklärung von Metadaten in [63] folgende Beschreibungen: „*Metadaten sind beschreibende Informationen*“ und „*Metadaten beschreiben irgendetwas über Daten*“. Demnach kann man fast alles als Metadaten bezeichnen, was auch nur irgendeine Aussage über irgendetwas macht. Erst die folgende Definition von T. Berners-Lee in [30] stellt durch die Verwendung des Begriffs *maschinenverständlich* den Bezug zur digitalen Datenverarbeitung her.

Definition:

Metadaten sind maschinenverständliche Informationen über Web-Ressourcen oder andere Dinge. (T. Berners-Lee in [30])

Im folgenden werden unter Metadaten in diesem Sinne digital gespeicherte und auswertbare Daten über Daten verstanden.

Ursprünglich wurden Metadaten im wesentlichen zur Definition von Datenfeldern und den ihnen entsprechenden Wertebereichen in Datenbanken verwendet. Weil allein durch diese Angaben aber ein Objekt nicht inhaltlich interpretiert werden kann, werden nicht nur technische, sondern auch interpretierende, inhaltliche Informationen als Metadaten abgelegt. Sie beschränken sich nicht nur auf das Anwendungsgebiet von Datenbanken.

Metadaten können folgende Eigenschaften haben:

- **Metadaten sind kontextabhängig.** Unter einem Kontext versteht man hier verschiedene Sichtweisen auf die gleichen Daten, d.h. je nachdem aus welchem Blickwinkel man Daten betrachtet, können sie Daten oder Metadaten sein. Deshalb fällt es häufig schwer, Daten und Metadaten zu unterscheiden. Metadaten können auch selbst wieder durch Metadaten beschrieben werden, die dann oft auch als *MetaMetaDaten* bezeichnet werden. Dies entspricht dann allerdings nur einer höheren Abstraktionsstufe aus einer bestimmten Sicht. T. Berners-Lee formuliert in [30] die Aussage „*Metadaten sind Daten*“ als Axiom.
- **Metadaten sind nicht eindeutig.** Für ein und dieselbe Ressource können also beliebig viele Sätze von Metadaten bestehen („*Anyone can say anything about anything*“ in [30]).

Metadaten müssen nicht in einer einfachen, flachen Form vorliegen, sie können hierarchisch strukturiert sein. Beispielsweise hat eine Person unter anderem einen Namen, der wiederum aus dem Vor- und Nachnamen besteht, und eine Adresse, die aus Straße, Postleitzahl, usw. besteht. Will man den Autor eines Werks angeben, so kann dieser aus mehreren Personen bestehen. Die Struktur vom Metadaten wird auch als das *Datenmodell* bezeichnet. Die einzelnen Elemente des Datenmodells bezeichnet man als *Metadatenfelder*, bei der Beschreibung der Metadaten kann für jedes Feld eine Definition angegeben werden, die das Interpretieren des Inhaltes ermöglicht.

Die konkrete Angabe von Metadaten zu einer bestimmten Ressource bezeichnet man als *Metadateninstanz*.

Metadaten können auf unterschiedliche Weise dargestellt werden, die technische Darstellung oder Umsetzung von Metadaten wird als *Bindung* bezeichnet, sie kann zum Beispiel mit Hilfe der *eXtensible Markup Language* (XML) [69], des *Resource Description Framework* (RDF) [68] oder in Form einer Datenbank erfolgen. Zum Austausch von Metadaten bieten sich aber XML und RDF an, da beides vom W3C Consortium zur Beschreibung strukturierter Daten entwickelt und empfohlen wurde.

In XML-Bindung sieht eine Instanz eines Metadaten-Satzes zur Beschreibung von Autoren zum Beispiel folgendermaßen aus:

```
<autor>
  <person>
    <vorname>Max</vorname>
    <name>Mustermann</name>
    <adresse>
      <strasse>Musterweg</strasse><nr>1</nr>
      <plz>10000</plz><ort>Musterstadt</ort>
    </adresse>
  </person>
</autor>
```

Ein **Metadaten**system ist die Kombination von Feldern, ihren Definitionen, Datenformaten, Strukturen und Bindung. Durch ein Metadaten system werden also nicht nur die Eigenschaften von Feldern, sondern auch ihre Bedeutung und ihre Beziehungen untereinander definiert.

Die Hauptgründe, Objekte mit Hilfe von Metadaten zu beschreiben, sind:

(1) Autarkie

Oft kann eine Ressource sich selbst nicht ausreichend beschreiben. Zum Beispiel kann nach den Inhalten von Bildern nicht oder nur schwer gesucht werden, wenn die entsprechenden beschreibenden Daten nicht in textueller Form abgelegt werden. Der Einsatz von Metadaten soll letztendlich zu *selbstbeschreibenden Ressourcen* führen.

(2) Skalierbarkeit

Oft sind Recherchen auf sehr großen Datenbeständen nötig. Solche Suchanfragen lassen sich enorm beschleunigen, wenn nicht mit einer Volltextsuche der gesamte Datenbestand, sondern wenn durch Auswahl definierter Metadatenfelder nur ein Teil durchsucht werden muß.

(3) Kompatibilität

Der Austausch von Informationen und Ressourcen zwischen verschiedenen Suchmaschinen, Applikationen, Systemen, Lernumgebungen wird ermöglicht, wenn die verschiedenen Systeme die jeweiligen Metainformationen des anderen Systems auf ihr eigenes Metadaten system abbilden können.

Abschnitt 2.2.1 stellt eines der ersten und bekanntesten Metadaten systeme, den *Dublin Metadata Core Element Set*, vor. Abschnitt 2.2.2 beschreibt die Arbeit des *IEEE Learning Technology Standards Committee (LTSC)*, das einen *Standard for Information Technology – Education and Training Systems – Learning Objects and Metadata* erarbeitet. Abschnitt 2.2.3 faßt Arbeiten der Fakultät für Mathematik und Informatik der Universität Passau an

einer Auszeichnungssprache für Lerninhalte, der *Learning Material Markup Language LMML*, zusammen.

2.2.1 Dublin Core Metadata Element Set (DCMES)

Das wohl bekannteste Metadaten-System ist der *Dublin Core*. Im März 1995 fand in Dublin, Ohio, der erste Metadata Workshop statt. Ergebnis des Treffens war unter anderem ein minimaler Satz von Metadaten zum leichten Auffinden von dokumentenähnlichen Objekten in Netzwerkumgebungen wie dem Internet, der sogenannte *Dublin Core Metadata Element Set (DCMES)*. Inzwischen enthält der von der *Dublin Core Metadata Initiative* [37] gepflegte *Dublin Core* die folgenden 15 optionalen Elemente:

Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage, Rights

In XML-Bindung sieht eine Instanz des DCMES zum Beispiel folgendermaßen aus:

```
<dcmes>
<title>
  Multimediale, datenbankgestützte Lehr- und Lernumgebungen
</title>
<creator>Holger Höhn</creator>
<subject>e-learning, Lernumgebung, Lernplattform</subject>
<description>Die Arbeit beschreibt den Aufbau einer multimedialen,
  datenbankgestützten Lehr- und Lernumgebung
  am Anwendungsbeispiel der Dermatologie....
</description>
<publisher>unbekannt</publisher>
<contributor></contributor>
<date>05/2002</date>
<type>file</type>
<format>PDF</format>
<language>de</language>
</dcmes>
```

Alle Elemente dürfen für eine Ressource mehrfach aufgeführt werden. Nach eigenen Angaben erfüllt der DCMES die Eigenschaften Einfachheit, semantische Kompatibilität, internationale Übereinstimmung und Erweiterbarkeit mit dem Ziel der Ergänzung bestehender Methoden zur Suche und Indexierung von webbasierten Metadaten.

Anwendung findet DCMES unter anderem im Metadaten-Projekt deutscher Bibliotheken *META-LIB* [38]. Das von der Deutschen Forschungsgemeinschaft DFG geförderte Projekt lief von 1997 bis 2000. Beteiligt waren unter anderem das deutsche Bibliotheksinstitut, die deutsche Bibliothek (DDB) und die Bayerische Staatsbibliothek. Die Elemente des DCMES

wurden auch (teilweise unter anderer Bezeichnung) in den *IEEE Learning Object Metadata-Standard* (Abschnitt 2.2.2) integriert.

2.2.2 Der IEEE LOM – Standard

2.2.2.1 Einführung

Während sich der *Dublin Core Metadata Element Set* auf Metadaten und -systeme für dokumentenähnliche Objekte bezieht, entwickelt das *IEEE Learning Technology Standards Committee (LTSC)* speziell für Lernumgebungen technische Standards, Empfehlungen für praktische Umsetzungen und Leitfäden für Software-Komponenten, Tools, Technologien und Design-Methoden. Unter anderem erarbeitet die Arbeitsgruppe P1484.12 (*Learning Objects Metadata Working Group*) des LTSC einen *Standard for Information Technology – Education and Training Systems – Learning Objects and Metadata*, im folgenden auch kurz **LOM – Standard** für *Learning Object Metadata Standard* genannt. Er spezifiziert die Syntax und Semantik von Lernobjekt-Metadaten. Die folgenden Ausführungen beziehen sich auf den entsprechenden Working Draft in der Version 6.1 vom 18.04.2001 [44].

Bereits 1997 rief das *IMS Global Learning Konsortium* [47] eine Initiative zur Entwicklung eines offenen, marktorientierten Standards für Online-Lernen inklusive einer Spezifikation für Metadaten ins Leben. Ähnliche Bestrebungen gab es zur gleichen Zeit durch eine IEEE Gruppe, aus der später das LTSC entstand. Noch im gleichen Jahr startete IMS eine Zusammenarbeit mit dem europäischen *ARIADNE*-Projekt. *ARIADNE* (1995) und *ARIADNE II* (1998) waren durch die Europäische Union finanzierte Projekte, die sich mit der Entwicklung von Tools und Methoden zur Erstellung, Verwaltung und Wiederverwendung von computerbasierten pädagogischen Elementen beschäftigten. Die Validierung fand unter anderem in verschiedenen akademischen Institutionen in Europa statt. Inzwischen ist aus den Projekten die *ARIADNE Foundation* [28] entstanden.

1998 reichten *IMS* und *ARIADNE* einen gemeinsamen Vorschlag für eine Spezifikation bei der IEEE ein, die Grundlage für die weitere Arbeit der IEEE LTSC Arbeitsgruppe am LOM-Standard. 1999 veröffentlichte *IMS* die Arbeit des LTSC in den USA, Europa, Australien und Singapur und brachte das Feedback in die weitere Entwicklung des Standards ein.

Der Standard beschränkt sich auf einen minimalen Satz von Eigenschaften, die nötig sind, um Lernobjekte zu verwalten, lokalisieren und zu evaluieren. Das *konzeptuelle Daten-*

schema des Standards definiert die Datenfelder, aus denen eine Metadateninstanz für ein Lernobjekt zusammengesetzt ist.

Ein Lernobjekt ist im Sinn des Standards jede digitale oder nicht-digitale Einheit, die während des Technologie-unterstützten Lernens (wiederholt) benutzt werden kann.

2.2.2.2 Das konzeptuelle Datenschema des LOM – Standards

Das konzeptuelle Datenschema spezifiziert die Datenfelder, aus denen eine Metadateninstanz für ein Lernobjekt zusammengesetzt ist:

Definition:

Ein *LOM-Datenfeld* ist ein Datenfeld, für das Name, Erklärung, Größe, Reihenfolge, Wertebereich und Datentyp gemäß der Beschreibung im LOM-Standard definiert sind.

Verschiedene LOM-Datenfelder, die zueinander in engeren Beziehungen stehen, werden hierarchisch zusammengefaßt. Die oberste Hierarchie-Ebene (das Basisschema) bilden neun sogenannte *Kategorien*, die in der folgenden Liste stichpunktartig jeweils mit Beispielen von untergeordneten Datenfeldern erklärt werden:

- (1) *general*: allgemeine Informationen zum Lernobjekt insgesamt
enthält: Titel, Sprache, Schlüsselworte, ...
- (2) *lifecycle*: Historie und aktueller Zustand des Lernobjekts; Personen, die irgend etwas mit dem Lernobjekt zu tun haben.
enthält: Version; Status; beteiligte Personen, ...
- (3) *Meta-metadata*: bezieht sich nicht auf das Lernobjekt selber, sondern auf Informationen über den vorliegenden Metadaten-Satz
enthält: Autor; Sprache, ... (jeweils der Metadaten, nicht der Daten!)
- (4) *technical*: technische Anforderungen und Eigenschaften des Lernobjekts
enthält: Format; Größe; Speicherort, ...
- (5) *educational*: erzieherische und pädagogische Eigenschaften des Lernobjekts
enthält: Interaktivitätsgrad; Lernobjekt-Typ (z.B. Übung, Experiment, ...); Kontext (z.B. Grundschule, Hauptschule, ...); Schwierigkeitsgrad, ...
- (6) *rights*: Urheberrechte / Kosten / Nutzungsbedingungen des Lernobjekts
enthält: Kosten; Copyright, ...

(7) *relation*: Beziehungen des vorliegenden Lernobjekts zu anderen Lernobjekten

enthält: Art der Beziehung (z.B. Teil von, basiert auf, ...); Beschreibung, ...

(8) *annotation*: Kommentare (inkl. dessen Autor) zu diesem Lernobjekt

(9) *classification*: Einordnung des Lernobjekts in bestimmte Klassifikationen

enthält: Ziel und Quelle der Klassifikation; ID innerhalb der Klassifikation,

Auch die 15 Elemente des Dublin Core (Kap. 2.2.1) sind im LOM-Standard, allerdings unter anderen Bezeichnungen, enthalten. Im Anhang des LOM-Standards -WD 6.1 [44] findet man eine Mapping-Tabelle zwischen den LOM-Datenelementen und den Elementen des *Dublin Core*. Eine Auflistung aller LOM-Datenelemente ist in Anhang A, Seite 139, abgedruckt.

Eine Metadaten-Instanz wird als *konform* zum LOM-Standard bezeichnet, wenn sie ausschließlich aus LOM-Datenelementen besteht. Die Kategorie *classification* kann ausdrücklich dazu benutzt werden, das Datenschema zu erweitern. Dies gilt allerdings nur, wenn solche Erweiterungen nicht bereits bestehende Datenelemente ersetzen. Jedes Lernobjekt kann mehrere Sätze von Lernobjekt-Metadaten haben.

Über die Repräsentation von Metadaten macht das IEEE LTSC keine Vorschrift, sondern schlägt für den Moment XML als Austauschformat vor. Allerdings wird nicht ausgeschlossen, daß in Zukunft eine Art der gemeinsamen Repräsentation definiert wird.

2.2.2.3 Ziele

Mit Hilfe des LOM-Standards soll ermöglicht bzw. erleichtert werden:

- die Entwicklung, Verwaltung und der Einsatz und die Kompatibilität von computerbasierten Lern- und Trainingskomponenten und Systemen.
- das Suchen, Auswerten, Benutzen von Lernobjekten durch Lernende und Lehrende
- die Entwicklung von Lernobjekten in Einheiten, die sinnvoll kombiniert und zerlegt werden können
- das Ausdrücken von Bildungsinhalten und Leistungsstandards in einem standardisierten Format (unabhängig vom Inhalt)
- die Versorgung von Forschungseinrichtungen mit Standards, die das Erfassen und gemeinsame Verwenden von vergleichbaren Daten unterstützen

Der Standard soll von anderen Standards referenziert werden, um den Aufbau von Lernsystemen, Katalogen etc. zu ermöglichen.

2.2.2.4 Modifikationen und Spezifikation des IMS

Das an der Entwicklung des LOM-Standards beteiligte *IMS Global Learning Consortium* faßt seine Arbeit auf dem Gebiet der Lernobjekt - Metadaten in drei Dokumenten zusammen:

- (1) Das „*IMS Learning Resource Meta-data Information Model*“ enthält neben dem IEEE-LOM-Standard WD 6.1 eine relativ kleine Liste von Modifikationen durch das IMS-Konsortium. Die meisten dieser Änderungen beziehen sich auf den Datentyp von Feldern. Die IMS-Spezifikation ersetzt häufig den Wertebereich *restricted vocabular* (wie eigentlich im LOM-Standard vorgesehen) durch einen *LangString*, also einen Text inklusive Kennzeichnung der Sprache.
- (2) Das „*IMS Learning Resource Meta-data XML Binding*“ beschreibt, wie die Lernobjekt-Metadaten mit Hilfe von XML dargestellt werden können.
- (3) Der „*IMS Learning Resource Meta-data Best Practice and Implementation Guide*“ beschreibt die Vorgehensweise zur Beschreibung von Lernobjekten mit dem Standard in der Praxis. In früheren Versionen (z.B. Version 1.1. final Specification) verfolgte das IMS-Konsortium den Ansatz, den mit über 80 Datenfeldern sehr umfangreichen IEEE LOM Standard in unbedingt notwendige Elemente (*Core of Elements*) und Erweiterungen (*Standard Extensions*) aufzuteilen. Als Begründung wurde angeführt, daß eine so große Anzahl von Feldern die einfache Implementierung von standard-konformen Anwendungen eher hindern würde. Die Core-Elemente umfaßten nur noch 19 Elemente. In Version 1.2 ließ man diese Aufteilung in Core und Extension-Elemente mit der Begründung wieder fallen, daß die Erwartungen durch eine solche Vereinfachung nicht erfüllt wurden.

Außerdem stellt IMS auch DTDs und XML-Schemata auf seiner Homepage [47] zum Download bereit, mit denen generierte XML-Datensätze auf Korrektheit geprüft werden können.

Das IMS-Konsortium beschreibt genauer als der IEEE-LOM-Standard, wann eine Metadaten-Instanz bzw. eine Metadaten-Applikation konform zur IMS-LOM-Spezifikation ist:

Eine **Metadaten-Instanz A** ist konform zur IMS Metadata Spezifikation, wenn folgende vier Punkte erfüllt sind:

- (1) A enthält mindestens ein IMS-LOM-Datenfeld
- (2) (a sei ein IMS-LOM-Datenfeld in A) =>
(a wird wie in der IMS-LOM Spezifikation definiert benutzt.)
- (3) (a sei ein Element in A und a ist ein Element der IMS-LOM Spezifikation) =>
(die Werte von a sind in A strukturiert wie in der IMS-LOM Spezifikation und diese Struktur ist in A enthalten.)
- (4) (a sei ein Element in A, a sei nicht in der IMS-LOM Spezifikation) =>
(Es gibt kein b in der IMS-LOM-Spezifikation, so daß $a = b$. Das heißt: es darf kein Element ergänzt werden, das (unter anderer Bezeichnung) bereits in der IMS-LOM Spezifikation enthalten ist.)

Eine **Metadaten-Applikation B** ist IMS-LOM konform, wenn folgendes gilt:

- (1) B muß mindestens ein Datenfeld verarbeiten können, das in der IMS-LOM Spezifikation enthalten ist.
- (2) (B bekommt als Input eine zur IMS-LOM Spezifikation konforme Instanz A) =>
(bei Weiterübertragung muß B die Original-Metadaten-Instanz A exakt erhalten.)

2.2.3 Learning Material Markup Language - LMML

An der Fakultät für Mathematik und Informatik der Universität Passau wurde eine eigene XML-basierte Auszeichnungssprache für Lehr-/Lerninhalte entwickelt, die *Learning Material Markup Language (LMML)*. Ziel des Projekts ist es, in Lernsystemen anspruchsvolle Anfragemöglichkeiten, thematisches/typspezifisches Filtern für individuellen Zugang, adaptive Präsentation und Navigation zu ermöglichen.

In verschiedenen Publikationen zu LMML ([20], [19]) werden die Eigenschaften von Lehrmaterial folgendermaßen zusammengefaßt:

- Lehrmaterial besteht aus unterschiedlichen, kleinen (konzeptuellen) Einheiten unterschiedlicher Granularität.
- Die konzeptuellen Einheiten haben eine innere Struktur.
- Konzeptuelle Einheiten können auf andere Einheiten in unterschiedlicher Weise verweisen.

- Metainformationen gibt es auch zu Inhaltsobjekten von konzeptuellen Einheiten.

Außerdem gibt es eine modulare Grobstruktur von konzeptuellen Einheiten in Bezug auf Präsentation, Komposition, Navigation und Beziehungen. Außer diesen allgemeinen Eigenschaften gibt es domänenspezifische Eigenschaften, nämlich domänenspezifische

- Arten von konzeptuellen Einheiten
- Informationen zu konzeptuellen Einheiten
- Strukturierung/Zugriff
- Typen von Verweisen.

Konform zu diesen Eigenschaften von Lehrmaterial wurde ein Meta-Modellierungsansatz für Lehrmaterial auf verschiedenen Ebenen gewählt:

- das sogenannte *Abstract Meta-Modell* ist von Fachgebieten unabhängig, beschreibt Lehrmaterial allgemein, stellt also eine allgemeine Art der Strukturbeschreibung für Lernsysteme dar.
- Die *Domain-Specific Models* beschreiben modulare Grob- und konzeptuelle Feinstruktur, definieren Schemata, die die erlaubte Form festlegen. Hier können z.B. auch Guided Tours (Sequenzen von konzeptuellen Einheiten) festgelegt werden.

Es wird nicht der Inhalt, sondern die Dokumentenstruktur des Lehrmaterials modelliert. Dadurch soll der Aufwand für die Modellierung des Inhalts vermieden werden. LMML wurde zur Repräsentation von konzeptuellen Einheiten definiert.

Die Spezifikation von LMML umfaßt:

- Inhaltsmodule (z.B. conclusion, definition, example, paragraph, ...)
- Medienobjekte (text, image, sound, animation, code, math, ...)
- Strukturobjekte (Listen und Tabellen)
- Texterweiterungen (Texthervorhebung, Formeln, Code, Links, ...)
- Strukturmodule (section, collection, glossary, bibliography, ...)

Das folgende Beispiel zeigt ein einfaches LMML-Dokument mit einem eingebundenen Bild und Latex-Code:

Beispiel:

```
<?xml-version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE lmmml SYSTEM „...“>
<lmmml version="-//DE.UNI-PASSAU.DAISY//DTD LMML-CS 1.1//EN">
<paragraph title="Paragraph mit Medienobjekten">
<LMMLtext>Das ist LMML-Text. Es folgt ein Bild:</LMMLtext>
<image type="image/gif" uri="../images/bild2.gif">
<text type="text/latex">
<![CDATA[
Das ist eine Liste in Latex:
\begin{itemize}
\item Punkt 1
\item Punkt 2
\end{itemize}
] ]>
</text>
</paragraph>
</lmmml>
```

Eine Instanz von LMML ist LMML-CS, eine Auszeichnungssprache für Lehrmaterial im Informatik-Umfeld [40]. Nach eigenen Angaben geht der Passauer Ansatz über die IEEE/IMS-LOM Spezifikation hinaus, weil auch domänenspezifische Beschreibung sowie die Beschreibung von kleineren konzeptuellen Einheiten (Definitionen, Beweise) unterstützt wird. Das Passauer Framework erlaubt auch, andere Metadaten-Spezifikationen einzubinden. Allerdings wird nicht näher erläutert, wie diese Einbindung umzusetzen ist.

2.3 IMS - Spezifikationen

Das *IMS Global Learning Consortium* beschäftigt sich neben der Mitarbeit am IEEE-LOM-Standard unter anderem auch mit der Erarbeitung eines standardisierten Satzes von Strukturen zum Austausch von Lerninhalten, dem *IMS Content Packaging Information Model* und einer Spezifikation zur Repräsentation von Fragen und Testdaten, der *Question & Test Interoperability (QTI) Spezifikation*.

2.3.1 IMS Content Packaging Information Model

Die folgenden Ausführungen beziehen sich auf die Version 1.1 vom April 2001 [46]. Ziel des Modells ist es, einen standardisierten Satz von Strukturen zu definieren, der zum Austausch von kompletten Inhalten zwischen verschiedenen Lernumgebungen verwendet werden kann. Dazu dienen *Packages*, die neben den wirklichen physikalischen Dateien (den Inhalten) ein Manifest-File enthalten (Abbildung 1).

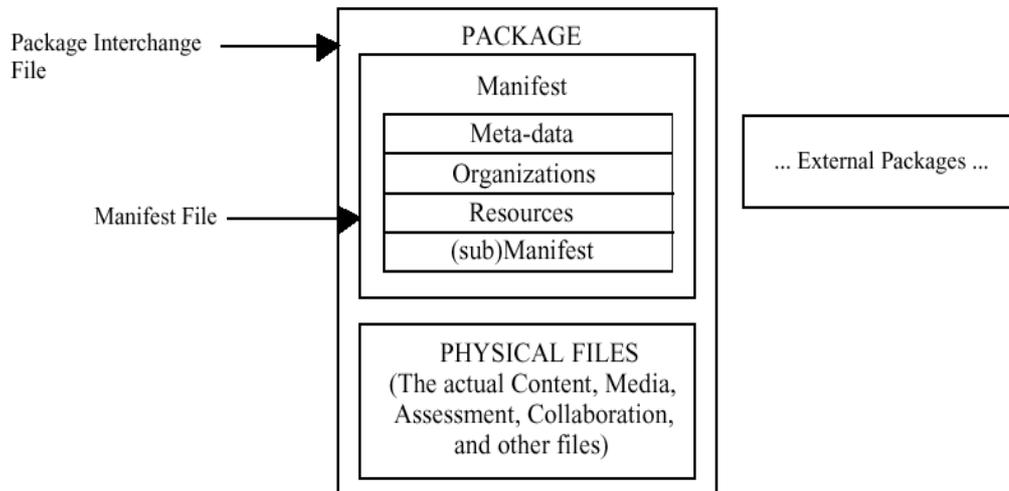


Abbildung 1: Modell des IMS Content Packaging Information Model aus [46].

Das Manifest-File (siehe Beispiel unten in der empfohlenen XML-Bindung) enthält im `<metadata>`-Tag Metadaten, die das Paket beschreiben, im `<organizations>`-Tag die über den Inhalten liegenden Hierarchie/Navigationsstrukturen und im `<resources>`-Tag eine Liste aller verwendeten Ressourcen. Werden die Ressourcen zusammen mit dem Manifest in eine Datei gepackt (zip, cab, o.ä.), wird diese Datei als *Package Interchange File* bezeichnet. Ressourcen können zum Beispiel Webseiten, Mediendateien, Textdateien, Quizelemente oder sonstige Daten in Form von Dateien sein.

Beispiel für Manifest-File:

```
<?xml version="1.0"?>
<manifest identifier="M01" version="1.0"
  xmlns="x-schema:IMS_CONTENTv1p0.xdr">
  <metadata>
    <schema>IMS_CONTENT</schema>
    <schemaversion>1.0</schemaversion>
    <record xmlns="x-schema:IMS_METADATAv1p1.xdr">
      <general>
        <title><langstring lang="de_DE">Dermatologie</langstring></title>
        <description>
          <langstring lang="de_DE">Skript zur Dermatologie</langstring>
        </description>
        <keywords>
          <langstring lang="en_US">dermatology</langstring>
        </keywords>
      </general>
    </record>
  </metadata>
```

```

<organizations>
  <tableofcontents identifier="MT0" title="eintitle">
    <item identifier="MI1" identifierref="MR1" title="Einleitung">
    </item>
    <item identifier="MI2" identifierref="MR2" title="All. Erkrankungen">
      <item identifier="MI3" identifierref="MR3" title="Allgemeines">
      </item>
      <item identifier="MI4" identifierref="MR4" title="Typ I-Reaktionen">
        <item identifier="MI5" identifierref="MR5" title="Urtikaria ">
        </item>
      </item>
    </item>
  </tableofcontents>
</organizations>
<resources>
  <resource identifier="M1" type="webcontent"><file href="ein.htm"/>
</resource>
  <resource identifier="MR2" type="webcontent"><file href="all.htm"/>
</resource>
  <resource identifier="MR3" type="webcontent"><file href="allg.htm"/>
</resource>
  <resource identifier="MR4" type="webcontent"><file href="typ1.htm"/>
</resource>
  <resource identifier="MR5" type="webcontent"><file href="urt.htm"/>
</resource>
</resources>
</manifest>

```

2.3.2 IMS Question & Test Interoperability (QTI) Specification

Die *Question & Test Interoperability (QTI) Spezifikation* [45] beschreibt Basisstrukturen zur Repräsentation von Fragen und Testdaten. Dadurch wird die Austauschbarkeit solcher Daten zwischen Lernumgebungen und Lernmanagement-Systemen ermöglicht. Die kleinste austauschbare Einheit ist ein *Item*. Ein Item kann nicht aus anderen Items kombiniert werden. Eine *Section* besteht aus einer beliebigen Mischung von Items oder anderen Sections. Ein *Assessment* muß mindestens eine Section enthalten, Items können nicht direkt in einem Assessment enthalten sein. Beziehungen zwischen Assessments sind nicht vorgesehen. Für Items sind unter anderem folgende Typen vorgesehen:

- Standard wahr / falsch Fragen
- Standard Multiple Choice Fragen. Dabei können Texte, Bilder oder Tonspuren als Antwortalternativen angegeben werden.
- Fragen mit Freitextantworten.

Zur Repräsentation wird XML verwendet. Das folgende Beispiel stellt eine Standard-Multiple-Choice Frage mit (mehreren) richtigen Textantwort-Vorgaben dar:

Beispiel eines Items:

```

1<questestinterop>
2  <item title="Multiple Response Item" ident="Beispiel1">
3    <presentation label="BasicExample005b">
4      <material>
5        <mattext>Wann kommen typische
6          Nagelveränderungen vor?</mattext>
7      </material>
8      <response_lid ident="MR01" rcardinality="Multiple">
9        <render_choice>
10         <response_label ident="A">
11           <material><mattext>Psoriasis</mattext></material>
12         </response_label>
13         <response_label ident="B">
14           <material><mattext>Lichen ruber</mattext>
15           </material>
16         </response_label>
17         <response_label ident="C">
18           <material><mattext>Alopecia areata</mattext>
19           </material>
20         </response_label>
21       </render_choice>
22     </response_lid>
23   </presentation>
24   <resprocessing>
25     <outcomes>
26       <decvar varname="SCO" vartype="Integer" defaultval="1"/>
27     </outcomes>
28     <respcondition>
29       <conditionvar>
30         <varequal respident="MR01">A</varequal>
31       </conditionvar>
32       <setvar action="Add" varname="SCO">5</setvar>
33       <displayfeedback linkrefid="Correct"/>
34     </respcondition>
35   </resprocessing>
36   <itemfeedback ident="Correct">
37     <material>
38       <mattext>Ja, richtige Antwort, weil ...</mattext>
39     </material>
40   </itemfeedback>
41</questestinterop>

```

Aus dem Beispiel wird deutlich, daß nicht nur die Fragestellung (Zeilen 4-7) und Antwortalternativen (8-21) in XML-Schreibweise dargestellt werden können. Vielmehr können auch die richtigen Antworten bestimmt (29) und mit Kommentaren (36-38) versehen werden. Mit Hilfe von Variablen ist auch eine Punktebewertung möglich (31). Daneben bietet die Spezifikation unter anderem auch Methoden zur Beschreibung von Zeitvorgaben, Layouts von Fragen und Antworten.

2.4 Zusammenfassung und Bewertung

Mit Hilfe von Metadaten können Objekte in einfacher Weise sowohl für den Menschen lesbar als auch rechnergestützt auswertbar beschrieben werden. Der IMS Meta-Data Best Practice and Implementation Guide [27] beschreibt den Zweck einer Metadaten-Spezifikation für Lernobjekte folgendermaßen: „*Eine Metadaten-Spezifikation ermöglicht effizientes Finden und Benutzen von Ressourcen, indem eine Struktur von definierten Elementen bereitgestellt wird, die die Lernressource zusammen mit den Nutzungs- und Darstellungsanforderungen beschreiben oder katalogisieren.*“

Für die Austauschbarkeit von Lernobjekten zwischen verschiedenen Anwendungen bzw. Lernumgebungen ist es nötig, daß auch die Metadaten-Informationen zueinander kompatibel sind. Besonders gute Voraussetzungen dafür bietet der *IEEE-LOM Standard* bzw. die Modifikationen durch das IMS Global Learning Consortium:

- (1) Er ist bereits recht weit gereift und wird von verschiedenen Institutionen und Firmen verwendet bzw. referenziert. Nicht zuletzt gibt der Projektträger *Neue Medien in der Bildung* des bmb+f den IEEE-LOM-Standard als Referenz an [60].
- (2) Er wird mit der IEEE von einer allgemein anerkannten Organisation entwickelt.
- (3) Er ist weitgehend deckungsgleich mit den Anforderungen, die man zur Beschreibung von Lernobjekten (in der Dermatologie) hat (siehe Abschnitt 6.3).
- (4) Er ist andererseits so flexibel, daß er beinahe beliebig erweiterbar ist.
- (5) Das *IMS Global Learning Consortium* bietet DTDs / XML-Schemata sowie Anleitungen zur praktischen Umsetzung des Standards an, die die Arbeit mit dem Standard erleichtern.

Deshalb wird in dieser Arbeit versucht, die Beschreibung von Lernobjekten möglichst nahe an den *IEEE-LOM Standard* anzulehnen. Dabei wird nicht so sehr auf die exakte Abbildung des Standards als auf eine möglichst einfache Konvertierbarkeit einer Lernobjekt-Beschreibung in eine zum LOM-Standard konforme Metadateninstanz Wert gelegt. Das ermöglicht den nötigen Freiraum für eigene Entwicklungen bei gleichzeitiger Bindung an den Standard. So wird bei der Beschreibung der Eigenschaften von Lernobjekten in Abschnitt 6.3 auf die Abbildbarkeit auf den LOM-Standard geachtet. In Abschnitt 6.3.9 ist das Matching der Lernobjekt-Attribute mit den LOM-Datenfeldern zu finden. Der LOM-Standard bildet dann auch die Grundlage für das Design der Lernobjekt-Datenbank in Abschnitt 6.4. Außer-

dem wird in Abschnitt 6.7 der Export von Lernobjekten als zum LOM-Standard konforme Metadateninstanz beschrieben.

Die *Learning Material Markup Language LMML* erscheint trotz des vielversprechenden Ansatzes der verschiedenen Ebenen sehr auf spezielle Anwendungen angepaßt und für unsere Anforderungen zunächst nicht geeignet. Es müßte hier zunächst eine eigene LMML-Instanz entwickelt werden. Dies würde dann zwar die Beschreibung von Lernobjekten mit Hilfe von LMML ermöglichen. Da aber LMML keinerlei internationalen Standard darstellt, würde die Implementierung einer Insellösung entsprechen. Der in Zukunft eventuell erwünschte Austausch von Lernobjekten kann dann wieder nur durch eine unter Umständen aufwendige Konvertierung gewährleistet werden. Ein kombinierter Absatz scheint hier sinnvoller: die Beschreibung eines Lernobjekts erfolgt grundsätzlich mit dem LOM-Standard. Ist das zu beschreibende Lernobjekt ein Dokument, so wird es in XML mit einer entsprechend entwickelten DTD repräsentiert.

Die beiden anderen Spezifikationen des IMS Global Learning Consortium scheinen durchdacht und sinnvoll. Abschnitt 6.7 beschreibt, wie Lerninhalte der erstellten Lernumgebung konform zum IMS Content Packaging Information Model exportiert und in anderen Lernumgebungen integriert werden können. In Zukunft sollten auch die in Abschnitt 6.3 vorgestellten Quizelemente konform zur IMS QTI Specification exportiert werden können.

3 Lehr- und Lernumgebungen

3.1 Einführung und Definition

Definition: Ein *Autorenwerkzeug* ist jedes Werkzeug, das der Erstellung von (multimedialen) Lernobjekten dient.

Definition: Eine *Lehrumgebung (Lehrplattform)* ist eine Sammlung von Tools für den Einsatz in der Lehre.

Definition: Eine *Lernumgebung (Lernplattform)* ist ein (eine Sammlung von) Tool(s) zur (interaktiven) Präsentation von Lernobjekten zum Zweck des (Selbst-)Lernens.

Die Begriffe Autorenwerkzeug und Lernumgebung implizieren schon die verschiedenen Benutzergruppen: *Autorenwerkzeuge* dienen den Dozenten und Autoren zur Erstellung von Lernobjekten. Zur Generierung der verschiedenen Lernobjekt-Typen können dabei verschiedene Werkzeuge verwendet werden, solange die gemeinsame Verwendung der erstellten Objekte in einer gemeinsamen Lernumgebung für die Studenten gewährleistet werden kann. Dies kann zum Beispiel durch die Konformität der Autorenwerkzeuge zum LOM-Standard realisiert werden. Eine *Lehrplattform* besteht aus diesen Autorenwerkzeugen zusammen mit den Tools, die von Dozenten während des Lehrens eingesetzt werden. Dazu

gehören zum Beispiel Präsentationsprogramme wie Powerpoint.

Auf studentischer Seite bietet sich die Integration der unterschiedlichen Lernobjekte in eine gemeinsame Benutzeroberfläche einer *Lernumgebung* an. So muß sich der Student nicht an unterschiedliche Benutzerführungen gewöhnen und Verweise zwischen Lernobjekten werden erleichtert. Das schließt natürlich nicht aus, daß für spezielle Objekttypen eigene Viewer gestartet werden, dies sollte aber aus einer gemeinsamen Navigationsengine transparent für den Nutzer geschehen.

Eine umfangreiche Aufzählung wichtiger Merkmale von Lernplattformen und eine lange Liste vorhandener Lernplattformen findet man zum Beispiel im Gutachten *Selektions- und Entscheidungskriterien für die Auswahl von Lernplattformen und Autorenwerkzeugen* [17], das Prof. Dr. R. Schulmeister im Auftrag des österreichischen Bundesministeriums für Bildung, Wissenschaft und Kultur (bm:bwk) erstellt hat. Er teilt die wichtigsten Merkmale von Lernplattformen in die 10 Kategorien *Administration, Kursmanagement, Didaktik, Kommunikation, Medien, Design, Evaluation, Technologie & Technik, Support* und *wirtschaftliche Gesichtspunkte* ein, nimmt aber kaum eine Bewertung einzelner Produkte vor, sondern gibt eine detaillierte Liste von über 200 Kriterien vor. Der jeweilige Anwender muß daraus auswählen, welche für ihn wie wichtig sind und dann vergleichen, ob eine Lernplattform diese Kriterien ausreichend erfüllt.

Ein wichtiges Feature von Lernumgebungen ist die Möglichkeit der Definition von unterschiedlichen *Lernpfaden*.

Definition: Ein *Lernpfad* ist eine Folge von Lernobjekten.

Zum einen sollten Autoren die Möglichkeit haben, Lernpfade festzulegen, die man auch als *Guided Tours* bezeichnen könnte. Um zum Beispiel einen ganzen Themenbereich zu erarbeiten, können dem Lernenden Lernpfade in unterschiedlichen Schwierigkeitsgraden oder zu unterschiedlichen Zwecken (erstmaliges Lernen oder Wiederholung) angeboten werden. Dabei soll dem Student aber nicht ein Pfad aufgezwungen werden. Vielmehr kann er den Weg an beliebigen Stellen verlassen, um an für ihn besonders interessanten oder unklaren Bereichen weitergehende Informationen zu suchen.

Zum anderen kann aber auch der Lernende selbst Lernpfade definieren, indem er sich zum Beispiel in einer Art editierbaren History merken kann, in welcher Reihenfolge er die Lernobjekte besucht hat.

Die folgenden Abschnitte beschreiben anhand einiger Beispiele von Autorenwerkzeugen und Lernumgebungen aktuelle Entwicklungen auf diesem Gebiet.

3.2 Microsoft Learning Resource iNterchange (LRN) -Toolkit

Microsoft bezeichnet das *Learning Resource iNterchange (LRN) -Toolkit* [53] als die erste kommerzielle Applikation der *IMS Content Packaging Spezifikation* (Abschnitt 2.3.1). Es soll zur Identifizierung, zum Verteilen, Updaten und zur Generierung von Online Inhalten und Kursen dienen. Es enthält Tools zur Organisation, zum Betrachten und zur Validierung von bestehenden Lernobjekten sowie einige Beispiele und Dokumentationen. Die Lernobjekte selbst können aber nicht mit dem Toolkit erstellt werden. Über das Tool zur Zusammenstellung von Lerninhalten, dem *LRN-Editor*, werden zur *IMS Content Packaging Spezifikation* konforme XML-Manifest-Dateien (siehe Abschnitt 2.3.1) erzeugt und verwaltet (Abbildung 2). Es wird links der Navigationsbaum mit Metadaten, Inhaltsverzeichnissen und Ressourcenverwaltung und rechts die Ressource im Detail mit HTML-Preview angezeigt, Voraussetzung dafür ist ein installierter Microsoft Internet Explorer, Version 5.5 oder höher. Der Spezifikation entsprechend lassen sich Metadaten, Inhaltsverzeichnisse (Organizations) und Ressourcen verwalten.

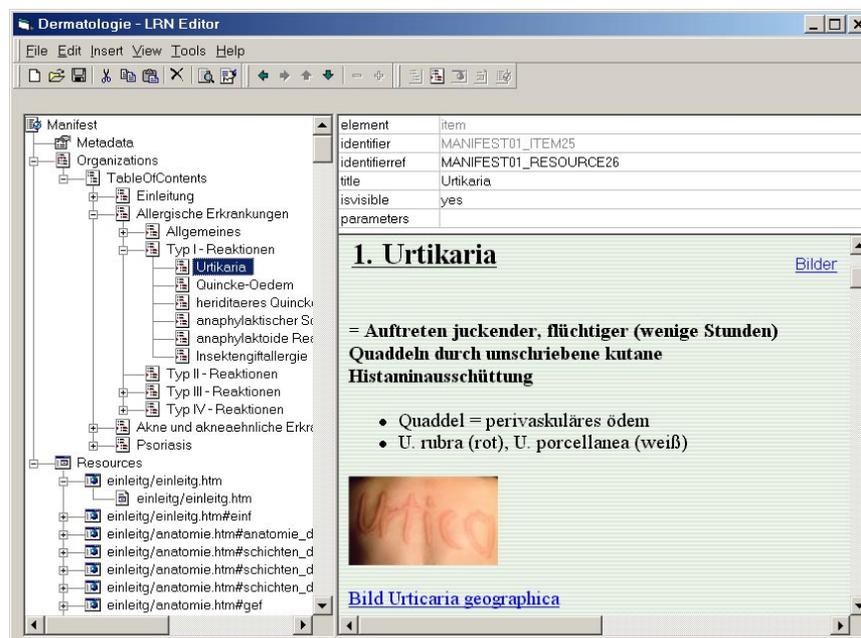


Abbildung 2: Microsoft LRN Editor.

Allerdings ist das Hinzufügen von Ressourcen auf HTML-Dateien mit den Variationen *Erstellen einer neuen Datei* oder *Angabe einer bestehenden HTML-Datei* bzw. *absoluten*

URL beschränkt. Das Einfügen von Ankern als eigene Ressource bzw. Eintrag im Inhaltsverzeichnis wird dabei nicht unterstützt.

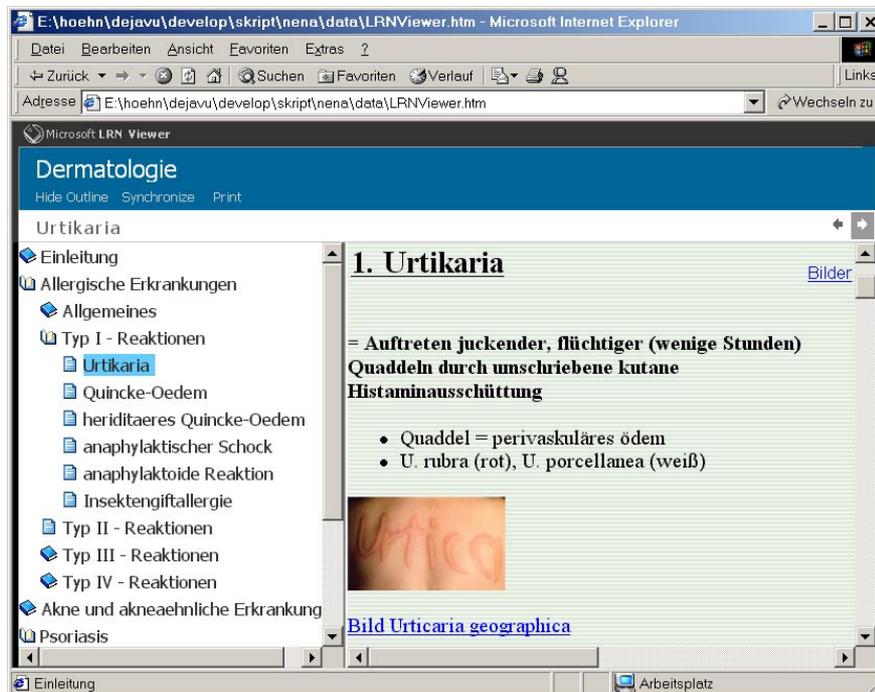


Abbildung 3: Microsoft LRN-Viewer.

Der Lernende verwendet dann den Internet Explorer (ab Version 5.5) als *LRN-Viewer*. Dort wird das XML-Manifest-File zusammen mit den Ressourcen in eine Oberfläche mit Navigationsbaum und Blättermöglichkeit umgesetzt (Abbildung 3). Den Viewer kann man im Moment allerdings nur als sehr einfache Lernumgebung bezeichnen. Es fehlen einige nützliche Funktionen. So hat der Nutzer zum Beispiel keine Möglichkeit, Bemerkungen einzugeben oder Textstellen zu markieren.

3.3 e-Learn

In der *e-Learn Software* der Firma *E-Teach GmbH* [39] entsprechen die Navigationsbäume Lernpfaden. Sie können ähnlich wie beim *Microsoft LRN-Toolkit* durch Einfügen von inneren Knoten und Blättern erstellt werden. Blätter können ausgehend von einer begrenzten Auswahl von vordefinierten Layouts generiert werden.

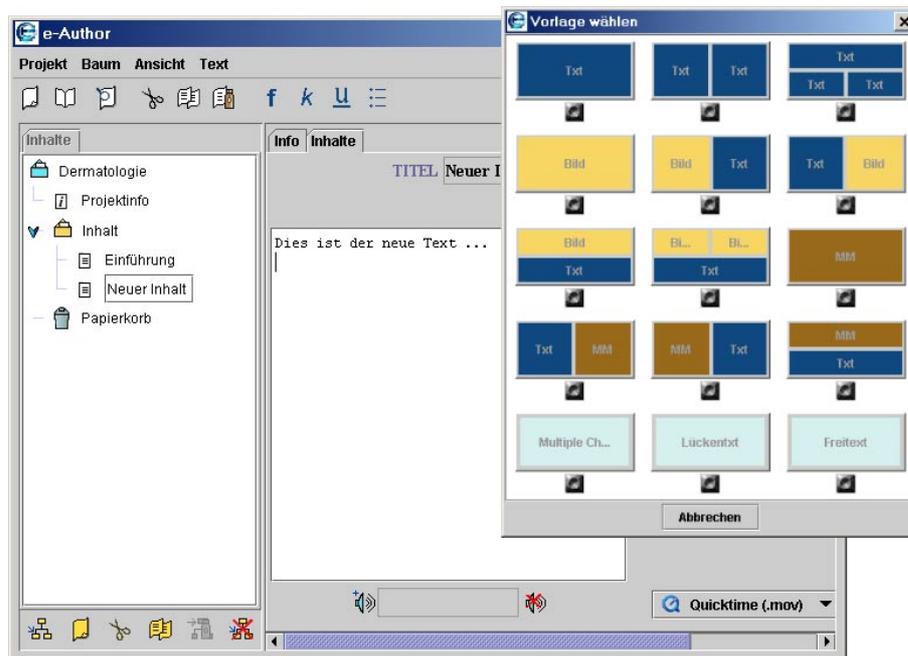


Abbildung 4: e-Author: Navigationsbaum und Vorlagen.

Dabei ist das Einbinden bereits bestehender Lernobjekte nur bedingt möglich, dies muß bei Texten zum Beispiel über die Zwischenablage erfolgen, bei Powerpoint-Vorlesungen über einen Export der Folien als Pixelbilder. Neben der Erstellung von Texten mit einfachen Formatierungen und Tabellen können nur Audiodateien und Videos eingefügt werden. Als Prüfungselemente können Freitext- oder Multiple Choice Fragen oder ein Lückentext erstellt werden. Diese werden dann aber nicht vom System, sondern von einem Tutor bewertet.

Die *e-Learn Software* will in 5 Modulen alle am Lernprozeß Beteiligten (Kursplaner, Kursautor, Tutoren, Verwalter und Lernende) unterstützen:

- *e-Author* ist ein Tool zur Erstellung von Kursen (Abbildung 4).
- *e-Tutor* stellt Funktionen zur Kontrolle des Lernfortschritts und Ausbildungslösungen bereit.
- *e-Editor* bietet zusätzliche Möglichkeiten, Lernpfade festzulegen.
- *e-Admin* ermöglicht die Kontrolle des Lernfortschritts.
- *e-Student* stellt die Lernumgebung des Pakets dar (Abbildung 5). Sie wird innerhalb eines Browsers dargestellt. Neben der Abarbeitung von Kursen entlang der Lernpfade hat der Lernende über entsprechende Buttons Zugang zu den Funktionen Lerngeschichte, E-Mail, schwarzes Brett, Chat und Pause. Als Pausen-Funktion werden Audiodateien

abgespielt oder einfache Spiele gestartet. Auf einzelne Lerninhalte können Notizen eingefügt werden und Lesezeichen gesetzt werden.



Abbildung 5: e-Student. Lernumgebung der e-Teach GmbH

3.4 Blackboard 5: Learning System

Das Lernsystem der *Blackboard Inc.* [31] ist eine webbasierte Server-Software, bestehend aus einem Content Management System, Kommunikationskomponenten, Prüfungs- und Kontrollelementen. Blackboard ist an der Entwicklung von IMS-Standards beteiligt. Autoren können webbasiert Basis-Lerninhalte direkt mit Formularen erzeugen oder Dateien per Upload in Kurse einbinden. Kurse werden in Form von Lernpfaden durch Lerninhalte und Prüfungen definiert. Den Studenten stehen persönliche Werkzeuge wie Terminplaner, Aufgabenplaner und ein Adreßbuch zur Verfügung. Sie können in den Lerninhalten browsen, Kurse belegen, mit anderen Teilnehmern über Chat oder in Foren diskutieren und die Seiten an ihre persönlichen Bedürfnisse und Vorlieben anpassen.

Nach eigenen Angaben wird das Lernsystem unter anderem in über 40 Universitäten in den USA, Europa, Australien und Asien eingesetzt. Es stehen bereits fertige, editierbare Lerninhalte in Form von Paketen, sogenannten *Course Cartridges* zur Verfügung. Sie enthalten die unterschiedlichsten Inhalte wie Folien, Dokumente oder Multimedia-Files. *Course Cartridges* sind von unterschiedlichen Verlagen erhältlich.

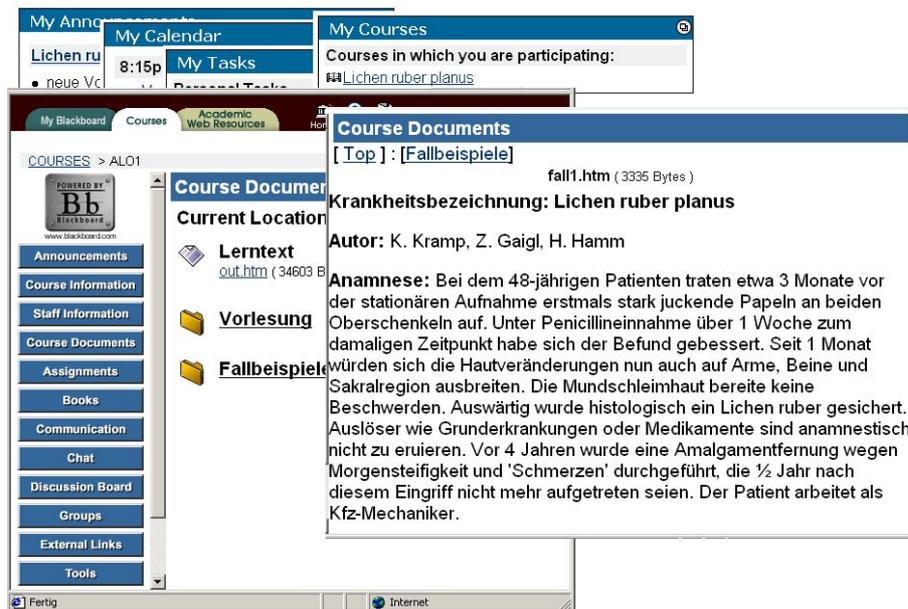


Abbildung 6: Blackboard 5. Lernumgebung.

3.5 Authoring on the Fly

An der Universität Freiburg wird *Authoring on the Fly (AOF)* [26] entwickelt. Das System besteht im wesentlichen aus dem Tool *mlb* zur Präsentation von Vorträgen oder Vorlesungen (Abbildung 7). Es dient selbst als Server, das heißt: die Vorträge können über Netz live an beliebig vielen Clients mitverfolgt werden.

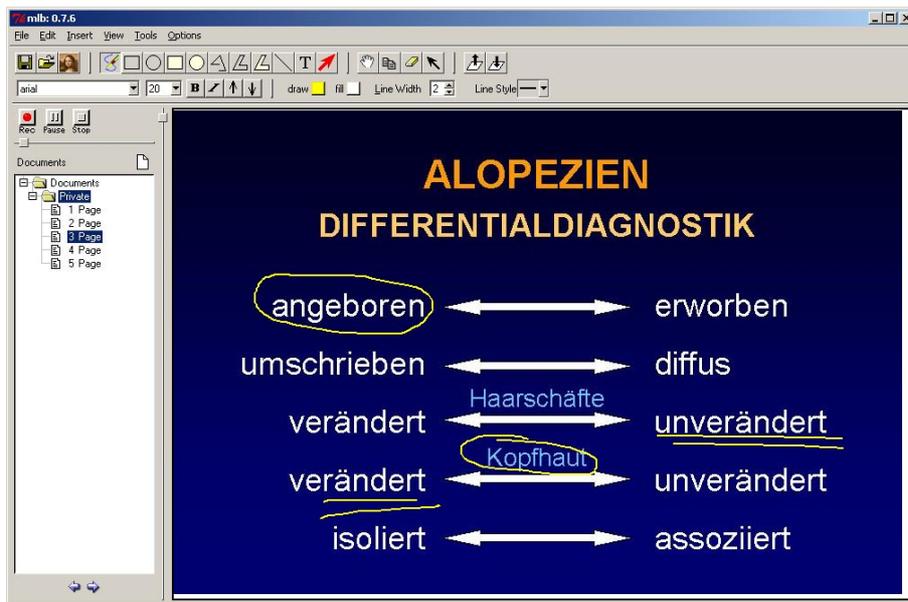


Abbildung 7: AOF. Erstellung und Präsentation von Vorträgen.

Dazu werden weitere Features wie eine Chat-Funktion oder das Hand-Raising-Tool (die Möglichkeit, Fragen von beliebigen Clients aus zu stellen) angeboten. Zusätzlich kann ein Vortrag aufgezeichnet werden. Dabei werden neben den Präsentationsfolien auch Notizen, die der Dozent auf den Folien gemacht hat sowie die Tonspur aufgezeichnet. Der Student hat auf diese Weise die Möglichkeit, Vorlesungen inklusive Ton und Bemerkungen zeitlich navigierbar nachzubereiten. Dazu stellt AOF das Tool *aofSync* bereit. AOF bietet aber keine Möglichkeit der Verwaltung von Lernobjekten oder Erstellung anderer Lernobjekte. Da *mlb* bisher nur rudimentäre Werkzeuge anbietet, müssen auch die Präsentationsfolien selbst in anderen Applikationen (zum Beispiel Powerpoint) erstellt werden und dann als Pixelbilder in AOF importiert werden.

3.6 Medizinische Lernumgebungen

In diesem Abschnitt werden einige weitere Lernumgebungen aus dem medizinischen Bereich mit dem Schwerpunkt Dermatologie kurz beschrieben. Auf alle folgenden Umgebungen greift der Lerner über einen Standard-WWW-Browser zu. Allen gemeinsam ist außerdem, daß jeweils nur ein Typ von Lernobjekten verwendet wird. Die Plattformen beschränken sich auf vom jeweiligen Server abrufbare Lerninhalte, d.h. es werden keine Installationsroutinen für Autorenwerkzeuge zur Erstellung von Lernobjekten oder für die Lernumgebung angeboten. Man ist also auf die vorhandenen Inhalte angewiesen.

Die *Dermatology Image Bank* [34] und der *Dermatology Online Atlas (DOIA)* [35] stellen eigentlich nur dermatologische Bilddatenbanken dar. Es werden keine Lernpfade oder Kurse angeboten. Sie sind deshalb auch nur bedingt als Lernumgebung einzuordnen. *DOIA* bietet immerhin ein Bilderquiz, in dem der Lerner anhand eines Bildes die dargestellte Diagnose nennen muß.

Im *Dermatologie Praktikum 2000 (Derma2000)* [33] werden dem Lerner Patienten vorgestellt, die er richtig behandeln soll. Dazu müssen unter anderem die richtigen Verfahren angewendet werden.

Warren Piette's *Introduction to Basic Dermatology* [59] ist eine Sammlung von HTML-Seiten, die mit Bildern wichtige Hautkrankheiten sehr knapp beschreiben. Wie der Name schon sagt, stellt es nur eine Einführung dar und wurde zuletzt 1996 überarbeitet.

Das Java Applet *CAMPUS* [62] dient der „computergestützten Aus- und Weiterbildung in der Medizin mit einem flexiblen und simulativen fallbasierten System“.

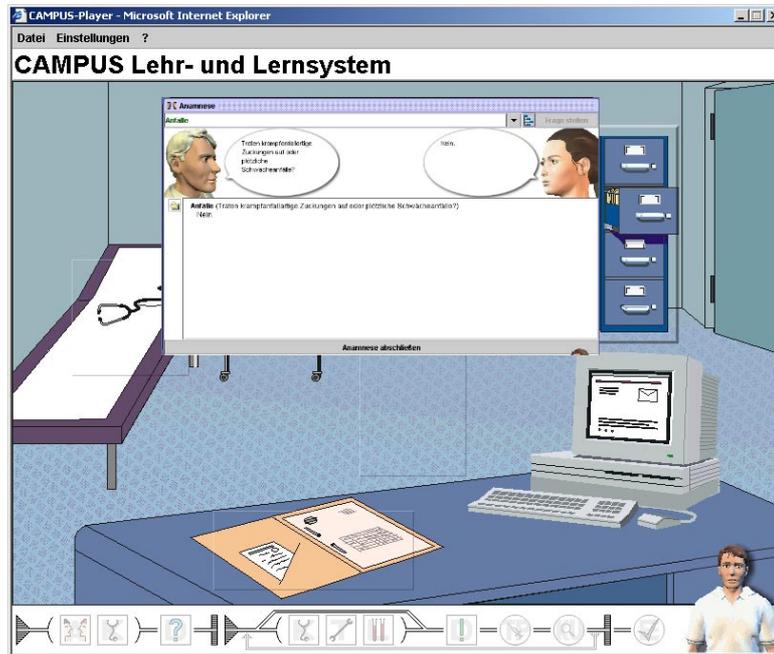


Abbildung 8: CAMPUS Lehr- und Lernsystem

Das auf der Internetseite zur Verfügung stehende Beispiel ist sehr ausführlich und graphisch sehr ansprechend aufbereitet (Abbildung 8). Der Lernende befindet sich mit dem Patienten in einem virtuellen Behandlungszimmer. Durch Anklicken der unterschiedlichen Gegenstände werden Aktionen ausgelöst. So wird zum Beispiel durch Anklicken des Aktenschrankes die Akte des Patienten geöffnet.

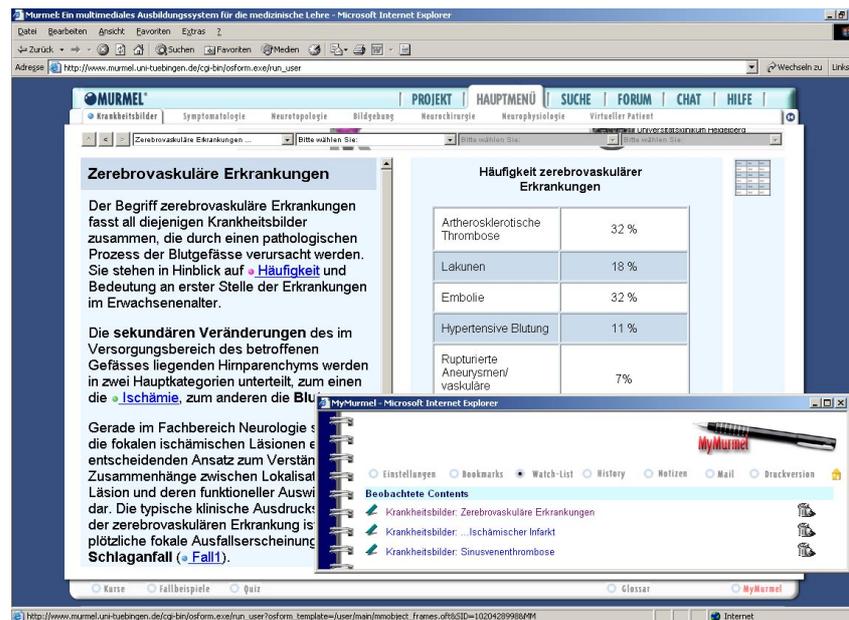


Abbildung 9: MURMEL Ausbildungssystem für die medizinische Lehre.

Das *Multimediale Ausbildungssystem für die Medizinische Lehre MURMEL* [54] bietet neben erklärenden Texten auch Fallbeispiele und Kommunikationskomponenten wie Foren und Chat. Die Navigation ist nicht wie bei den meisten Systemen direkt als Baum abgebildet und deshalb auch etwas gewöhnungsbedürftig (Abbildung 9). Jeder Eintrag kann auf eine Lesezeichenliste und auf eine Beobachtungsliste gesetzt werden. Die Eingabe von eigenen Bemerkungen zu einzelnen Einträgen ist aber nicht möglich.

3.7 Zusammenfassung und Bewertung

Das Microsoft LRN Toolkit befindet sich noch in der Entwicklung und hat sicher noch nicht ausreichend Funktionalität, es enthält zum Beispiel keine Autorenwerkzeuge zur Erstellung von Lernobjekten. Außerdem muß der Lerner auf seinem System zunächst eine aktuelle Version des Internet-Explorer installieren. Blackboard 5 scheint vom Funktionsumfang sehr ausgereift zu sein und bietet an einigen Stellen eher zuviel Funktionalität. Am nächsten an der gewünschten Lehr- und Lernumgebung ist von den beschriebenen Paketen *e-Learn*. Allerdings kann hier nur eine begrenzte Anzahl von Lernobjekt-Typen eingefügt werden. Die Lernobjekte können nicht einfach in anderen Objekten wiederverwendet werden. Alle Lernobjekte sind textbasiert (oder müssen in Textdokumente eingebunden werden) und die Layout-Möglichkeiten sind durch die vorgegebenen Templates doch stark eingeschränkt. Der vielversprechende *AOF* Ansatz ist sehr speziell auf Vorlesungen zugeschnitten und würde sich deshalb eigentlich nur für den Einsatz als ein Modul einer Lernumgebung eignen.

Die vorgestellten Lernumgebungen im medizinischen Bereich sind vielfach einfache Skripte oder Bilddatenbanken und werden dem Anspruch einer Lernumgebung nicht gerecht. Sehr weit gereift ist allerdings das *CAMPUS*-Projekt mit seiner guten graphischen Aufbereitung zusammen mit der sehr umfangreichen Hinterlegung mit medizinischem Datenmaterial und Fachwissen. Allerdings findet man auf der *CAMPUS*-Internetseite [62] nur ein Demo für die Seite des Lernenden. Ein Autorensystem konnte nicht getestet werden. Die Plattformen sind nicht installierbar sondern nur vom jeweiligen Server abrufbar. Sie beschränken sich jeweils auf bestimmte Lernobjekt-Typen. Eine Anpassung oder gar Integration eigener Lerninhalte scheint nicht möglich zu sein.

Im Rahmen des DEJAVU-Projekts [48] hat das Institut für medizinische Psychologie der Universität Jena eine Auflistung von 262 medizinischen Lernprogrammen im Internet erstellt [61]. Dabei wurden unter anderem die jeweiligen Fachgebiete und die Anzahl der Lernsysteme im jeweiligen Fach erfaßt:

Anatomie	36	Gynäkologie	6
Innere Medizin	37	Orthopädie	5
Radiologie	22	Schmerztherapie	5
fächerübergreifend	20	Immunologie	4
Neurologie	20	Urologie	4
Naturwiss.(Biologie/Chemie/Physik)	16	HNO	4
Histologie	14	Onkologie	4
Physiologie	14	Anamnese	3
Pathologie	13	Arbeitsmedizin	3
Pharmakologie	11	Informationssysteme	3
Biochemie	10	Mathematik/Statistik	3
Dermatologie	10	Mikrobiologie	3
Ophthalmologie	10	Anästhesie	2
Notfallmedizin	9	Psychiatrie	2
Pädiatrie	7	Tropenmedizin	2
Allgemeinmedizin	6	Geriatrie	1
Chirurgie	6	Umweltmedizin	1
Genetik	6		

Die meisten dieser Lernumgebungen sind im Prinzip allerdings nur Internetpräsentationen. Außerdem wurde festgestellt, daß multimediale Inhalte wie Videos, Animationen, Sprache und Töne nur sehr selten eingesetzt werden.

Gegen den Einsatz eines kompletten Pakets spricht auch, daß es zum einen teuer ist (Blackboard verlangt allein für das Online-Stellen eines Kurses mit maximal 25 MB Speicherplatz für ein Jahr 295US\$) und zum anderen in vielen Punkten dann doch nicht den gesetzten Anforderungen entspricht. Man hat meist keinen Einfluß auf das Design, die Möglichkeiten der Verlinkung sind beschränkt und man kann verschiedene Lernobjekt-Typen nicht integrieren. Auch eine Verwaltung der Lernobjekte in einer Datenbank wird von keinem der Systeme unterstützt.

Weil zusätzlich auf geleistete Vorarbeiten innerhalb verschiedener Projekte am Lehrstuhl für Informatik II der Universität Würzburg (Kapitel 5, [2], [3] und [16]) aufgebaut werden kann, ist es sinnvoll, eine eigene Lehr- und Lernumgebung zu entwickeln. Sie soll zum einen die konkreten Bedürfnisse von Dermatologen zufriedenstellen, zum anderen aber auch auf andere Disziplinen anpaßbar sein. Das ist sicher für andere bildorientierte Fachrichtungen möglich, sollte aber auch noch allgemeiner anwendbar sein. Andererseits soll

auch darauf geachtet werden, daß die Qualität der Lehr- und Lernumgebung nicht durch ihre Einsatzmöglichkeit in möglichst allen Gebieten leidet. Dies scheint auch ein Problem einiger der vorgestellten Umgebungen zu sein. Sie werden aus kommerziellen Gründen so konzipiert, daß sie in möglichst vielen Fachrichtungen eingesetzt werden können. Dadurch fehlt ihnen aber oft die Möglichkeit, fachspezifische Informationen zu verarbeiten.

4 Medizinische Klassifikationen

Medizinische Klassifikationen spielen innerhalb dieser Arbeit eine wichtige Rolle zur Beschreibung von dermatologischen Lernobjekten mit Hilfe von Metadaten. Durch die Verwendung von verbreiteten Klassifikationen zusammen mit standardisierten Metadaten wird der Austausch von Lernobjekten mit anderen Anwendungen und Datenbanken, die die gleichen Klassifikationen verwenden, möglich. Im folgenden werden die wichtigsten medizinischen Klassifikationen vorgestellt: die ICD-Codes der WHO (Abschnitt 4.1), der Operationsschlüssel nach § 301 SGB V (Abschnitt 4.2), die MeSH-Codes (Abschnitt 4.3) und das Unified Medical Language System der National Library of Medicine (Abschnitt 4.4).

4.1 ICD – Codes der WHO und Modifikationen

4.1.1 Überblick

Die Masterversion der International Statistical Classification of Diseases and Related Health Problems (ICD) wird von der World Health Organization (WHO) [64] gepflegt und basiert auf der Bertillon Classification of International List of Causes of Death (1893). Aktuell ist zum Zeitpunkt der Erstellung dieser Arbeit die zehnte ICD-Revision (ICD-10). Inzwischen gibt es unter anderem folgende Modifikationen der WHO-ICD-Klassifikation:

- Die ICD-9-CM (Clinical Modification der 9. Revision) wurde in den USA speziell für den Einsatz in Krankenhäusern vom National Center for Health Statistics (NCHS) und der Health Care Financing Administration entwickelt und verwaltet.
- Die ICD10-AM (Australian Modification der 10. Revision) ist die australische Modifikation des WHO-ICD-10 Codes.
- Die ICD-10-SGBV (ICD-10, Sozialgesetzbuch V) dient in Deutschland zur Verschlüsselung von Diagnosen und damit letztendlich auch zur Abrechnung von Arztkosten. Die ICD-10-SGBV wird in Abschnitt 4.1.2 genauer beschrieben.
- Der Dermatologische Diagnosenkatalog DDK ist eine auf die Bedürfnisse der Dermatologie abgewandelte ICD (Abschnitt 4.1.3).

4.1.2 ICD-10-SGBV

Am 1. Januar 1993 ist in Deutschland das Gesundheitsstrukturgesetz in Kraft getreten, in dem unter anderem die Verschlüsselung von Diagnosen und Operationen mit internationalen medizinischen Klassifikationen vorgeschrieben wird, die vom Deutschen Institut für medizinische Dokumentation und Information (DIMDI) [36] im Auftrag des Bundesministeriums für Gesundheit herauszugeben sind. Dabei handelt es sich um die jeweils gültigen Ausgaben der Internationalen Klassifikation der Krankheiten.

In der Todesursachenstatistik wird die ICD-10 seit dem 1. Januar 1998 eingesetzt. In der ambulanten und stationären Versorgung werden Diagnosen seit dem 1. Januar 2000 nach der ICD-10 in der SGB-V-Ausgabe verschlüsselt. Die ICD10-AM (Australian Modification der 10. Revision) wird auch in Deutschland die Grundlage für Arzt-Abrechnungen ab 2003 sein. Dies ist auch der Grund, weshalb sich das DIMDI bei der Formulierung der ICD-10-SGB-V Klassifikation an der ICD-10-AM orientiert hat.

Die ICD-10-SGB-V ist gemeinfrei und kann kostenlos über die Homepage des DIMDI [36] in unterschiedlichen Formaten (als Sammlung von ASCII- oder SGML-Dateien) heruntergeladen werden. Die ASCII-Dateien sind zusammen ca. 2,3 MByte groß, die Diagnosen sind hierarchisch gegliedert. Eine Zeile steht dabei für einen Eintrag, in der ersten Spalte steht jeweils der ICD-Code für die Diagnosenbezeichnung der zweiten Spalte. Die Diagnosenhierarchie wird durch gemeinsame Code-Präfixe abgebildet.

Im Beispiel gehören alle Diagnosen mit L64.* zur Diagnosengruppe L64 – Alopecia androgenetica.

Beispiel:

L63	Alopecia areata
L63.0	Alopecia (cranialis) totalis
L63.1	Alopecia universalis
L63.2	Ophiasis.
L63.8	Sonstige Alopecia areata
L63.9	Alopecia areata, nicht näher bezeichnet.
L64	Alopecia androgenetica
L64.0	Arzneimittelinduzierte Alopecia androgenetica
L64.8	Sonstige Alopecia androgenetica
L64.9	Alopecia androgenetica, nicht näher bezeichnet

4.1.3 Dermatologischer Diagnosenkatalog - DDK

Der *Dermatologische Diagnosenkatalog DDK* basiert auf der 9. und 10. Revision der ICD Codes und wurde von einer Gruppe deutscher Dermatologen entwickelt und 1996 in Buchform herausgegeben [5]. Hintergrund war, daß die ICD-Codes für das dermatologische Umfeld oftmals zu allgemein waren. Deshalb wurden an die ICD9 und ICD10 Codes jeweils zwei Stellen angefügt, um eine detailliertere Beschreibung von Diagnosen geben zu können. Der DDK wird in der vorliegenden Arbeit zur Verschlagwortung von Lernobjekten mit Diagnosen eingesetzt, da er sich durch diese Anpassung besonders eignete. Leider beschränkt sich der DDK bei der Bezeichnung der Diagnosen auf die deutsche Sprache, was sich auf eine einfache Verlinkung zu anderen medizinischen Datenquellen nachteilig auswirkt (siehe Abschnitt 5.3.1).

Beispiel:

Ausschnitt aus Abschnitt 4.1.2 mit den erweiterten DDK – ICD-Codes

L63.000	Alopecia cranialis totalis
L63.100	Alopecia areata universalis
L63.200	S Ophiasis
L63.200	S Alopecia areata, Ophiasistyp
L63.810	Alopecia areata diffusa
L63.820	Alopecia areata fere totalis
L63.840	Alopecia areata barbae
L63.880	Alopecia areata, sonstige
L63.900	Alopecia areata o. n. A.
L64.000	S Arzneimittelinduzierte Alopecia androgenetica
L64.000	S Alopecia androgenetica, arzneimittelinduziert
L64.831	Alopezie, androgenetische mit Androgenüberschuß
L64.832	Alopezie, androgenetische vom männlichen Typ
L64.833	Alopezie, androgenetische vom weiblichen Typ
L64.835	Effluvium, androgenetisches
L64.885	Effluvium, telogenes
L64.900	Alopecia androgenetica o. n. A.

Der originale ICD-10-Code L63.8 (*Sonstige Alopecia areata*) wurde also zum Beispiel um weitere, genauer beschriebene Alopecia-Diagnosen erweitert. Außerdem wurden Synonyme in den DDK aufgenommen. Sie sind im obigen Beispiel durch ein „S“ in der zweiten Spalte gekennzeichnet.

4.2 Operationenschlüssel nach Paragraph 301 SGB V

Der Operationenschlüssel nach Paragraph 301 SGB V wurde ebenfalls vom *DIMDI* [36] erstellt. Er ist ähnlich wie die ICD-Codes hierarchisch gegliedert und wurde zunächst nur zur Verschlüsselung operativer Eingriffe eingesetzt. Seit Januar 2002 wird die Version 2.1 verwendet, um allgemein medizinische Prozeduren im Krankenhaus zu verschlüsseln. Im Bereich der dermatologischen Lernobjekte kann sie zum Beispiel zur Kodierung von Videofilmen über Operationstechniken eingesetzt werden.

Beispiel:

```
5-89 ... 5-92    Operationen an Haut und Unterhaut
5-894.1      Exzision, lokal, mit primärem Wundverschluß
5-894.18     Exzision, lokal, mit primärem Wundverschluß, Unterarm
```

4.3 Medical Subject Headings – MeSH

Die *Medical Subject Headings* [56] enthalten das von der *National Library of Medicine* (*NLM*) [58] kontrollierte Vokabular, das zur Indexierung von Artikeln, zur Katalogisierung von Büchern und zur Suche in MeSH-indizierten Datenbanken wie *PubMed* [55] verwendet wird. Die MeSH sind hierarchisch gegliedert. Einträge können dabei an verschiedenen Stellen in der Hierarchie erscheinen. *Alopecia* taucht zum Beispiel an den folgenden beiden Stellen auf (in Klammern jeweils der zugeordnete MeSH-Tree-Code):

```
Skin and Connective Tissue Diseases [C17]
  Skin Diseases [C17.800]
    Skin Appendage Diseases [C17.800.805]
      Hair Diseases [C17.800.805.305]
        Hypotrichosis [C17.800.805.305.535]
          Alopecia [C17.800.805.305.535.122]
            Alopecia Areata [C17.800.805.305.535.122.147]
            Mucinosi s, Follicular [C17.800.805.305.535.122.550]
```

und:

```
Pathological Conditions, Signs and Symptoms [C23]
  Pathological Conditions, Anatomical [C23.300]
    Alopecia [C23.300.035]
```

Besonders wichtig sind die MeSH-Terms, weil sie in der Medizin sehr bekannt und weit verbreitet sind, wenn es darum geht, Schlagworte zu Artikeln anzugeben und weil sie bei der *PubMed* – Artikelrecherche als Suchkriterium (siehe Abschnitt 5.3.5) dienen.

4.4 Unified Medical Language System - UMLS

Das *Unified Medical Language System (UMLS)* [57] wurde 1986 von der *National Library of Medicine* [58] als Forschungs- und Entwicklungsprojekt ins Leben gerufen. Ziel von UMLS ist das einfache Retrieval und die einfache Integration von elektronischer medizinischer Information aus verschiedensten Quellen. Außerdem ermöglicht das UMLS die Verlinkung von Patientendaten, Fallstudien und ähnlichem zu Expertensystemen. Weiterhin soll das UMLS Probleme durch unterschiedliche Terminologie in unterschiedlichen Klassifikationen lösen. UMLS besteht im wesentlichen aus den drei Grundbausteinen *Metathesaurus* (Abschnitt 4.4.1), *SPECIALIST Lexikon* (Abschnitt 4.4.2) und *Semantic Network* (Abschnitt 4.4.3).

Das UMLS ist nach Registrierung und Anerkennung der Nutzungsbedingungen direkt von der *National Library of Medicine* auf CD zu beziehen oder kostenlos downzuladen. Allerdings gelten für die in UMLS eingebrachten medizinischen Quellen die jeweiligen Lizenzvereinbarungen der Quelle.

4.4.1 UMLS Metathesaurus

Der Metathesaurus vereint über 60 verschiedene medizinische Vokabulare und Klassifikationen (unter anderem auch MeSH, ICD-10, ICD-9-CM) und faßt dabei verschiedene Namen für das gleiche Konzept zusammen. Die Konzepte enthalten zum Beispiel Bezeichnungen in verschiedenen Sprachen, Definitionen und Einordnungen in Hierarchien. In der Ausgabe des Jahres 2001 wurden ungefähr 800.000 Konzepte mit 1,9 Millionen Konzeptnamen erfaßt. Der Metathesaurus ist durch diese Konzepte organisiert, zwischen ihnen existieren folgende verschiedene Arten von bewerteten Beziehungen (Links):

- *broader (RB)*: eine weitläufigere Beziehung
- *narrower (RN)*: eine nähere Beziehung
- *other related (AQ)*: eine andere Beziehung als Synonym, weiter oder enger

- *like (LIKE)*: die beiden Konzepte sind „ähnlich“. UMLS gibt dazu an, daß mit „like“ verbundene Konzepte in zukünftigen Versionen Kandidaten für eine Zusammenfassung zu einem Konzept sind.
- *parent (PAR)*: eine Eltern-Beziehung in einem Quell-Vokabular
- *child (CHD)*: eine Kind-Beziehung in einem Quell-Vokabular
- *sibling (SIB)*: eine Bruder-Beziehung in einem Quell-Vokabular

Zu jedem Link zwischen zwei Konzepten gibt UMLS jeweils neben dem Typ auch die Quelle an, also aus welchem Vokabular oder aus welcher Klassifikation der Link abgeleitet wurde.

Der folgende Abschnitt veranschaulicht die Menge der aus dem UMLS Metathesaurus extrahierbaren Informationen in Auszügen am Beispiel des Konzepts *Alopecia areata*:

Beispiel: Konzept C0002171

Bezeichnung: englisch: Alopecia Areata; finnisch: paelvikaljuisuus; deutsch: Alopecia areata, ...

UMLS – Quellen: MeSH, ICD10-AM, WHO Adverse Drug Reaction Terminology (WHOART), Clinical Terms Version 3 (Read Codes) (Q199), DXplain (An expert diagnosis program), ...

Definition: A microscopically inflammatory, usually reversible, patchy hair loss occurring in sharply defined areas and usually involving the beard or scalp. (Dorland, 27th ed)

Relationen:

Insgesamt gibt es für das Konzept 209 Relationen, davon als haben es 106 als Quelle
 - mit folgenden Arten von Beziehungen : AQ: 37, CHD: 13, PAR: 9, RB: 8, RL: 2, RN: 2, RO: 11, SIB: 24
 - u.a. mit folgenden Quellen: 38 aus Mesh, 57 aus ICD10-AM, ...

Einordnung in Hierarchien:

das Konzept wird unter anderem in die folgenden Hierarchien eingeordnet:

- MSH2001: als CHILDREN: -; als ANCESTOR: Medical Subject Headings, Diseases (MeSH Category), Skin and Connective Tissue Diseases, Skin Diseases, Skin Appendage Diseases, Hair Diseases, Hypotrichosis, Alopecia; als SIBLING: Mucinosi, Follicular
- ICD10-AM: als CHILDREN: Alopecia areata, unspecified (L63.9), Alopecia universalis (L63.1), Ophiasis (L63.2), Other alopecia areata (L63.8); als ANCESTOR: ICD10-AM, Diseases of the skin and subcutaneous tissue, Disorders of skin appendages, Alopecia (capitis) totalis (L63.0); als SIBLING: Acne (L70), Androgenic alopecia (L64), Apocrine sweat disorders (L75), Cicatricial alopecia [scarring hair loss] (L66), Eccrine sweat disorders (L74), Follicular cysts of skin and subcutaneous tissue (L72), ...

Informationsquellen, in denen das Konzept gefunden wurde:

UMLS gibt an, daß das Konzept in *OMIM Online Mendelian Inheritance in Man* (Victor McKusick, Johns Hopkins University) (1996) und 110 mal in den Citations von *Medline* gefunden wurde

4.4.2 SPECIALIST Lexikon

Das Specialist Lexikon ist ein allgemeines englisches Lexikon mit medizinischen Fachbegriffen. Es wurde bisher nicht in die Lehr-/Lernumgebung eingebracht. Denkbar wäre allerdings ein Einsatz bei der Verlinkung von medizinischen Texten.

Beispiel:

```
{base=alopecia
  entry=E0008235
  cat=noun
  variants=uncount
  variants=reg}
{base=alopecia areata
  entry=E0008236
  cat=noun
  variants=uncount}
{base=alopecia circumscripta
  entry=E0008237
  cat=noun
  variants=uncount}
{base=alopecia mucinosa
  entry=E0008238
  cat=noun
  variants=uncount}
```

4.4.3 UMLS Semantic Network

Das UMLS Semantic Network enthält 134 semantische Typen, die über 54 Links (diese sind nicht mit den Links im Metathesaurus zu verwechseln!) verbunden sind. Für den Einsatz innerhalb der Lernumgebung bzw. zur Generierung von Lernobjekten erweist es sich allerdings als nicht geeignet: Durch die geringe Anzahl verschiedener Typen im Semantic Network einerseits und die Einschränkung auf das Fachgebiet der Dermatologie auf der Seite der Lernumgebung andererseits ist die Hierarchie von den in der Arbeit meist verwendeten Klassifikationen von Diagnosen so weit entfernt, daß jeder Diagnose nur genau der UMLS semantische Typ „*Disease or Syndrome*“ zugeordnet werden kann.

Beispiel: Konzept C0002171, Alopecia areata

das Konzept hat genau eine Zuordnung zu einem semantischen Typ, nämlich zu *Disease or Syndrome*

5 Toolbox zur Erstellung von Vorlesungen

Wichtige Vorarbeiten zur Erstellung der im folgenden Kapitel 6 beschriebenen Lehr- und Lernumgebung wurden im Rahmen von *SENTIMED* (*Semantic Net Toolbox for Images in Medical Education* [7], [50]) geleistet. SENTIMED war ein disziplinübergreifendes Projekt zwischen der Universitätshautklinik Würzburg und dem Lehrstuhl für Informatik II und wurde im Rahmen der *Multimedia in der Lehre*-Initiative (MEILE) vom Herbst 1997 bis Ende 1998 vom Freistaat Bayern gefördert. Ziel des Projekts war die Unterstützung von Dozenten bei der Vorbereitung und Präsentation von Vorlesungen und Vorträgen in bildorientierten Disziplinen. Die konkrete Umsetzung erfolgte am Beispiel der Dermatologie.

SENTIMED vereinfacht und beschleunigt den Arbeitsablauf zur Vorbereitung und Zusammenstellung von Vorlesungen. Als Grundlage wurde zunächst ein digitales, damals allerdings noch nicht datenbankgestütztes Bildarchiv aufgebaut. Darauf aufsetzend wurden Tools implementiert, die es dem Dozenten ermöglichen, von seinem Rechner-Arbeitsplatz bequem eine wiederverwendbare, digitale und multimediale Vorlesung zu erstellen, die dann mit PC und Multimedia-Projektor vorgeführt werden kann.

Bei der Implementierung wurde darauf geachtet, daß die Werkzeuge nicht zu sehr auf die speziellen Anforderungen der Dermatologie zugeschnitten wurden und somit mit vertret-

barem Aufwand dem Projektziel entsprechend auch an andere, bildorientierte Disziplinen anzupassen sind.

SENTIMED wurde an den Hautkliniken der Universitäten Würzburg und Jena mit Erfolg eingesetzt und unter anderem auf der *Medica98* [23], im *Hautarzt* [11], auf dem *4. World Congress on the Internet in Medicine in Heidelberg* [9], auf dem *2nd European Symposium on Teledermatology* [6] und im *Biomedical Journal* [7] vorgestellt.

Der folgende Abschnitt 5.1 beschreibt den Aufbau des Pakets sowie verschiedene Installationsvarianten. In Abschnitt 5.2 wird das digitale **Bildobjekt-Archiv** von der Auswahl geeigneter Bilder bis zur Erfassung der Daten zu den Bildern vorgestellt. Das **semantische Netz** (Abschnitt 5.3) unterstützt den Dozenten bei der Suche im Bildarchiv und bei der weiterführenden Recherche im Internet. Zum Aufbau des Netzes werden die Daten des digitalen Bildarchivs mit weiteren Informationsquellen wie dem UMLS (siehe Abschnitt 4.4) ergänzt. Das Modul zur **Vorlesungserstellung** (Abschnitt 5.4) erlaubt es dem Dozenten, einfach und komfortabel Bilder aus dem Archiv in Vorlesungen zu integrieren, die dann mit dem Modul zur **Vorlesungspräsentation** (Abschnitt 5.5) vorgeführt werden können.

5.1 Architektur, Installationsvarianten und Kopierschutz

Abbildung 10 zeigt die Client-Server-Architektur der Toolbox: Auf einem WWW-Server liegen die Bilder und verschiedene CGI-Skripte sowie das semantische Netz. Von den Clients wird über einen Standard-WWW-Browser auf das semantische Netz des Servers zugegriffen. Die Tools zur Vorlesungserstellung und -präsentation arbeiten auf den Clients mit lokalen ASCII-Dateien. Die Bilder können zusätzlich auch lokal auf dem Client liegen, um die Verarbeitung zu beschleunigen. Das Paket ist ein semistatisches System. Das heißt: es ist so ausgelegt, daß die Verwendung der Tools temporär nach der Datenerfassung und weiteren Verarbeitungsschritten vorgesehen ist.

Server	Client
WWW-Server	WWW-Browser
Semantisches Netz + CGI-Skripte (Kap. 5.3)	Vorlesungserstellung (Kap.5.4)
Bilder in versch. Auflösungen (Kap. 5.2.5)	Vorlesungspräsentation (Kap. 5.5)
	ASCII-Datenbank (Kap. 5.2.5)

Abbildung 10: Client-Server-Architektur

Nach der Erfassung neuer Daten ist die Durchführung einiger Skripte notwendig, um die Daten ins System einzubringen. In der erstellten Lehr-/Lernumgebung (Abschnitt 5.5) wurde diese lokale, semistatische Lösung durch eine dynamische Datenbank-Lösung ersetzt.

Alle Tools sind in JAVA oder mit Perl-Skripten implementiert, d.h. das Paket ist sowohl server- als auch clientseitig plattformunabhängig. Getestet wurde das System unter Windows (95/98/NT) und Linux Betriebssystemen. Als WWW-Server wird jeweils ein Apache Web-Server eingesetzt. Für die Windows-Plattformen existiert eine Installations-CDROM, die das automatische Einrichten aller in Abbildung 10 aufgelisteten Server- und Clientmodule auf demselben Rechner ermöglicht. Das heißt: der Rechner ist gleichzeitig Server und Client. Das Bildmaterial ist ebenfalls auf der CD in mehreren Auflösungsstufen abgelegt. Mit dieser CD können Dozenten das System also auch am heimischen PC ohne Internetanbindung nutzen (siehe Anhang B.1, Seite 147).

Da die Bilder des Bildarchivs einen nicht unerheblichen Wert darstellen, sind sie mit Hilfe eines Hardware-Dongles vor unerlaubter Benutzung geschützt: Auf dem Dongle der Firma Wibu befindet sich fest einkodiert ein Schlüssel, mit dessen Hilfe die Bilddaten kodiert werden. Zum Dekodieren der Bilder beim Abrufen muß der Schlüssel vom Dongle erneut verwendet werden. Diese Lösung hat den Vorteil der hohen Sicherheit, da die Bilder nur zusammen mit einem Dongle dekodiert werden können. Sie hat aber auch den Nachteil, daß jeder Nutzer Zugriff auf einen Dongle haben muß. Innerhalb eines Klinik-Intranets stellt das kein Problem dar, da die Dongles als Netzwerk-Dongles eingesetzt werden können. Das heißt: es muß nur ein Dongle innerhalb eines Intranets vorhanden sein, der von jedem Client aus erreichbar sein muß. Sinnvollerweise kann dieser Netzwerk-Dongle am Server selbst angebracht werden. Es muß aber jedem Dozenten, der das Paket außerhalb der Klinik benutzen will, ein Dongle zum Preis von ca. 50 € zur Verfügung stehen.

5.2 Bildarchiv

Abbildung 11 skizziert die Erfassung der Bilder. In den folgenden Abschnitten werden die einzelnen Schritte detailliert beschrieben. Eine Beschreibung der Konfiguration und der Implementierung ist im Anhang B.3, Seite 148 zu finden.

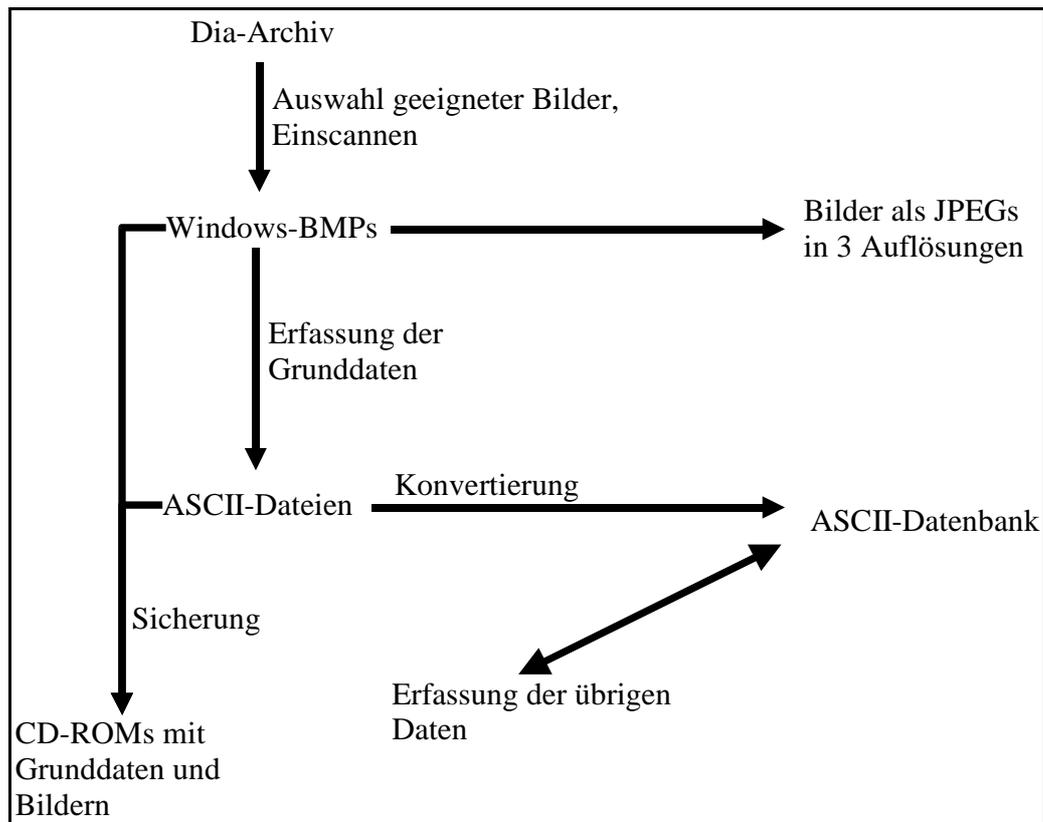


Abbildung 11: Ablauf der Datenerfassung bei den Bildobjekten

5.2.1 Auswahl geeigneter Bilder

Weil die Diaarchive in der Regel sehr große Stückzahlen umfassen und es meist von der gleichen Diagnose mehrere Bilder unterschiedlicher Qualität gibt, können und sollen nicht alle Bilder aufgenommen werden. Vielmehr muß ein Experte, zum Beispiel ein erfahrener Arzt, in einem ersten Schritt geeignete Bilder auswählen. In die Auswahlentscheidung fließen unter anderen die folgende Aspekte ein:

- *Vollständigkeit*: das resultierende Bildobjekt-Archiv soll möglichst alle in der Dermatologie vorkommenden Diagnosen abdecken.
- *Diaqualität*: alle Bilder sollten in guter Qualität vorliegen. Gute Qualität ist dabei aber in Relation zu anderen, zur gleichen Diagnose vorhandenen Bildern zu sehen. Bei seltenen Erkrankungen, zu denen es nicht viele Bilder gibt, kann es durchaus sinnvoll sein, auch qualitativ schlechtere Bilder aufzunehmen.
- *Lehrwert*: das Archiv soll primär für die Lehre genutzt werden. Es ist deshalb sinnvoll, Bilder aufzunehmen, die besonders typisch für eine Diagnose sind. Außerdem sollten jeweils auch Bilder von Differentialdiagnosen (Diagnosen, die mit der tatsächlichen Dia-

gnose leicht zu verwechseln sind) erfaßt werden, um den Lernenden auch Abgrenzungen zu anderen Diagnosen veranschaulichen zu können.

5.2.2 Reinigen und Einscannen der Dias

Da für das Reinigen und Einscannen der Bilder kein Expertenwissen benötigt wird, kann dieser Arbeitsschritt auch von einer Hilfskraft ohne große medizinische Kenntnisse übernommen werden.

In der folgenden Tabelle sind die technischen Daten beim Einscannen der Bilder in den Hautkliniken Würzburg und Jena zusammengefaßt:

	Würzburg	Jena
Rechner	233MHz, 128 MB RAM, 4fach Brenner, 18 GB Festplatten	500 MHz, 64 MB RAM, 4fach Brenner, 12 GB Festplatten
Diascanner	Nikon LS 1000	Nikon LS 2000
Anzahl Bilder	ca. 4500	ca. 6000
Auflösung	3072 x 2048 x 24 Bit (2350 dpi)	1947 x 1296 x 24 Bit (1500 dpi)
Speicherbedarf/Bild	ca. 18 MB	ca. 8 MB
Speicherbedarf (gesamt)	ca. 80 GB	ca. 50 GB

Die Entscheidung, die Bilder mit einer recht hohen Auflösung einzuscannen, hat folgende Gründe:

- Die Bilder sollen in der späteren Anwendung mindestens *monitorfüllend* präsentiert werden können. Man muß aber davon ausgehen, daß sich die üblichen Auflösungen von Grafikkarten bzw. Monitoren in der Zukunft weiter erhöhen werden. Es werden also die Voraussetzungen geschaffen, möglichst lange hochqualitatives Bildmaterial anzubieten.
- Der *Ausdruck* von Bildern in hoher Qualität soll möglich sein. Allein während der Projektlaufzeit stiegen die üblichen Druckerauflösungen von ca. 300 dpi auf ca. 2880 dpi.
- Es sollen qualitativ hochwertige *Vergrößerungen* von Bildausschnitten möglich sein. Um dabei nicht auf berechnete Vergrößerungen angewiesen zu sein, können diese auch aus den originalen, hochaufgelösten Bildern ausgeschnitten werden (siehe Abschnitt 5.3.3)

In Jena wurde die Qualität der Bilder zugunsten schnellerer Verarbeitung beim Einscannen

und bei der Konvertierung reduziert.

Zum Teil sind die Dias so stark verunreinigt, daß sie einzeln mit einem Pinsel gereinigt, teilweise sogar aus dem Diarahmen entnommen werden müssen, um Verunreinigungen innerhalb der Glasrahmen zu entfernen. Für das Einscannen steht jeweils ein Autofeeder zur Verfügung, mit dem bis zu 50 Dias im Batchbetrieb, also unbeaufsichtigt, eingescannt werden können. Da das Einscannen eines einzelnen Bildes bis zu 60 Sekunden in Anspruch nimmt, ist dies eine nicht zu vernachlässigende Arbeitszeiterparnis. In der Praxis konnte der Autofeeder in Würzburg leider nicht eingesetzt werden, da die Dias hier stark unterschiedliche Diarahmen haben. Das hat zur Folge, daß in der Tat jedes einzelne Dia manuell in den Scanner eingeführt und gescannt werden muß. In Jena dagegen haben alle Dias die gleichen, Autofeeder-geeigneten Rahmen, was das Einscannen der Bilder wesentlich vereinfacht. Die Dias jeweils einer Serie werden von der Scan-Software automatisch in einem Verzeichnis gespeichert. Die Dateinamen erhalten dabei ein einheitliches Präfix und ein fortlaufendes dreistelliges Suffix (001,002,...). Im Prinzip kann eine solche Serie beliebig groß sein, in der Praxis wurden aber immer ca. 30 Bilder in einer Serie gescannt.

Die Qualität der so produzierten digitalen Bilder wird von Ärzten durchweg als sehr gut und auf jeden Fall für den Verwendungszweck in der Lehre ausreichend bewertet.

5.2.3 Erfassen der Grunddaten

Zur Vorbereitung der Verschlagwortung werden im Batch-Betrieb kleine Vorschau-bilder der gescannten, hochauflösten Bilder generiert. Dadurch wird die Ladezeit der Bilder beim eigentlichen Verschlagworten drastisch reduziert. Die Erfassung der Grunddaten erfolgt mit HTML-Formularen und CGI-Skripten in einem Standard-WWW-Browser. Es wird ein Apache-WWW-Server verwendet, der auch die Verwaltung der Benutzerrechte übernimmt. Benutzergruppen sind Administratoren und Erfasser. Alle CGI-Skripte wurden in Perl implementiert und sind konfigurierbar (siehe Anhang B.3).

Eine **Übersicht über alle Scan-Verzeichnisse** ist der Einstiegspunkt für die Grundverschlagwortung (Abbildung 12).

Verzeichnis	Präfix	bearbeitet	Datum/Zeit/von	Index	Bearbeitung
scan1	A5	19/51	2001-01-30/10:26/Arzt1	X	sperren
scan2	D4	0/2		X	sperren
scan3	A1	17/17	2001-05-23/10:27/Arzt3	X	sperren
scan4	A2	4/21	2001-07-10/10:28/Arzt2	X	sperren
scan5	B6	18/18	2001-03-01/10:28/Arzt1	X	sperren
scan6	B12	0/29		X	sperren
scan7	Lichen	0/42		X	freigeben
scan8	Alopezie	0/35		X	freigeben

Abbildung 12: Datenerfassung. Übersicht über alle Scan-Verzeichnisse.

Ein Perl-Skript scannt alle Unterverzeichnisse des Scan-Verzeichnisses und ermittelt das jeweilige gemeinsame Präfix, die Anzahl bereits verschlagworteter Bilder und den Zeitpunkt und Autor der letzten Bearbeitung. Es wird eine Übersichtstabelle über alle Verzeichnisse generiert und Links zum Verschlagworten und zu einer Übersichtsseite einer Serie (Index) angeboten. Defaultmäßig werden neue Verzeichnisse gesperrt. Grund dafür ist, daß beim Scannen der Unterverzeichnisse nicht erkannt werden kann, ob das jeweilige Verzeichnis bereits vollständig ist, d.h. ob das Einscannen bereits abgeschlossen ist. Ein Administrator gibt das Verzeichnis erst frei, wenn das Einscannen der jeweiligen Serie abgeschlossen ist.

Bei der Grundverschlagwortung werden zunächst nur die auf dem Dia direkt vermerkten Daten erfaßt (siehe Abschnitt 6.3.1). Da die Dias nur vertikal orientiert und auch spiegelverkehrt gescannt werden konnten, werden zusätzlich zu den Grunddaten die notwendige Drehung und Spiegelung des Bildes erfaßt (Abbildung 13).

[Vorheriges](#)
(ohne Speichern)
[Nächstes](#)
(ohne Speichern)
[Index der Serie](#)
[Übersicht](#)

Bild: A5034

ICD10

Pat.

Bem.

Spiegeln

Drehen



Abbildung 13: Datenerfassung: Eingabe der Grunddaten für ein Bild

Von der Übersicht über alle Scanverzeichnisse (Abbildung 12) kann durch Anklicken des Index-Links zu einer Übersicht über alle Bilder einer Scan-Serie mit Angabe der bereits erfaßten Daten gewechselt werden (Abbildung 14). Von hier aus können die einzelnen Bildobjekte ausgewählt und editiert werden.

A5001 ICD10: L63.820 Pat.: xxx/99 Bem.:  Editieren ...	A5002 ICD10: L63.900 Pat.: xxx/99 Bem.:  Editieren ...	A5003 ICD10: L63.900 Pat.: xxx/99 Bem.:  Editieren ...	A5004 ICD10: L63.900 Pat.: xxx/99 Bem.: in Remission  Editieren ...
A5005 ICD10: L63.900 Pat.: xxx/99 Bem.: Kadaverhaare  Editieren ...	A5006 ICD10: L60.920;L63.900 Pat.: xxx/99 Bem.: Areata-Nagel  Editieren ...	A5007 ICD10: L63.900;L60.920 Pat.: xxx/99 Bem.: Areata-Nagel  Editieren ...	A5008 ICD10: L63.900 Pat.: xxx/99 Bem.:  Editieren ...
A5009 ICD10: L63.900 Pat.: xxx/99	A5010 ICD10: L63.900 Pat.: xxx/99	A5011 ICD10: L65.810;L20.990 Pat.: xxx/99	A5012 ICD10: L65.810 Pat.: xxx/99

Abbildung 14: Datenerfassung: Übersicht über ein Scan-Verzeichnis.

5.2.4 Sicherung der Daten

Die Bilder werden in der hohen Auflösung unkomprimiert oder verlustfrei komprimiert auf CDs oder Festplatten mit den jeweiligen Grunddaten gesichert. Diese CDs/Platten dienen dann als Master für alle weiteren Verarbeitungsschritte wie Generierung Internet-geeigneter Auflösungen (siehe nächster Abschnitt) und Einfügen von Anonymisierung (Abschnitt 5.2.7). Außerdem ist eine ständige Weiterentwicklung verlustbehafteter Bildkompressionsverfahren zu erwarten. Demnächst wird wahrscheinlich zum Beispiel JPEG von JPEG2000 als Standard abgelöst. Durch das Vorhalten der Masterversionen können die Bilder jederzeit durch Batchkonvertierungen (siehe nächster Abschnitt)

- mit anderen Kompressionsverfahren
- mit anderen Qualitätseinstellungen
- in anderen Auflösungen

gespeichert werden.

5.2.5 Aufbereitung der Bildobjekte

Für die weitere Verwendung werden die Bilder mit Perl-Skripten und geeigneten Tools im Batch-Betrieb aufbereitet. Für die Präsentation im Inter-/Intranet, die Generierung und die Präsentation von Vorlesungen werden sowohl monitorfüllende Versionen als auch kleine, schnell ladbare Vorschaubilder (Thumbnails, ca. 120x120 Pixel) benötigt.

Außerdem werden die Daten der Bildobjekte in ASCII-Tabellen zusammengefaßt. Ein Übersicht der Tabellen und der jeweils gespeicherten Daten findet man in Anhang B.3.

5.2.6 Einfache Suche im Bildarchiv

Sobald die Erfassung der Grunddaten abgeschlossen ist, können die Bilder auf einem Server für die Benutzer suchbar nach den jeweiligen Kriterien angeboten werden (Abbildung 15).



Abbildung 15: CGI-Suche im Bildarchiv

Dazu wird abhängig von der Konfiguration ein HTML-Formular mit den erfaßten Datenfeldern generiert. Die auf die Suchanfrage passenden Bilder werden zusammen mit den erfaßten Daten übersichtlich in einer HTML-Tabelle vergrößerbar dargestellt. Eine Beschreibung der Konfiguration und Skripte ist in Anhang B.3, Seite 148 zu finden.

5.2.7 Eingabe der sonstigen Daten

Zur Eingabe aller weiteren Daten wurde eine Java-Applikation implementiert (Abbildung 16).

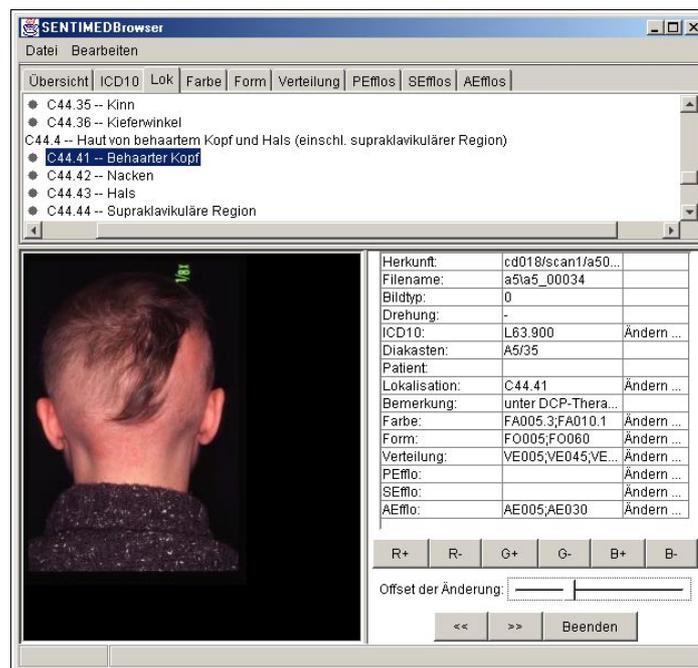


Abbildung 16: Datenerfassung: Java-Applikation zur Eingabe der übrigen Daten.

Sie verwendet als Datenquelle die gleichen Tabellen wie die Skripte zur Erfassung der Grunddaten und erlaubt das Browsen über die bereits eingegebenen Bildobjekte. Im linken unteren Bereich wird das Bild angezeigt, rechts unten findet sich eine Übersicht über alle bisher eingegebenen Daten zu diesem Bild. Über die Karteikarten im oberen Bereich des Fensters können alle relevanten Daten erfasst bzw. editiert werden. Lokalisationen und Diagnosen werden mit Hilfe von hierarchischen Bäumen angezeigt, die anderen Eingabefelder werden als Picklisten dargestellt. Mehrfachauswahl ist bei allen Eingaben möglich. Durch Aufziehen von Rechtecken im Bild ist das Schwärzen von Regionen möglich. Dadurch können auf Bildern identifizierbare Personen zum Beispiel durch Abdeckung der Augenregion anonymisiert werden. Die so markierten Regionen werden nicht fest in das Original eingefügt, sondern als Rechteck-Koordinaten abgelegt. So kann das Einfügen bei der Konvertierung der Bilder für die Verwendung geschehen und auch wieder rückgängig gemacht bzw. verändert werden.

Eine Beschreibung der Konfiguration und der Implementierung ist im Anhang B.4, Seite 153 zu finden.

5.3 Semantisches Netz

Semantisches Netz enthält die beiden Begriffe Semantik und Netz. Unter einem Netz versteht man eine Menge von untereinander in irgendeiner Weise verbundener Knoten. Semantik läßt sich nicht ganz so einfach abgrenzen. Aus dem Griechischen übersetzt ist Semantik die *Lehre von der Bedeutung der Zeichen*, in der Logik versteht man unter Semantik eine *Bezeichnung für die Theorie der Wahrheit logischer Sätze und Folgerungen und entsprechender Begriffe* und in der Sprachwissenschaft *die Bedeutung und den Inhalt sprachlicher Ausdrucksformen sowie deren Beziehungen zu den bezeichneten Gegenständen und Tatsachen*.

Dem entsprechend gibt es auch unterschiedliche Auffassungen darüber, was ein semantisches Netz ist. Allen gemeinsam sind aber folgende Kriterien:

1. Ein semantisches Netz dient der Wissensrepräsentation.
2. Ein semantisches Netz besteht aus Konzepten und Relationen. Konzepte sind unäre, Relationen binäre Prädikate.
3. Aus graphentheoretischer Sicht ist ein semantisches Netz ein gerichteter Graph. Es besteht also aus Knoten und Kanten. Knoten entsprechen den Konzepten, die bewerteten Kanten den Beziehungen zwischen den Konzepten. Es gelten z.B. die Bedingungen, daß jedem Konzept genau ein Knoten entspricht und daß bestimmte Beziehungen („is_a“) keine Zyklen bilden.
4. Die Konzepte (also Knoten) sind in einer Hierarchie gegliedert.
5. Beziehungen werden vererbt.

Daneben dienen semantische Netze im Bereich der künstlichen Intelligenz auch oft der Ableitung neuer Fakten durch Inferenzmechanismen oder werden zur Modellierung von assoziativen, dem menschlichen Denken ähnlichen Speichern verwandt. J. F. Sowa liefert in [18] folgende Definition von semantischen Netzen:

Definition:

Ein **semantisches Netz** ist eine Struktur zur Repräsentation von Wissen als ein Muster von untereinander verbundenen Knoten und Pfaden.

Semantische Netze gibt es bereits seit den 1960er Jahren. Quillian war 1968 mit seinem *Semantic Memory* der erste, der semantische Netze in der künstlichen Intelligenz zur Über-

setzung und zum Verständnis natürlicher Sprachen angewandt hat. Damals beschränkte sich das Anwendungsgebiet semantischer Netze im wesentlichen auf das Gebiet natürlicher Sprachen. Obwohl in den 70er bereits sehr weit entwickelt, wurden semantische Netze in den 80er Jahren oft durch lineare Beschreibungen wie den Programmiersprachen Lisp und Prolog oder durch regelbasierte Expertensysteme ersetzt. Erst als Ende der 80er Jahre die Anforderungen in Größe und Komplexität wuchsen und Netze auf dem Bildschirm visualisiert werden konnten, wurde wieder verstärkt auf semantische Netze zurückgegriffen. Heute dienen sie in regelbasierten Expertensystem oft der Unterstützung bei der Notation der Objekthierarchien. Auch im Bereich der Programmiersprachen werden Typen- und Klassenhierarchien ähnlich wie in semantischen Netzen beschrieben. Schließlich kann auch das Design einer Datenbank durch vereinfachte Darstellungen von semantischen Netzen, zum Beispiel in Entity-Relationship Diagrammen, visualisiert werden. Auch der Mitbegründer des WWW, Tim Berners-Lee, bringt semantische Netze mit seinem Konzept des *Semantic Web* aktuell in die Diskussion [22]. Er versteht unter dem Semantic Web ein intelligentes WWW, in dem die Pages durch maschinenverständliche Metainformationen angereichert werden:

„The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.“ -- Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web [67]

Mit Hilfe von Ontologien könnten dann zum Beispiel Suchmaschinen einfache Aussagen machen oder sogar Schlüsse ziehen.

In der vorgestellten Toolbox wird ein semantisches Netz zur Unterstützung bei der Recherche benutzt. Es soll also nicht neue Fakten generieren, sondern vorhandenes Wissen möglichst strukturiert und gut navigierbar darstellen. Letztendlich ist es die Intention des semantischen Netzes, dem Nutzer, hier also dem Dozenten, möglichst viele relevante Informationen übersichtlich und vor allem in sinnvoller Weise verlinkt zu präsentieren. Es soll dabei unter anderem auch als Ausgangspunkt für weitere Internetrecherchen dienen.

Abschnitt 5.3.1 beschreibt, wie das semantische Netz aus dem Bildarchiv zusammen mit dem *Dermatologischen Diagnosenkatalog* (Kap. 4.1.3), den *Medical Subject Headings* (Abschnitt 4.3), dem *Unified Medical Language System* (Kap 4.4) und dem *ICD-9-CM* generiert wird. Angelehnt an die Definition wird das semantische Netz durch die (dynamische

sche) Generierung verlinkter Hypertext-Seiten umgesetzt, d.h. die Knoten sind HTML-Seiten, Relationen sind Links. Die einzelnen Seiten sind einheitlich strukturiert. Die Links enthalten als „Semantik“ jeweils die Datenquelle, aus der der Link abgeleitet wurde, und das Verhältnis (Up, Down, Sibling) des Zielknotens zum aktuellen Knoten (siehe Abschnitt 5.3.1). Das semantische Netz kann zentral auf einem (Intranet-) Server abgelegt werden und die Nutzer benötigen keine zusätzliche Software außer dem ihnen bereits vertrauten Standard-WWW-Browser. Als Einstiegspunkte werden hierarchische und alphabetische Listen von Diagnosen oder Lokalisationen angeboten (Abschnitt 5.3.2). Zum Betrachten des hochqualitativen Bildmaterials dient ein Java-Applet (Abschnitt 5.3.3). Zum Wiederfinden und Einbinden von im semantischen Netz gefundenen Objekten in Vorlesungen dient eine interne Zwischenablage (Abschnitt 5.3.4). Außerdem wird dem Dozenten durch die Verlinkung mit externen Datenquellen die Möglichkeit zur weiteren Recherche im Internet gegeben (Abschnitt 5.3.5).

5.3.1 Generierung

Als Quellen für den Aufbau des semantischen Netzes dienen das Bildobjekt-Archiv (Abschnitt 5.2), die ICD-9-CM Klassifikation (4.1.1), der Dermatologische Diagnosenkatalog DDK (4.1.3), die Medical Subject Headings MeSH (4.3) und das Unified Medical Language System UMLS (4.4). Um die aus diesen Quellen extrahierbaren Informationen sinnvoll präsentieren zu können, ist es nötig, zusammengehörende Daten aus verschiedenen Quellen zu Knoten zusammenzufassen. Diese Knoten haben untereinander dann bewertete Beziehungen, die ihrerseits wieder aus den Quellen extrahiert werden.

Grundlage für das endgültige, webfähige semantische Netz ist eine aus den Quellen generierte XML-Datenbank, in der die Relationen zwischen Einträgen als XML-Links umgesetzt sind (Abbildung 17). Zum einen erzeugen Perl-Skripte aus der XML-Datenbank statische HTML-Seiten, zum anderen aber auch dynamische HTML-Seiten direkt bei Anfrage durch den Benutzer.

Zur Erstellung der Verknüpfungen der einzelnen Datenquellen zu Knoten in der XML-Datenbank wird nach folgenden Strategien vorgegangen:

- DDK und UMLS: Zuerst wird ein direkter Match zwischen der (deutschen) UMLS-Konzeptbezeichnung und der DDK-Bezeichnung versucht. Liefert das keinen Treffer, werden Heuristiken angewandt. Vom UMLS wird nur der UMLS Metathesaurus einge-

bracht. Das UMLS Semantic Network eignet sich nicht, da es aufgrund der Einschränkung auf das Fachgebiet Dermatologie keine sinnvollen Knoten und Relationen mehr bietet (siehe Abschnitt 4.4.3).

- DDK und MeSH: Es werden Oberbegriffe, speziellere Begriffe und exakte Treffer durch Stringvergleiche ermittelt.
- DDK und ICD9-CM: der DDK stellt nur eine Erweiterung der ICD-9-CM dar, d.h. das Matching kann durch einfaches Abschneiden der letzten beiden Ziffern des DDK-Codes durchgeführt werden.
- ICD0 / UMLS / MeSH / ICD9CM: die Zuordnung erfolgt durch paarweisen Vergleich der jeweiligen Konzeptbezeichnungen.

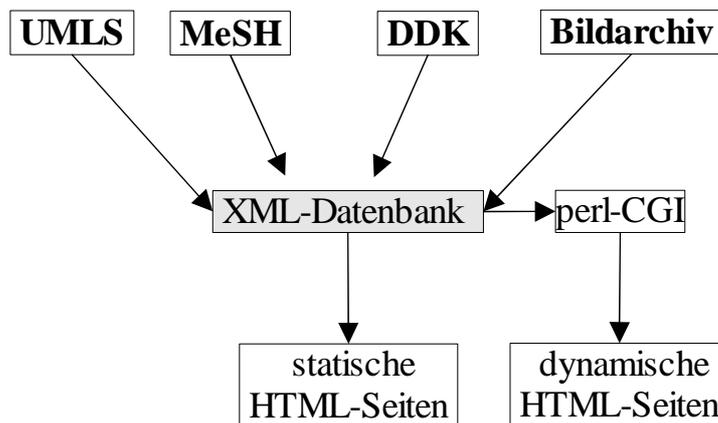


Abbildung 17: Generierung des semantischen Netzes.

Insgesamt ergibt sich damit der in Abbildung 18 dargestellte Aufbau eines Knotens des semantischen Netzes.

Durch die Einbindung der verschiedenen Datenquellen kommt es leider auch zu Inkonsistenzen, wenn in verschiedenen Datenquellen sich widersprechende Relationen eingebunden werden.

Für detailliertere Informationen zur Generierung des semantischen Netzes siehe Anhang B.5, Seite 154.

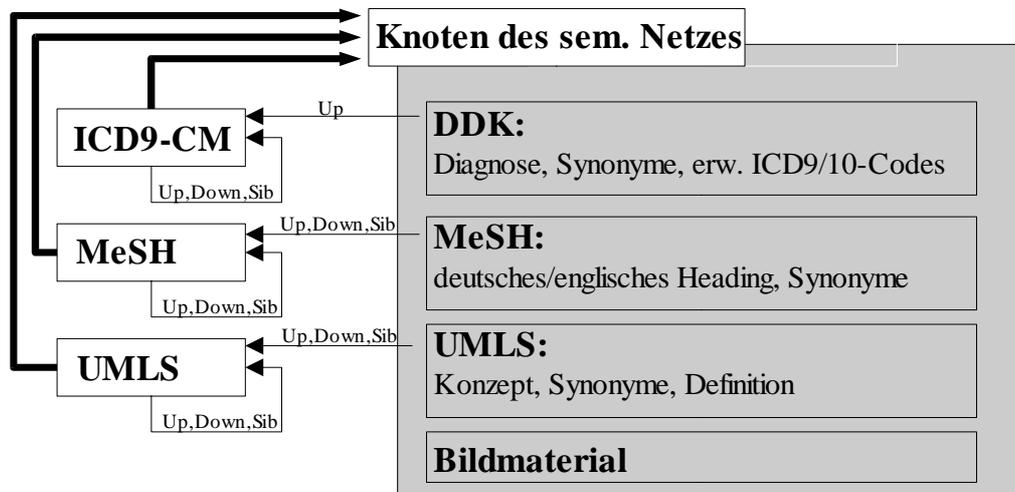


Abbildung 18: Aufbau der Knoten im semantischen Netz

5.3.2 Navigation

Zur Navigation in den HTML-Seiten steht ein Frameset zur Verfügung (Abbildung 19). Die Links im linken Steuerframe dienen dabei als Einstiegspunkte in das semantische Netz:

Diagnosen alphabetisch: Der Benutzer bekommt als Einstieg eine alphabetische Liste aller Diagnosen angezeigt, zu denen Bildmaterial im Archiv vorhanden ist.

Diagnosen hierarchisch: Der Benutzer steigt in der höchsten Hierarchie-Ebene in das semantische Netz ein und kann sich dann über zwei Ebenen zum Bildmaterial der speziellen Diagnosen durchklicken.

Lokalisationshierarchie: Der Benutzer kann sich in der Körperregionen-Hierarchie von der höchsten Ebene bis zum Bildmaterial durchklicken.

Diakastenverzeichnis: Um den Dozenten der Hautklinik Würzburg den für sie bisher üblichen Weg der Suche nach Dias nachzubilden, wird eine Liste aller Diakasten-Schlüssel als Einstieg generiert.

Über diese Einstiegspunkte gelangt der Nutzer schließlich zu den Knoten des semantischen Netzes (in Abbildung 19 zum Beispiel *Alopecia areata o.n.A.*). Dort werden die Informationen aus den verschiedenen Datenquellen mit den weiterführenden Links angezeigt.

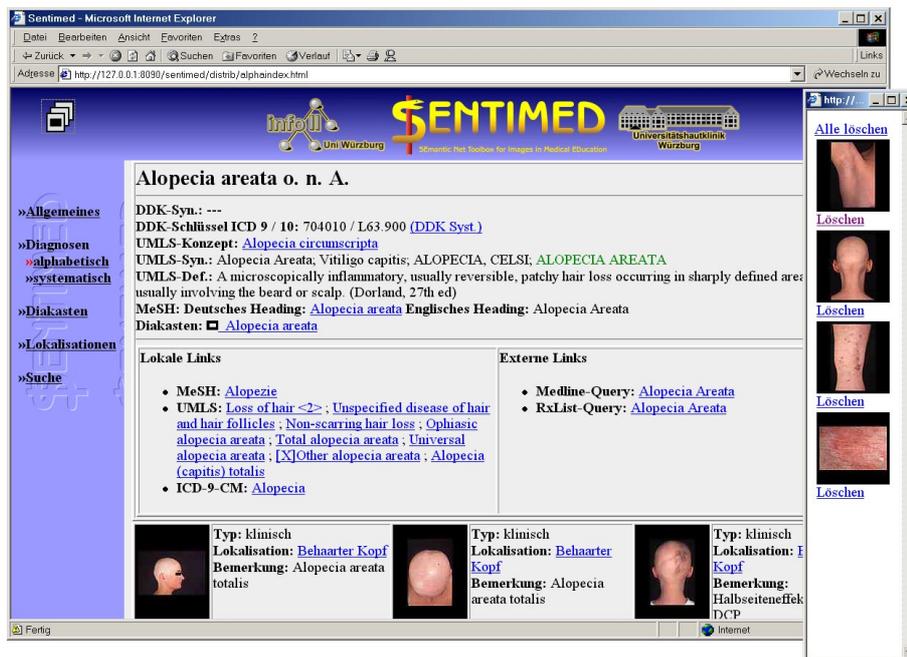


Abbildung 19: Semantisches Netz im Browser mit interner Zwischenablage

5.3.3 Präsentation des Bildmaterials

Zur Präsentation des hochqualitativen Bildmaterials dient ein Java-Applet. Es zeigt neben dem Bild auch alle verfügbaren Daten eines Bildobjekts an (Abbildung 20, links). Wurden zum Bild Diagnosen angegeben, führen Links vom Applet zurück auf den entsprechenden Knoten im semantischen Netz. Außerdem wurde jeweils eine client- und eine serverseitige Methode des Zoomings implementiert:

Bei der *clientseitigen Vergrößerung* (Abbildung 20, Mitte) werden die Vergrößerungen ohne Nachladen von Daten allein aus den Pixeldaten des Originalbildes berechnet. Zur Berechnung von Zooms mit doppelter Kantenlänge wird als Grundlage eine Arbeit von Said und Pearlman verwendet. In [15] wird dort (mit dem Ziel der Bildkompression) folgendes Vorgehen bei der Prädiktion von Pixelwerten vorgeschlagen:

Ausgangspunkt ist die S-Transformation: eine Folge von Integer-Zahlen $c[n]$, $n=0, \dots, N-1$, kann durch zwei Folgen (jeweils der Mittelwert und die Differenz zweier benachbarter Folgeelemente) repräsentiert werden:

$$lo[n] = \lfloor (c[2n] + c[2n+1]) / 2 \rfloor, n=0, \dots, N/2-1 \quad \text{und}$$

$$hi[n] = c[2n] - c[2n+1], n=0, \dots, N/2-1$$

Said/Pearlman geben nun einen Algorithmus zur Prädiktion der Werte von $h[n]$, d.h. Werte, die von $h[n]$ mit relativ hoher Wahrscheinlichkeit angenommen werden:

$$(1) \hat{hi}[n] = \sum_{i=-1}^1 a_i (lo[n-1+i] - lo[n+i]) - b_1 hi[n+1]$$

Für die Koeffizienten werden verschiedene Möglichkeiten vorgeschlagen, unter anderem die folgende:

$$(2) a_{-1}=0; a_0=2/8; a_1=3/8; b_1=2/8$$

Die inverse S-Transformation wird dann für $n = 0, \dots, N/2-1$ berechnet durch

$$c[2n] = lo[n] + \lfloor (hi[n] + 1) / 2 \rfloor \quad \text{und} \\ c[2n+1] = c[2n] - hi[n]$$

d.h. man berechnet aus zwei Folgen der Länge $N/2$ eine Folge der Länge N . Dies kann man dazu nutzen, Vergrößerungen zu berechnen. Als Ausgangsfolge l werden erst spalten-, dann zeilenweise die Originalpixelwerte des Bildes benutzt, und die Folge h mit Hilfe von (1) prädiziert. Allerdings hat man nun die Werte für $h[n+1]$ nicht zur Verfügung. Deshalb wird von $h[n+1] = 0$ ausgegangen (ein wahrscheinlicher Wert, da $h[n+1]$ die Differenz benachbarter Pixelwerte ist). Die Prädiktion (1) mit den Koeffizienten aus (2) ergibt dann folgende Formel:

$$\hat{hi}[n] = \frac{2lo[n-1] + lo[n] - 3lo[n+1]}{8}$$

oder, nach Umformen:

$$\hat{hi}[n] = \frac{2(lo[n-1] - lo[n+1]) + lo[n] - lo[n+1]}{8}$$

Damit kann nun der Prädiktionwert performant allein durch Summe, Differenz und Shift-Operatoren in Java berechnet werden (hier am Beispiel für den Rot-Anteil inklusive Extraktion des Byte-Wertes aus dem gesamten Pixelwert):

```
hi[n] = (((((pixels[n-1] & 0x00ff0000)>>16)
- ((pixels[n+1] & 0x00ff0000)>>16))<<1)
+ ((pixels[n] & 0x00ff0000)>>16)
- ((pixels[n+1] & 0x00ff0000)>>16)) >>3;
```

Außerdem kann diese Berechnung speichersparend direkt auf dem Array, in dem die Pixel-daten der Vergrößerung abgelegt werden sollen, erfolgen. Dazu müssen die Originalpixel-werte lediglich in das letzte Viertel des Arrays gelegt werden. Die berechneten Pixel wer-den dann von vorne in den gleichen Array gespeichert.

Bei der *serverseitigen Vergrößerung* (Abbildung 20, rechts) werden die Koordinaten des vom Nutzer durch Aufziehen bestimmten Rechtecks an den Server übermittelt, auf dem das Bild in hoher Auflösung bereitliegt. Durch ein CGI-Skript wird aus diesem der gewünschte Bildbereich ausgeschnitten und an das Applet geschickt.

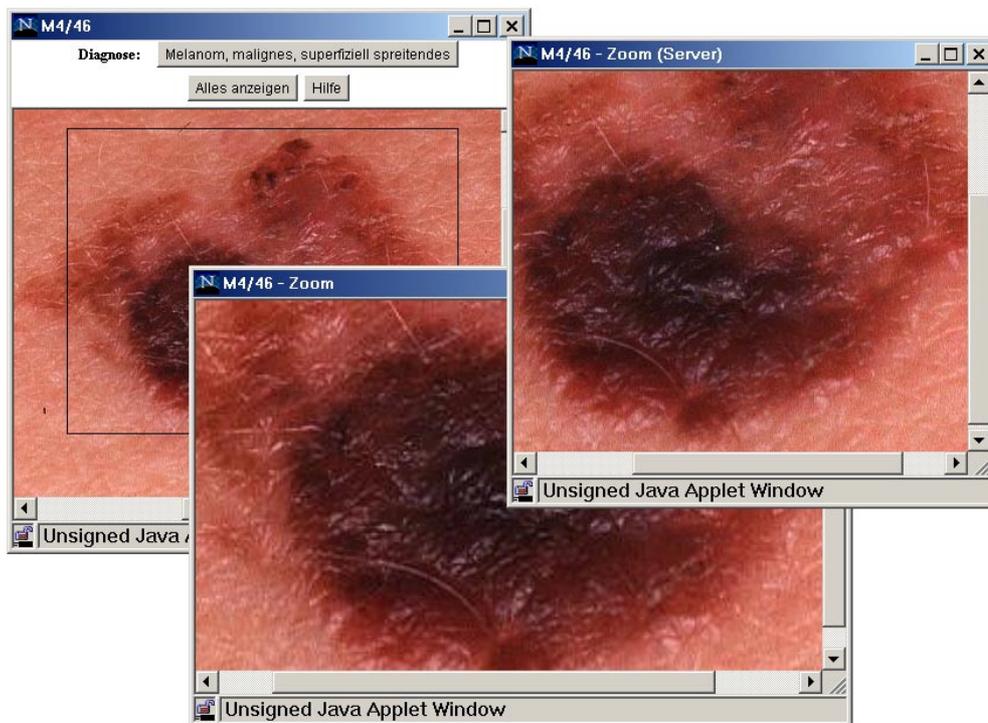


Abbildung 20: Applet zur Bildbetrachtung.

Eine exaktere Beschreibung der Konfiguration und der Implementierung ist im Anhang B.5 Seite 154, zu finden.

5.3.4 Interne Zwischenablage von Bildobjekten

Bildobjekte, auf die der Nutzer während des Browsens im semantischen Netz trifft, können in einer internen Zwischenablage gespeichert werden. Hierfür wird für den Nutzer auf dem Server eine Datei angelegt, in dem die Verweise zu den Bildern abgelegt werden. Im Browser werden die Inhalte in einer Thumbnail-Übersicht dargestellt (Abbildung 19, rechts). Aus der internen Zwischenablage können die Objekte dann komfortabel in Vorlesungen eingefügt werden (siehe Abschnitt 5.4).

5.3.5 Verlinkung mit externen WWW-Datenquellen

Schnittstellen vom semantischen Netz zu weiterer Recherche im WWW stellen Links zu externen Datenquellen dar. *PubMed* [55] ist ein Service der *National Library of Medicine*

(NLM) [58]. Hinter *PubMed* steckt eine große Literatur-Datenbank von medizinischen Artikeln der ganzen Welt. Suchkriterien für die Recherche sind zum Beispiel MeSH-Terms, Journals, in denen der Artikel erschienen ist, Jahr der Erscheinung oder der Autor. Als Ergebnis wird eine Übersicht passender Artikel präsentiert, zu denen oft auch kurze Zusammenfassungen (Abstracts), aber nicht der Volltext angezeigt werden kann.

Nachdem bereits zur Generierung der Knoten des semantischen Netzes ein Matching zu MeSH-Bezeichnungen durchgeführt wird, kann auch eine einfache Anbindung an *PubMed* erfolgen. Um dem Dozenten dabei möglichst die Formulierung der Suchanfrage zu erleichtern, wurden die Knoten um Links zu *PubMed* erweitert. Es wird ein Applet gestartet, das das Editieren einer vorformulierten Anfrage erlaubt (Abbildung 21).

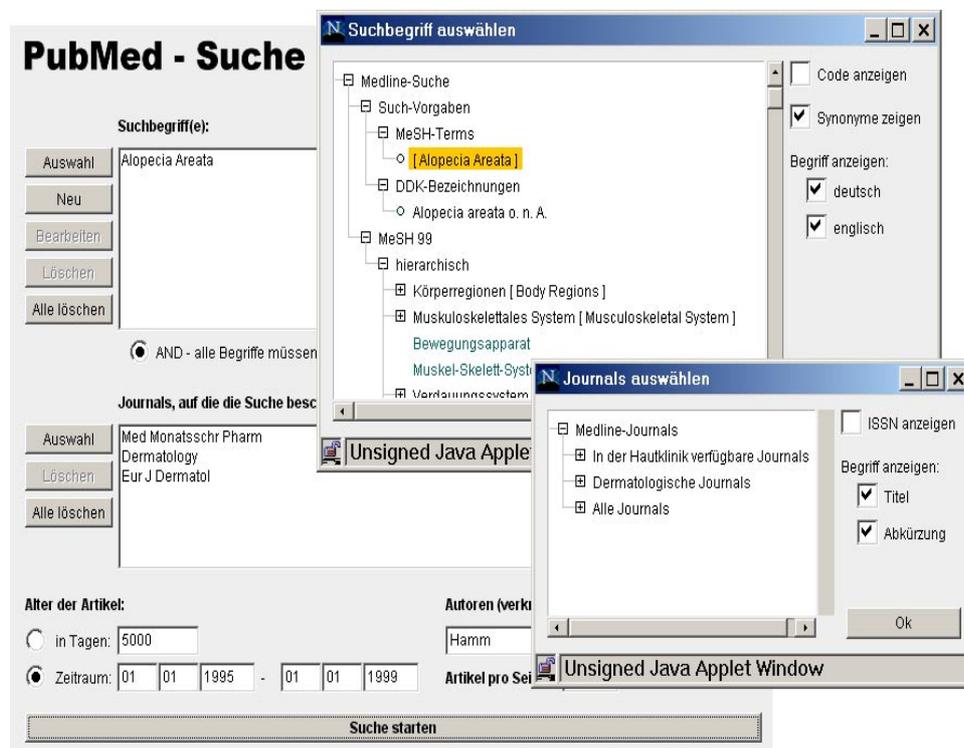


Abbildung 21: Applet zum Editieren einer PubMed-Suchanfrage.

Dazu können die Begriffe in einer Baumansicht der MeSH ausgewählt werden (Abbildung 21, rechts oben). Journals können zur besseren Auswahl auf dem Server in verschiedene Kategorien aufgeteilt werden (Abbildung 21, rechts unten). So kann nach entsprechender Definition eine Suchanfrage bei PubMed zum Beispiel nach Artikeln in allen Journals gestellt werden, die in der lokalen Bibliothek vorhanden sind.

Die eingegebenen Daten werden über eine von der NLM definierte CGI-Schnittstelle direkt an PubMed weitergeleitet und das Suchergebnis zu weiteren Nachforschungen in einem neuem Browser-Fenster dargestellt (Abbildung 22). Genauere Angaben zur Implementierung des Applets sind in Anhang B.5, Seite 154 zu finden.

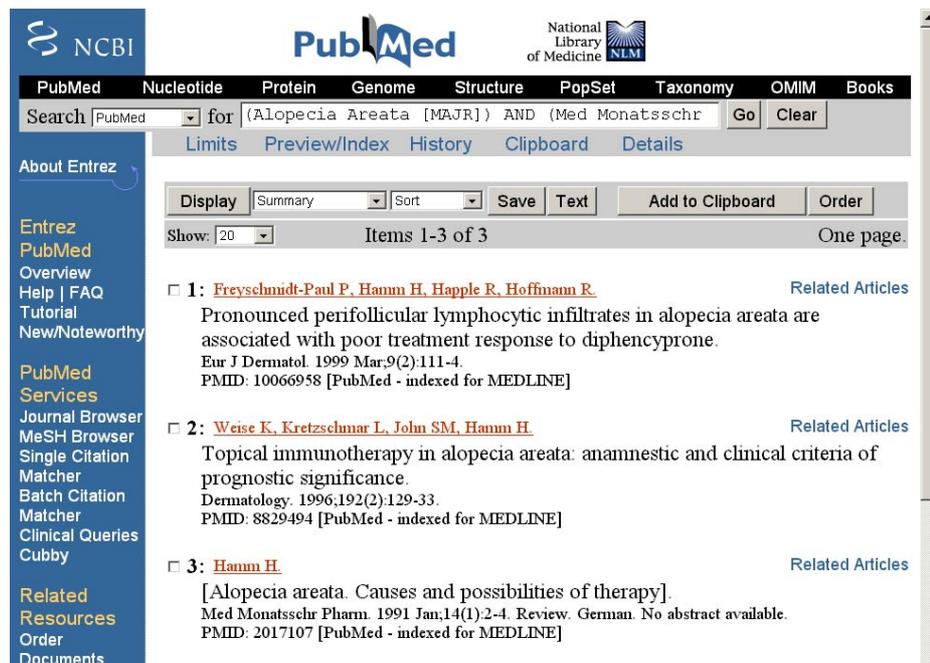


Abbildung 22: Ergebnis der PubMed-Suchanfrage aus Abbildung 21

5.4 Vorlesungserstellung

Zur Erstellung von Vorlesungen wurde eine Java-Applikation erstellt, die es dem Autor ermöglicht,

- in einfacher und komfortabler Weise aus dem großen Fundus vorhandener Bilder die gewünschten Bilder auszuwählen,
- erläuternde Texte, zum Beispiel in Form von Powerpoint-Folien, einzufügen,
- Vergrößerungen von Bildern bzw. Bildausschnitten einzubinden,
- Recherchen im semantischen Netz mit Bildobjekten als Einstiegspunkt zu starten,
- Bildobjekte aus der Zwischenablage des semantischen Netzes einzufügen,
- eigene digitale Bilder zu integrieren,
- bereits vorhandene Vorlesungen wieder zu verwenden,
- die aktuelle Version der Vorlesung in einer Vorschau zu betrachten und

- die (fertige) Vorlesung auf einem geeigneten Datenträger wie einer CD-ROM in kompakter Form abzuspeichern, um sie an anderen Orten vorzuführen.

Die Oberfläche der Applikation besteht im wesentlichen aus vier Karteikarten, nämlich der *Suche*, der *Vor-*, *Zwischen-* und *Präsentationsauswahl*. Die einzelnen Auswahlen dienen jeweils der Darstellung von Sammlungen von Vorschau Bildern (Abbildung 23). Alle Bilder sind durch Anklicken vergrößerbar. Die *Vorauswahl* hat den Zweck, Treffer der Suche anzuzeigen. Befinden sich bereits Bilder in der Vorauswahl, werden sie ohne Nachfrage durch die Trefferanzeige ersetzt. Die *Zwischenauswahl* ist als Sammlung von Kandidaten für die endgültige Auswahl zu betrachten. Durch die *Zwischenauswahl* wird dem Nutzer auf einfache Weise die Möglichkeit gegeben, Bilder miteinander zu vergleichen, ohne sie vorher in die endgültige Auswahl einfügen zu müssen. Die Sammlung der Bilder bleibt auch über mehrere Suchanfragen hinweg erhalten. Die *Präsentationsauswahl* enthält die aktuell für die Vorlesung ausgewählten Bildobjekte. In ihr können Bilder oder Gruppen von Bildern selektiert und per Drag'n'Drop auf einfache Weise umsortiert werden.

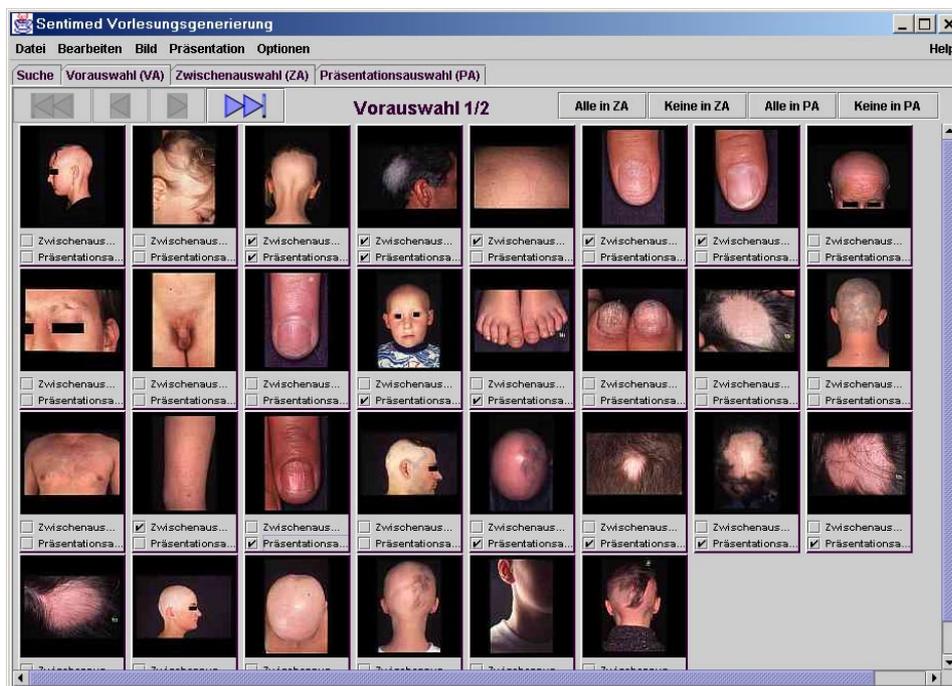


Abbildung 23: Vorlesungsgenerierung. Vorauswahl.

In den folgenden Abschnitten werden nun die einzelne Funktionen der Applikation beschrieben.

5.4.1 Suche im Bildarchiv und Auswahl der Bilder

Für die Recherche im Bildarchiv wird dem Nutzer die Möglichkeit gegeben, nach Kombinationen von Bildtypen, Diakästen, Lokalisationen und Diagnosen zu suchen (Abbildung 24).

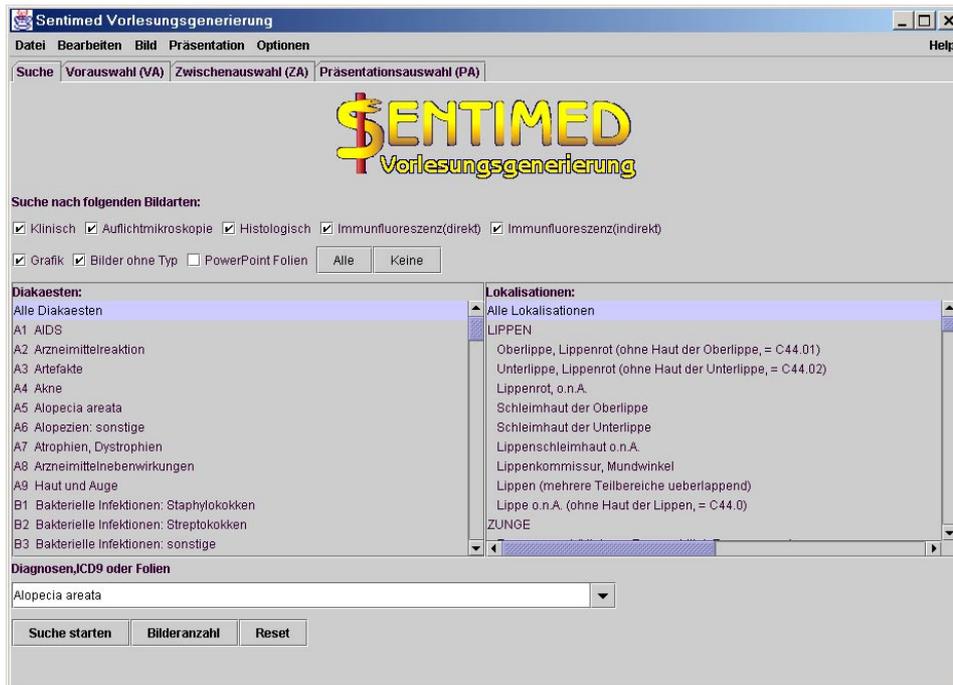


Abbildung 24: Vorlesungserstellung. Suchdialog mit Picklisten.

Die Suche nach Diakästen ist dabei als Tribut an die Nutzer der Hautklinik Würzburg zu sehen, die so möglichst nahe an ihrer bisherigen Suche im Dia-Archiv arbeiten konnten. Zur Erleichterung der Sucheingabe werden Picklisten angeboten, in denen auch Mehrfach-Selektion möglich ist. Dabei werden die Verknüpfungen so behandelt, wie der Nutzer sie auch intuitiv verstehen würde: Innerhalb einer Kategorie (Diagnose, Bildtyp, ...) werden die ausgewählten Elemente mit *oder* verknüpft, zwischen den Kategorien mit *und*.

Beispiel: bei Auswahl von:

Bildtyp: klinisch, histologisch; Lokalisation: Oberlippe, Unterlippe; Diagnose: Basalzellkarzinom.

werden alle Bildobjekte gefunden, für die gilt

(Bildtyp=klinisch oder Bildtyp = histologisch) und (Lokalisation=Oberlippe oder Lokalisation=Unterlippe) und (Diagnose=Basalzellkarzinom).

Die zur Suchanfrage gefundenen Bilder werden in der Vorauswahl angezeigt. Von hier können einzelne oder alle Bildobjekte in die Zwischenauswahl übernommen werden. Ist sich der Autor allerdings schon sicher, daß ein Objekt in die Präsentation eingefügt werden soll, kann er es auch direkt in die Präsentationsauswahl einfügen.

5.4.2 Import von Powerpoint-Präsentationen

Da das Tool selbst keine Möglichkeit zur Erstellung von Textfolien bietet, können einfache Powerpoint-Präsentationen importiert werden. Allerdings muß dazu aus Powerpoint ein GIF/JPEG-Export durchgeführt werden. Dieser Export kann nun seinerseits in eine Vorlesung importiert werden. Durch diesen Umweg gehen aber alle eventuell in Powerpoint generierten Animationen verloren. Dieser Umstand war auch ein Hauptgrund, warum für die in Abschnitt 5.5 beschriebene Lehr-/Lernumgebung zur Vorlesungserstellung zusätzlich ein Powerpoint-AddIn (Abschnitt 6.5) implementiert wurde, das im Gegensatz zur obigen Vorgehensweise Bildobjekte in Powerpoint-Präsentationen integriert.

5.4.3 Vergrößerungen von Bildern bzw. Bildausschnitten

Da die Bilder digital in hoher Auflösung vorliegen, besteht die Möglichkeit, qualitativ hochwertige Vergrößerungen zu generieren. Dies ist im Bereich der Dermatologie besonders wichtig, da oft erst bei hohen Vergrößerungen kleine Unterschiede zwischen Hauterkrankungen zu erkennen sind. Durch Anklicken eines Vorschaubildes in einer der Auswahlen wird das Bild monitorfüllend in der Größe, in der es auch später in der Vorlesung verwendet wird, angezeigt. In dieser Darstellung können beliebige, rechteckige Bereiche markiert werden. Die Vorlesungserstellung stellt dann eine Verbindung zum WWW-Server her und testet mit einem CGI-Skript, ob zum vorliegenden Bild auf dem Server ein Bild höherer Auflösung vorhanden ist. Ist das der Fall, können durch ein weiteres CGI-Skript die entsprechenden Bildausschnitte auf dem Server in hoher Auflösung ausgeschnitten und an das Vorlesungserstellungstool übertragen werden. Diese Vergrößerung wird dem Nutzer in einem eigenen Fenster angezeigt (Abbildung 25).

5.4.4 Import von eigenen Bildern

Es kommt recht häufig vor, daß Vorlesungen durch weitere, nicht im Bildarchiv gespeicherte Bilder ergänzt werden sollen. In der Vorlesungserstellung können JPEG- und

GIF-Bilder von der lokalen Platte des Benutzers importiert und somit automatisch in die Zwischenauswahl eingefügt werden.

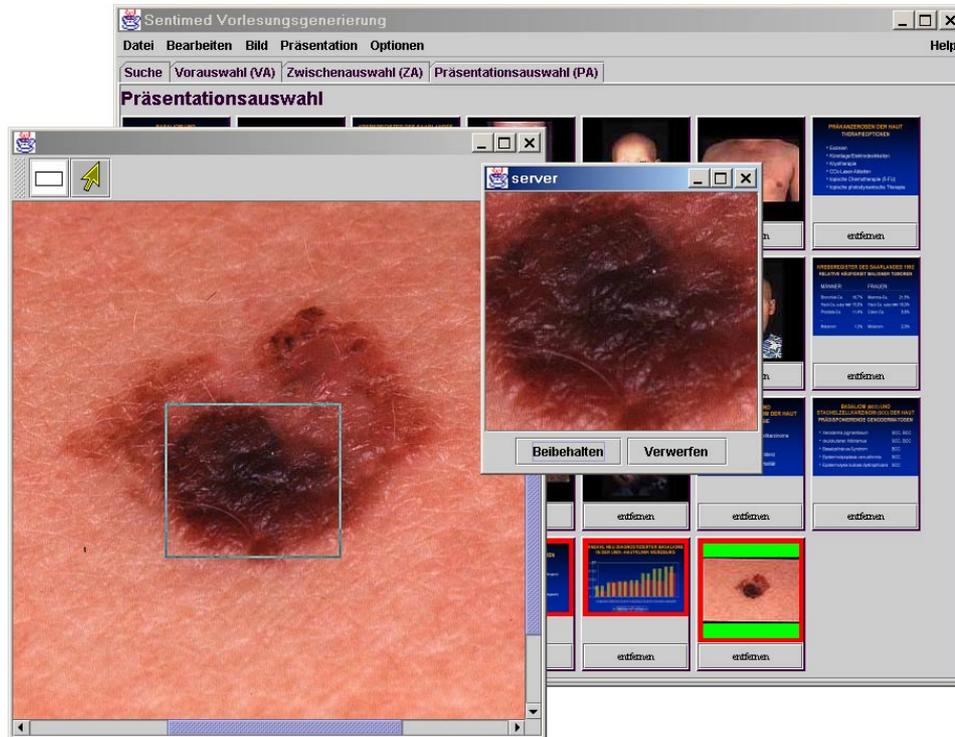


Abbildung 25: Vorlesungserstellung.

5.4.5 Import aus dem semantischen Netz

Während der Recherche im semantischen Netz kann der Nutzer beliebige Bildobjekte für das Einfügen in die Vorlesungserstellung vormerken. Diese vorgemerkten Bildobjekte werden auf Wunsch in die Zwischenauswahl eingefügt.

5.4.6 Recherche im semantischen Netz

Jedes Bild in einer der Auswahlen kann als Einstiegspunkt für eine Recherche im semantischen Netz dienen. Mit Hilfe eines Kontextmenues kann man den Diagnosenknoten im semantischen Netz, dem das Bild zugeordnet ist, aufrufen und von dort weiter navigieren.

5.4.7 Wiederverwendung von Vorlesungen

Zu jeder Zeit können Vorlesungen gespeichert werden. Dabei werden die aktuelle Zwischen- und Präsentationsauswahl gespeichert. Um Speicherplatz zu sparen, werden zu Bildern, die aus dem Bildarchiv kommen, lediglich Verweise abgelegt. Alle anderen Bilder, aus Powerpoint importierte oder eigene Bilder, werden in der erzeugten Datei mit

abgespeichert. Die so generierten Dateien können dann als Vorlage für zukünftige Vorlesungen verwendet oder in andere Vorlesungen integriert werden.

5.4.8 Vorschau von Vorlesungen

Für den aktuellen Stand der generierten Vorlesungen kann jederzeit eine Vorschau gestartet werden. Sie bietet die gleiche Funktionalität wie die Vorlesungspräsentation (siehe Abschnitt 5.5). Der einzige Unterschied besteht darin, daß die Vorschau nicht im Vollbildmodus sondern in einem Fenster abläuft.

5.4.9 Export von Vorlesungen

Zur Vorführung einer Vorlesung kann diese in ein Verzeichnis exportiert werden. Dabei werden nicht nur, wie beim Abspeichern, Verweise auf Bilder aus dem Archiv abgelegt, sondern die Bilder selbst werden zusammen mit allen für die Präsentation notwendigen Daten in das Verzeichnis kopiert. Auf diese Weise kann durch Kopieren des Inhalts dieses Verzeichnisses (zum Beispiel auf CD-ROM) eine Vorlesung an jedem beliebigen Rechner vorgeführt werden, auf dem das Tool zur Vorlesungspräsentation (siehe Abschnitt 5.5), auch ohne Zugriff auf das Bildarchiv, installiert ist.

Eine exaktere Beschreibung der Konfiguration und der Implementierung ist im Anhang B.6, Seite 158 zu finden.

5.5 Vorlesungspräsentation

Zur Präsentation der mit dem in Abschnitt 5.4 vorgestellten Tool erzeugten Vorlesungen gibt es eine weitere Java-Applikation, die im Vollbildmodus arbeitet. Beim Start wird zunächst eine Übersicht mit Vorschaubildern zu allen Bildern geladen (Abbildung 26).

Hier werden Bilder, für die Vergrößerungen vorbereitet wurden, mit einer blauen Umrahmung markiert (im Beispiel: Zeile 4, Spalte 3). Ein beliebiges Bild kann durch Anklicken als Vollbild dargestellt werden. Von dort aus kann durch einfache Maus- oder Tastaturbefehle vorwärts und rückwärts zum jeweils nächsten Bild gesprungen werden. Sind zu einem Bild Vergrößerungen vorhanden, wird oben links eine Lupe eingeblendet. Durch Tastendruck ('z' für Zoom) werden die Bildbereiche, zu denen die Vergrößerungen gehören, sicht- und anklickbar gemacht. Zu jedem Zeitpunkt kann auch zur Übersichtsseite zurückgekehrt werden. Dies ist insbesondere dann interessant, wenn Zwischenfragen aus dem

Auditorium kommen, oder wenn der Dozent noch einmal einen Vergleich zu einem früheren Bild ziehen möchte.

Eine exaktere Beschreibung der Konfiguration und der Implementierung ist im Anhang B.7, Seite 159 zu finden.

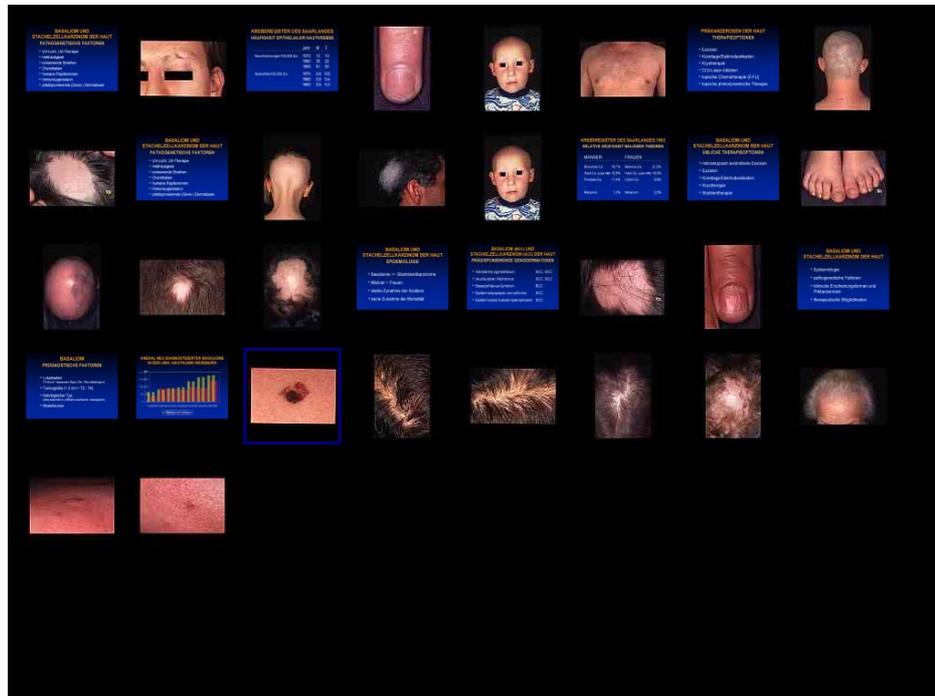


Abbildung 26: Vorlesungspräsentation: Übersicht

6 Datenbankgestützte Lehr- und Lernumgebung

Kapitel 2 und 3 haben die Begriffe Lernobjekt bzw. Lehr-/Lernplattformen definiert und aktuelle Entwicklungen beschrieben. Im Rahmen von SENTIMED [50] (Kapitel 5) wurden auf der Dozenten/Autorensseite wichtige Vorarbeiten zur Erstellung einer Lehr-/Lernplattform geleistet. Insbesondere das Bildarchiv stellt eine gute Grundlage für den Aufbau einer datenbankgestützten Umgebung dar. Auch plattformunabhängige effiziente Tools zum schnellen Retrieval in großen Bildarchiven wurden generiert. Allerdings wird beim Einsatz immer wieder festgestellt, daß sich die Nutzer mit der Einarbeitung in neue Tools (z.B. Vorlesungserstellung in Abschnitt 5.4) schwer tun. Auch das Einbringen von Textfolien über einen Powerpoint-Export ist ein unerwünschter Umweg: die Dozenten müssen also zur Erstellung einer Vorlesung zwei verschiedene Autorenwerkzeuge benutzen. Bisher existiert auch noch keinerlei Unterstützung für die Seite des Lernenden, also keine Lernumgebung.

Unter anderem setzt die entwickelte Lehr- und Lernumgebung folgende Aufgaben um:

- Die Erstellung von Vorlesungen wird für den Dozenten noch weiter erleichtert, zusätzlich können neben Bildern auch weitere Lernobjekte wie Videos in Vorlesungen integriert werden.

- Das textbasierte Bildarchiv wird Grundlage einer echten Lernobjekt-Datenbank, in der auch **andere Lernobjekt-Typen** verwaltet werden können. Dazu sind natürlich auch Werkzeuge nötig, mit denen die Lernobjekte erstellt und in die Datenbank eingebracht werden können.
- Erstellte Lernobjekte können auch zwischen verschiedenen Institutionen ausgetauscht werden.
- Es wird eine Lernumgebung geschaffen, mit der der Lernende dann auch die vorhandenen Lernobjekte der Datenbank in sinnvoller Weise nutzen kann.

Ergebnis ist eine umfassende, multimediale und datenbankgestützte Lehr- und Lernumgebung. Umgesetzt werden die Entwicklungen in dem vom Bundesministerium für Bildung und Forschung geförderten Projekt *DEJAVU (Dermatological Education as Joint Accomplishment of Virtual Universities)* [48], in dessen Beschreibung man zum Beispiel folgende Passage findet:

„Das Projekt integriert universitätsübergreifend und netzgestützt die Aufbereitung multimedialer Vorlesungen für Dozent(inn)en der Dermatologie mit der Präsentation in Präsenz-Vorlesungen und Praktika bis hin zur Generierung interaktiver Repetitorien und Prüfungselemente aus den Lehrveranstaltungen inkl. deren didaktischer Evaluierung“.

Projektbeteiligte sind neben dem Lehrstuhl für Informatik II der Universität Würzburg auch die Hautkliniken von Würzburg, Jena und Mannheim sowie das Institut für Medizinische Psychologie der Universität Jena.

Abschnitt 6.1 diskutiert die Voraussetzungen für den Einsatz einer Lehr-/Lernplattform in der Dermatologie auf der Seite der Autoren (in der Klinik) und die technische Ausstattung (PC, Internetzugang,..) der Studenten. Abschnitt 6.2 faßt das Konzept der entwickelten Plattform zusammen und stellt das Zusammenspiel der einzelnen Bausteine Lernobjekt-Datenbank, Autorenwerkzeuge und Lernumgebung kurz vor. In Abschnitt 6.3 werden die verschiedenen betrachteten Lernobjekt-Typen mit ihren Eigenschaften gelistet, die in der in Abschnitt 6.4 beschriebenen Lernobjekt-Datenbank verwaltet werden. Zur Erstellung der Lernobjekte wurden verschiedene Autorenwerkzeuge entwickelt, die Abschnitt 6.5 vorstellt. Die Präsentation der Lernobjekte, also die Lernumgebung für den Studenten behandelt Abschnitt 6.6. Schließlich wird in Abschnitt 6.7 dargelegt, wie Lernobjekte bzw. Lerninhaltspakete konform zu dem in Kapitel 2 vorgestellten LOM-Standard bzw. IMS Content Packaging Information Model exportiert werden können.

6.1 Technische Voraussetzungen und Akzeptanz

Die grundsätzlichen Voraussetzungen für den Einsatz von Datenbanken sind durch die vorhandenen Vernetzungen in den (beteiligten) Institutionen gegeben. Auch Arbeitsplatzrechner sind in ausreichender Anzahl und Ausstattung vorhanden. Auf der Seite der Dozenten sollen Berührungspunkte durch möglichst einfache und vertraute Oberflächen bei maximalem Nutzen vermindert werden. Maximaler Nutzen bedeutet dabei zum Beispiel Zeitersparnis bei der Vorbereitung oder höherer Lehrwert der generierten Lernobjekte. Auf studentischer Seite wurden bereits nach Abschluß der Vorarbeiten (siehe Kapitel 5) und zu Beginn des DEJAVU-Projekts Umfragen unter den Studenten in Würzburg, Jena und Mannheim durchgeführt, um die Akzeptanz von multimedialen Präsenzvorlesungen sowie die private technische Ausstattung im Hinblick auf die Nutzung einer digitalen Lernumgebung festzustellen. In Würzburg hatten zu diesem Zeitpunkt bereits etwa die Hälfte der Befragten multimediale Vorlesungen im Rahmen des SENTIMED-Projekts gehört und konnten deshalb nach ihren Erfahrungen befragt werden. Die einzelnen Fragen sowie ihre Auswertung sind in Anhang C.1 abgedruckt. Zusammenfassend kann man festhalten:

- ein Großteil der Studierenden verfügt über Fähigkeiten in Textverarbeitung und Internetnutzung, allerdings bezeichnet sich ein Drittel aller Befragten selbst als Anfänger in diesen Bereichen.
- die technischen Voraussetzungen für die Nutzung (Internetzugang privat / Universität, privater PC) sind größtenteils gegeben, allerdings beklagen Studierende teilweise lange Wartezeiten auf freie Arbeitsplätze in den Computer-Pools der Universitäten.
- die meisten Studenten haben privat lediglich ein analoges Modem zur Verfügung, was die Entwicklung einer reinen Online-Version nicht sinnvoll erscheinen läßt.
- es besteht grundsätzlich Interesse an einer digitalen Nachbearbeitung von Vorlesungen, besonders gewünscht werden hier Bildmaterial mit Zoom-Möglichkeit und der Vorlesungstext.
- über 95% der befragten Studenten, die bereits digitale Vorlesungen gehört haben, beurteilen die Qualität der digitalen Vorlesungen gleich gut oder besser als die „normaler“ Vorlesungen.
- es wird die Einrichtung eines Email-Verteilers für Fragen gewünscht.

- in Vorlesungen sollten mehr bewegte Bilder in Form von Videos oder Animationen integriert werden.
- Bilder sollen auch nebeneinander gezeigt werden können, damit Unterschiede zwischen Diagnosen besser erklärt werden können.
- Als Konsequenzen ergeben sich die folgenden Aspekte:
- der Ansatz des SENTIMED-Projekts, nämlich die Durchführung multimedialer Vorlesungen wird von den Studenten sehr gut akzeptiert. Weiterführende Hilfsmittel zur Nachbereitung werden auch von studentischer Seite ausdrücklich begrüßt.
- Veranstaltungen, die in den Umgang mit der Lernumgebung einführen, sind sinnvoll. Diese sollten sich vor allem auch an Frauen richten, da diese selbst ihre computerbezogenen Fähigkeiten im Vergleich zu Männern als wesentlich geringer einschätzen.
- die Zugangsmöglichkeiten zu PCs an der Universität für die Nutzung der Lernumgebung sollten verbessert werden.
- eine Offline (CD-basierte) Unterstützung ist nötig.

Studenten, die bereits digitale, multimediale Vorlesungen gehört hatten, wiesen zudem auf die Notwendigkeit einer Backup-Lösung für den Fall von technischen Ausfällen hin. Außerdem seien lichtstarke Multimediaprojektoren wünschenswert, damit der Vorlesungssaal nicht zu stark verdunkelt werden muß.

6.2 Konzept, Module und Aufbau

Die Lehr- und Lernumgebung gliedert sich in drei Hauptbausteine *Lernobjekt-Datenbank*, *Autorenwerkzeuge* für die Dozenten und eine *Lernumgebung* für den Studenten.

Für die Verwaltung der Lernobjekt-Typen und Lernobjekte (siehe Abschnitt 6.3) wird aus den folgenden Gründen eine Datenbank verwendet:

- Die Benutzerverwaltung der Datenbank kann genutzt werden.
- Der Mehrbenutzerbetrieb wird bereits durch das Datenbanksystem geregelt.
- Der Zugriff von verschiedenen Arbeitsplätzen/Institutionen und Applikationen ist ohne Probleme möglich.
- Die Suche nach Lernobjekten kann durch Datenbankabfragen realisiert werden.
- Die Lernobjekte und Lernobjekt-Typen sind einfach verwaltbar.
- Dadurch ist auch die einfache Erweiterbarkeit auf andere Lernobjekt-Typen möglich.

- Lernobjekte können nach einer Datenbankrecherche einfach in andere Lernobjekte integriert werden und somit jederzeit wiederverwendet werden.
- Die Redundanz wird minimiert. Ein Lernobjekt, das in mehreren anderen Lernobjekten verwendet wird (zum Beispiel ein Bild), muß nicht mehrfach abgelegt werden.
- Lernobjekte können normiert mit Attributen versehen werden.
- Dadurch wird eine Konformität mit internationalen Standards (s. Kapitel 2) erreicht.

Diese Lernobjekt-Datenbank (Abschnitt 6.4) ist gleichzeitig der zentrale Baustein des Pakets. Auf Dozentenseite stehen verschiedene Autorenwerkzeuge (Abschnitt 6.5) zur Verfügung. Diese sind teilweise direkt über JDBC (Java Database Connectivity) bzw. ODBC (Open Database Connectivity), teilweise indirekt über WWW-Server und Perl-CGI-Skripten mit DBI/DBD (Database Interface/Database Driver) mit der Datenbank verbunden. Den Studenten steht schließlich eine personalisierbare Lernumgebung (Abschnitt 6.6) zur Verfügung.

Die Lernobjekte der verschiedenen beteiligten Zentren werden nicht in einer zentralen Datenbank, sondern aus den folgenden Gründen jeweils in lokalen Datenbanken verwaltet (Abbildung 27):

- In den jeweiligen Zentren (Hautklinken) stehen bereits Firewalls zur Verfügung, die unerlaubten Zugriff auf die Daten von außen verhindern. Aufgrund dieser Firewalls ist unter Umständen auch schon der Zugriff auf eine Datenbank außerhalb der Einrichtung beschränkt.
- Lernobjekte sind Eigentum der jeweiligen Institutionen. Die Verwendung zum Beispiel eines Bildes einer anderen beteiligten Institution kann nur nach Anforderung erfolgen.
- Der Zugriff über das Netz an einen entfernten Server ist langsamer und fehleranfälliger.
- Eine Anpassung der Datenbank an Institutions-Spezifika ist möglich. Allerdings sollte dies nur in Ausnahmefällen und mit Bedacht geschehen, damit eine Zusammenführung der lokalen Datenbanken zu einer zentralen Datenbank möglich bleibt.

Um auch den Austausch von Lernobjekten zwischen den beteiligten Zentren zu gewährleisten, werden die jeweiligen lokalen Datenbanken regelmäßig zu einer zentralen Datenbank zusammengefügt, auf die alle Beteiligten Zugriff haben. Hier werden allerdings nicht die vollständigen Lernobjekte gehalten, für Bilder oder Videos werden zum Beispiel nur

Vorschaubilder zentral gespeichert. Lernobjekte, die von anderen Partnern verwendet werden sollen, müssen über die zentrale Datenbank vom Eigentümer angefordert werden.

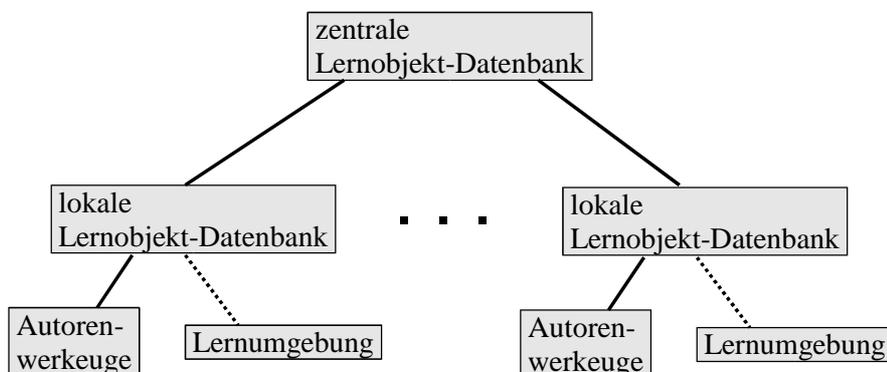


Abbildung 27: Module der Lernumgebung, lokale und zentrale Datenbanken.

Zur Installation der einzelnen Module steht eine Installations-CD für Windows-Plattformen zu Verfügung (siehe Anhang E.1, Seite 181).

6.3 Lernobjekte in der Dermatologie

Die aus dem Bereich der Dermatologie innerhalb dieser Arbeit betrachteten Lernobjekte sind *Bild-* und *Videoobjekte*, *Moulag*en, *Vorlesungen*, *Lerntexte*, *Skripte*, *Fallbeispiele* und *Quiz*elemente. Die folgenden Abschnitte resultieren zum großen Teil aus Erfahrungen in der Zusammenarbeit mit den Hautklinken der Universitäten Würzburg, Jena und Mannheim, an denen auch erste Prototypen der entwickelten Module getestet wurden.

In enger Zusammenarbeit mit den Hautärzten wurden die Datenfelder definiert, die die einzelnen Lernobjekt-Typen beschreiben. Als Grundlage für die Verschlagwortung werden jeweils möglichst internationale Standards verwendet. Sind keine Standards vorhanden, werden selbstdefinierte Codes verwendet. Die folgende Aufzählung beschreibt die Datenfelder, die allen Typen von Lernobjekten zuordenbar sind sowie ihre Bedeutung und die jeweils verwendete Codierung mit einem Beispiel. Die Eigenschaften, die nur auf speziellen Lernobjekt-Typen zutreffen, werden in den entsprechenden Unterabschnitten erläutert.

A) **Diagnose(n)**. Den meisten Lernobjekten kann eine oder mehrere Diagnosen zugeordnet werden.

Ausnahmen sind hier Grafiken, die zum Beispiel den Aufbau der Haut beschreiben oder Videos, die Operationstechniken zeigen. Bei Angabe mehrerer Diagnosen kann es sinnvoll sein, die Reihenfolge der Diagnosen zu berücksichtigen, wenn zum Beispiel eine

Diagnose eine Folgeerscheinung der anderen ist.

Grundlage für die Verschlagwortung: erweiterte ICD10-Codierung des Dermatologischen Diagnosenkatalogs (siehe Abschnitt 4.1.3)

Beispiel: L63.100 (für *Alopecia universalis*)

B) **Lokalisation(en).** In vielen Fällen kann bestimmt werden, welche Körperregion(en) von den in A) angegebenen Erkrankungen betroffen sind.

Grundlage für die Verschlagwortung: Zur Codierung wurde der sogenannte Tumorphistologieschlüssel (ICD-O) [1] herangezogen.

Beispiel: C44.41 (für *behaarter Kopf*)

C) **Effloreszenzen.** Effloreszenzen sind die sichtbaren Grundelemente einer Hauterkrankung. Man unterscheidet Primär-, Sekundär- und andere Effloreszenzen.

Grundlage für die Verschlagwortung: eigene Codierung

Beispiel: PE070 für *Pustel*, SE070 für *Schorf*

D) **Farbe.** Die Farbe einer abgebildeten Hauterkrankung ist nur schwierig zu bestimmen, weil sie unter anderem vom subjektiven Eindruck des Betrachters, von den Hardware-Einstellungen (Monitor, Grafikkarte), von der Qualität des Dias bzw. des eingescannten Bildes und von der verwendeten Filmmarke abhängt. Nachdem zu Beginn eine sehr feinkörnige Farbskala erarbeitet wurde (... *mittelbraun, dunkelbraun, braunschwarz, ...*) hat man sich später auf eine relativ grobe Farbbestimmung festgelegt (... *rot, braun, ...*).

Grundlage für die Verschlagwortung: eigene Codierung.

Beispiel: FA010 für *rot*

E) **Form.** Sie beschreibt das Aussehen der Umrandung einer Hauterkrankung.

Grundlage für die Verschlagwortung: eigene Codierung

Beispiel: FO005 für *bizarrr*

F) **Verteilung.** Hier kann angegeben werden, wie die Effloreszenzen einer Erkrankung verteilt sind, also zum Beispiel gruppiert oder vereinzelt.

Grundlage für die Verschlagwortung: eigene Codierung

Beispiel: VE010 für *disseminiert*, VE015 für *einzel(solitär)*

G) **Zielgruppe.** Sie beschreibt die Personengruppe, für die die Verwendung des Lernobjekts vorgesehen ist.

Grundlage für die Verschlagwortung: es wird zwischen *Studenten* (Verwendung in der Lehre) und *Ärzte* (Verwendung für Weiterbildung / für Vorträge) unterschieden.

- H) **Eigentümer.** Das Feld gibt das Institut bzw. die Person, die das Bild in die Datenbank eingebracht hat, an.
- I) **Datum.** Hier wird das Datum, zu dem das Lernobjekt in die Datenbank aufgenommen wurde, abgelegt.
- J) **Bemerkung.** Dieses Freitext-Datenfeld nimmt alle durch die übrigen Datenfelder evtl. nicht abgedeckten Daten auf.

6.3.1 Bildobjekte

In der klassischen, nicht digitalen Lernumgebung ist ein Bildobjekt in der Regel ein Diapositiv mit handschriftlichen Bemerkungen, in der digitalen Lernumgebung zum Beispiel ein eingescanntes Pixelbild mit den dazu erfaßten Daten.

Bilder spielen in der Dermatologie eine sehr wichtige Rolle. Damit Studenten Hauterkrankungen diagnostizieren und unterscheiden können, müssen sie sie vorher gesehen haben. Man spricht dabei oft auch von der Schulung des sogenannten *diagnostischen Blicks*. Je geschulter der diagnostische Blick eines Arztes ist, desto schneller kommt er durch einfaches Betrachten einer Hauterkrankung zur richtigen Diagnose oder zumindest zu guten Verdachtsdiagnosen und kann diese dann mit weiteren Tests bestätigen oder widerlegen. Je besser die Verdachtsdiagnosen sind, desto schneller und billiger – man kann sich eventuell teure Tests sparen - kommt man zur richtigen Diagnose

Für die Schulung stehen aber nicht immer reelle Patienten zur Verfügung, sondern nur Bilder. Deshalb bestehen auch Vorlesungen in der Dermatologie im wesentlichen aus Bildern von Hauterkrankungen. Sowohl in Würzburg als auch in Jena umfaßt das Diaarchiv inzwischen mehrere tausend analoge Diapositive. Teilweise, zum Beispiel in der Hautklinik Würzburg, werden die Dias in sogenannten Diakästen gruppiert. Diese haben eine Bezeichnung, die das Themengebiet, zu dem die Bilder gehören, beschreiben (Bsp.: Diakasten L2 entspricht *Lichen ruber*). Zu jedem Themengebiet kann es mehrere Diakästen geben.

Neben diesen klinischen Bildern werden auch sogenannte histologische Bilder und Grafiken verwendet. Histologische Bilder sind mikroskopische Aufnahmen von (eingefärbten) Schnitten, die von entnommenen Hautpartien erstellt werden. Grafiken werden meist zur Veranschaulichung von theoretischen Sachverhalten verwendet, zum Beispiel zur Darstel-

lung des Aufbaus der Haut.

Auf den meisten Dias des Archivs befinden sich handschriftlich u.a. die Patienten-Nummer (Würzburg), der Patientennamen (Jena) und die Bezeichnung der Diagnose(n). In Würzburg wird auch der Diakasten und eine laufende Nummer innerhalb des Diakastens vermerkt. Leider ist die in Würzburg vergebene Patienten-Nummer nicht so zu verstehen, daß Bilder des gleichen Patienten auch immer die gleiche Patienten-Nummer erhalten. Vielmehr kann die Nummer nur dazu dienen, eventuell über den Umweg der Akten andere Bilder des Patienten zu finden. Das ist allerdings mit einem relativ hohen Zeitaufwand verbunden. So können zusammengehörende Bilder nur dann als solche identifiziert werden, wenn dies bei der Datenerfassung mehr oder weniger zufällig auffällt. Solche Serien von Bildern sind für die Dokumentation eines Krankheitsverlaufs interessant.

Zusätzlich zu den allgemeinen Attributen von Lernobjekten sind für Bildobjekte folgende Datenfelder von Interesse:

- **Patient.** Die Identität der auf den Bildern dargestellten Patienten spielt für die Anwendung in der Lehre oder Weiterbildung eine untergeordnete Rolle. Wichtig wird die Information nur, wenn Bilder zu einer Gruppe zusammengefaßt werden sollen, d.h. wenn zum Beispiel alle Bilder eines Patienten gezeigt werden, um einen Krankheitsverlauf darzustellen.

Grundlage für die Verschlagwortung: Bildnummer, über die letztendlich auch der Patient ermittelt werden kann.

Beispiel: 1200/98

- **Einverständnis.** Je nach Aufnahme ist es möglich, auf klinischen Bildern Personen zu erkennen und zu identifizieren. Liegt das Einverständnis eines Patienten vor, können Bilder auch ohne Anonymisierung verwendet werden, andernfalls muß die Person unkenntlich gemacht werden.

Grundlage für die Verschlagwortung: es wird nur zwischen „Anonymisierung nicht notwendig“, wenn keine Person erkennbar ist und „Einverständnis liegt vor / liegt nicht vor“ unterschieden.

- **Koordinaten von Anonymisierungsrechtecken.** Ist auf einem Bild ein Patient identifizierbar, muß es anonymisiert werden. Dies geschieht in der Regel durch Abdecken des Augenbereichs mit (schwarzen) Rechtecken. Statt die Rechtecke direkt in die Bilder ein-

zufügen, werden nur ihre Pixelkoordinaten abgelegt. Zur Präsentation des Bildmaterials für den Lernenden können die Rechtecke dann im Batchbetrieb tatsächlich in die Bilder eingefügt werden.

- **Bildqualität.** Durch dieses Datenfeld wird die tatsächliche Qualität bewertet. Auf die Bildqualität haben neben der Qualität des Diapositivs auch das Einscannen oder eventuelle Verschmutzungen Einfluß.

Grundlage für die Verschlagwortung: eigene Codierung, Unterscheidung nach *sehr gut, gut, befriedigend*

- **Drehfaktor/Spiegelung.** Das Feld wird nur für interne Zwecke benötigt, falls das Bild zum Zeitpunkt der Verschlagwortung nicht korrekt gedreht ist bzw. spiegelverkehrt gescannt wurde.
- **Aufnahmedatum.** Es ist das Datum, zu dem das Bild selbst gemacht wurde. Diese Daten können beispielsweise zur Dokumentation von Krankheitsverläufen verwendet werden.

Weitere Informationen zu Bildobjekten: Autorenwerkzeug: Abschnitt 6.5.1, Seite 102, Verwendung von Bildobjekten in der Lernumgebung: Abschnitt 6.6.4, Seite 123.

6.3.2 Moulagen

Moulagen (französisch: „Maske“) sind Wachsabbildungen und werden in der Dermatologie zur plastischen, realitätsnahen Veranschaulichung von Hauterkrankungen verwendet. Besonders interessant sind Moulagen für selten auftretende Diagnosen und für Diagnosen, für die ein zweidimensionales Bild nicht anschaulich genug ist.

Für den Einsatz in der Lehr- und Lernumgebung werden Moulagen aus verschiedenen Richtungen mit einer digitalen Kamera fotografiert.

Siehe auch: Autorenwerkzeug: Abschnitt 6.5.4, Seite 107, Verwendung in der Lernumgebung: Abschnitt 6.6.5, Seite 123.

6.3.3 Videoobjekte

In der klassischen Lernumgebung wurden Videoobjekte bisher nicht eingesetzt. Videos können im dermatologischen Umfeld dazu dienen, dem Lernenden zum Beispiel Operationstechniken oder die Durchführung von sogenannten Testungen zu erläutern. Zusätzlich zu den allgemeinen Attributen von Lernobjekten können bei Videos von Operationstechniken

Codes des Operationenschlüssels (siehe Abschnitt 4.2) angegeben werden.

Siehe auch: Autorenwerkzeug: Abschnitt 6.5.1, Seite 102, Verwendung in der Lernumgebung: Abschnitt 6.6.4, Seite 123.

6.3.4 Vorlesungen

Klassische Dermatologie-Vorlesungen bestehen aus einer Reihe von Textdias und Bildern, die mit einem Diaprojektor gezeigt werden. An der Vorbereitung und Durchführung solcher Präsenzvorlesungen sind i.a. mindestens zwei Personen beteiligt. Zum einen muß der Dozent geeignete Bilder aus dem Diaarchiv der Klinik auswählen. Oft werden diese noch durch weitere Bilder aus seinem privaten Fundus ergänzt. Zur weiteren Erläuterung von Diagnosen werden zum Beispiel in Powerpoint digitale Folien erstellt, von denen wiederum Dias erstellt werden. Beides, Powerpoint-Dias und Krankheitsbilder, wird dann von einem Vorlesungsassistenten in richtiger Reihenfolge und Orientierung in den Diaschlitten des Projektors einsortiert und im Hörsaal gezeigt. Dabei bedient der Assistent den Projektor auf Anweisung des Dozenten.

Für die Entwicklung einer Lehr-/Lernumgebung muß man zwischen diesen üblichen *Präsenzvorlesungen* und *Vorlesungen zur Nachbereitung* unterscheiden. Beide Arten sollen neben den bekannten Bildobjekten auch andere multimediale Lernobjekte wie zum Beispiel Videos enthalten. Vorlesungen wurden zur Nachbereitung bisher nicht angeboten: dazu hätte man den Studenten die Dias zur Verfügung stellen müssen. Hat man nun aber schon die Präsenzvorlesung in digitaler, multimedialer Form, ist es einfach, sie zur Nacharbeit aufzubereiten.

Bemerkung: Bei der Verschlagwortung können oft Diagnosen höherer Hierarchie-Ebenen angegeben werden, da eine Vorlesung meist eine Diagnosengruppe abdeckt. Vorlesungen enthalten in der Regel andere Lernobjekte wie Bild- oder Videoobjekte und „erben“ deren Attribute.

Siehe auch: Autorenwerkzeug: Abschnitt 6.5.5, Seite 108, Verwendung in der Lernumgebung: Abschnitt 6.6.5, Seite 124.

6.3.5 Lerntexte und Skripte

Lerntexte stellen dem Lernenden den Stoff eines gewissen Themengebietes in strukturierter Form dar. Die klassische Form von Lerntexten findet man in Lehrbüchern. Als Skript bezeichnet man einen sehr kompakten, übersichtlichen Auszug aus den Lerntexten. Lerntexte dienen in diesem Sinne eher zum Erlernen von neuen Lerninhalten, Skripte zum Nacharbeiten und Auffrischen von bereits bekanntem Wissen oder zur Prüfungsvorbereitung.

Bemerkung: Ähnlich wie Vorlesungen fassen Lerntexte oft verschiedene Diagnosen zusammen und können deshalb mit Diagnosen höheren Hierarchie-Ebenen verschlagwortet werden. Auch Lerntexte und Skripte enthalten andere Lernobjekte (Bild, Video).

Siehe auch: Autorenwerkzeug: Abschnitt 6.5.6, Seite 111, Verwendung in der Lernumgebung: Abschnitt 6.6.7, Seite 126.

6.3.6 Fallbeispiele

Zur Vertiefung von erlerntem Wissen werden den Dermatologie-Studenten konkrete Patienten-Fallbeispiele von der Anamnese bis zum Entlassungsbefund präsentiert:

1. *Anamnese:* Beschreibung eines Patienten. Dazu gehören alle Daten, die man von einem realen Patienten wissen oder erfragen kann, zum Beispiel: Arbeitsplatz, Hautkrankheiten in der Verwandtschaft usw.
2. *Aufnahmebefund:* Beschreibung der Erscheinung der Hauterkrankung (Färbung, Lokalisation, ...)
3. *Diagnostik:* Beschreibung der durchgeführten Tests
4. *Diagnose:* Welche Diagnose wurde aufgrund der vorhergehenden Punkte gestellt?
5. *Therapie:* Welche Therapie-Maßnahmen wurden ergriffen?
6. *Entlassungsbefund:* Welche Wirkung haben die angewendeten Therapien gezeigt?

In der klassischen Lehre werden Fallbeispiele simuliert, indem den Studenten am Ende einer Vorlesung ein realer Patient vorgestellt wird. Eine kleine Gruppe von Studenten hat dann die Chance, den Patienten zu untersuchen und eine Diagnose zu stellen. Das hat zwar den Vorteil der Realitätsnähe, aber auch den Nachteil, daß erst Patienten mit passenden Erkrankungen gefunden werden müssen und daß immer nur eine sehr kleine Gruppe von Studenten einen direkten Nutzen zieht.

Im digitalen Abbild eines Fallbeispiels kann der Ablauf eines Falls durch Texte, angereichert durch Bild- oder Videoobjekte veranschaulicht werden. Zusätzlich können dem Studenten individuelle Hilfestellungen geboten werden.

Bemerkung: Alle nötigen Eigenschaften werden bereits durch die Attribute der Lernobjekte allgemein abgedeckt. Fallbeispiele können Bild- und Videoobjekte enthalten.

Siehe auch: Autorenwerkzeug: Abschnitt 6.5.7, Seite 114, Verwendung in der Lernumgebung: Abschnitt 6.6.7, Seite 126.

6.3.7 Quizelemente

Mit Ausnahme von Klausurfragen und der schon beschriebenen Vorstellung eines Patientenfalls werden bisher in der Lehre kaum Quizelemente verwendet. Sie sind aber wichtig, weil sie die Lernumgebung auflockern und es dem Lernenden ermöglichen, in einer spielerischen Umgebung zu lernen. Besonders auf diesem Gebiet kann man von den digitalen, multimedialen Möglichkeiten profitieren. Im Moment werden drei Arten von Quizelementen unterschieden:

Das **Bilderquiz** zeigt dem Studenten Bildobjekte, zu denen jeweils die korrekte Diagnose zu stellen ist. Dabei kann man die Schwierigkeitsstufe variieren, indem man unterschiedliche Anzahl von Lösungsvorschlägen anbietet.

Übungsfragen sind Quizelemente, die (Multiple-Choice-)Klausurfragen nachstellen.

Im **Fallquiz** wird dem Studenten ein Patient mit einem Aufnahmebefund vorgestellt. Er muß dann eine Anamnese durchführen, zu der er die richtigen Fragen stellen muß. Der Student muß notwendige Tests durchführen, eine Diagnose stellen und schließlich Therapien vorschlagen. Das Fallquiz entspricht einem durch Aufgaben angereicherten Fallbeispiel.

Siehe auch: Autorenwerkzeug: Abschnitt 6.5.8, Seite 115, Verwendung in der Lernumgebung: Abschnitt 6.6.8, Seite 126.

6.3.8 Beziehungen der Lernobjekte untereinander

Wie aus den vorigen Abschnitten klar wird, stehen Lernobjekte meist nicht für sich alleine, sondern bestehen aus anderen Lernobjekten. Abbildung 28 veranschaulicht diese Beziehungen der Lernobjekte untereinander: *Bildobjekte*, *Moulagen* und *Videoobjekte* bilden die unterste, erste Ebene. Sie kann man im vorliegenden Kontext als atomare Lernobjekte bezeichnen. Sie beinhalten keine weiteren Lernobjekte und sind jeweils Teil der Lern-

objekte auf zweiter Ebene: *Vorlesungen*, *Lerntexte/Skripte*, *Fallbeispiele* und *Quizelemente*. Diese Lernobjekte lassen sich wiederum *Themenbereichen* zuordnen. Bei allen Beziehungen kann ein Lernobjekt selbstverständlich nicht nur Teil eines anderen sondern beliebig vieler Lernobjekte sein. Für die Verschlagwortung gilt dann natürlich, daß sich alle Eigenschaften nach oben vererben, d.h. besteht zum Beispiel eine Vorlesung aus Bild- und Videoobjekten erbt die Vorlesung alle Eigenschaften dieser Teil-Lernobjekte.

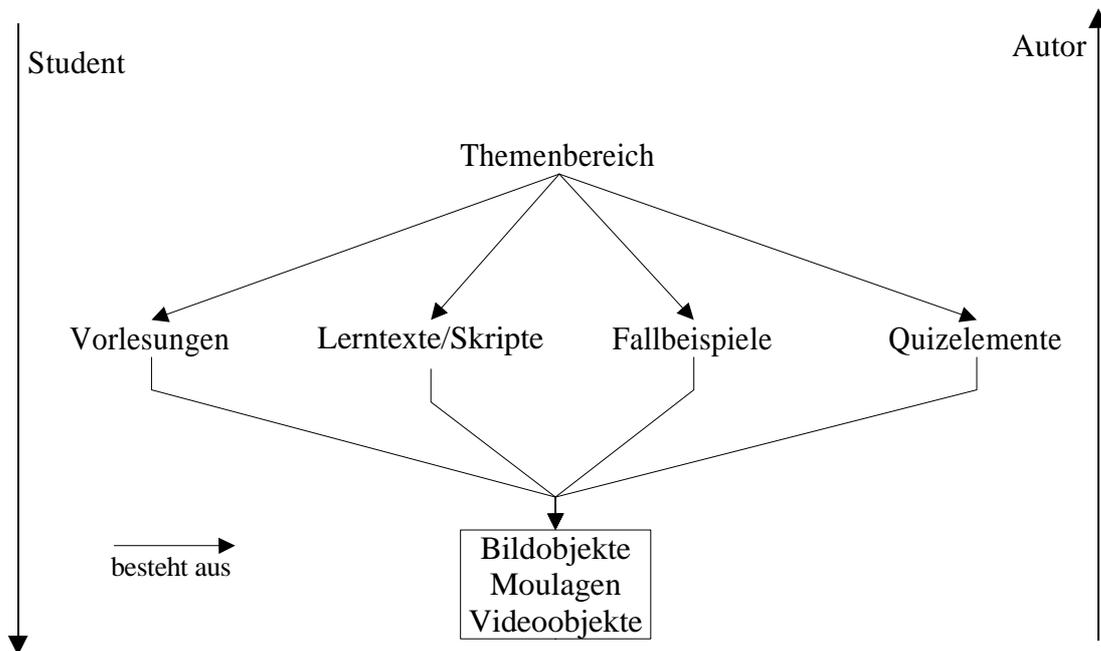


Abbildung 28: Lernobjekt-Hierarchie.

Die unterste Ebene der atomaren Lernobjekte ist alleinstehend nicht für den Studenten geeignet. Studenten durchlaufen die in Abbildung 28 dargestellte Lernobjekt-Hierarchie also eher von oben nach unten. Abbildung 29 veranschaulicht ein typisches Beispiel für die Abfolge der Abarbeitung von Lernobjekten eines Studenten (vgl. Lernpfad, Abschnitt 3).

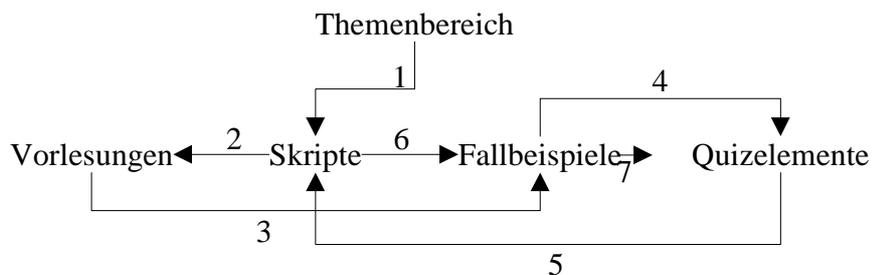


Abbildung 29: Beispiel für einen Lernpfad

Autoren werden dagegen zuerst die atomaren Lernobjekte erstellen, um aus diesen dann die Lernobjekte höherer Ebenen zusammenzusetzen.

6.3.9 Matching mit dem LOM-Standard

Die Attribute der Lernobjekte lassen sich recht einfach auf die Datenfelder des LOM-Standards (Abschnitt) mappen. Die meisten zu erfassenden Daten lassen sich mit dem Standard modellieren. Im folgenden ein exemplarischer Auszug aus dem Matching der Lernobjekt-Attribute mit dem LOM-Standard (siehe Anhang A):

<i>Dermatologie-Lernobjekt</i>	<i>LOM-Standard</i>
„Ebene“ (vgl. Abbildung 28)	1.9 general.aggregationlevel
Patient, Eigentümer	2.3 lifecycle.contribute
Typ (Bild, Video, Lerntext, ...)	5.2 educational.learningresourcetype
Zielgruppe	5.6 educational.context
Beziehung zwischen den Lernobjekten	7 relation
Diagnose, Lokalisation, Effloreszenzen, Farbe, Form, Verteilung	9 classification

Daneben gibt es allerdings auch einige Attribute, die sich nicht in den LOM-Standard einordnen lassen, wie zum Beispiel

- das Einverständnis eines Patienten zur Verwendung eines Bildobjekts in der Lehre.
- die Speicherung der Koordinaten der Anonymisierungs-Rechtecke. Liegt keine Einverständnis-Erklärung des Patienten vor, müssen Regionen des Bildes durch Einfügen von Rechtecken unkenntlich gemacht werden. Die Speicherung solcher Koordinaten ist im LOM Standard nicht trivial abbildbar.

Bis auf wenige Ausnahmen ist also die Darstellung der Lernobjekte konform zum LOM-Standard möglich (siehe Abschnitt 6.7).

6.4 Lernobjekt – Datenbank

6.4.1 Anforderungen

Die Datenbank soll die folgenden Anforderungen erfüllen:

- Die Datenbank soll im Hinblick auf eine spätere Verwendung der Lernobjekte in anderen Lernumgebungen möglichst konform mit dem *IEEE LOM Standard* (vgl. Abschnitt 2.2.2) sein. Zumindest soll ein einfacher Import/Export von LOM-konformen XML-Datensätzen möglich sein.

- Die Hierarchie der Lernobjekte (vgl. Kap. 6.3.8) soll gut abgebildet werden.
- Die Datenbank sollte auf einfache Art und Weise erweiterbar sein, d.h. neue Lernobjekt-Typen sollen ohne großen Aufwand eingefügt werden können. So kann auch die Anpassung auf andere Anwendungsbeispiele gewährleistet werden.
- Die Anbindung an (in unterschiedlichen Programmiersprachen erstellten) Applikationen soll möglichst einfach und flexibel sein.

6.4.2 Umsetzung

Abbildung 30 veranschaulicht die Struktur der Datenbank: Die Tabelle *LearningObjects* ist die zentrale Tabelle der Datenbank. In ihr sind alle Lernobjekte mit ihrer Lernobjekt-ID (im folgenden mit LOID abgekürzt) enthalten. Die verschiedenen Typen von Lernobjekten werden in der Tabelle *SpecialTables* verwaltet, in der neben dem Namen des Typs (*Picture*, *Videoclip*, ...) auch der jeweilige Name der Haupt-Tabelle steht, in der die Haupteigenschaften der Lernobjekte dieses Typs gespeichert werden. Die Tabellen *LearningObjects* sowie die jeweiligen Typ-Tabellen sind sehr dynamisch, d.h. ihre Inhalte werden bei jedem Einfügen, Löschen und Bearbeiten von Lernobjekten modifiziert. Die Codierungstabellen, die im wesentlichen die Zuordnung zwischen einem Code und der damit codierten Bezeichnung enthalten, sind dagegen eher statisch.

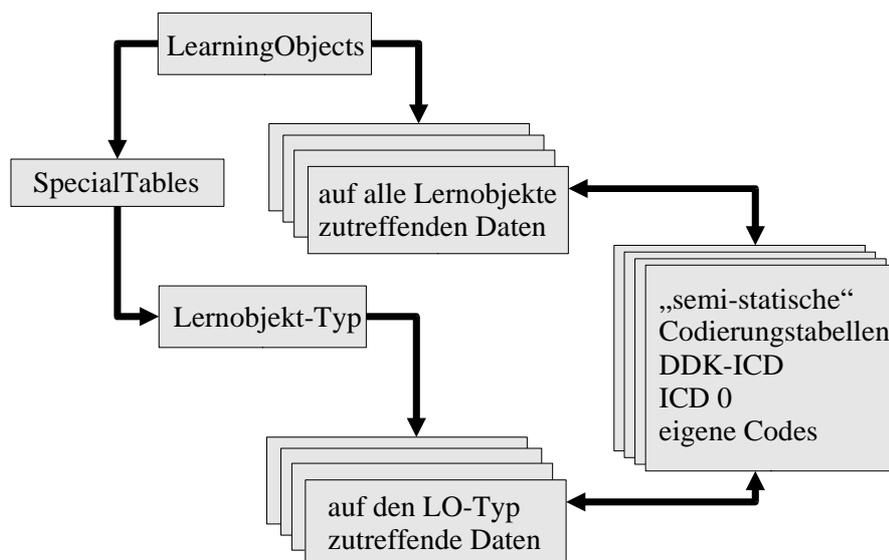


Abbildung 30: Verschiedene Arten von Tabellen in der Datenbank

Sie ändern sich beim Einfügen, Löschen oder Bearbeiten von Lernobjekten nicht, sondern nur beim Einfügen oder Aktualisieren der Codetabellen. Dies ist zum Beispiel der Fall, wenn

- eine bisher unbekannte Diagnose aufgenommen werden muß,
- ein neuer Lernobjekt-Typ aufgenommen wird, der mit anderen, noch nicht in die Datenbank integrierten Eigenschaften verschlagwortet werden soll oder
- Codes aktualisiert werden (zum Beispiel bei der Umstellung von ICD9 auf ICD10).

In der Datenbank gibt es für jedes Lernobjekt einen Verweis zum Lernobjekt selbst, d.h. die Lernobjekte werden nicht als BLOBs (Binary Large Objects) oder CLOBs (Character Data Large Objects) in der Datenbank gespeichert, sondern im Server- oder User-Objektpfad. Abbildung 31 skizziert die Client-Server-Architektur der Umgebung. Der *Server-Objektpfad* ist ein Verzeichnis auf einem Netzlaufwerk, unter dem alle Lernobjekte liegen. Autoren, die an den Clients arbeiten müssen Schreibrechte, Lernende Leserechte auf diesem Verzeichnis haben.

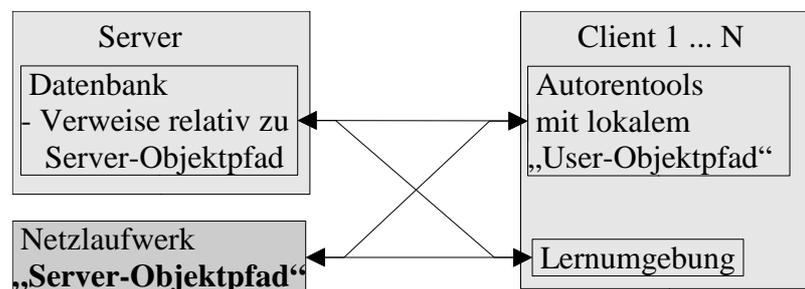


Abbildung 31: Client/Server-Architektur mit Zugriff auf ein Netzlaufwerk

Der nur für Autoren wichtige *User-Objektpfad* ist ein lokales Verzeichnis auf dem jeweiligen Client-Rechner, unter dem die Lernobjekte liegen, die in die Datenbank eingefügt werden sollen. In der Datenbank werden aber nicht absolute Pfade auf dem Netzlaufwerk abgelegt, sondern die Position relativ zum Server-Objektpfad. Die Autorentools sind konfigurierbar, d.h. in ihnen wird der Server-Objektpfad als absoluter Pfad einmal angegeben und jedesmal, wenn auf ein Lernobjekt zugegriffen werden soll, wird der in der Datenbank gespeicherte relative Pfad angehängt. Diese Auslagerung der Objekte hat zwar den Nachteil, daß es durch Löschen oder Umbenennen von Dateien zu Inkonsistenzen kommen kann, aber auch den Vorteil, daß Lernobjekte leicht außerhalb der Datenbank bearbeitet werden können. Wird zum Beispiel entschieden, daß die Anonymisierung von Bildobjekten nicht durch das Einblenden von schwarzen Rechtecken, sondern durch eine Rasterung der ent-

sprechenden Bildbereiche geschehen soll, kann dies unabhängig von der Datenbank im Batchbetrieb passieren.

Insgesamt umfaßt die Datenbank über 50 Tabellen, die in Anhang D, Seite 169, mit ihrer Zuordnung zu den Lernobjekt-Typen beschrieben sind.

6.4.3 Auswahl des Datenbanksystems

Zur Verwaltung der Lernobjekte und Lernobjekt-Typen wird eine relationale Datenbank verwendet. Da einige Lernobjekt-Typen wie Lerntexte oder Fallbeispiele durch XML-Dokumente repräsentiert werden, wäre die Nutzung eines relationalen Datenbanksystems mit *XML-Unterstützung*, z.B. IBM DB2 oder Oracle 8i/9i, ebenfalls denkbar. Solche Systeme verfolgen meist zwei Ansätze: die Speicherung des XML-Dokuments als CLOB oder die Abbildung der Dokumentenstruktur auf Tabellen und Spalten, was allerdings nur für einheitlich strukturierte XML-Dokumente möglich ist [4].

Im Moment wurde auf den Einsatz eines solchen Systems verzichtet, weil nur externe Referenzen und nicht die Lernobjekte selbst in der Datenbank gespeichert werden und Diskussionen mit den Ärzten ergaben, daß die Dokumente nur als Gesamtheit präsentiert werden sollen. Auch ein Update von einzelnen Teilelementen von Dokumenten ist nicht geplant.

Für die Zukunft sollte man den Einsatz aber nicht ausschließen, da sich

- die XML-Unterstützung in den Datenbanken noch weiter entwickeln wird (auch das SQL-Standardisierungskomitee der ISO arbeitet an einer Erweiterung von SQL um SQL/XML [51])
- man durch das Speichern der XML-Daten die volle Datenbank-Funktionalität erhält. Durch den Einsatz einer entsprechenden Anfragesprache (XML-QL, XQL) können folgende Anfragetypen effizient bearbeitet werden und liefern XML-Dokumente zurück:
 - wertorientiert: Vergleich von ein Tag-/Attributwerten,
z.B. @attr >5 bei <tag attr="4">, tag < 7.1 bei <tag>6.1</tag>
 - textorientiert: Dokumente, die einen Text (nicht) enthalten.
 - strukturorientiert:
z.B. /Buch/Titel: das Dokument enthält <buch>..<titel>...</titel>..</buch>
 - linkorientiert: Dokumente, die auf ein bestimmtes Ziel verweisen oder die von Dokumenten referenziert werden.

- die betrachteten Dokumente strukturiert und damit einfach auf Relationenschemata abbildbar sind

Auch hybride Ansätze wie in [4] erwähnt, sind denkbar: man könnte zum einen das XML-Dokument als ganzes (CLOB oder Verweis) speichern, die einzelnen Abschnitte aber zusätzlich in die Datenbank einfügen. Dadurch erspart man sich das Wiederzusammensetzen des Dokuments bei der Präsentation und die Inhalte werden in der Datenbank mit den üblichen Datenbankabfrage-Möglichkeiten suchbar. Auf der anderen Seite wird Redundanz in das System eingebracht.

Weil auch viele große Mengen an multimedialen Inhalten, zum Beispiel Bilder und Videos, verwaltet werden, bietet sich auch der Einsatz von *Multimedia-Datenbanksystemen* an. Sie bieten zusätzlich zu den üblichen SQL-Anfragen auch Ähnlichkeitsanfragen. So können zu einem Bild alle zu diesem Bild ähnliche Bilder gefunden werden. Dazu werden für die unterschiedlichen Medientypen Ähnlichkeitsmaße definiert. Für die Zukunft ist die Verwendung eines Multimedia-Datenbanksystems vor allem für Autoren bei der Erstellung komplexerer Lernobjekte interessant.

Während der Entwicklung wurde allerdings eine MySQL-Datenbank gewählt, da sie

- kostenlos zur Verfügung steht,
- leicht installierbar ist,
- für Windows und Linux-Plattformen zur Verfügung steht,
- hohe Performance bietet,
- große Datenmengen verwalten kann,
- alle benötigten Features bereitstellt
- und am Lehrstuhl für Informatik II bereits installiert ist und in anderen Projekten ohne Probleme eingesetzt wird.

Alle Schnittstellen zwischen Anwendungen und der Datenbank basieren auf:

- JDBC (Java Database Connectivity) bei Zugriffen von Java-Applikationen
- ODBC (Open Database Connectivity) beim Zugriff über VisualBasic
- DBI/DBD (Database Interface/Database Driver) bei Zugriff über Perl-Skripte

Deswegen ist das eigentliche Datenbanksystem austauschbar, solange für sie entsprechende JDBC/ODBC/DBI-Treiber existieren.

6.4.4 Konvertierung des textbasierten Bildarchivs

Natürlich sollte das im Rahmen der Vorarbeiten (Abschnitt 5) erstellte textbasierte Bildarchiv als Grundlage für die Lernobjekt-Datenbank dienen. Da die Bildobjekte dort mit den gleichen Attributen versehen waren, stellt das Einfügen der Datensätze kein Problem dar. Nach dem Einlesen und Parsen der Textdateien können für das Einfügen in die Datenbank die entsprechenden, für das Autorentool *DEJAVUCapt* (siehe Abschnitt 6.5.1) erstellten Klassen verwendet werden.

6.4.5 Mergen der lokalen Datenbanken

Beim Mergen der lokalen Datenbanken zu einer zentralen wird entweder eine Datenbank ohne Lernobjekte, aber mit den Codetabellen, oder die Kopie einer lokalen Datenbank als Ausgangspunkt genommen. In diese werden dann sukzessive alle weiteren Objekte eingefügt. Beim Mergen trifft man auf das Problem doppelter Lernobjekt-IDs: Nachdem in den lokalen Datenbanken die IDs automatisch fortlaufend vom Datenbanksystem vergeben werden, gibt es natürlich in den verschiedenen Datenbanken identische IDs. Um Konflikte beim Mergen zu verhindern und gleichzeitig die Datenbankstruktur zu erhalten, wird eine zusätzliche Tabelle *matching* in die zentrale Datenbank eingefügt. Beim Hinzufügen eines Lernobjekts aus einer lokalen Datenbank wird eine neue Lernobjekt-ID vergeben und alle Verweise in den Tabellen auf die alte ID durch diese neue ersetzt:

- 1 Generiere Datenbank D_z ohne Lernobjekte mit Codetabellen
- 2 Für alle lokalen Datenbanken D_1 :
 - 2.1 Für alle Lernobjekte LO mit ID $loid_1$ aus D_1 :
 - 2.1.1 Generiere neue Lernobjekt-ID $loid_z$ für Datenbank D_z
 - 2.1.2 Füge LO in D_z mit $loid_z$ ein, ersetze beim Einfügen alle Tabelleneinträge $loid_1$ durch $loid_z$
 - 2.1.3 Füge in die Tabelle *matching* von D_z das Paar $(loid_z, loid_1)$ ein

Man erhält somit eine zentrale Datenbank, auf die genauso wie auf die lokalen Datenbanken zugegriffen werden kann. Will ein Autor einer Institution X ein Lernobjekt der Institution Y anfordern, das er in der zentralen Datenbank gefunden hat, ist diese Anforderungen aufgrund der *matching*-Tabelle problemlos möglich.

Eine exaktere Beschreibung des Skriptes ist im Anhang D.4, Seite 180 zu finden.

6.4.6 Suche in der Datenbank

Zur Suche in der Lernobjekt-Datenbank wurde ein HTML-Formular generiert (Abbildung 32). Für die am häufigsten verwendeten Suchkriterien, Diagnose und Lokalisation, steht eine Speedtyping-Funktion zur Verfügung: Ab einer Eingabelänge von zwei Zeichen werden bei jedem Tastendruck in einem der beiden Eingabefelder in der Datenbank die Diagnosen bzw. Lokalisationen ermittelt, die noch zum angegebenen Präfix passen und im rechten Frame angezeigt. Für andere Datenfelder, für die Codetabellen mit relativ wenigen Einträgen zur Verfügung stehen (zum Beispiel *Farbe*), wird das Suchformular dynamisch erstellt, d.h. in den DropDown-Boxen werden jeweils die in der Tabelle eingetragenen möglichen Werte angeboten.

Geben Sie in einem der folgenden Felder den Anfang der Codes bzw. Beschreibungen ein. Im rechten Frame werden dann die passenden Einträge aus der Datenbank angezeigt.

Diagnose (ICD9/10/Beschr):

Lokalisation:

- [Alopecia areata fere totalis](#)
- [Alopecia areata o. n. A.](#)
- [Alopecia totalis o. n. A.](#)
- [Alopecia areata_Ophiasistyp](#)
- [Alopecia areata universalis](#)
- [Alopecia areata diffusa](#)
- [Alopecia diffusa o. n. A.](#)
- [Alopecia androgenetica o. n.](#)

Lernobjekt-Typ:

Primäreffloreszenz:

Sekundäreffloreszenz:

andere Effloreszenzen:

Farbe:

Form:

Verteilung:

Daten zu Thumbnails anzeigen

Präfix-Suche (Diagnose, IC)

Ihre Suche lieferte folgende 3 Treffer in der Datenbank:

	Diagnose(n): - Alopecia diffusa o. n. A.	Effloreszenzen: Prim.: - Sek.: - Andere: Alopezie (Alopecia)	Farbe: hautfarben Form: bizarr Verteilung: in behaarten Arealen
	Diagnose(n): - Alopecia diffusa o. n. A.	Effloreszenzen: Prim.: - Sek.: Schuppe (Squama) Andere: Alopezie (Alopecia)	Farbe: gelblich Form: bizarr Verteilung: in behaarten Arealen
	Diagnose(n): - Alopecia diffusa o. n. A. - Arzneimittel, unerwünschte Nebenwirkung	Effloreszenzen: Prim.: - Sek.: - Andere: Alopezie (Alopecia)	Farbe: hautfarben Form: bizarr Verteilung: in behaarten Arealen

Abbildung 32: Suche in der Lernobjekt-Datenbank.

6.5 Autorenwerkzeuge zur Erstellung von Lernobjekten

Die Erstellung von Lernobjekten ist vom jeweiligen Lernobjekt-Typ abhängig. Für einige Lernobjekt-Typen existieren bereits bekannte, teilweise kommerzielle Erstellungstools. Diese werden dann verwendet, wenn sie entweder alle Anforderungen erfüllen oder wenn sie den Dozenten bereits vertraut im Umgang sind. Zum Beispiel werden die unter den Dozenten weit verbreiteten *Microsoft Office Tools* wie *Powerpoint* oder *Word* eingesetzt, auch wenn einige gewünschte Funktionen fehlen oder das Einfügen in die Lernobjekt-

Datenbank nur mit einer Nachbearbeitung möglich ist. So fällt es Dozenten leichter, das System zu benutzen, weil die Hürde, sich in ein neues Programm einzuarbeiten zu müssen, wegfällt.

Abbildung 33 zeigt die Erstellung und das Einfügen von Lernobjekten in die Datenbank. Als zentrales Werkzeug dient das Tool *DEJAVUCapt*. Mit ihm werden die durch andere Werkzeuge vorbereiteten Lernobjekte in die Datenbank eingetragen. Eine Ausnahme bilden nur die Quizelemente. Hier greifen Perl-CGIs direkt bei der Erstellung auf die Datenbank zu. In den folgenden Abschnitten wird zunächst *DEJAVUCapt* und dann der Ablauf der Erstellung für die einzelnen Lernobjekt-Typen detailliert beschrieben.

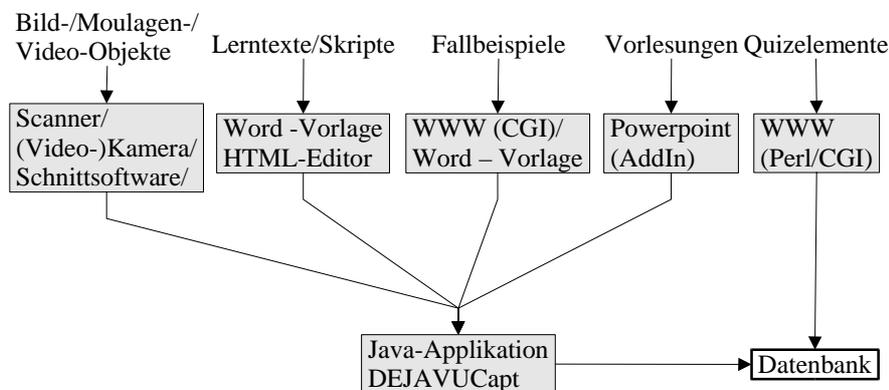


Abbildung 33: Autorenwerkzeuge. Übersicht.

6.5.1 DEJAVUCapt – Einfügen von Lernobjekten in die Datenbank

Zur Verschlagwortung der Lernobjekte und Einfügen in die Datenbank wurde *DEJAVUCapt*, eine Java-Applikation mit direkter Datenbankbindung über JDBC implementiert. Nach der Verbindung zur Datenbank und der Identifikation des Nutzers durch seine Datenbankzugangsdaten, stehen die folgenden Funktionen zur Verfügung:

(1) Hinzufügen von Lernobjekten

Es können Lernobjekte, die in einem beliebigen Unterverzeichnis des User-Objektpfads liegen, eingefügt werden. Die Objekte stehen anderen Nutzern allerdings erst dann zur Verfügung, wenn sie durch die *Upload*-Funktion der Applikation hochgeladen werden. Dabei werden Bilder als JPGs in zwei verschiedenen Auflösungen (als Vorschaubild mit kleiner und Großbild in monitorfüllender Auflösung) und Videos (ebenfalls mit Vorschaubild, siehe (2)) inklusive ihrer Unterverzeichnisstruktur vom User-Objektpfad in den Server-Objektpfad in das Verzeichnis des Datenbanknutzers kopiert. Auf dem

Netzlaufwerk werden also alle Lernobjekte eines Nutzers in einem Unterverzeichnis gespeichert. Sollen zum Beispiel Objekte aufgenommen werden, die dem Institut gehören, kann ein Datenbanknutzer mit dem Namen des Instituts angelegt werden. Bei anderen Lernobjekt-Typen besteht der Upload in einem einfachen Kopieren des Objekts.

Zur Veranschaulichung der Dateioperationen beim Upload von Bildern und Videos diene das folgende Beispiel:

```
User-Objektpfad: c:\input\  
Server-Objektpfad: s:\images,  
Nutzer: Holger Höhn, Datenbank-Id: 13  
Aktion: Hochladen des  
Bildes c:\input\beispiel1\bsp.tif  
Videos c:\input\video\videol.avi  
Dadurch werden folgende Dateien im Server-Objektpfad erzeugt:  
s:\images\13\beispiel\bsp.jpg  
s:\images\13\beispiel\bspt.jpg  
s:\images\13\video\videol.avi  
s:\images\13\video\videolt.jpg
```

(2) Erstellen eines Vorschaubildes für Video-Objekte

Für Video-Objekte ist das Erstellen eines repräsentativen Vorschaubildes nicht so einfach wie für Bilder.

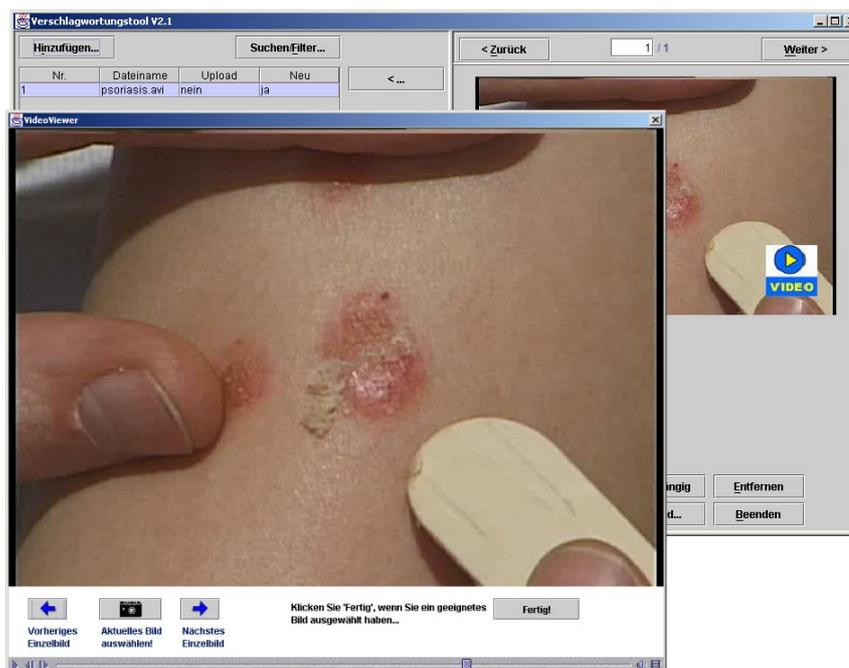


Abbildung 34: Festlegung eines Bildes als Vorschaubild für ein Video

Man kann nicht einfach das erste Bild des Films verwenden: Es ist schwarz, wenn zu Beginn erst aufgeblendet wird oder stellt keine aussagekräftige Szene dar. Deshalb entscheidet der Verschlagworter selbst, welches Einzelbild den Inhalt des Videos repräsentativ genug darstellt. Dazu kann ein Video in *DEJAVUCapt* abgespielt und an jeder beliebigen Stelle angehalten werden (Abbildung 34). Zur exakteren Positionierung ist das Blättern in Einzelbildschritten möglich. Vom jeweils aktuellen Bild kann dann ein Snapshot erstellt werden, der als Vorschaubild verwendet wird. In den Snapshot wird automatisch ein kleines Logo eingeblendet, das kennzeichnet, daß es sich bei dem Lernobjekt um ein Video handelt.

(3) Anonymisierung von Bildobjekten

Solange das Bild lokal im User-Objektpfad vorhanden ist, können Bildbereiche, die es erlauben, Patienten zu identifizieren, anonymisiert, d.h. mit schwarzen Rechtecken abgedeckt werden. Dazu wird das Bild durch Anklicken vergrößert. In dieser Darstellung werden mit der Maus Rechtecke aufgezogen. Die Koordinaten dieser Rechtecke werden in der Datenbank vermerkt. Erst beim Upload des Bildes werden dann tatsächlich schwarze Rechtecke in die Bilder eingefügt.

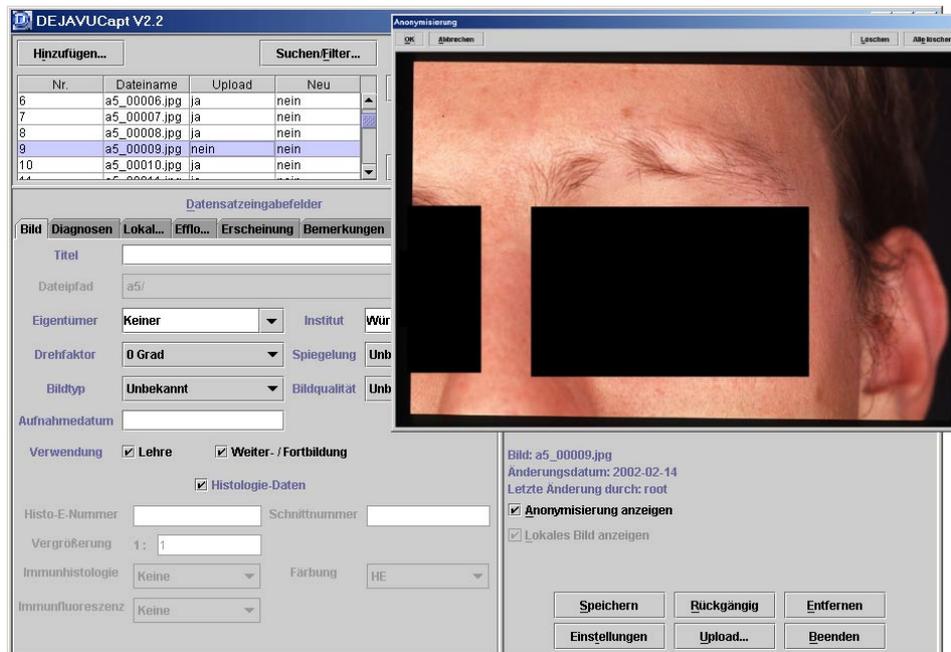


Abbildung 35: DEJAVUCapt. Verschlagwortung/Anonymisierung.

Bemerkung: Das Einfügen von Anonymisierungsrechtecken ist nur dann möglich, wenn das gerade verschlagwortete Bild auch im lokalen User-Objektpfad (in möglichst guter Qualität, am besten verlustfrei) vorhanden ist. Der Grund ist, daß die Rechtecke erst beim Upload des Bildes tatsächlich ins Bild eingefügt werden. Würde man die Rechtecke jeweils in die JPEG-Versionen auf dem Server-Objektpfad einfügen, würde sich die Qualität der Bilder mit jeder dadurch notwendigen JPEG-Neukomprimierung sukzessive verschlechtern.

(4) Verschlagworten und Verwalten der Objekte

Mit den Karteikarten (Abbildung 35, links unten) können die verschiedenen Eigenschaften von Objekten erfaßt bzw. editiert werden. Wo immer möglich, werden dem Nutzer Auswahllisten angeboten, um die Möglichkeit von Tippfehlern zu vermeiden und die Bearbeitungsgeschwindigkeit zu erhöhen. Beim nachträglichen Editieren von Datensätzen wird der Nutzer durch Such- bzw. Filtermöglichkeiten unterstützt, er kann hier zum Beispiel Objekte nach bestimmten Filenamen oder zugeordneten Diagnosen finden.

Zur Konfiguration von *DEJAVUCapt* steht eine eigene Oberfläche zur Verfügung (Abbildung 36). Hier können je nach Wunsch einzelne Karteikarten deaktiviert werden, wenn die entsprechenden Datenfelder nicht erfaßt werden sollen. Außerdem können aus den einzelnen Picklisten Untermengen ausgewählt werden.

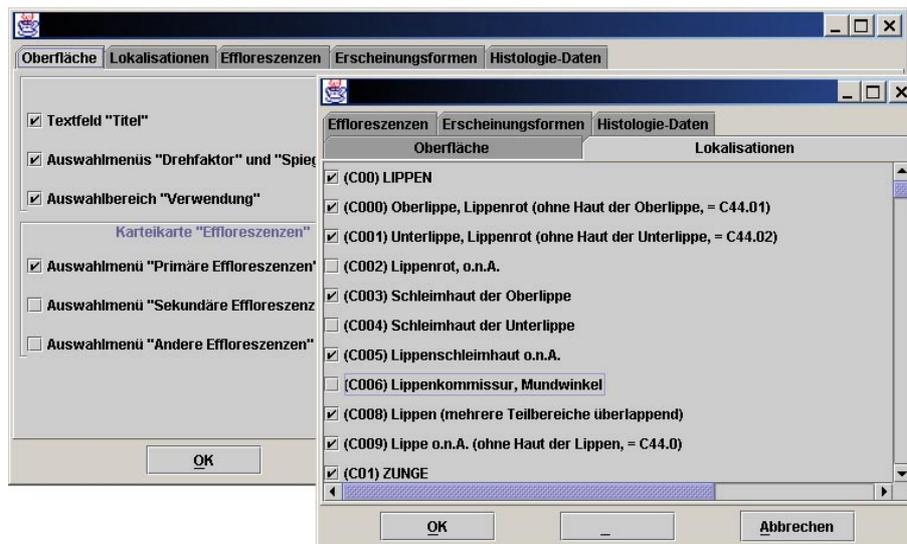


Abbildung 36: Konfiguration von *DEJAVUCapt*

Dadurch wird die Verschlagwortung zum einen übersichtlicher, zum anderen nicht zu feinkörnig, was bei der späteren Recherche ungünstig sein kann. Schließlich können so auch einheitliche klinikinterne Untermengen der Listen festgelegt werden.

Eine Beschreibung der Konfiguration und der Klassen ist im Anhang E.2.1, Seite 182 zu finden.

6.5.2 Nachverschlagwortung von Attributen

Die Anwendung von *DEJAVUCapt* liegt hauptsächlich bei der Integration neuer Objekte und beim Editieren einzelner Objekte. Im Gegensatz dazu kann es aber auch sinnvoll sein, eine Eigenschaft ganzer Gruppen von Lernobjekten zu editieren oder zu verschlagworten. So kann man eventuell nur in Abhängigkeit von anderen Bildern entscheiden, ob ein Objekt eher für Studenten oder Ärzte geeignet ist. Daneben kann die Geschwindigkeit der Verschlagwortung erheblich erhöht werden, wenn man nur eine bestimmte Eigenschaft von Objekten editieren will. Deshalb wurden zusätzlich Perl-CGIs implementiert, die das Editieren einer Eigenschaft von vielen Bildern innerhalb einer einzigen HTML-Seite erlauben. Die Datenbankbindung erfolgt hierbei über Perl-DBI/DBD. Die Umsetzung erfolgte zum Beispiel bei der Verschlagwortung der Zielgruppe und der Bildqualität.

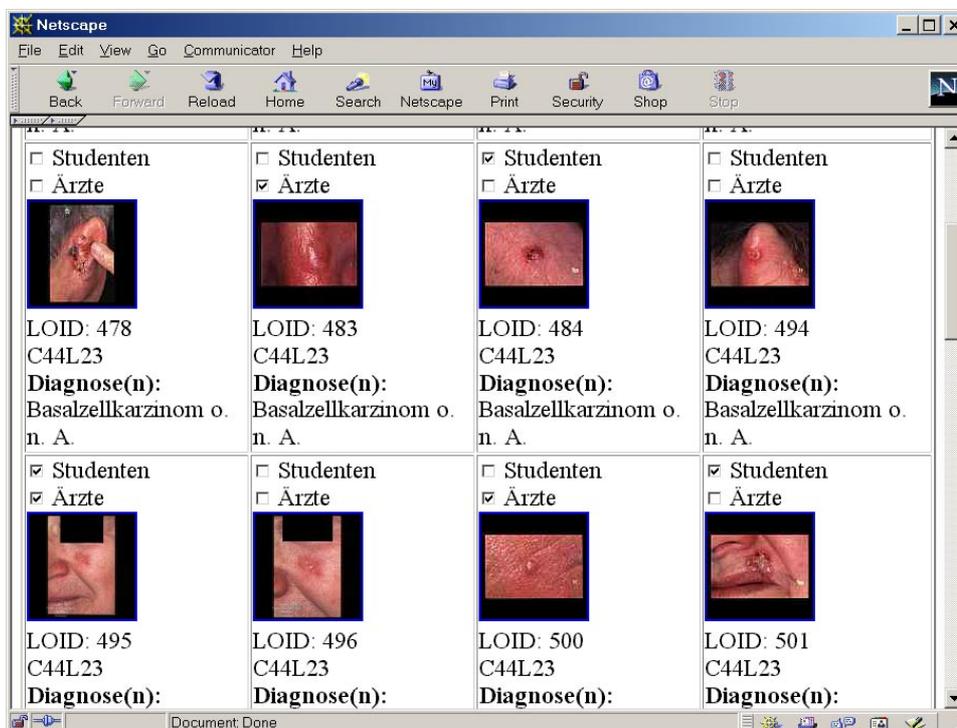


Abbildung 37: Nachverschlagwortung, hier am Beispiel Zielgruppe

Die Bildobjekte werden zunächst nach einem Kriterium sortiert (in der Regel ist das der DDK-Diagnosen-Code). Eine feste, aber konfigurierbare Anzahl N von Bildern wird dann auf jeweils einer HTML-Seite in Formularform angezeigt (Abbildung 37). Sind in der Datenbank bereits Daten eingetragen, werden die Formularfelder entsprechend vorbelegt. Hat der Nutzer die Einträge für die HTML-Seite vervollständigt, werden die Daten gespeichert und die nächsten N Objekte angezeigt.

Bemerkung: Der Nutzer muß in der aktuellen Sortierung nicht bei Objekt 1 mit der Verschlagwortung beginnen, es wird eine Link-Übersicht über alle HTML-Seiten generiert. Auf diese Weise kann die Arbeit unterbrochen und an derselben Stelle wieder aufgenommen werden.

Eine exaktere Beschreibung der Konfiguration und der Perl-Skripte ist im Anhang E.2.2, Seite 184 zu finden.

6.5.3 Bild- und Videoobjekte

Bildobjekte werden in der Regel durch Einscannen von Dias generiert (siehe Abschnitt 5.2.2). Die Videos werden mit einer digitalen Videokamera aufgenommen, über eine Firewire-Schnittstelle in einen PC eingelesen, geschnitten und in ein geeignetes Videoformat (z.B. DivX konvertiert). Zusätzlich wird *AviSynth* [29] verwendet, um Szenen des Videos mit Untertiteln zu versehen. *AviSynth* bietet den Vorteil, daß die Untertitel nicht direkt in das Video als Pixeldaten gesetzt werden. Die Texte werden vielmehr zur Abspielzeit eingeblendet. Dadurch kann das Video zum Beispiel ohne Untertitel in einer Vorlesung eingesetzt werden, das gleiche Video aber für die Studenten zur Nachbereitung mit erklärenden Texten angeboten werden.

6.5.4 Moulagen

Um einen 3-dimensionalen Eindruck einer Moulage am Bildschirm vermitteln zu können, muß sie von verschiedenen Seiten aufgenommen werden. Je mehr Bilder vom Objekt aufgenommen werden, desto realitätsnäher kann das Objekt am Bildschirm in verschiedene Richtungen gedreht werden. Die in Abbildung 38 skizzierte Konstruktion gewährleistet, daß die Bilder jeweils vom gleichen Standpunkt und aus der gleichen Entfernung aufgenommen werden. Zur Aufnahme wird eine Digitalkamera verwendet, die über die USB-Schnittstelle von einem PC aus gesteuert wird. Auch der Drehteller wird mit einem Schritt-

motor vom PC aus über die parallele Schnittstelle gedreht. Die zweite Drehrichtung wird im Moment per Hand verändert, die entwickelte Software kann aber auch sie steuern, wenn ein zweiter Motor vorhanden ist. Der Ersteller kann vor der Aufnahme die Anzahl der Bilder pro Achse sowie die gewünschte Auflösung der Bilder einstellen. Danach werden automatisch alle Bilder geschossen. Ausnahme: Ist für eine Drehachse kein Schrittmotor vorhanden, fordert die Software den Nutzer auf, den Drehwinkel zu verändern. Nach Erstellen aller Aufnahmen, können zusätzlich HotSpots auf einzelnen Bilder einer Serie eingefügt werden. HotSpots sind ausgezeichnete Regionen (Rechtecke) im Bild, die mit Links versehen werden. Die Links können dann insbesondere auch auf die 3D-Ansicht eines vergrößerten Ausschnitts der Moulage verweisen.

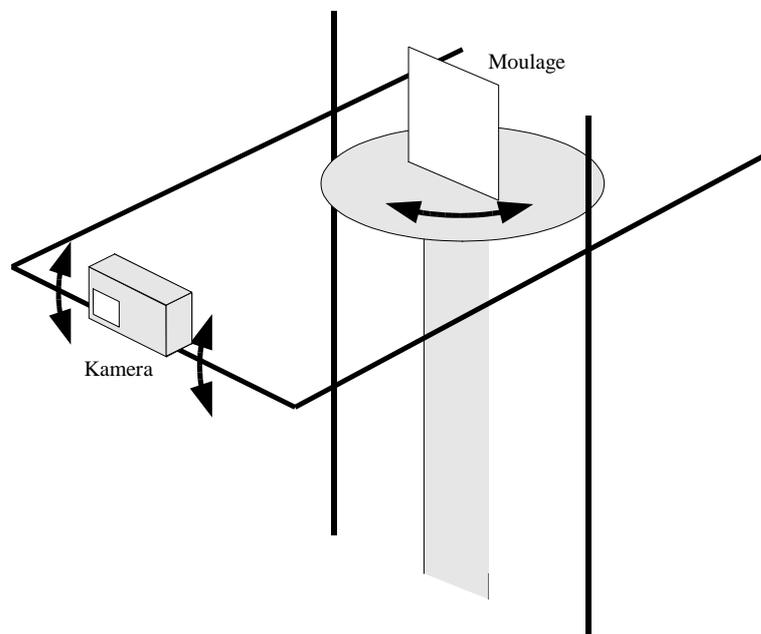


Abbildung 38: Aufnahme von Moulagen zur 3D-Darstellung

6.5.5 Präsenzvorlesungen

Zur einfachen und komfortablen Erstellung von Präsenzvorlesungen enthält das Paket ein AddIn für Powerpoint. Über das in *VisualBasic for Applications* implementierte Tool werden nach einer entsprechenden Recherche in der Lernobjekt-Datenbank Bilder und Videos in Powerpoint-Präsentationen eingefügt. Dazu stehen dem Dozenten ein Standardsuche-Dialog und ein Dialog zur erweiterten Suche zur Verfügung. Die Standardsuche (Abbildung 39, links) deckt den am meisten verwendeten Anfragemodus, nämlich die Suche nach Dia-

gnosen, ab. Dabei können die Diagnosen sowohl über ICD9/10-Codes als auch über ihre Bezeichnung eingegeben werden. Bei jedem Tastendruck im Eingabefeld wird in der Datenbank nachgefragt, welche Diagnosen zum aktuellen Präfix passen. Diese werden in der oberen Auswahlbox gelistet. Es können nun entweder Diagnosen dieser Liste über den Button *Diagnosen hinzufügen* zur Anfrage hinzugefügt werden oder es kann eine Präfixsuche gewählt werden, d.h. es werden alle Bilder/Videos gefunden, denen eine Diagnose zugeordnet wurde, die mit dem angegebenen Präfix beginnt.

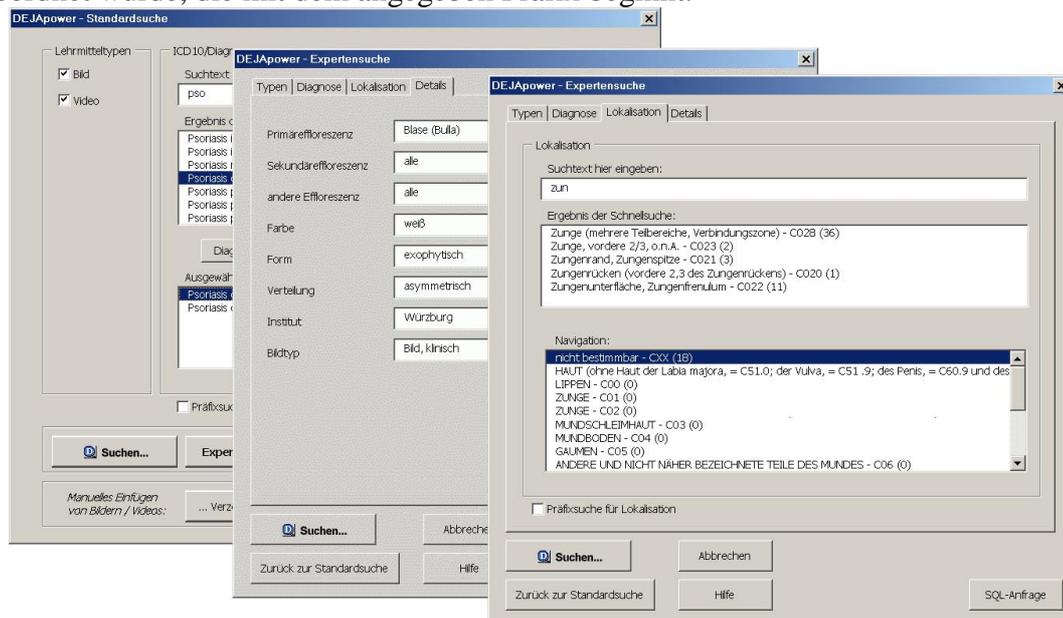


Abbildung 39: Powerpoint-AddIn: Standardsuche und erweiterte Suche

Schließlich kann die Anfrage auf einen Lernobjekt-Typ (Bild oder Video) eingeschränkt werden. In der Expertensuche (Abbildung 39, Mitte und rechts) kann eine Auswahl aus den übrigen erfaßten Datenfeldern getroffen werden. Die Eingaben bzw. Auswahlen werden zu einer Suchanfrage *und*-verknüpft.

Die zur Anfrage gefundenen Objekte werden in einer Vorschau-Übersicht präsentiert (Abbildung 40, links). Durch Anklicken können sie in Großdarstellung angezeigt werden, wobei auch das Abspielen von Videos möglich ist (Abbildung 40, Mitte). Zum Einfügen der Objekte in die aktuelle Powerpoint-Präsentation stehen neben dem Einfügen als neue Folie auch einige Layout-Vorlagen zur Verfügung (Abbildung 40, rechts):

- *Standard*: das Objekt wird auf einer neuen Folie zentriert eingefügt

- *mit Datenbankdaten:* es wird links ein Textfeld mit den Daten aus der Datenbank eingefügt, das Objekt rechts daneben
- *mit leerem Textfeld:* links wird ein leeres Textfeld, daneben das Objekt platziert.
- *horizontale/vertikale Doppelprojektion:* Bei Wahl dieses Layouts wird das aktuelle Objekt auf eine neue Folie eingefügt, das nächste Objekt, das eingefügt wird, wird dann automatisch darunter/daneben platziert. Dies entspricht der in den klassischen Dermatologie-Vorlesungen häufig verwendeten Doppelprojektion (Einsatz zweier Diaprojektoren) zum Vergleich zweier Bilder.

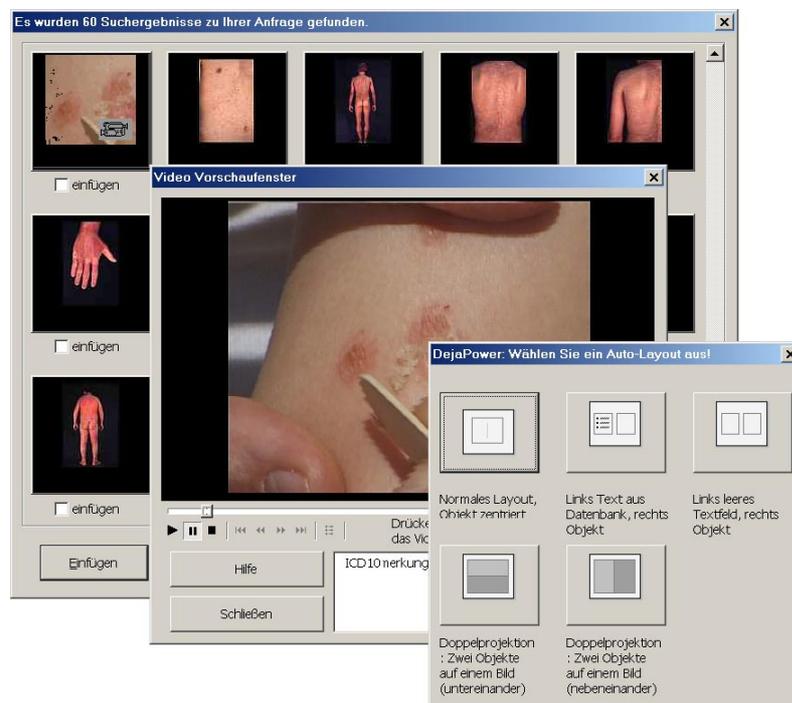


Abbildung 40: Powerpoint-AddIn: Suchergebnis und Einfügen

Der Ersteller hat außerdem die Wahl, ob er seinen Präsentationshintergrund beibehalten will oder ob das Objekt auf einem schwarzen Hintergrund eingefügt wird.

Wird ein Lernobjekt eingefügt, so wird die Lernobjekt-ID automatisch als Foliennotiz abgelegt. Mit Hilfe dieser Notizen kann ein zweites AddIn aus einer so erstellten Präsentation eine Textdatei exportieren, in der vermerkt ist, welches Lernobjekt (welche Lernobjekt-ID) auf welcher Folie eingesetzt wurde. Diese Textdatei dient dann dazu, dem Studenten **Vorlesungen zur Nachbereitung** (siehe Abschnitt 6.6.5) anzubieten. Die folgende Abbildung 41 veranschaulicht den Ablauf:

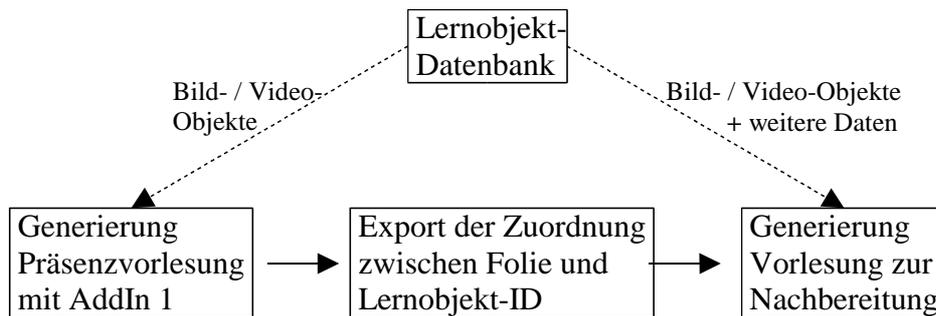


Abbildung 41: Vorlesung zur Nachbereitung

Da die jeweiligen Lernobjekt-IDs bekannt sind, können dabei alle aus der Datenbank bekannten Informationen eingebracht werden. Zum Beispiel kann auf jedes Objekt ein Link zu einer Zusammenfassung aller Informationen oder zum entsprechenden Knoten im semantischen Netz gelegt werden.

Eine Beschreibung der Konfiguration und der Implementierung ist im Anhang E.2.3, Seite 185 zu finden.

6.5.6 Lerntexte

Zusammen mit den beteiligten Dermatologen wurde eine flache, in die folgende DTD umgesetzte Struktur für Lerntexte erarbeitet:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- DTD zur Beschreibung von Lerntexten im DEJAVU-Projekt -->
<!-- Holger Hoehn, 2001-10-24 -->
<!-- einfache Elemente -->
<!ELEMENT title (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT synonym (#PCDATA)>
<!ELEMENT definition (#PCDATA)>
<!ELEMENT zusammenfassung (#PCDATA)>
<!ELEMENT epidem (#PCDATA)>
<!ELEMENT patho (#PCDATA)>
<!ELEMENT klinik (#PCDATA)>
<!ELEMENT sonder (#PCDATA)>
<!ELEMENT kompli (#PCDATA)>
<!ELEMENT histo (#PCDATA)>
<!ELEMENT diagnostik_cont (#PCDATA)>
<!ELEMENT dif (#PCDATA)>
<!ELEMENT labor (#PCDATA)>
<!ELEMENT erreger (#PCDATA)>
<!ELEMENT bilddiagnostik (#PCDATA)>
<!ELEMENT labandere (#PCDATA)>
<!ELEMENT diffdiag (#PCDATA)>
<!ELEMENT therapie_cont (#PCDATA)>
<!ELEMENT topisch (#PCDATA)>
  
```

```

<!ELEMENT syst (#PCDATA)>
<!ELEMENT bes (#PCDATA)>
<!ELEMENT verlauf (#PCDATA)>
<!ELEMENT literatur (#PCDATA)>

<!--Elemente mit Elementen-->
<!ELEMENT diagnostik (diagnostik_cont?, histo, dif, labor,
    erreger, bilddiagnostik, labandere)>
<!ELEMENT therapie (therapie_cont?, topisch, syst,
    bes, verlauf?)>

<!--root - Element -->
<!ELEMENT lerntext (title, autor, synonym?, definition,
    zusammenfassung, epidem?, patho?, klinik, sonder?,
    kompli?, diagnostik, diffdiag?, therapie?,
    verlauf?, literatur?)>

```

In der Datenbank werden Verweise auf zu dieser DTD gültige XML-Dateien verwaltet. Die folgende Grafik stellt den gesamten Ablauf von der Erstellung in Word bis zum Einfügen in die Datenbank dar:



In Lerntexten kommen zum Beispiel Fließtexte, Aufzählungen, Bildmaterial vor, auch auf Formatierungen wie Fettdruck etc. soll nicht verzichtet werden. Da Autoren in der Regel mit der Verwendung von Word vertraut sind, wird dies zur Eingabe der Lerntexte verwendet (1). Um ein Mindestmaß an Einheitlichkeit zu erreichen, wurde eine Word-Vorlage mit der Lerntext-Struktur erstellt (Abbildung 42). Die Einbindung von Bild- und Videoobjekten in Lerntexte erfolgt analog zur Generierung von Präsenzvorlesungen mit einem AddIn. Der Autor kann so komfortabel aus Word eine Datenbankanfrage starten und die entsprechenden Objekte in Kleindarstellung direkt in Word integrieren. Dabei wird die Lernobjekt-ID versteckt als alternativer Text eingefügt, der auch beim Export nach HTML erhalten bleibt. Der Lerntext wird aus Word nach HTML exportiert (2). Allerdings ist der defaultmäßig in Word vorhandene HTML-Export kaum zu verwenden, da das produzierte HTML-Dokument unnötig viele Tags enthält. Alternativ wird das von Microsoft kostenlos downloadbare Plugin *Export to Compact HTML* [52] eingesetzt. In Schritt (3) werden durch ein Perl-Skript unnötige Tags entfernt und die Konvertierung von HTML nach XHTML erfolgt mit Hilfe des Tools *Tidy* [66] des W3C. Das generierte XHTML-File wird durch ein weiteres Perl-Skript nach XML konvertiert (4). Dabei werden lediglich die Abschnitte entsprechend der DTD getagt, die Inhalte der einzelnen Bereiche werden als CDATA eingebunden. Auf

diese Weise werden die vorhandenen (X)HTML-Formatierungen einfach übernommen. Im folgenden Beispiel der Abschnitt *diagnostik*:

```
<diagnostik>
  <![CDATA[ <ul><li><i>klassische Form</i> mit typischer
    Klinik; ...</li>.
  <li>Probeexzision mit Histologie und direkter
    Immunfluoreszenz </li>
  <li>Suche nach Ausl&ouml;sern (insbesondere ...</li></ul>
  ]]>
</diagnostik>
```

Eine exaktere Beschreibung der Konfiguration und der Konvertierung ist im Anhang E.2.4, Seite 186 zu finden.

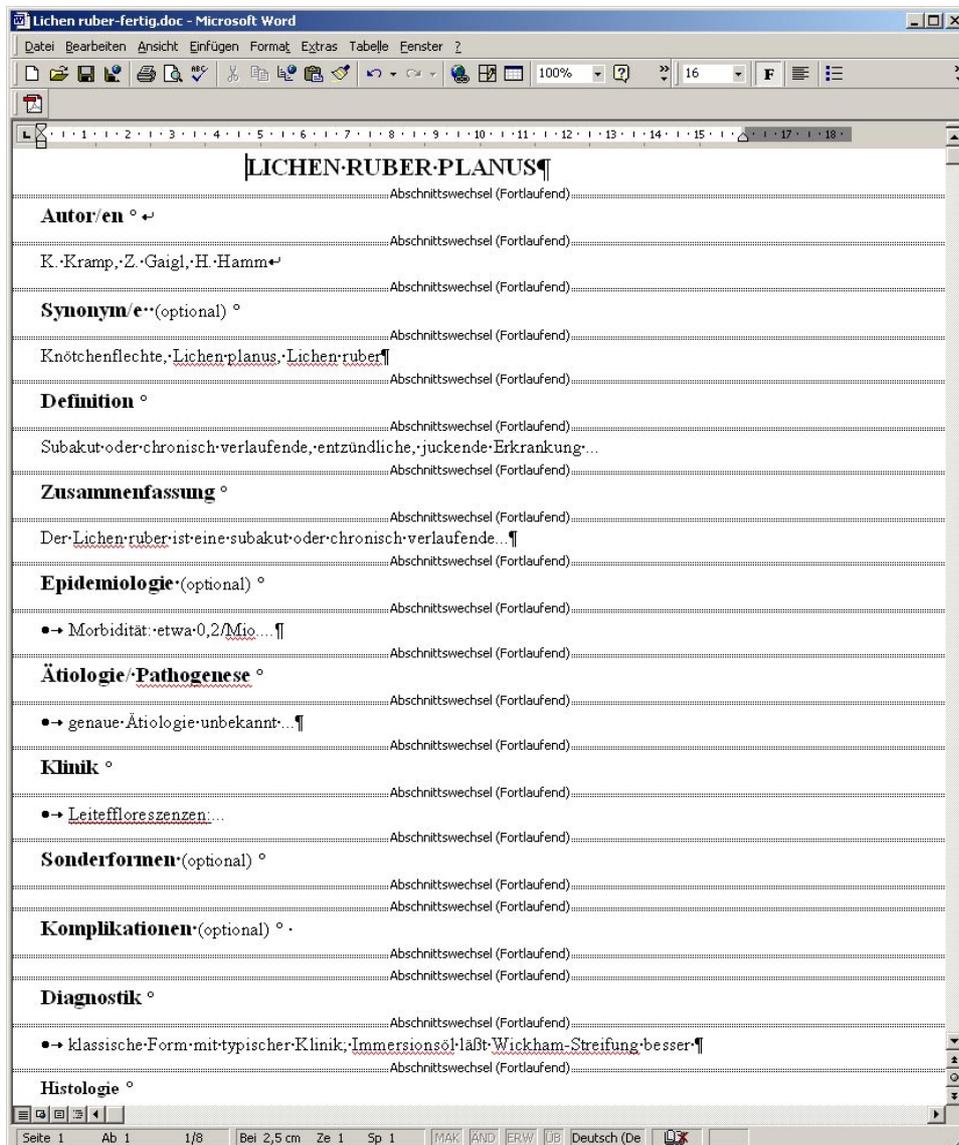


Abbildung 42: World-Vorlage zur Erfassung von Lerntexten.

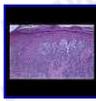
Krankheitsbezeichnung: Lichen ruber Autor: Autor3 Anamnese Bei dem 48-jährigen Patienten t erstmals stark juckende Papeln Penicillineinnahme über 1 Woche gebessert. Seit 1 Monat würden Hautbefund bei Aufnahme Bei Aufnahme finden sich an der sakral zahlreiche rot-livide, Mundschleimhaut sind weißliche Streifen zu sehen. Nägel und betroffen. Diagnostik Auf eine weitere Probebiopsie wird verzichtet. Auswärtiger Befund: Hyperorthokeratose. Epidermis z.T. plump akanthotisch. Subepidermal bandförmiges entzündliches Infiltrat, vorwiegend Lymphozyten. Deutliche Zerstörung des Basalzellagers z.T. Therapie Es wird eine Bade-PUVA-Therapie eingeleitet, die 4x/Woche d Unter dieser Behandlung sistiert der Juckreiz rasch. Die Ha insbesondere an den Handgelenken, heilen jedoch nur sehr z diesen Es wird eine Bade-PUVA- Therapie eingeleitet, die 4x/ Diskussion Bei Entlassung sind die Hautver änderungen deutlich abgeflac mehr. Die Schleimhautveränderun gen sind abgeheilt. Die Bade ambulant fortgesetzt. Ein mit dem Patienten vereinbarter Wiedervorstellungstermin wird nicht wahrgenommen. <input type="button" value="Speichern"/>	Fallberichte <table border="1"> <thead> <tr> <th>Autor</th> <th>Bezeichnung</th> <th>Aktion</th> </tr> </thead> <tbody> <tr> <td>Autor1:</td> <td>Scabies</td> <td>[Editieren] [HTML] [Löschen] [Bilder]</td> </tr> <tr> <td>Autor2:</td> <td>Scabies</td> <td>[Editieren] [HTML] [Löschen] [Bilder]</td> </tr> <tr> <td>Autor3:</td> <td>Lichen ruber plaus</td> <td>[Editieren] [HTML] [Löschen] [Bilder]</td> </tr> </tbody> </table> Neuer Fallbericht ... Startseite	Autor	Bezeichnung	Aktion	Autor1:	Scabies	[Editieren] [HTML] [Löschen] [Bilder]	Autor2:	Scabies	[Editieren] [HTML] [Löschen] [Bilder]	Autor3:	Lichen ruber plaus	[Editieren] [HTML] [Löschen] [Bilder]	Zugeordnete Bilder:  aus Fallbericht entfernen!  aus Fallbericht entfernen!  aus Fallbericht entfernen! Bilder zuordnen Startseite Fallberichte
Autor	Bezeichnung	Aktion												
Autor1:	Scabies	[Editieren] [HTML] [Löschen] [Bilder]												
Autor2:	Scabies	[Editieren] [HTML] [Löschen] [Bilder]												
Autor3:	Lichen ruber plaus	[Editieren] [HTML] [Löschen] [Bilder]												

Abbildung 43: Erstellung von Fallbeispielen.

Das Einfügen des Lerntextes in die Datenbank (5) erfolgt mit dem in Abschnitt 6.5.1 vorgestellten *DEJAVUCapt*. Beim Hinzufügen kann hier als Lernobjekt-Typ *Lerntext* ausgewählt werden. Danach kann der Lerntext, zum Beispiel mit Diagnosen, verschlagwortet werden und analog zu Bild- und Videoobjekten auf den Server-Objektpfad hochgeladen werden. Als Vorschaubild wird in diesem Fall nur *Lerntext* angezeigt, auf Klick öffnet sich ein Fenster in dem der Lerntext mit Hilfe von XSL-Stylesheets nach HTML konvertiert angezeigt wird.

6.5.7 Fallbeispiele

Auch für Fallbeispiele wurde eine einfache DTD erstellt:

```
<!ELEMENT krankheitsbezeichnung (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT anamnese (#PCDATA)>
<!ELEMENT aufnahmebefund (#PCDATA)>
<!ELEMENT diagnostik (#PCDATA)>
<!ELEMENT therapie (#PCDATA)>
<!ELEMENT entlassungsbefund (#PCDATA)>

<!-- ***** root - Element -->
<!ELEMENT fallbericht (autor, anamnese, aufnahmebefund,
    diagnostik, therapie, entlassungsbefund)>
```

Im Gegensatz zu Lerntexten bestehen Fallberichte überwiegend aus Fließtext. Neben der Erfassung von Fallberichten mit einer Vorlage – ähnlich zur Erfassung von Lerntexten – kann die Eingabe alternativ browsergestützt erfolgen. Dazu wird eine Übersicht über alle Fallbeispiele generiert (Abbildung 43, rechts oben). Der jeweils gewählte Fall kann dann editiert werden. Mit Hilfe einer Suchanfrage können Bilder zugeordnet werden (Abbildung 43, rechts unten). CGI-Skripte erzeugen dann aus den eingegebenen Daten HTML-Dateien. Die Weiterverarbeitung (Konvertierung nach XHTML (3), XML (4) und Einfügen in die Datenbank (5)) erfolgt analog zur Verarbeitungskette bei Lerntexten.

6.5.8 Quizelemente

6.5.8.1 Bilderquiz

Das Bilderquiz der aktuellen Form (Abschnitt 6.6.8) erfordert kein eigenes Autorenwerkzeug. Es werden lediglich in zufälliger Reihenfolge Bildobjekte präsentiert, zu der der Lernende aus einer vorgegebenen Auswahl die richtige Diagnose auswählen muß. Diese Informationen sind aber bereits durch das Verschlagworten der Bildobjekte vorhanden. In einer zukünftigen Form des Bilderquiz könnte der Autor zum Beispiel die Bilder oder Antwortalternativen auswählen, die dem Lernenden angezeigt werden.

6.5.8.2 Übungsfragen

Die Übungsfragen beschränken sich auf Multiple Choice – Fragen, die verschiedenen Themengebieten zuordenbar sind. Neben dem *Fragentext* und dem *Themengebiet* sind die *möglichen Antworten*, die *korrekte(n) Antwort(e)n*, evtl. zum Fragentext gehörende *Bilder* und eine *Erläuterung* zu erfassen. Die Erläuterung soll dem Lernenden weitere Informationen dazu geben, warum die jeweilige Antwort korrekt ist oder nicht. Die Erfassung der Fragen geschieht in einem Standard-WWW-Browser mit Perl/CGI Webseiten (Abbildung 44).

In den Eingabefeldern der linken Seite kann jeweils HTML-Text inklusive Tags eingegeben werden, rechts wird dann der bereits formatierte Text angezeigt. In jedem Schritt kann eine zusätzliche Antwortalternative hinzugefügt werden. Das Einfügen der Daten in die Datenbank erfolgt direkt über die Perl-CGI-Skripte.

Eine Beschreibung der Konfiguration und der Implementierung ist im Anhang E.2.5, Seite 187 zu finden.

Fragestellung

Welche Aussage trifft nicht zuNULL
Eine 22jährige Patientin bemerkt seit einer Woche das Auftreten von druckschmerzhaften Hautveränderungen (siehe Abbildung). Sie hat Fieber bis zu 38 °C.
Folgende Maßnahmen sind für die Erkennung einer Grundkrankheit von Bedeutung:

Welche Aussage trifft nicht zuNULL
Eine 22jährige Patientin bemerkt seit einer Woche das Auftreten von druckschmerzhaften Hautveränderungen (siehe Abbildung). Sie hat Fieber bis zu 38 °C.
Folgende Maßnahmen sind für die Erkennung einer Grundkrankheit von Bedeutung:

Antworttext **korrekt? löschen**

A. Bestimmung des Antistreptolysintiters

B. Röntgenuntersuchung des Thorax

C. Thrombozytenzählung

F.

Erläuterung:

Die Abbildung zeigt ein Erythema nodosum. Gehäuft bei jungen Frauen bilden sich bevorzugt an den Unterschenkelstreckseiten schmerzhafte, münzgroße Knoten, die sich von rot über livide bis zu gelb-grün verfärben (wie ein Hämatom). Dabei befinden sich nicht alle im selben Stadium (s. Abbildung: unterschiedliche Färbung der Herde).
Es besteht eine medikamenteninduzierte (D) oder häufiger eine infektiologische Genese. V.a. bei der Sarkoidose (B), Tuberkulose (B) (C) und Streptokokkeninfektionen (A) kommt es zuder akuten Vasculitis im subcutanen Fettgewebe (Panniculitis) des E. nodosum.

Die Abbildung zeigt ein Erythema nodosum. Gehäuft bei jungen Frauen bilden sich bevorzugt an den Unterschenkelstreckseiten schmerzhafte, münzgroße Knoten, die sich von rot über livide bis zu gelb-grün verfärben (wie ein Hämatom). Dabei befinden sich nicht alle im selben Stadium (s. Abbildung: unterschiedliche Färbung der Herde).
Es besteht eine medikamenteninduzierte (D) oder häufiger eine infektiologische Genese. V.a. bei der Sarkoidose (B), Tuberkulose (B) (C) und Streptokokkeninfektionen (A) kommt es zuder akuten Vasculitis im subcutanen Fettgewebe (Panniculitis) des E. nodosum.

Abbildung 44: Eingabe von Übungsfragen in einem WWW-Browser

6.5.8.3 Fallquiz

Als Autorenwerkzeug für das Fallquiz wurden Perl-CGIs erstellt, die über DBI mit DBD::mysql mit der Datenbank kommunizieren. Das heißt: auch hier werden die Daten direkt in der Datenbank verwaltet, die Verwendung eines weiteren Tools ist nicht nötig.

Auswahl - Kurzanamnese - Bilder - Anamnese - Effloreszenzen - Diagnostik - Diagnose

Titel: Alopecia areata

Erstellender: Holger Höhn
(Geben Sie hier Ihren Namen ein.)

Kurzanamnese: Der Patient hat Haare an seinen Fingern.
(Kurze Anamnesebeschreibung, die später dem Studenten als erste Information zum Fall dient.)

Bemerkung:

Auswahl - Kurzanamnese - Bilder - Anamnese - Effloreszenzen - Diagnostik - Diagnose

Folgende Bilder sind bereits ausgewählt:

Entfernen Entfernen Entfernen

Geben Sie hier eine Diagnose ein, um nach weiteren Bildern zu suchen.

Diagnose: Alopecia areata

Abbildung 45: Fallquiz. Eingabe der Kurzanamnese und Auswahl der Bilder.

Der Dozent gibt bequem über den ihm vertrauten Webbrowser nach und nach die Daten zu *Kurzanamnese*, *Bildmaterial*, *Anamnese*, *Effloreszenzen*, *Diagnostik* und *Diagnose* ein.

Für die *Kurzanamnese* (Abbildung 45, links) wird ein Fließtext eingegeben, der später dem Lernenden das Bild eines Patienten im Krankenzimmer vermitteln soll. Zur weiteren Veranschaulichung können aus der Datenbank *Bildobjekte* über eine Datenbankanfrage zugeordnet werden (Abbildung 45, rechts)

Da während der *Anamnese* (Abbildung 46) bei unterschiedlichen Diagnosen (Fällen) häufig gleiche Fragen gestellt werden, werden diese ebenfalls in der Datenbank verwaltet. Ist eine Frage noch nicht vorhanden, kann sie neu aufgenommen werden.

Auswahl - [Kurzanamnese](#) - [Bilder](#) - **Anamnese** - [Effloreszenzen](#) - [Diagnostik](#) - [Diagnose](#)

Folgende Anamnesefragen gibt es bereits in der Datenbank:

Haben Sie in der Familie Personen, die auch unter Haarausfall leiden?	Neu ...
Nehmen Sie Medikamente ein?	
Fühlen Sie sich krank?	
Haben Sie Gelenk-, Kopf- und/oder Gliederschmerzen?	
Haben Sie manchmal kleine Bläschen an der Lippe?	

DieFrage ist in diesem Fall sinnvoll.

Patientenantwort:
(Diese Antwort erhält später der Student, wenn er diese Anamnesefrage stellt.)

Erläuterung:
(Diese Erläuterung soll dem Studenten als Hilfestellung dienen aus welchem Grund das Stellen dieser Frage sinnvoll/nichtsinvoll ist.)

Abbildung 46: Fallquiz. Eingabe der Anamnese.

Aus dem Fragenpool kann dann jeweils eine Frage gewählt werden, zu der dann festgelegt wird, ob es sinnvoll ist, diese Frage dem Patienten im vorliegenden Fall zu stellen. Ziel ist es, den Studenten darauf zu trainieren, möglichst nur die notwendigen Fragen zu stellen, um den Patienten nicht unnötig zu verängstigen. Zu jeder sinnvollen Frage muß natürlich auch die Antwort des Patienten eingegeben werden. Außerdem kann der Autor dann auch noch eine Erläuterung hinzufügen, die dem Lernenden weitere Hinweise gibt, warum die Frage in diesem Fall (nicht) sinnvoll ist. Während die Fragen des Pools selbst für alle Fälle zu Verfügung stehen, können die Antworten bzw. die Erläuterungen zur gleichen Frage selbstverständlich von Fall zu Fall unterschiedlich ausfallen.

Auswahl - Kurzanamnese - Bilder - Anamnese - **Effloreszenzen** - Diagnostik - Diagnose

Zu folgenden Bilder soll eine Effloreszenzauswahl festgelegt werden:



Anzeigen	Sinnvoll	Primäreffloreszenz
<input type="checkbox"/>	<input type="checkbox"/>	Blase (Bulla)
<input type="checkbox"/>	<input type="checkbox"/>	Bläschen (Vesicula)
<input type="checkbox"/>	<input type="checkbox"/>	Fleck (Macula)
<input type="checkbox"/>	<input type="checkbox"/>	Knoten (Nodus)
<input type="checkbox"/>	<input type="checkbox"/>	Papel (Papula)
<input type="checkbox"/>	<input type="checkbox"/>	Plaque

Abbildung 47: Fallquiz. Eingabe der Effloreszenzen.

Sie müssen in der Datenbank also separat gehalten werden.

Der nächste Schritt der Erstellung ist die Eingabe der *Effloreszenzen* (Abbildung 47). Dazu werden dem Autor die vorher gewählten Bilder angezeigt. Er muß dann entscheiden, welche Effloreszenzen aus der vorgegebenen Liste dem Lernenden angezeigt werden und welche davon richtig sind.

Auswahl - Kurzanamnese - Bilder - Anamnese - Effloreszenzen - **Diagnostik** - Diagnose

Es stehen folgende Diagnostische Verfahren zur Verfügung:

Anzeigen	Sinnvoll	Diagnostisches Verfahren	Erläuterung	Ergebnis
<input type="checkbox"/>	<input type="checkbox"/>	Blutuntersuchung		
<input type="checkbox"/>	<input type="checkbox"/>	EKG		
<input type="checkbox"/>	<input type="checkbox"/>	Abstrich		
<input type="checkbox"/>	<input type="checkbox"/>	Lungentest		
<input type="checkbox"/>	<input type="checkbox"/>	Allergietest		

Auswahl - Allgemein - Bilder - Anamnese - Effloreszenzen - Diagnostik - **Diagnose**

Wählen Sie aus folgender Diagnoseliste die gewünschten Diagnosen aus:

- Alopecia areata fere totalis
- Alopecia areata o. n. A.**
- Alopecia areata universalis
- Alopecia areata, Ophiasistyp
- Alopecia areata, sonstige
- Alopecia areolaris syphilitica
- Alopecia cranialis totalis
- Alopecia diffusa o. n. A.
- Alopecia mucinosa Pinkus
- Alopecia parvimaclata
- Alopecia seborrhoica
- Alopecia specifica
- Alopecia totalis o. n. A.
- Alopecia triangularis congenitalis
- Alopezie o. n. A.
- Alopezie, androgenetische mit Androgenüberschuß
- Alopezie, androgenetische vom männlichen Typ
- Alonezie androgenetische vom weiblichen Typ

Abbildung 48: Fallquiz. Auswahl der Diagnostik und der Diagnose.

Für die *Diagnostik* (Abbildung 48, links) wählt der Autor aus einer erweiterbaren Liste von diagnostischen Verfahren diejenigen aus, die dem Studenten angezeigt werden sollen. Ähnlich wie bei den Anamnesefragen wird erfaßt, ob das Verfahren im vorliegenden Fall sinnvoll ist oder nicht. Außerdem wird ggf. das Ergebnis des Verfahrens und eine Erläuterung (Begründung) dafür aufgenommen, warum das Verfahren (nicht) sinnvoll ist. Für die *Diagnose* (Abbildung 48, rechts) schließlich werden aus einer Liste von Diagnosen, nämlich den in der Datenbank gespeicherten DDK-Diagnosen, die Diagnosen ausgewählt, aus denen dann später der Student die richtige(n) Diagnose(n) finden soll.

Eine Beschreibung der Konfiguration und der Perl-Implementierung ist im Anhang E.2.6, Seite 188 zu finden.

6.6 Lernumgebung DEJAVULearn

6.6.1 Einleitung

Zunächst mußte entschieden werden, in welche Lernumgebung die Objekte integriert werden sollten. Trotz der Vielfalt der angebotenen Umgebungen (siehe Kapitel 3), fällt eine Entscheidung für bereits bestehende Umgebungen schwer:

- Die Auswahl ist fast unüberschaubar. Selbst bereits vorhandene Bewertungen ermöglichen keine schnelle Entscheidung für ein spezielles Produkt.
- Nicht für alle Produkte stehen Demoversionen zum Testen des Funktionsumfangs zur Verfügung.
- Alle getesteten Produkte hatten Funktionsbeschränkungen und stellen nur Teillösungen dar.
- Der Preis der kommerziellen Produkte ist durchweg sehr hoch.
- Der Installationsaufwand der Produkte ist schwer abzuschätzen.

Aufgrund von Vorarbeiten in anderen Projekten (Schmoeckel [16], HagerROM 2001 [2], HagerROM 2002 [3]) gibt es am Lehrstuhl für Informatik II bereits eine Benutzeroberfläche, die die Grundlage für die im folgenden vorgestellte Lernumgebung *DEJAVULearn* bildet.

6.6.2 Systemvoraussetzungen

DEJAVULearn wurde unter Windows2000/ME mit der Entwicklungsumgebung *IBM Visual Age for Java*, Version 3.3, entwickelt. Durch den Einsatz von Java als Programmie-

rsprache ist *DEJAVULearn* plattformunabhängig. Getestet wurde die Umgebung unter Windows95/98/2000/XP zusammen mit der Java Runtime Umgebung JRE 1.3. Aktuell ist noch keine reine Offline-Version verfügbar, die Lernobjekte werden noch direkt aus der Datenbank geholt, teilweise auch indirekt über einen WWW-Server. Das heißt: *DEJAVULearn* benötigt eine Netzverbindung zur Datenbank und zu einem WWW-Server. Die Generierung einer Offline-Version ist allerdings in Vorbereitung, Lerntexte und Fallbeispiele können bereits offline integriert werden, es fehlen allerdings noch Module, die die Quizelemente aus der Datenbank extrahieren bzw. ohne WWW-Server arbeiten.

Als Workaround ist die Offline-Nutzung jetzt schon möglich, wenn auf einem Rechner neben *DEJAVULearn* auch ein WWW-Server und die Datenbank installiert werden.

Eine Beschreibung der Konfiguration und Implementierung ist dem Anhang E.3.1, Seite 190, zu entnehmen.

6.6.3 Funktionen und Aufbau

Wie in Abbildung 49 zu sehen, sind die Hauptbestandteile der *DEJAVULearn* – Benutzeroberfläche der Navigationsframe links und der Lernobjekt-Präsentationsframe rechts. Zur Darstellung von textorientierten Objekten (z.B. Lerntexte) wurde das in der Java-API bereits vorhandene *javax.swing.JEditorPane* erweitert.

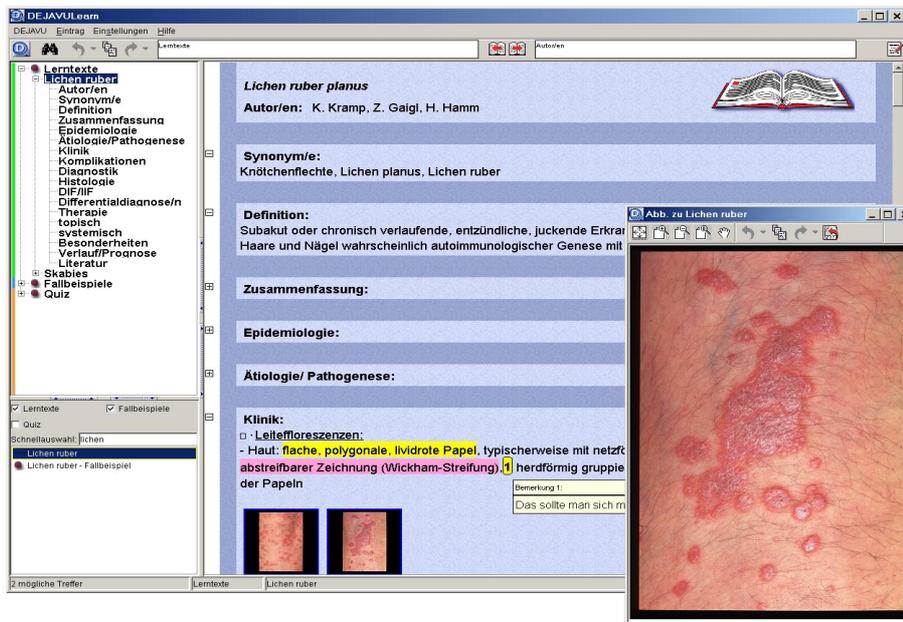


Abbildung 49: *DEJAVULearn* - Umgebung

Insgesamt sind folgende Funktionen von *DEJAVULearn* hervorzuheben:

- **Notizen.** Bemerkungen / Markierungen und Lesezeichen werden zusammen in einem Notizendialog übersichtlich verwaltet. Alle Arten von Notizen werden gespeichert und beim Start des Programms wieder geladen.
Bemerkungen sind die Texte des Nutzers, die auf den HTML-Seiten an beliebigen Stellen vom Benutzer eingegeben werden können.
Markierungen simulieren den Gebrauch von Textmarkern. Text kann auf den Seiten in verschiedenen Farben (auch über Sitzungen hinweg) markiert werden.
Lesezeichen können auf beliebige angezeigte Lernobjekte zum schnellen Wiederauffinden gesetzt werden.
- **History.** Ähnlich wie bei bekannten Internetbrowsern wird eine History der bereits besuchten Lernobjekte angelegt.
- **Bildbetrachter.** Zur Anzeige von Bildern startet der integrierte Bildbetrachter (Abbildung 49, rechts) mit folgenden Funktionen:
 - *Zooming.* Die Bilder können mehrfach über eine Lasso-Funktion (d.h. genau der markierte Bereich wird vergrößert) oder im ganzen gezoomt werden.
Bemerkung: Das serverseitige Zooming aus Abschnitt 5.3.3 wurde bisher noch nicht integriert, ist aber ohne großen Aufwand möglich.
 - *Zugriff auf die Bilder-History.*
 - *Bild in neuem Fenster öffnen.* Bei der Anzeige eines Bildes wird das gerade im Hauptfenster des Betrachters angezeigte Bild durch das neue ersetzt. Durch diese Funktion kann ein Bild in einem eigenen Fenster dauerhaft angezeigt und so mit anderen Bildern verglichen werden.
 - *Rücksprung zur Herkunft:* Das Lernobjekt, von dem aus das Bild aufgerufen wurde (zum Beispiel ein Lerntext), wird im *DEJAVULearn* – Hauptfenster geladen.
- **Druckfunktion.** Es können neben ganzen HTML-Seiten auch nur markierte Textbereiche gedruckt werden. Der Ausdruck von Bildern aus dem integrierten Bildbetrachter ist ebenfalls möglich.
- **Navigationsbaum.** Der Navigationsbaum ist ein sehr mächtiges Werkzeug. Es können damit unterschiedliche Lernpfade abgebildet werden oder ein lexikalischer Zugang zum Lernmaterial ermöglicht werden. Die Generierung der Navigationsbäume ist in Anhang E.3.1, Seite 190, beschrieben.

- *Definition von Lernpfaden:* *DEJAVULearn* enthält eine Blätterfunktion. Durch Blättern nach vorn (rechts) wird der Baum im Navigationsframe von oben nach unten (also in der Reihenfolge einer Tiefensuche) durchlaufen. Autoren können diesen Baum festlegen und damit auch eine Blätterreihenfolge, also einen Lernpfad (siehe Abschnitt 3) vorgeben, den der Lerner aber jederzeit (zum Beispiel durch Verfolgen eines Links) verlassen kann. Es wird also eine beliebige Untermenge der vorhandenen Lernobjekte innerhalb eines Baumes referenziert. Da mehrere Bäume in *DEJAVULearn* integriert werden können, stehen verschiedene Bäume für verschiedene Lernpfade (zum Beispiel angepaßt an den jeweiligen Wissensstand des Lerners) zur Verfügung.
- *Systematischer Zugriff.* Daneben können die Bäume auch eine alphabetische oder systematische Navigation auf den Lernobjekten ermöglichen,. So sind zum Beispiel hierarchische Bäume sortiert nach Lernobjekt-Typen (Abbildung 49), Diagnosen, Lokalisationen usw. denkbar.
- **Schnellauswahl.** die Schnellauswahl beschleunigt den Zugriff auf Einträge in den Bäumen. Bei jeder Eingabe eines Zeichens im Schnellauswahlfeld werden die Einträge aus dem Baum gefiltert, die den bisher eingegeben Substring noch enthalten.
- **Einklappen von Textbereichen.** Innerhalb von HTML-Seiten können zur kompakteren Darstellung vordefinierte Textbereiche eingeklappt werden, d.h. es bleibt nur noch eine Überschrift als Platzhalter sichtbar. In Abbildung 49 sind zum Beispiel im Lerntext zu *Lichen ruber* die Bereiche *Zusammenfassung*, *Epidemiologie* und *Ätiologie/ Pathogenese* eingeklappt. Bereiche können lokal ein-/ausgeklappt werden, es ist aber auch eine globale Vorauswahl möglich. Der Nutzer kann also zum Beispiel entscheiden, daß der Bereich *Epidemiologie* standardmäßig eingeklappt wird. Ihm bleibt dann immer noch offen, den eingeklappten Bereich im jeweiligen Eintrag wieder sichtbar zu machen.
- **Wiedergabe von Video und Audio.** Werden Video- oder Audioobjekte eingebunden, wird zur Wiedergabe auf Windows-Plattformen der auf dem System eingestellte Video- bzw. Audioplayer gestartet.

Die Konfiguration und Implementierung ist in Anhang E.3.1, Seite 190 beschrieben.

Im folgenden wird nur geschildert, wie die einzelnen Lernobjekt-Typen innerhalb der Lernumgebung verwendet werden.

6.6.4 Bild- und Video-Objekte

Die typische Anwendung von Bild- und Video-Objekten besteht nicht in ihrer Präsentation im einzelnen, sondern eher innerhalb anderer Objekte, zum Beispiel in Lerntexten (siehe Abschnitt 6.3). In Lernobjekten, die in der Lernumgebung als HTML-Seiten dargestellt werden, können Vorschaubilder für Bilder und Videos in der üblichen Form zweistufig integriert werden: Man bindet in der HTML-Seite selbst ein kleines Vorschaubild ein, das auf Anklicken vergrößert wird. In der Lernumgebung wird auf diesen Klick hin allerdings der integrierte Bildbetrachter gestartet (Abbildung 49, rechts), für Videoclips wird der auf dem Rechner voreingestellte Betrachter für Videos gestartet (zum Beispiel der Microsoft Windows Media Player).

6.6.5 Moulagen

Zu Präsentation von Moulagen wird aktuell das Java-Applet *Freedom VR* [43] verwendet. Es wurde dazu entwickelt, fotografische virtuelle Realität in Webseiten einzubinden. Das Applet lädt beim Starten alle Bilder einer Szene (also zum Beispiel alle Bilder aus den verschiedenen Blickwinkeln einer Moulage) und stellt jeweils ein Bild dar. Der räumliche Eindruck entsteht dann beim Ziehen der Maus in verschiedenen Richtungen, wodurch jeweils die entsprechenden Bilder angezeigt werden. Abbildung 50 zeigt in der oberen Reihe 4 Bilder einer Moulage (insgesamt wurden 230 Bilder aufgenommen), in der unteren Reihe sind zwei Bilder einer Ausschnittvergrößerung zu sehen.



Abbildung 50: Moulagen. 3D-Darstellung.

Leider lädt *Freedom VR* bei Programmstart alle Bilder eine Szene komplett als Java-Images in den Speicher. Dadurch stößt man relativ schnell an die Grenzen des Arbeitsspeichers auch moderner Rechner. Nachdem *Freedom VR* im Quellcode verfügbar ist, wären Ansätze zur Verbesserung darin zu suchen, Bilder rund um die gerade angezeigte Region geschickt nachzuladen oder die Bilder in komprimierter Form im Speicher zu halten und dann erst bei Anforderung auszupacken. Alternativ können auch andere Tools, zum Beispiel *Quicktime VR*, eingesetzt werden.

6.6.6 Vorlesungen zur Nachbereitung

Im Verlauf der Arbeiten an DEJAVU wurden verschiedene Möglichkeiten diskutiert, wie den Studenten gehaltene Vorlesungen zur Nachbereitung zur Verfügung gestellt werden können:

- **digitales Video:** Dazu muß während der Vorlesung eine Kamera zur Aufzeichnung mitlaufen. Die Nachteile dieser Methode sind der hohe Speicherbedarf für ein bis zu 90-minütiges Video und die schlechte Bildqualität. Dadurch ist sie ungeeignet dazu, Diagnosen zu erlernen.
- **Bildfolge mit Tonsequenzen:** Während der Vorlesung wird der Ton mit aufgezeichnet. Die Vorlesung selbst wird mit Powerpoint gehalten, das selbst dann schon die Funktionalität bietet, die Tonspur für jede Folie getrennt zu speichern. Nachteil bei diesem automatisierten Splitten ist, daß der Dozent häufig keine Pause macht, wenn er eine Folie wechselt oder, was noch schlimmer ist, den Text einer Folie in die nächste hineinspricht. Alternativ dazu bleibt dann nur das manuelle Schneiden der Tonspur. Mit einem AddIn (siehe Abschnitt 6.5.5) werden die in der Powerpoint-Präsentation verwendeten Bild- und Videoobjekte aus der Datenbank in Originalqualität extrahiert. Präsentiert werden kann die Vorlesung zum Beispiel in HTML-Form. Der Ton zum jeweiligen hochwertigen Bild wird als Link angeboten. Der Student kann dann selbst entscheiden, ob er den Ton zum Bild hören will oder nicht. Außerdem können Links zu weiteren Datenbankinformationen zum Lernobjekt bzw. zu Knoten im semantischen Netz generiert werden.
- **Bildfolge mit Untertiteln:** Ähnlich wie bei der vorangegangenen Methode werden hier die Bilder aus der Datenbank gezeigt. Statt des Tons wird aber entweder ein Fließtext –

der um Versprecher oder ähnliches bereinigte Sprechtext des Dozenten – oder nur Schlagworte zum Bild gezeigt.

Natürlich sind auch beliebige Kombinationen der drei Varianten denkbar und innerhalb von *DEJAVULearn* darstellbar.

Allerdings verliert man Informationen über vom Dozenten mit einem Laserpointer oder Stock hervorgehobene Bildregionen („...*Läsionen* sehen wir zum Beispiel hier und hier...“), wenn man die Vorlesung nicht mit einer Videokamera aufnimmt. Diese Auszeichnungen sind bei der Präsentation mit statischen Bildern auch nur schwer nachzubilden. Man müsste dann im Bild Regionen numerieren und dann entsprechend verlinken (mit dem Originalton des Dozenten nicht möglich) oder dasselbe Bild mehrmals mit verschiedenen Hervorhebungen hintereinander anzeigen. Technisch stellt das kein Problem dar, ist aber vom Platzbedarf her natürlich deutlich ungünstiger als das Bild angezeigt zu lassen und die gewünschte Region jeweils hervorzuheben. Dies ist aber mit reinen HTML-Mitteln nicht möglich. Allerdings gibt es auch hierzu bereits Ansätze.

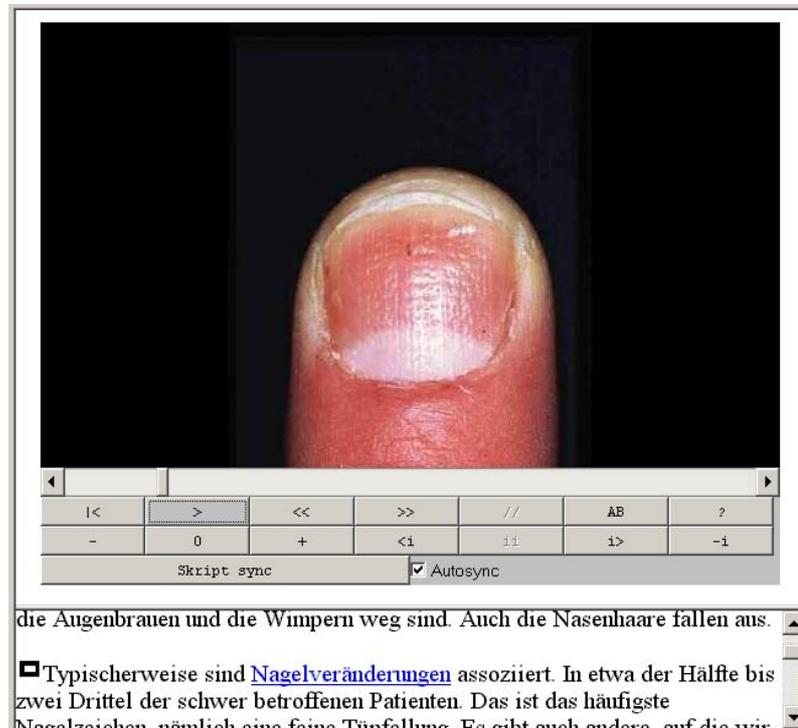


Abbildung 51: Vorlesungen zur Nachbereitung mit Bild, Text, Ton.

Im Rahmen einer Diplomarbeit [12] und weiteren, darauf aufsetzenden Arbeiten [6] wurde ein Prototyp eines Java-Applets mit folgenden Funktionen erstellt (Abbildung 51):

- zeitlich gesteuerte Anzeige von Bildern
- Sprung zu beliebigen Bildern innerhalb der Abfolge
- synchrones Abspielen von Tonsequenzen zu den Bildern
- synchronisierte Anzeige von Text
- Einblendung von Markierungen in Bilder zu beliebigen Zeitpunkten. Diese Markierungen können auch mit Links angereichert werden.

6.6.7 Lerntexte und Fallbeispiele

Lerntexte und Fallbeispiele sind in der Datenbank als Verweise zu XML-Dateien gespeichert (Abschnitt 6.5.6). Zur Darstellung in *DEJAVULearn* werden die XML-Dateien mit Hilfe von XSL-Stylesheets nach HTML konvertiert und im erweiterten *JEditorPane* angezeigt. Abbildung 49 zeigt den Lerntext zu *Lichen ruber*.

6.6.8 Quizelemente

Die in *DEJAVULearn* integrierten Quizelemente dienen (zumindest im Moment) nicht zur Durchführung von Prüfungen, sondern dem Nutzer zum Selbststudium. Auch eine Bewertung der erzielten Ergebnisse wurde noch nicht implementiert. Dazu bedarf es allerdings auch einer genauen Absprache mit den Experten: In Diskussionen ergaben sich zum Beispiel folgende Aspekte:

- Welche Fragen geben wieviele Punkte?
- Ist die Reihenfolge gegebener Antworten oder durchgeführter diagnostischer Verfahren wichtig?
- Soll in die Bewertung einfließen, wie teuer ein durchgeführtes Verfahren ist (eine für die Praxis durchaus wichtiger Aspekt)?

Ein erster Prototyp für die Studentenseite des **Fallquiz** wurde mit Perl-CGIs implementiert, Schnittstelle zur Datenbank ist DBI mit DBD::mysql. Nacheinander muß der Nutzer die sinnvollen Anamnesefragen, die Effloreszenzen, die sinnvollen diagnostischen Verfahren und schließlich die richtige Diagnose auswählen (Abbildung 52).

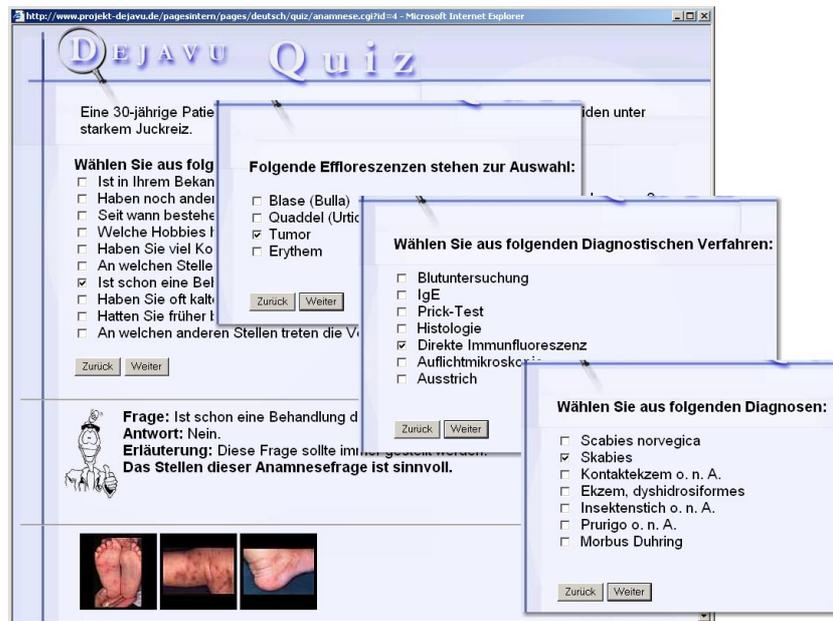


Abbildung 52: Ablauf des Fallquiz.

Diese funktionsfähige Version soll aber in Zukunft für eine Offline-Variante durch eine reine Java-Implementierung ersetzt werden. Dann muß kein WWW-Server zur Ausführung der CGI-Skripte mehr im Zugriff sein, evtl. kann auch der Zugriff auf die Datenbank ersetzt werden. Außerdem bieten sich bei der Programmierung flexiblere Layout-Möglichkeiten.

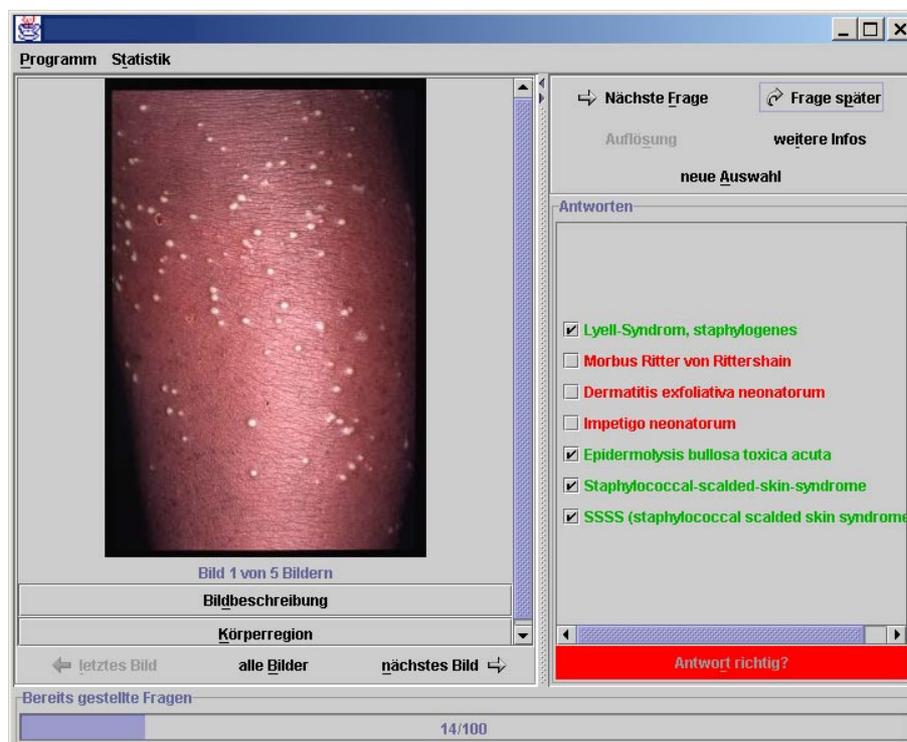


Abbildung 53: DEJAVULearn. Bilderquiz.

Das **Bilderquiz** (Abbildung 53) dient vor allem der Schulung des diagnostischen Blicks. Es steht als eigene Java-Applikation zur Verfügung, kann aber auch aus *DEJAVULearn* gestartet werden. Beim Starten des Quiz hat der Anwender die Möglichkeit, die ihm präsentierten Bilder auf bestimmte Themengebiete oder Lokalisationen einzuschränken. Aus den vorhandenen Bildobjekten zu diesen Bereichen werden dann zufällige Bilder angezeigt, zu denen die jeweilige Diagnose angegeben werden muß. Das Quiz ist in der Schwierigkeitsstufe konfigurierbar. Je nachdem wie fortgeschritten der Nutzer ist, werden ihm neben der richtigen Diagnose mehr oder weniger falsche zur Auswahl gestellt. Die dem Nutzer als falsche Diagnosen angezeigte Auswahl wird dabei nicht zufällig getroffen, es wird jeweils eine Auswahl der in der Datenbank gespeicherten Differentialdiagnosen gezeigt. Als Differentialdiagnosen bezeichnet man die Diagnosen, die mit einer Diagnose verwechselbar sind. Bei rein zufälliger Auswahl der Antworten könnte dagegen zum Beispiel *Zungenfibrom* angeboten werden, auf dem Bild ist aber keine Zunge abgebildet. Als Hilfestellung können alle Bildobjekte, denen die gleiche Diagnose zugeordnet wurde, angezeigt werden. Der Lernende hat nach der Lösung zudem die Möglichkeit, weitere Informationen (Skripte, Lerntexte, Videos, usw.) zur dargestellten Diagnose aus der Datenbank anzufordern.

Eine Beschreibung der Konfiguration und der Implementierung ist im Anhang E.3.2, Seite 193 zu finden.

Zur Durchführung von **Übungsaufgaben** dienen Perl-CGI-Skripte (Abbildung 54). Zufällig aus dem gesamten Pool ausgewählte Fragen werden nacheinander als HTML-Formulare zusammen mit den Antwortalternativen angezeigt. Dabei können die Fragen auf ein Themengebiet eingeschränkt werden. Der Lerner kann nach Auswahl einer der Alternativen entscheiden, ob er direkt die nächste Frage beantworten will oder vorher die Erläuterung zur aktuellen Frage einsehen will. Nach der letzten Frage wird dem Lernenden eine Zusammenfassung seiner Fragen präsentiert. Dabei werden alle gestellten Fragen, die Antworten und optional die Erläuterungen angezeigt. Bei jeder Frage wird die vom Nutzer gewählte Antwort und die richtige Antwort markiert. Auf diese Weise kann der Lerner die Fragen noch einmal nachbearbeiten und gegebenenfalls auch ausdrucken. Als Abwandlung können die Skripte auch so konfiguriert werden, daß falsch beantwortete Fragen später erneut gestellt werden.

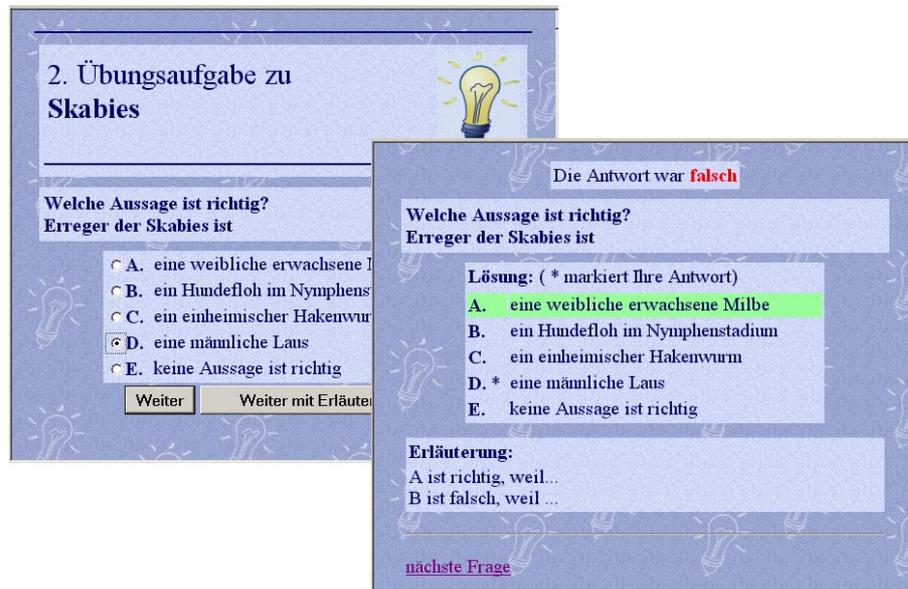


Abbildung 54: DEJAVULearn. Übungsaufgaben.

Für die spätere Generierung einer Offline-Version der Lernumgebung sollen die im Moment auf Perl-CGI-Skripten basierenden Quizelemente (Fallquiz und Übungsaufgaben) komplett in JAVA implementiert werden, da andernfalls stets ein WWW-Server erreichbar bzw. lokal installiert sein muß.

Eine Beschreibung der Konfiguration und der Implementierung ist im Anhang E.3.4, Seite 194 zu finden.

6.7 Konformität zu bestehenden Standards

Bei der Modellierung der Datenbank wurde darauf geachtet, daß die Datenfelder mit dem LOM-Standard (Abschnitt 2.2.2) konform sind. Mit einem Perl-Skript (siehe Anhang F, S. 197) kann ein Lernobjekt aus der Datenbank als XML-Record exportiert werden. Die generierten XML-Dateien sind mit der DTD der Version 1.2, Final Specification, der *IMS Metadata Specification* vom Juni 2001 (*imsmd_rootv1p2.dtd*) gültig. Im folgenden ein Beispiel für einen XML-Record eines Bildobjekts, einige Abschnitte (zum Beispiel *classification* für *Farbe, Form, Verteilung*) sind hier nicht aufgenommen.

```

<?xml version="1.0" ?>
<!DOCTYPE lom SYSTEM "imsmd_rootv1p2.dtd">
<lom>
  <general>
    <title><langstring>Bild a3_00001.jpg</langstring></title>
    <catalogentry>
      <catalog>LO Datenbank</catalog>
      <entry><langstring>101</langstring></entry>
    </catalogentry>
  </general>
  <lifecycle>
    <contribute>
      <role><source><langstring>unbekannt</langstring></source>
        <value><langstring>Bildscanner</langstring></value></role>
      <date><datetime>1998-02-11</datetime></date>
    </contribute>
  </lifecycle>
  <technical><format>jpg</format><location>a3/a3_00001.jpg</location>
</technical>
  <classification>
    <taxonpath>
      <source><langstring>ICD10</langstring></source>
      <taxon><id>L98110</id>
        <entry><langstring>Artefakt </langstring></entry>
      </taxon>
    </taxonpath>
    <description><langstring>Diagnose</langstring></description>
  </classification>
  <classification>
    <taxonpath>
      <source><langstring>ICD0</langstring></source>
      <taxon><id>C4451</id>
        <entry><langstring>Brustwand</langstring></entry></taxon>
      </taxonpath>
      <description><langstring>Lokalisation</langstring></description>
    </classification>
</lom>

```

Daneben können für *DEJAVU Learn* generierte Lernpfade als XML-Record konform zum *IMS Content Packaging Information Model, Version 1.1.1*, (Abschnitt 2.3.1) exportiert werden.

Sofern die am Lernpfad angehängten Lernobjekte als HTML-Seiten vorhanden sind, können diese Inhalte dann auch mit dem Viewer des *Microsoft LRN Toolkit* (Abschnitt 3.2) dargestellt werden (Abbildung 55). Umgekehrt können auch mit dem *LRN Editor* erstellte Lernpfade in *DEJAVU Learn* importiert werden.

Die Exportfunktion und die Konvertierung der Lernpfade sind in Anhang F, Seite 197, beschrieben.

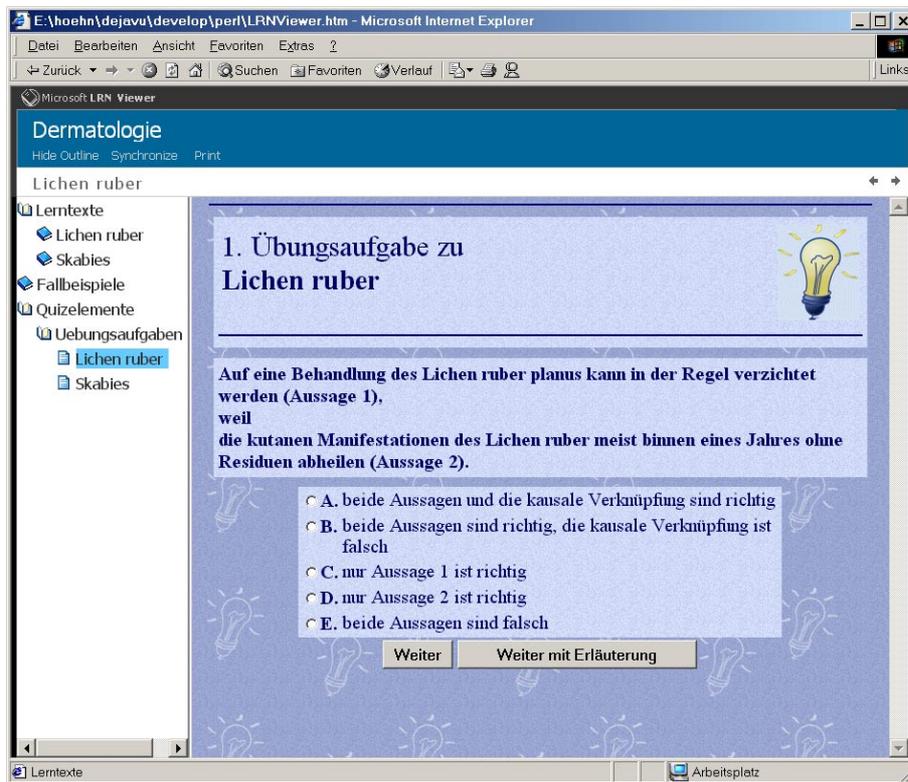


Abbildung 55: Lernpfade und Lernobjekte im MS LRN-Viewer

7 Zusammenfassung und Ausblick

7.1 Ergebnisse

Neben den Vorteilen, die der Einsatz von E-Learning im allgemeinen bietet, wie die flexible Lernorganisation oder die Nutzung von Lerninhalten unabhängig von Ort und Zeit, zeichnet sich die entwickelte Lehr- und Lernplattform gegenüber den in Kapitel 3 vorgestellten Produkte besonders durch folgende Punkte aus:

- **Lerninhalte höherer Qualität.** Die Möglichkeit der Erstellung von Lernobjekten an verschiedenen Zentren und der *Vereinigung zu einer Lernobjekt-Datenbank* führt zu Lerninhalten höherer Qualität. Zum einen kommt es zu einer *multizentrische Bündelung von Expertenwissen*, zum anderen zu einer Arbeitsteilung bei der Erstellung von Lernobjekten, wodurch sich der einzelne Autor mehr auf seine Inhalte konzentrieren kann.
- **Erweiterbarkeit.** Durch das Design der Datenbank ist eine Erweiterung auf andere Lernobjekt-Typen möglich. Die Applikation zum Einfügen der Objekte in die Datenbank *DEJAVUCapt* ist so modular gehalten, daß auch hier andere Lernobjekt-Typen vorschlagwortet werden können. Alternativ steht auch der Weg über andere Tools (zum Beispiel Perl-CGIs) zur Verfügung.

- **Verwaltbarkeit.** Durch die Abstützung auf die relationale Datenbank wird der Bestand von Lernobjekten effizient verwaltbar. Vorhandene Lernobjekte können mit Datenbank-anfragen durch die Verschlagwortung effizient und gezielt gefunden werden.
- **Objektorientierung.** Durch die objektorientierte Modellierung der Datenbank können einmal erstellte Lernobjekte in anderen Lernobjekten jederzeit wiederverwendet werden.
- **Konsistenz.** Durch die Einbindung von Lernobjekten in andere Objekte werden Änderungen in den eingebundenen Objekten automatisch übernommen.
- **Flexibilität.** Die Erstellung von Lernobjekten geschieht durch unterschiedliche Autorenwerkzeuge. Selbst Objekte vom gleichen Lernobjekt-Typ können über verschiedene Autorenwerkzeuge erstellt werden. Autoren können also oft auf ihre favorisierten Werkzeuge zurückgreifen.
- **Geringer Verwaltungsaufwand.** Die Autorenwerkzeuge sowie die Applikation zum Einfügen von Objekten in die Datenbank *DEJAVUCapt* können von den Dozenten/Autoren selbst bedient werden. Die gesamte Lehr- und Lernplattform ist weitgehend ein selbstlaufendes System. Eine Ausnahme bildet hier zum Beispiel das Hinzufügen eines neuen Lernobjekt-Typs. Hier muß der Typ in die Datenbank eingetragen werden und auch die Applikation zum Einfügen in die Datenbank *DEJAVUCapt* muß angepaßt werden.
- **Navigationsmöglichkeiten für den Studenten.** Die flexible Lernumgebung *DEJAVU-Learn* erlaubt mit den unterschiedlichen Navigationsmöglichkeiten den effizienten Zugriff des Studenten auf die Lerninhalte, sei es in Form vorgeschlagener Lernpfade oder zum Nachschlagen.
- **Personalisierbarkeit.** Durch die Verwaltung persönlicher Notizen in Form von Bemerkungen und Lesezeichen und die Konfigurierbarkeit kann die Lernumgebung auf die persönlichen Bedürfnisse und Vorstellungen des Lernenden angepaßt werden.
- **Einfache Installation.** Alle Module können auf Windows-Plattformen mit Setup-Programmen installiert werden.
- **Konformität zu internationalen Standards.** Der Austausch von Lernobjekten oder ganzen Kursen in Form von Inhaltspaketen ist aufgrund der Konformität zu bekannten E-Learning Standards wie IEEE *Learning Object Metadata Standard* und *IMS Content Packaging Information Model* durch das Exportieren von XML-Datensätzen möglich.

Speziell im Anwendungsbeispiel Dermatologie bieten sich weitere Vorteile:

- **Erleichterung der Vorbereitung und Durchführung von Vorlesungen.** Statt in jedem Vorlesungszyklus erneut in einem Diaarchiv manuell passende Bilder suchen zu müssen, kann komfortabel in der Datenbank gesucht werden. Zudem können Folien in einmal erstellten Vorlesungen leicht ausgetauscht oder aktualisiert werden. Außerdem werden der Vorlesungsassistent und der Fotograf, der Textdias für Vorlesungen belichten mußte, überflüssig und stehen damit für andere Aufgaben zu Verfügung.
- **Neue Lernobjekt-Typen.** Erst durch die Verwendung neuer Lernobjekt-Typen wie Videoclips wird es möglich, in Vorlesungen Operationstechniken anschaulich zu präsentieren.
- **Verschlagwortung anhand internationaler Klassifikationen.** Dozenten und Studenten haben durch die Abstützung der Verschlagwortung auf verbreitete internationale Klassifikationen wie ICD und MeSH schnellen Zugriff auf die Lernobjekte.
- **Vorlesungen zur Nachbereitung.** Erst durch die Bereitstellung von Vorlesungen zur Nachbereitung haben Dermatologie – Studenten die Chance, Vorlesungen mit dem Bildmaterial nachzuvollziehen.
- **Entlastung von Patienten.** Patienten werden durch Lernobjekt-Typen wie Fallbeispiel und -quiz entlastet. Es kann weniger häufig notwendig sein, am konkreten Patienten zu lehren, er wird durch einen virtuellen Patienten ersetzt. Das sollte aber nur unterstützend geschehen.

Sowohl bei der Modellierung als auch bei der Umsetzung wurde darauf geachtet, möglichst gut die Anforderungen bei gleichzeitiger Erweiterbarkeit auf andere, ähnliche Szenarien zu erfüllen. Zwar werden nicht alle dermatologischen Lernobjekt-Typen auch in anderen Fachrichtungen vorkommen, aber Vorlesungen und Quizelemente wird es beinahe überall geben. Besonders einfach sollte die Anpassung der Plattform für andere bildorientierte Disziplinen wie der Radiologie sein.

Die vorgestellte Lehr- und Lernumgebung wurde zusammen mit den Dozenten bei gleichzeitiger Evaluation durch Psychologen entwickelt. Schon während der Entwicklung des Pakets wurden die unterschiedlichen Module an den Hautkliniken der Universitäten Würzburg, Jena und Mannheim mit Erfolg eingesetzt.

7.2 Ausblick

Raum für weitere Entwicklungen findet sich insbesondere in den folgenden Bereichen:

- Automatischer Abgleich der lokalen Datenbank mit der zentralen Datenbank: Der Abgleich wird im Moment manuell vorgenommen, die Daten der lokalen Datenbanken werden von den einzelnen Institutionen an den Verwalter der zentralen Datenbank übermittelt, dieser führt diese dann mit einem Perl-Skript zur zentralen Datenbank zusammen. Geplant ist der regelmäßige, automatische Abgleich der Datenbanken. Dieser müßte aufgrund der lokalen Firewalls von den einzelnen Institutionen ausgehen.
- Einsatz eines Multimedia-Datenbanksystems: Dadurch könnten direkte Ähnlichkeitsanfragen an die Datenbank gestellt werden. Dies ist allein durch die große Anzahl an Bildern interessant. Durch eine entsprechende Anfrage ist es dann eventuell automatisch möglich, Bilder von Differentialdiagnosen zu finden.
- Erstellung eines Autorentools für das Bilderquiz: Das bisher auf zufälliger Auswahl von Bildern aus der Datenbank beruhende Bilderquiz kann an Qualität gewinnen, wenn durch einen Autor zum Beispiel bestimmte Bilder oder Antwortalternativen ausgewählt werden können.
- Implementierung eines Lernpfad-Editors: Für den Autor gibt es im Moment die Möglichkeit, Lernpfade in einem Text-Editor einzugeben und dann mit einem Perl-Skript so zu konvertieren, daß sie in *DEJAVULearn* eingelesen werden. Alternativ kann ein Lernpfad auch mit dem *LRN Editor* des *MS LRN Toolkit* erstellt werden. Allerdings gelten dabei die bereits in Abschnitt 3.2 erwähnten Einschränkungen. Hier wäre ein komfortabler Editor mit Datenbankbindung, der das Generieren von Lernpfaden mittels Drag'n'Drop erlaubt, erstrebenswert. Auch Studenten könnten diesen Editor zur Festlegung ihres persönlichen Lernpfads, zum Beispiel als Teil eines vorgegebenen Lernpfads, nutzen.
- Andere VR-Präsentationsmöglichkeiten für Moulagen: Das eingesetzte *Freedom VR* Applet hat den zum einen den Nachteil des hohen Arbeitsspeicherbedarfs, zum anderen kann es nicht in *DEJAVULearn* angezeigt werden. Hier gibt es zwar bestehende Präsentationsalternativen. Allerdings wäre für die gewünschte Einbindung in *DEJAVULearn* eine Java-Implementierung (auf *Freedom VR* aufsetzend oder völlig neu) sinnvoll.

- Scoring / Zeitmessung / Zeitvorgaben: Die Quizelemente dienen im Moment nur dem Selbststudium. Selbst hierfür wird dem Lernenden im Moment kaum Feedback gegeben, das eine Bewertung seiner Leistung ermöglicht. In zukünftigen Versionen ist die Möglichkeit eines Scorings sowie des Setzens von Zeitvorgaben für die Bearbeitung von Lernobjekten erstrebenswert. Spätestens wenn das System auch für Tests verwendet werden soll, müssen zur Bearbeitung von Lernobjekten auch Zeitmessungen möglich sein. Bisher gibt es auch keine Möglichkeit der Lernfortschrittskontrolle. In diesem Zusammenhang würde sich eine Orientierung an der *IMS Question & Test Interoperability (QTI) Spezifikation* (Abschnitt 2.3.2, S. 31) anbieten.
- Offline-Variante von DEJAVULearn: Für eine wirkliche Offline-Version der Lernumgebung sind noch einige Anpassungen notwendig. Der Aufwand ist davon abhängig, ob diese Version auch ohne die Datenbank (oder nur ohne WWW-Server) auskommen soll. Auf jeden Fall müssen aber die Quizelemente, die bisher auf CGIs beruhen (Fallquiz, Übungsfragen), auf andere Weise umgesetzt werden. Wegen der wünschenswerten Integration in *DEJAVULearn* bietet sich eine Implementierung in Java an.
- Kommunikations-Komponenten: *DEJAVULearn* beschränkt sich im Moment auf das Selbstlernen. Eine Bereicherung wären sicher eine Chat-Funktion oder Diskussionsforen wie sie von anderen Systemen bereits angeboten werden (siehe Abschnitt 3). Sie könnten zur Diskussion mit anderen Lernenden, aber auch mit einem Tutor zu regelmäßigen Zeiten genutzt werden.
- Graphische Aufbereitung: Die graphische Aufbereitung der Lernobjekte bei ihrer Darstellung wurde bisher noch nicht sehr forciert. Bei Lerntexten und Fallbeispielen ist das durch die Verwendung von Stylesheets abgedeckt. Die Präsentation der Quizelemente erfordert aber noch einen wesentlich höheren Aufwand. Zum Beispiel könnte man für das Fallquiz ein virtuelles Arztzimmer entwerfen.
- Volltextsuche: Die auf anderen Projekten basierende Lernumgebung *DEJAVULearn* besitzt grundsätzlich die Möglichkeit, einer konfigurierbaren Volltextsuche. Bisher war sie auf das Durchsuchen statischer HTML-Seiten mit Hilfe eines generierten q-Gramm Index beschränkt. Sie müßte so aufgebohrt werden, daß sie auch die Datenbank mit durchsucht.

- Anpassung des semantischen Netzes: Das semantische Netz wurde nur im Rahmen der Vorarbeiten generiert. Es ist bisher nicht mit der Lernobjekt-Datenbank gekoppelt und nicht in der Lernumgebung integriert.
- Konformität zur *QTI*-Spezifikation des IMS: Skripte zum Import/Export von Quiz-elementen konform zur *Question & Test Interoperability (QTI) Spezifikation* des IMS Global Learning Consortium existieren bisher nicht. Da die Datenfelder der integrierten Quizelemente (Übungsfragen, Fallquiz und Bilderquiz) relativ einfach auf die Spezifikation abbildbar sind, sollte sich der Aufwand aber in Grenzen halten.

Anhang A - LOM-Standard

In der nachstehenden Tabelle sind alle Felder des „Standard for Information Technology – Education and Training Systems – Learning Objects and Metadata“ – WD 6.1 [44] mit ihren Erläuterungen und Beispielen enthalten. Um die Übersichtlichkeit zu erhöhen, wurden aber nicht alle Spalten aus der Original-Tabelle übernommen. Es fehlen: *Valuespace* und *Datatype*. Die Spalte *Example* wurde teilweise gekürzt.

	Name	Size	Order	Example
	Explanation			
1	General	1	N/A	-
	This category groups the general information that describes this learning object as a whole.			
1.1	Identifier	1	N/A	-
	A globally unique label that identifies this learning object. This data element is not and shall not be used, because there is no specified method for the creation of a globally unique identifier.			
1.2	Title	1	N/A	-
	Name given to this learning object.			

1.3	Catalog Entry	smallest permitted maximum: 10 items	No.	-
	This data element defines an entry within a catalog (i.e. a listing identification system) assigned to this learning object. This sub-category shall describe this learning object according to some known cataloging system so that it may be externally searched for and located according to the methodology of the specified system. This sub-category should be used as a functional replacement for the currently reserved data element 1.1:General.Identifier, as that is currently reserved. NOTE--One of the catalog entries could be generated automatically by a tool.			
1.3.1	Catalog	1	N/A	ISBN, ARIADNE
	The name of the catalog (i.e. listing identification system).			
1.3.2	Entry	1	N/A	(en,2-7342-0318), (fr,LEAO875)
	Actual string value of the entry within the catalog (i.e. listing identification system).			
1.4	Language	smallest permitted maximum: 10 items	No	"en", "en-GB", "de", "fr-CA", "it"
	The primary human language or languages used within this learning object to communicate to the intended user. NOTE--An indexation tool may provide a useful default.			
1.5	Description	smallest permitted maximum: 10 items	No	-
	A textual description of the content of this learning object.			
1.6	Keywords	smallest permitted maximum: 10 items	No	-
	Keywords or phrases describing this learning object. This data element should not be used for characteristics that can be described by other data elements.			
1.7	Coverage	smallest permitted maximum: 10 items	No	(en,Circa, 16th century France)
	The span or extent of such things as time, culture, geography or region that applies to this learning object.			
1.8	Structure	1	N/A	-
	Underlying organizational structure of this learning object.			
1.9	Aggregation Level	1	N/A	-
	The functional granularity of this learning object.			
2	Life Cycle	1	N/A	-
	This category describes the history and current state of this learning object and those who have affected this learning object during its evolution.			
2.1	Version	1	N/A	3.0, (en,1.2.alpha), (nl,voorlopige versie)
	The edition of this learning object.			
2.2	Status	1	N/A	-
	The state or condition of this learning object.			

2.3	Contribute	smallest permitted maximum: 30 items	No	-
	This data element describes those people or organizations that have affected the state of this learning object during its evolution (includes creation, edits and publication). NOTE:--This sub-category is different from 3.3:MetaMetaData. Contribute.			
2.3.1	Role	1	N/A	-
	Kind of contribution. At least the Author(s) of the learning object should be described.			
2.3.2	Entity	smallest permitted maximum: 40 items	Yes	-
	The identification of and information about people or organizations contributing to this learning object, most relevant first. If 2.3.1:LifeCycle.Contribute.Role equals Author, then the entity or entities shall be a person or persons. If 2.3.1:LifeCycle.Contribute.Role equals Publisher, then the entity shall be an organisation. If the entity is an organisation, then it should be a university department, company, agency, institute, etc. under whose auspices the contribution was made.			
2.3.3	Date	1	N/A	-
	The date of the contribution.			
3	Meta-Metadata	1	N/A	-
	This category describes this metadata record itself (rather than the learning object that this record describes). This category describes such things as who created this meta-data record, how, when and with what references. This is not the information that describes the learning object itself.			
3.1	Identifier	1	N/A	-
	A globally unique label that identifies this metadata record. This is not and shall not be used, as there is no specified method for the creation of a globally unique identifier.			
3.2	Catalog Entry	smallest permitted maximum: 10 items	No	-
	This sub-category defines an entry within a catalog (i.e. listing identification system), given to the metadata instance. This category should describe this metadata instance according to some known cataloging system so that it may be externally searched for and located according to that system. This data element may be used as a functional replacement for the currently reserved data element 3.1:MetaMetaData. Identifier. NOTE:--One of the catalog entries may be generated automatically by an indexation tool.			
3.2.1	Catalog	1	N/A	Ariadne
	The name of the Catalog (i.e. listing identification system). NOTE:--Generally system generated.			
3.2.2	Entry	1	N/A	(en,KUL532)
	Actual string value of the entry in the Catalog. NOTE:--This data element is usually generated by the system.			

3.3	Contribute	smallest permitted maximum: 10 items	Yes	-
	This sub-category describes those people or organizations that have affected the state of this metadata instance during its evolution (includes creator and validator). NOTE: This data element is different from 2.3:Lifecycle.Contribute.			
3.3.1	Role	1	N/A	-
	Kind of contribution. Exactly one instance of creator should exist.			
3.3.2	Entity	smallest permitted maximum: 10 items	Yes	-
	The identification of and information about the people or organizations contributing to this metadata instance, most relevant first.			
3.3.3	Date	1	N/A	-
	The date of the contribution.			
3.4	Metadata Scheme	smallest permitted maximum: 10 items	No	LOM-1.0
	The name and version of the authoritative specification used to create this metadata instance. NOTE:--This data element may be user selectable or system generated. If multiple values are provided, then the metadata instance shall conform to multiple metadata schemes.			
3.5	Language	1	N/A	-
	Language of this metadata instance. This is the default language for all LangString values in this metadata instance.			
4	Technical	1	N/A	-
	This category describes the technical requirements and characteristics of this learning object.			
4.1	Format	smallest permitted maximum: 40 items	No	video/ mpeg,application/ x-tool-book,text/html
	Technical data type(s) of (all the components of) this learning object. This data element shall be used to identify the software needed to access the learning object.			
4.2	Size	1	N/A	-
	The size of the digital learning object in bytes. Only the digits '0' through '9' should be used; the unit is bytes, not Mbytes, GB, etc. This data element shall refer to the actual size of this learning object. If the learning object is compressed, then this data element shall refer to the uncompressed size.			
4.3	Location	smallest permitted maximum: 10 items	Yes	http://host/id
	A string that is used to access this learning object. It may be a location (e.g. Universal Resource Locator), or a method that resolves to a location (e.g. Universal Resource Identifier). Preferable Location first. This is where the learning object described by this metadata instance is physically located.			

4.4	Requirements	smallest permitted maximum: 40 items	No	-
	This sub-category describes the technical capabilities required in order to use this learning object. If there are multiple requirements, then all are required, i.e. the logical connector is AND.			
4.4.1	Type	1	N/A	-
	The technology required to use this learning object, i.e. hardware, software, network, etc.			
4.4.2	Name	1	N/A	-
	Name of the required technology to use this learning object. NOTE:--The value for this data element may be derived from 4.1:Technical.Format automatically, e.g., "video/mpeg" implies "Multi-OS".			
4.4.3	Minimum Version	1	N/A	-
	Lowest possible version of the required technology to use this learning object.			
4.4.4	Maximum Version	1	N/A	-
	Highest version of the technology known to support the use of this learning object.			
4.5	Installation Remarks	1	N/A	-
	Description of how to install this learning object.			
4.6	Other Platform Requirements	1	N/A	sound card, runtime
	Information about other software and hardware requirements.			
4.7	Duration	1	N/A	PT1H30M, PT1M45S
	Time a continuous learning object takes when played at intended speed. This data element is especially useful for sounds, movies or animations.			
5	Educational	1	N/A	-
	This category describes the key educational or pedagogic characteristics of this learning object. This is the pedagogical information essential to those involved in achieving a quality learning experience. The audience for this metadata includes teachers, managers, authors and learners.			
5.1	Interactivity Type	1	N/A	Expositive documents include essays, video clips, all kinds of graphical
	The flow of interaction between this learning object and the intended user. In an expositive learning object, the information flows mainly from this learning object to the learner. Expositive documents are typically used for learning- by- reading. In an active learning object, information also flows from the learner to this learning object. Active documents are typically used for learning- by- doing. Activating links to navigate in hypertext documents is not considered as an information flow. Thus, hypertext documents are expositive.			
5.2	Learning Resource Type	smallest permitted maximum: 10 items	Yes	-
	Specific kind of learning object, most dominant kind first.			
5.3	Interactivity Level	1	N/A	-
	The degree of interactivity between the end user and this learning object.			

5.4	Semantic Density	1	N/A	-
	Amount of information conveyed by this learning object as compared to its size or duration.			
5.5	Intended End User Role	smallest permitted maximum: 10 items	Yes	-
	Principal user(s) for which this learning object was designed, most dominant first. A learner works with a learning object in order to learn something. An author creates or publishes a learning object. A manager manages the delivery of this learning object, e.g., a university or college. The document for a manager is typically a curriculum. A typical example of a learning object whose intended end user is an author is an authoring tool, specifically an authoring tool for learning objects, like a questionnaire authoring tool, or a pedagogical simulation authoring tool. NOTE:--In order to describe the intended end user role through the skills he is intended to master, or the tasks he is intended to be able to accomplish, the category 9:Classification can be used.			
5.6	Context	smallest permitted maximum: 10 items	No	-
	The principal environment within which the learning and use of this learning object is intended to take place.			
5.7	Typical Age Range	smallest permitted maximum: 5 items	No	7-9, 0-5, 15, 18-, (en,suitable for children over 7), (en,adults only)
	Age of the typical intended user. This data element shall refer to developmental age, if that would be different from chronological age. NOTE:--The age of the learner is important for finding learning objects, especially for school age learners and their teachers. When applicable, the string should be formatted as minage-maxage or minage-. (NOTE:--This is a compromise between adding three subfields (minAge, maxAge and description) and having just a free text field.) NOTE:--More specific schemes for what this data element tries to cover (such as various reading age or reading level schemes, IQ's or developmental age measures) should be represented through the 9:Classification category			
5.8	Difficulty	1	N/A	-
	This data element defines how hard it is to work through this learning object for the typical target audience.			
5.9	Typical Learning Time	1	N/A	PT1H30M, PT1M45S
	Approximate or typical time it takes to work with this learning object.			
5.10	Description	1	N/A	Teacher guidelines that come with a textbook.
	Comments on how this learning object is to be used.			
5.11	Language	smallest permitted maximum: 10 items	No	"en", "en-GB", "de", "fr-CA", "it"
	The human language used by the typical intended user of this learning object. NOTE:--As an example, for a learning object in French, intended for English speaking students, the value of 1.4:General.Language will be French, and the value of 5.11:Educational.Language will be English.			

6	Rights	1	N/A	-
	This category describes the intellectual property rights and conditions of use for this learning object. NOTE:--The intent is to reuse results of ongoing work in the Intellectual Property Right and e-commerce communities. This category currently provides the absolute minimum level of detail only.			
6.1	Cost	1	N/A	-
	Whether use of this learning object requires payment.			
6.2	Copyright and Other Restrictions	1	N/A	-
	Whether copyright or other restrictions apply to the use of this learning object.			
6.3	Description	1	N/A	-
	Comments on the conditions of use of this learning object.			
7	Relation	smallest permitted maximum: 100 items	No	-
	This category defines the relationship between this learning object and other learning objects, if any. To define multiple relationships there may be multiple instances of this category. If there is more than one target learning object, then each target is covered by a new relationship instance.			
7.1	Kind	1	N/A	-
	Nature of the relationship between this learning object and the target learning object, identified by 7.2:Relation.Resource.			
7.2	Resource	1	N/A	-
	The target learning object that this relationship references.			
7.2.1	Identifier	1	N/A	-
	Unique Identifier of the target learning object. This is not and shall not be used.			
7.2.2	Description	1	N/A	-
	Description of the target learning object.			
7.2.3	Catalog Entry	smallest permitted maximum: 10 items	No	-
	See 1.3:General.CatalogEntry.			
8	Annotation	smallest permitted maximum: 30 items	No	-
	This category provides comments on the educational use of this learning object, and information on when and by whom the comments were created. When multiple annotations are needed, multiple instances of this category may be used. This category enables educators to share their assessments of learning objects, suggestions for use, etc.			
8.1	Person	1	N/A	-
	The person who created this annotation.			
8.2	Date	1	N/A	-
	Date that this annotation was created.			
8.3	Description	1	N/A	-
	The content of this annotation.			

9	Classification	smallest permitted maximum: 40 items	No	-
	This category describes where this learning object falls within a particular classification system. To define multiple classifications, there may be multiple instances of this cat.			
9.1	Purpose	1	N/A	-
	The purpose of classifying this learning object.			
9.2	Taxon Path	smallest permitted maximum: 15 items	No	-
	This sub-category describes a taxonomic path in a specific classification system. Each succeeding level is a refinement in the definition of the higher level. There may be different paths, in the same or different classifications, which describe the same characteristic.			
9.2.1	Source	1	N/A	(en,ACM), (en,MESH), (en,ARI-ADNE)
	The name of the classification system. This data element may use any recognized "official" taxonomy or any user-defined taxonomy. NOTE:--An indexation or query tool may provide the top-level entries of a well-established classification (LOC, UDC, DDC, etc.).			
9.2.2	Taxon	smallest permitted maximum: 15 items	Yes	{[12,(en,Physics)], [23,(en,Acoustics)], [34,(en,Instruments)], [45,(en,Stethoscope)]}
	This sub-category describes a particular term within a taxonomy. A taxon is a node that has a defined label or term. A taxon may also have an alphanumeric designation or identifier for standardized reference. Either or both the label and the entry may be used to designate a particular taxon. An ordered list of taxons creates a taxonomic path, i.e. "taxonomic stairway": this is a path from a more general to more specific entry in a classification. A TaxonPath shall have a depth from 1 to 9. Normal values should be defined as values between 2 and 4.			
9.2.2.1	Id	1	N/A	320, 4.3.2 , BF180
	The identifier of the taxon, such as a number or letter combination provided by the source of the taxonomy.			
9.2.2.2	Entry	1	N/A	(en, Medical Sciences)
	The textual label of the taxon.			
9.3	Description	1	N/A	(en, A medical instrument for listening called a stethoscope.)
	This is the description of the learning object relative to the stated 9.1:Classification.Purpose of this specific classification, such as discipline, idea, skill level, educational objective, etc.			
9.4	Keywords	smallest permitted maximum: 40 items	Yes	-
	These are the keywords and phrases descriptive of the learning object relative to the stated 9.1:Classification.Purpose of this specific classification, such as accessibility, security level, etc., most relevant first.			

Anhang B - Textbasiertes Bildarchiv mit Toolbox

B.1 Installations CD „Textbasiertes Bildarchiv mit Toolbox“

Mit der CD „Textbasiertes Bildarchiv mit Toolbox“ können folgende Module, die im Rahmen der Vorarbeiten (SENTIMED) erstellt wurden, auf den Betriebssystemen Windows95/98/ME/NT/2000 installiert werden:

- Erfassung der Grunddaten durch CGI-Skripte inkl. Apache – WWW-Server (Abschnitt 5.2.3, Seite 60)
- Erfassung der restlichen Daten (Abschnitt 5.2.7, Seite 63)
- Semantisches Netz (Abschnitt 5.3, Seite 65)
- Vorlesungserstellung (Abschnitt 5.4, Seite 74)
- Vorlesungspräsentation (Abschnitt 5.5, Seite 79)

Zur Installation ist *Setup.exe* auf der CD auszuführen und den Anweisungen am Bildschirm zu folgen. Zum Starten der Module werden im Windows – Startmenu die entsprechenden Einträge erzeugt.

B.2 Verzeichnisstruktur

Für die Verschlagwortung und Suche im Bildarchiv wird der unten abgebildete Verzeichnisbaum angelegt.

Basisverzeichnis	# das in <i>cgi.ini</i> in Zeile 1 angegebene Verzeichnis
-eingabe	# Konf. Dateien, CGI-Skripte zur Verschlagwortung, Passwort-geschützt
-admin	# Skripte/Batch-Dateien für Administrator-Funktionen,
-scan	Admin-Passwort-geschützt
-scan1	# einzelne Scan-Verzeichnisse, in die jeweils Serien
-scan2	# von Bildern gescant werden
...	
-present	# Präsentations-Verzeichnis mit Zugriff für alle Nutzer
-daten	# Tabellen
-scans	
-scan1	# Verzeichnisse entsprechen denen aus eingabe/scan, hier sind die Bilder
-scan2	# aber als JPEGs abgelegt
...	
-suche	# CGI-Skripte für Bildarchiv-Suche

Da beides zum großen Teil mit Hilfe eines WWW-Servers und CGI-Skripten realisiert wurde, sind einzelne Verzeichnisse ohne Probleme durch verschiedene Passworte zu sichern, d.h. der Administrator hat Zugriff auf alle Verzeichnisse, die Gruppe der Verschlagwörter nur auf das *eingabe*-Verzeichnis (inkl. Unterverzeichnisse), die restlichen Nutzer auf das *present*-Verzeichnis (inkl. Unterverzeichnisse).

B.3 Grundverschlagwortung

Entwicklungs- umgebung	Windows NT mit: Apache-WWW-Server (http://www.apache.org/) ActivePerl (http://www.activestate.com/)
Siehe auch	Abschnitt 5.2, Seite 57

Die Grundverschlagwortung benutzt die Konfigurationsdateien *cgi.ini*, *tabellen.ini* und *felder.txt*:

a) *cgi.ini*

Zeile 1 gibt den Basispfad bis zu der im vorigen Abschnitt beschriebenen Verzeichnisstruktur an. Zeile 2 gibt die Extension der Bilddateien, die in die Datenbank aufgenommen werden sollen an (also das Bildformat, in dem die Bilder eingescannt wurden).

Beispiel:
 c:\sentimed\bilder\
 .bmp

b) *tabellen.ini*

In dieser Datei kann festgelegt werden, in welcher Tabellen-Datei welche Datenfelder abgelegt werden. Jede Zeile repräsentiert eine Tabelle, die Felder einer Zeile sind durch Semikolon getrennt. Das erste Feld ist der Name der Tabelle, alle weiteren Felder der Zeile sind die jeweiligen Datenfelder (Spalten) der Tabelle. Eine Sonderstellung nimmt die Spalte <src> ein. Sie ist der Pfad zum Bild relativ zu dem in *cgi.ini* angegebenen Basispfad (d.h. der absolute Pfad eines Bildes wird durch Aneinanderhängen des Basispfads und des in <src> angegebenen Pfads ermittelt).

Beispiel:
 tabelle.txt;icd10;icd9;diaka;dreh
 other.txt;pat;bem
 transtable.txt;<src>

c) *felder.txt*:

In dieser Datei kann konfiguriert werden, wie welche Felder erfasst werden sollen. Die Eingabe ist auf Textfelder und Drop-Down Listen eingeschränkt.

Eine Zeile *Feldname/Text/Beschreibung/Länge* wird bei der Erfassung als einzelliges Eingabefeld, eine Zeile *Feldname/drop/Beschreibung/wert1;wert2;wert3;...* als Drop-down-Liste dargestellt.

Beispiel:

```
icd10|Text|ICD10|30 # Zeile 1
pat|Text|Pat.|30
diaka|Text|Diakasten|10
bem|Text|Bem.|30
spieg|drop|Spiegeln|-; horz;vert
dreh|drop|Drehen|0;90;180;270 # Zeile 6
```

Die Umsetzung bei der Datenerfassung sieht dann so aus:

ICD10	<input type="text" value="L63.100"/>
Pat.	<input type="text" value="1024/53"/>
Diakasten	<input type="text" value="B1/1"/>
Ben.	<input type="text" value="Bemerkung ..."/>
Spiegeln	<input type="text" value="-"/>
Drehen	<input type="text" value="0"/>
	<ul style="list-style-type: none"> 0 90 180 270

Im folgenden werden die Skripte zur Grundverschlagwortung kurz beschrieben. Die Liste ist nach Verzeichnissen sortiert, innerhalb der Verzeichnisse alphabetisch. Einstiegspunkt in die Verschlagwortung ist das Skript *eingabe/checkDir.cgi*. Alle Skripte haben Zugriff auf die Konfigurationsdateien und verwenden diese, falls nötig.

- *eingabe/bearbeite.cgi*

startet die Verschlagwortung eines Verzeichnisses, ruft *konvJPG.cgi* auf, falls JPEG-Versionen für die Anzeige bei der Verschlagwortung fehlen. Nach der Abschluß dieser Vorbereitungen wird automatisch *next.cgi* zur Verschlagwortung des ersten Bildes aufgerufen.

Aufruf durch: Anklicken eines Links (von *checkDir.cgi* generiert)

ÜbergabeString: {v}&{p}

{v}: Verzeichnis, das bearbeitet werden soll

{p}: Präfix, mit dem die Bilddateien beginnen

- *eingabe/checkDir.cgi*

scannt die Unterverzeichnisse des Basisverzeichnisses. Das Skript stellt dabei fest, wieviele Bilder im jeweiligen Verzeichnis liegen, ob das Verzeichnis zur Bearbeitung freigegeben ist oder nicht und wieviele Bilder bereits verschlagwortet sind. Die Informationen werden dann als HTML-Tabelle ausgegeben.

Aufruf durch: direkt als URL (*checkDir.cgi* ist das Startskript)

ÜbergabeString: -

- *eingabe/createOver.cgi*

generiert eine Übersicht über die in einem Verzeichnis abgelegten Bildobjekte inkl. der eingegebenen Daten.

Aufruf durch: Anklicken eines Links (von *checkDir.cgi* generiert)

ÜbergabeString: {v}&{p}

{v}: Verzeichnis, das bearbeitet werden soll

{p}: Präfix, mit dem die Bilddateien beginnen

- *eingabe/eingabe.cgi*

zeigt die Daten eines Bildobjekts in einem HTML-Formular zur Verschlagwortung an (siehe *felder.txt*).

Aufruf durch: *next.cgi*

ÜbergabeString: siehe *next.cgi*

- *eingabe/konvJPG.cgi*

konvertiert ein Bild nach JPG (für die Anzeige in der Verschlagwortung)

Aufruf durch: *bearbeite.cgi*

ÜbergabeString: {v}&{p}&{n}&{g}

{v}: Verzeichnis, in dem das Bild liegt

{p}: Präfix, mit dem die Bilddatei beginnt

{n}: Bildnummer der Serie

{g}: Anzahl der Bilder in der Serie

- *eingabe/next.cgi*

generiert ein Frameset für ein Bildobjekt aus *eingabe.cgi* (HTML-Formular zur Eingabe der Daten) und *showRotImg.cgi* (Anzeige eines evtl. gedrehten Bildes)

Aufruf durch: Anklicken eines Links (von *createOver.cgi* oder *eingabe.cgi* erzeugt), automatisch durch *save.cgi* (nach Speicherung eines Datensatzes)

ÜbergabeString: {v}&{p}&{n}

{v}: Verzeichnis, in dem das Bild liegt

{p}: Präfix, mit dem die Bilddatei beginnt

{n}: Bildnummer der Serie

- *eingabe/save.cgi*

speichert die im Formular eingegebenen Daten zu einem Bildobjekt. Die Daten werden nicht direkt in den in *tabellen.ini* angegebenen Tabellen gespeichert, sondern es wird im gleichen Verzeichnis, in dem auch die Bilddatei liegt eine XML-ähnliche Datei (getagter Text) mit den Daten erzeugt. In die Tabellen werden diese Daten erst eingefügt, wenn der Administrator ein Update (siehe *update.bat*) ausführt.

Aufruf durch: Klick auf von *eingabe.cgi* erzeugten Button

ÜbergabeString: scanDir={v}&praeifix={p}&number={n}&{feldx}={wertx}& ...

{v}: Verzeichnis, in dem das Bild liegt

{p}: Präfix, mit dem die Bilddatei beginnt

{n}: Bildnummer der Serie

{feldx}: Feldname des x.ten Feldes (siehe *feldname.txt*)

{wertx}: Wert des x.ten Feldes

- *eingabe/showRotImg.cgi*

zeigt ein evtl. gedrehtes/gespiegeltes Bild an.

Aufruf durch: *eingabe.cgi* (durch Javascript, wenn vom Nutzer ein anderer Drehwinkel gewählt wird), *next.cgi*

ÜbergabeString: {d}&{f}&{src}&{dest}

{d}: Drehung des Bildes im Uhrzeigersinn (0|90|180|270)

{f}: 0: keine, 1: horizontale, 2: vertikale Spiegelung

{src}: Dateiname des Bildes inklusive absolutem Pfad

{dest}: Dateiname der Bildes, das zur Vorschau verwendet wird

- *admin\freigabe.cgi*

gibt ein Verzeichnis für die Verschlagnwortung frei oder sperrt es.

Aufruf durch: Anklicken eines Links (von *checkDir.cgi* generiert)

ÜbergabeString: {v}&{f}

{v}: Verzeichnis, das freigegeben/gesperrt werden soll

{f}: 0: {v} wird freigegeben, 1: {v} wird gesperrt

- *admin\prepare.bat;admin\prepare.pl*

bereitet die Verschlagnwortung eines/mehrerer Verzeichnisse vor. *prepare.bat* ist lediglich eine Batch-Datei, die *prepare.pl* startet. *prepare.pl* durchsucht alle Unterverzeichnisse des Basispfads nach Bilddateien mit der in *cgi.ini* angegebenen Extension, testet, ob es bereits eine JPEG-Version des Bildes gibt. Ist dies nicht der Fall, wird die JPEG-Version generiert.

Parameter: keine

Bemerkung: Die hier in kleiner Auflösung generierten JPGs sind nicht die später im Bildarchiv verwendeten Versionen, sondern werden nur zur Anzeige während der Verschlagnwortung benötigt.

- *admin\update.bat;admin\all.cgi*

Update.bat ruft *all.cgi* auf. *all.cgi* fügt die Bildobjekte der in *verz.txt* angegebenen Verzeichnisse in die Tabellen ein, konvertiert und kopiert die Bilder in ein entsprechendes Unterverzeichnis des Präsentationsverzeichnisses (*present\scan*). Bereits vorhandene

Verzeichnisse werden nur überschrieben, wenn als erster Parameter 1 übergeben wird!

Es wird eine Datei *update.log* erzeugt, in der evtl. Fehler dokumentiert werden.

verz.txt ist eine durch Zeilenumbruch getrennte Liste der aufzunehmenden Verzeichnisse (relativ zum Basispfad).

Parameter: [1]: optional, bereits vorhandene Daten werden überschrieben.

- *present\suche\suchform.cgi*

generiert ein HTML-Formular für eine Datenbankabfrage mit den in *felder.txt* angegebenen Feldern. Die Suche wird nach Anklicken des Submit-Buttons in *suche.cgi* durchgeführt.

ÜbergabeString: -

- *present\suche\suche.cgi*

führt eine Suche in den Datensätzen durch. Dazu wird für jedes Datenfeld, das vom HTML-Formular (siehe *suchform.cgi*) nichtleer übergeben wurde, eine Substring-Suche in den entsprechenden ASCII-Tabellen gestartet. Als Ergebnis wird eine Übersicht von Thumbnails mit den dazu erfaßten Daten generiert.

ÜbergabeString: der vom Formular aus *suchform.cgi* generierte ÜbergabeString.

B.4 Erweiterte Verschlagnwortung

Entwicklungs- umgebung	IBM Visual Age for Java (JDK 1.2.2) unter Windows ME/2000
Start	Aufruf der main-Methode von <i>sentimedeingabe.SENTIMEDBrowser</i>
Parameter	keine
Klassenpfad	Jar-Datei <i>sentimed.jar</i>
Siehe auch	Abschnitt 5.2.7, Seite 63

Die Applikation liest bei Programmstart die Datei *SENTIMEDBrowser.ini*. Sie besteht aus folgenden 3 Zeilen:

```
Pfad zu den Daten
Pfad zu den Bildern
Auflösung, in der Bilder in der Applikation angezeigt werden
```

Beispiel:

```
g:\cgi\present\daten\
g:\cgi\present\scans\
300x300
```

Die wichtigsten Klassen der Applikation im Paket *sentimedeingabe* sind:

– *SENTIMEDBrowser:*

Hauptklasse, enthält alle anderen Komponenten.

– *ImgData*

repräsentiert jeweils ein Bildobjekt mit den aktuellen Daten.

– *JDataList*

Unterklasse von *javax.swing.JList*, dient zur Anzeige der meisten Picklisten (zum Beispiel Effloreszenzen) in den Karteikarten.

B.5 Semantisches Netz

Generierung: Die Skripte zur Erstellung des semantischen Netzes wurden von S. Krippner im Rahmen einer Diplomarbeit am Lehrstuhl für Informatik II der Universität Würzburg erstellt und sind in [13] detailliert beschrieben.

Applet zur Bildbetrachtung

Entwicklungs- umgebung	JDK 1.1 unter Windows NT 4.0
Start	Einbinden auf HTML-Seite
Parameter	siehe unten (Einbindung in HTML)
Klassenpfad	Jar-Datei <i>sentimed.jar</i>
Siehe auch	Abschnitt 5.3.3, Seite 70

Das Applet wird folgendermaßen in HTML-Seiten eingebunden:

```
<APPLET code="JPGViewer" archive="sentimed.jar"
  name="JPGViewer" width=100 height=100>
<PARAM NAME=DATAPATH VALUE="sentimed/Daten/">
<PARAM NAME=IMGPATH VALUE="sentimed/bilder/">
<PARAM NAME=SERVER VALUE="linux">
<PARAM NAME=HIGHRESURL
  VALUE="http://127.0.0.1:8090/cgi-bin/server/">
<PARAM NAME=XMULT VALUE="1.554">
<PARAM NAME=YMULT VALUE="2.086">
<PARAM NAME=DIAGSEITEN VALUE="sentimed/DiagnosenSeiten/">
</APPLET>
```

Dabei haben die Parameter folgende Bedeutung:

- **DATAPATH**

Verzeichnis, in dem die erfaßten Daten zu den Bildern liegen, relativ zur URL der HTML-Seite

- **IMGPATH**

Verzeichnis, unter dem die Bilder liegen, relativ zur URL der HTML-Seite

- SERVER

windows: Server unter Windows (gleichzeitig wird davon ausgegangen, daß die Bilder Dongle-kodiert sind, siehe Abschnitt 5.1, S. 56)

linux: WWW-Server unter Linux

- HIGHRESURL

URL, unter der die Skripte *exists.cgi* und *cut.cgi* zum serverseitigen Vergrößern von Bildern liegen (s.u.)

- XMULT/YMULT:

Verhältnis zwischen Breite (Höhe) der Bilder auf dem Server und der angezeigten Bilder in Pixeln. Es wird benötigt, um die eingezeichneten Rechtecke auf die Koordinaten des Bildes auf dem Server zu skalieren.

- DIAGSEITEN

Verzeichnis, in dem die HTML-Seiten zu den Diagnosen (innere Knoten des semantischen Netzes) liegen, relativ zur URL der HTML-Seite. Mit Hilfe dieser Angabe kann ein Link von jedem Bild zur entsprechenden Diagnosen-seite des semantischen Netzes angelegt werden.

Im Bildbetrachter wird ein Bild geladen, indem über Javascript die Methode *showFrame(String)* der Klasse *JPGViewer* aufgerufen wird.

Hier eine knappe Erläuterung der zentralen Klassen:

- *JPGViewer*

Hauptklasse des Applets mit den Methoden *init()*, *start()* und *stop()*. Außerdem enthält sie die Methode *showFrame(String id)*, durch die ein Fenster mit dem zur übergebenen Id gehörenden Bild geöffnet wird.

- *ZoomFrame*

berechnet Vergrößerungen bzw. holt vergrößerte Ausschnitte vom Server und zeigt sie jeweils in einem eigenen Frame an.

Clientseitige Vergrößerung:

Zur Berechnung von Zooms mit doppelter Kantenlänge wird als Grundlage eine Arbeit von Said und Pearlman [15] verwendet. Die Beschreibung des Algorithmus ist in Abschnitt 5.3.3, Seite 70 zu finden.

Serverseitige Vergrößerung:

Um Vergrößerungen aus einer hohen Auflösung des Bildes auf einem Server auszuscheiden, wird zunächst durch den Aufruf des CGI-Skripts *exists.cgi* getestet, ob das Bild auf dem Server vorhanden ist (*highResUrl* ist die in der HTML-Seite als Parameter übergebene Url):

```
URLzoomurl = new URL(highResUrl+"/exists.cgi?pfad="+image);
BufferedReader in = new BufferedReader(
    new InputStreamReader (zoomurl.openStream()));
inputLine = in.readLine();
in.close();
if (inputLine.compareTo("true") == 0)
    serverZoom=true;
else
    serverZoom=false;
```

Fällt dieser Test positiv aus, kann der Ausschnitt vom Server über das CGI-Skript *cut.cgi* angefordert werden:

```
getImage(new URL(highResUrl+
    "/cut.cgi?pfad="+image
   +"&xwert="+zx+"&ywert="+zy
   +"&b="+width+"&h="+height));
```

Das Perl-Skript *cut.cgi* verwendet die Image Magick Tools, um den Bereich aus dem Bild auszuschneiden und als JPG auf STDOUT auszugeben:

```
system "convert -crop ".$width."x".$height."
    +$xwert+$ywert $image -";
```

- *InfoPanel*

lädt aus dem im Parameter DATAPATH übergebenen Verzeichnis alle Daten zum jeweiligen Bild und zeigt sie auf Wunsch an.

- *ImageCanvas*

zeigt das Bild selbst an und hat einen MouseListener als innere Klasse, der das Einzeichnen von Rechtecken verwaltet und das Anfordern von Vergrößerungen in der Klasse *ZoomFrame* auslöst.

Applet zur PubMed-Anfrage

Entwicklungs- umgebung	JDK 1.1 unter Windows NT 4.0
Start	<i>Einbinden auf HTML-Seite</i>
Parameter	keine
Klassenpfad	Jar-Datei <i>medline.jar</i>
Siehe auch	Abschnitt 5.3.5, Seite 72

Durch die Vorgabe, daß das Applet ohne Installation weiterer Plugins lauffähig sein sollte, konnten keine Swing-Klassen, speziell kein *javax.swing.JTree*, verwendet werden. Deshalb wurde zur hierarchischen Darstellung der Journals (und zur Aufteilung in die Kategorien *Alle/Dermatologische Journals/Favoriten*) eine eigene Baumdarstellung in *Tree.java* implementiert. Hauptklasse mit der *init*-Methode ist *MedlineSearcher*.

Das Applet wird folgendermaßen in HTML-Seiten eingebunden:

```
<APPLET ARCHIVE="medline.jar" CODE="MedlineSearcher.class">  
<PARAM NAME=script0 VALUE="search_terms.cgi?C84.440">  
</APPLET>
```

Das als Parameter übergebene Skript *search_terms.cgi* ermittelt die Suchbegriffe zum übergebenen ICD-Code, die beim Start des Applets vorgeschlagen werden.

Bemerkung: Die HTML-Seite, die das Applet anzeigt, wird jeweils selbst durch ein CGI-Skript dynamisch erzeugt. Dadurch ist auch eine dynamische Vorbelegung der Suchbegriffe möglich.

Das Applet liest seine Einstellung beim Starten aus der Datei *journals.ini*, in der unter anderem das Datenverzeichnis, in dem die Hierarchie der Journals abgebildet ist, angegeben wird. In diesem Verzeichnis wird *root.index* ausgewertet, in der die obersten Hierarchieebenen sowie Verweise auf index-Dateien mit den Inhalten der einzelnen Ebenen abgelegt sind.

B.6 Vorlesungserstellung

Entwicklungs- umgebung	JDK 1.1 unter Windows NT 4.0
Start	Aufruf der main-Methode von <i>MainClass</i>
Parameter	keine
Klassenpfad	Jar-Dateien <i>SlideShow.jar</i> und <i>MainClass.jar</i>
Siehe auch	Abschnitt 5.4, Seite 74

Die Applikation liest die Einstellungen bei Programmstart aus der Datei *vto.ini* mit folgendem Aufbau:

```
[pfad]
bildpfad= {Pfad zu den Bildern}
start_praes= {Befehl zum Start des Präsentationstools}

[server]
url= {url zum Verzeichnis, in dem die CGIs zum Ausschneiden
      von Rechtecken aus hochaufgelösten Bildern liegen}
horzMult= {(float)Größe des hochaufgelösten Bilds/Größe
           des angezeigten Bilds für Bilder mit Breite > Höhe}
vertMult= {(float)Größe des hochaufgelösten Bilds/Größe
           des angezeigten Bilds für Bilder mit Breite < Höhe}
```

Die textbasierten Datenbankdaten werden aus dem aktuellen Arbeitsverzeichnis geladen.

Hier eine knappe Erläuterung der wichtigsten Klassen:

- *MainClass*
Hauptklasse, enthält alle anderen Komponenten.
- *SuchePanel*
 - Unterklasse von *javax.swing.JPanel* zur Darstellung der Suchoberfläche
 - führt Suchanfragen aus (Methode: *diaSucheStarten(int)*)
- *VAZA*
Unterklasse von *javax.swing.JPanel*, dient als Oberklasse zur Darstellung der Vor- und
Zwischenauswahl (*VAuswahl* und *ZAuswahl*)
- *PAuswahl*
Unterklasse von *javax.swing.JPanel* zur Darstellung der Präsentationsauswahl
- *HighResImage*
zeigt den *javax.swing.JFrame* für Bilder in Großdarstellung an

- *ShapePopUp*

ruft für die aktuelle Region auf dem Server das CGI zum Ausschneiden einer Region auf. Der Bildausschnitt wird vom Server übertragen und lokal auf der Platte abgelegt.

Auf Serverseite werden die bereits in Anhang B.5 beschriebenen Perl-CGI-Skripte *exists.cgi* und *cut.cgi* verwendet

B.7 Vorlesungspräsentation

Entwicklungs- umgebung	JDK 1.1 unter Windows NT 4.0
Start	Aufruf der main-Methode von <i>SlideShow</i> oder Start aus der Vorlesungserstellung
Parameter	de DE [Verzeichnis oder Datei]
Klassenpfad	Jar-Datei <i>SlideShow.jar</i>
Siehe auch	Abschnitt 5.5, Seite 79

Wird die Applikation mit einem Verzeichnis als Argument gestartet, wird in diesem nach einem vom Vorlesungserstellungstool generierten Steuerfile gesucht. Alternativ kann auch das Steuerfile selbst als Argument angegeben werden. Wird kein Argument übergeben oder wird kein Steuerfile gefunden, liest das Programm die Datei *visited.lst* ein, in die bei Programmende Name, Datum und Speicherort der Vorlesung eingetragen werden. Beim Einlesen wird der Status der einzelnen Vorlesungen ermittelt: Existiert am angegebenen Speicherort die Vorlesung noch und sind alle referenzierten Objekte vorhanden? Das Ergebnis wird in einer Tabelle ausgegeben:



Vorlesung	Verzeichnis	zuletzt präsentiert am	Bemerkung
Basaliome	F:\SENTIMED\data\s...	-	nicht gefunden
Alopecia	E:\Programm\SENT...	17.02.2002, 15:16	3 Bilder fehlen
Aids	E:\Programm\SENT...	17.02.2002, 15:16	OK

Im Vorlesungs-Steuerfile steht für jedes Bild eine Zeile, für jede Vergrößerung wird eine Zeile der Form

```
#ZOOM:x;y;breite;höhe;dateiname
```

wobei $(x;y;breite;höhe)$ das Rechteck im Originalbild definiert, dessen Zoom in *dateiname* abgelegt ist.

Beispiel:

Die Vorlesung enthält 4 Folien, für die erste sind zwei Zoom-Rechtecke definiert.

m4_00040.jpg

#ZOOM: 249;210;323;255;m4_00040z1.jpg

#ZOOM: 463;151;225;222;m4_00040z0.jpg

m4_00044.jpg

m4_00045.jpg

a3_00001.jpg

Anhang C - Studentenbefragungen

C.1 Technische Voraussetzungen und Akzeptanz

Im Zeitraum Februar bis April 2001 fand an den Hautkliniken der Universitäten Würzburg, Mannheim und Jena eine Befragung von Medizinstudenten zum Thema „Digitale Dermatologie-Vorlesungen“ statt. In einem Fragebogen sollten die Studierenden ihre computerbezogenen Fähigkeiten einschätzen, Angaben über Computer- und Internetnutzung machen und ihren Standpunkt zur Nachbereitung von Lehrveranstaltungen äußern. Daneben wurde die Meinung der Teilnehmer zu bisher durchgeführten digitalen Vorlesungen erfragt. Es folgt eine Zusammenfassung der Ergebnisse. Die Auswertung wurde von der Dipl.-Psychologin J. Rosendahl, einer Mitarbeiterin der medizinischen Psychologie der Universität Jena, vorgenommen.

Befragt wurden:

334 Studierende der Medizin

- zwischen dem 2. und 15. Semester
 - 172 Frauen und 156 Männer (6 ohne Angabe ihres Geschlechtes)
 - im Alter zwischen 19 und 39 Jahren

Fähigkeiten und technische Voraussetzungen

Wie beurteilen Sie sich selbst hinsichtlich Ihrer Fähigkeiten im Umgang mit dem Computer?

- Textverarbeitung: 65,8% schätzen sich als Fortgeschrittener bzw. versierter Nutzer ein
- Internetnutzung: 61,5% sehen sich selbst als Fortgeschrittener bzw. versierter Nutzer

	Anfänger	Fortgeschrittener	versierter Nutzer	Gesamt (fehlende Angaben)
Textverarbeitung	113 34.2%	173 52.4%	44 13.3%	330 (4)
Internetnutzung	126 38.5%	155 47.4%	46 14.1%	327 (7)

Benutzen Sie privat einen PC?

- 91,3% der Befragten haben privat einen PC
- 98,4% der befragten Studenten, die einen PC besitzen, nutzen Win32
(restliche: 1x Mac, 1x Linux, 2x Win 3.11)
- 76,6% der Studierenden, die einen PC besitzen, haben auch eine Soundkarte

Welche Möglichkeiten des Internetzuges nutzen Sie?

- 94.6% der Befragten haben einen Internetzugang
- der größte Teil der Befragten (37.7%) hat sowohl privat als auch an der Universität einen Internetzugang:

Internetzugang	N	%
ausschließlich privat	109	32.6
ausschließlich Universität	81	24.3
privat & Universität	126	37.7
kein Zugang	18	5.4
Gesamt	334	100.0

- für ihren privaten Internetzugang nutzen
81.1% Modems (mehrheitlich 56K), 13.7% ISDN und 5.4% DSL

Welche Internetangebote nutzen Sie?

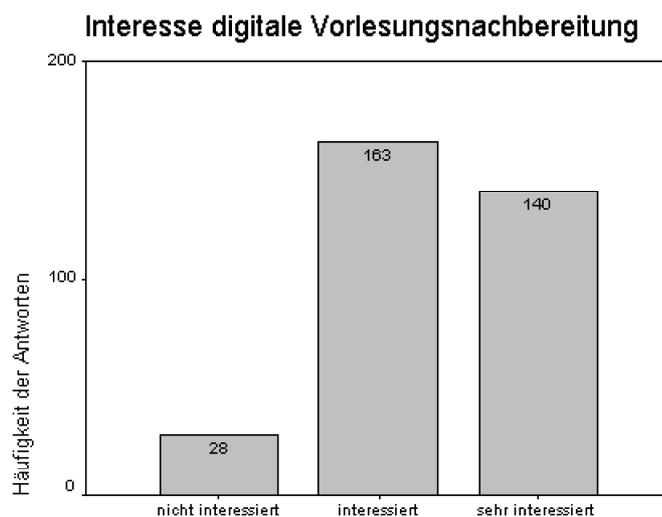
- von den befragten Studierenden nutzen 92.5% E-Mail, 89.2% WWW-Seiten und 58.3% Informationsdatenbanken

Nachbereitung von Vorlesungen

Wie interessiert wären Sie an der Möglichkeit, Vorlesungen digital mit Bildmaterial per WWW oder auf CD nacharbeiten zu können?

- 91.5% wären interessiert bzw. sehr interessiert an der Möglichkeit, Vorlesungen digital mit Bildmaterial per WWW oder auf CD nacharbeiten zu können.

nicht interessiert	interessiert	Sehr interessiert	Gesamt (fehlende Angaben)
28 8.5%	163 49.2%	140 42.3%	331 (3)

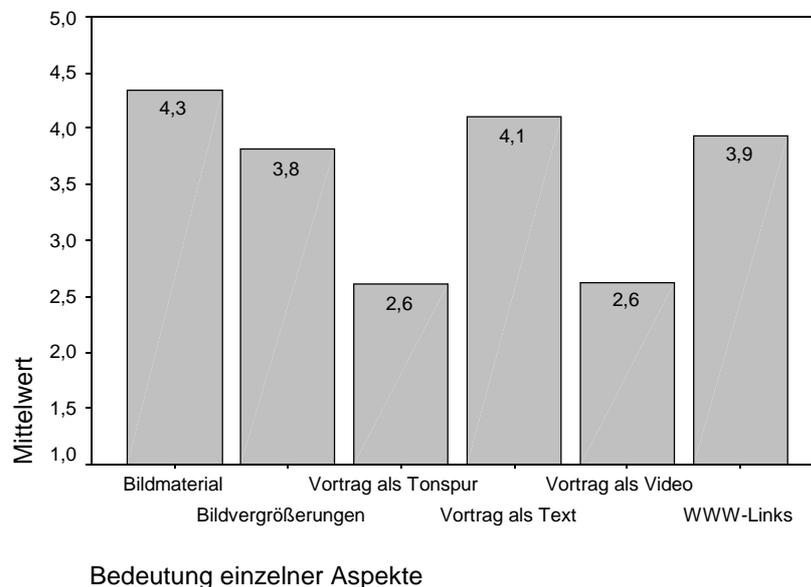


Wie beurteilen Sie die Bedeutung der folgenden Aspekte bei der Nachbereitung einer Vorlesung?

- am wichtigsten bei der Nachbereitung einer Vorlesung sind Bildmaterial und der Vortrag des Dozenten als Text
- auch die Vergrößerung von Bildern / Bildausschnitten sowie WWW-Links zu besprochenen Diagnosen sind für die Studenten von Bedeutung
- weniger wichtig ist den Studierenden der Vortrag des Dozenten als Tonspur oder als Video (Bewegtbild mit Ton)

Aspekte	durchschnittliche Bedeutung (1 = unwichtig, 5 = sehr wichtig)
Bildmaterial	4.28
Vergrößerung von Bildern / Bildausschnitten	3.78
Vortrag des Dozenten als Tonspur	2.61
Vortrag des Dozenten als Text	4.10
Vortrag des Dozenten als Video (Bewegtbild mit Ton)	2.70
WWW-Links zu besprochenen Diagnosen	3.89

Nachbereitung einer Vorlesung



digitale Vorlesungen

Haben Sie bereits digitale, mit Hilfe eines Multimedia-Projektors gehaltene Vorlesungen gehört?

- etwa die Hälfte der Befragten (168 Studierende; 50.3%) hat bereits digitale, mit Hilfe eines Multimedia-Projektors gehaltene Vorlesungen gehört

Wie beurteilen Sie die Qualität der in der Vorlesung per Multimedia-Projektor gezeigten Bilder im Vergleich zu „normalen“ Dias?

- 88.7% der Studierenden, die bereits digitale Vorlesungen gehört haben, schätzten die Qualität der gezeigten digitalen Bilder gleich gut oder besser ein als die normaler Dias

schlechter als „normale“ Dias	gleich gut	Besser als „normale“ Dias	Gesamt
19 11.3%	69 41.1%	80 47.6%	168 100.0%

Wie beurteilen Sie allgemein die Qualität digitaler, mit Hilfe eines Multimedia-Projektors gehaltenen Vorlesungen im Vergleich zu Dia-basierten?

- 96.2% der befragten Studenten, die bereits digitale Vorlesungen gehört haben, beurteilten die Qualität der digitalen Vorlesungen im allgemeinen gleich gut oder besser als die normaler Vorlesungen

schlechter als Dia-basierte	gleich gut	Besser als Dia-basierte	Gesamt
8 4.8%	63 37.5%	97 57.7%	168 100.0%

C.2 Befragung zu Lernobjekt-Typen

Diese Befragung wurde im Februar 2002 vom Institut für Medizinische Psychologie der Universität Jena in Jena mit Prototypen verschiedener Module durchgeführt. Obwohl nur 9 Probanden beteiligt waren, geben die Aussagen doch Anhaltspunkte für die weitere Entwicklung. Die folgenden Abschnitte stellen Auszüge aus dem Ergebnis dar.

Vorlesungen zur Nachbereitung

Es wurden vier unterschiedliche Versionen (Video, Folien mit Ton, Folien mit Text und Ton, Folien mit Schlagworten und Ton) vorgestellt. Es sollte angegeben werden, welche Version favorisiert wird und warum. Ebenso sollten verschiedene Eigenschaften des Moduls eingeschätzt und der Zweck der Nutzung angegeben werden.

Die beiden Versionen *Video* und *Folien mit Ton* wurden von keinem der Probanden ausgewählt und werden daher im weiteren nicht aufgeführt.

Folien mit Text und Ton favorisierten 5 Probanden. Als Vorteile dieser Version wurden genannt:

- Information ist komplett verfügbar

- Text ist ausführlicher als Schlagworte
- Text nötig zur besseren Erschließung des Stoffs
- Lesen besser als Zuhören
- Ton fördert Konzentration

Folien mit Schlagworten und optionalem Ton fanden 4 Befragte am besten und gaben folgende Vorteile an:

- Schlagworte fassen Gesagtes zusammen und sind ausreichend
- Stichpunkte/ Bilder bringen Gesagtes auf den Punkt

Allgemeine Beurteilung der Lernobjekt-Typen

Die Studenten sollten auch allgemein ihre Meinung zu den vorgestellten Lernobjekt-Typen äußern und mögliche Anwendungsgebiete, Vorteile und Nachteile der Nutzung sowie Änderungswünsche angeben.

Bemerkung: Die des öfteren genannten Fragen des Gegenstandskatalogs (GK) sind die Originalprüfungsfragen des 2. Staatsexamens und sind in Buchform zum Beispiel in der sogenannten schwarzen Reihe des Thieme-Verlags [21] erhältlich. Das Bettentestat ist eine Prüfungsform (in der Hautklinik der Universität Jena), in der Studenten am Patientenbett befragt werden.

a) Vorlesung zur Nachbereitung:

Vorteile / Anwendungsgebiet: als Überblick; Bilder anschauen; Vorlesungsbesuch ersetzen; wichtig zur Wiederholung der Lerninhalte

Nachteile / Änderungswünsche: Bild von Dozent einfügen

b) Lerntexte:

Vorteile / Anwendungsgebiet: zum Nachschlagen / Überblick

Nachteile / Änderungswünsche: zu umfangreich; Bücher werden bevorzugt; nützlich nur wenn an GK-Fragen orientiert

c) Fallbeispiele:

Vorteile / Anwendungsgebiet: nützlich zur Ergänzung

Nachteile / Änderungswünsche: zu speziell; als Text verfaßt: Bearbeitung mit zuviel Zeitaufwand (Vorschlag: in Schlagworten verfassen); Links zu Vorlesungen sinnvoll

d) Bilderquiz:

Vorteile / Anwendungsgebiet: nützlich zur Vorbereitung des Bettentests und "nebenbei"; besonders geeignet für visuellen Lerntyp

Nachteile / Änderungswünsche: zur Prüfungsvorbereitung werden andere Wege bevorzugt; zu zeitraubend; große Auswahl an Antwortmöglichkeiten und großer Bilderpool erwünscht; Bildbeschreibung selber geben (vorgegebene Liste); Freitextangaben wünschenswert

e) **Übungsfragen:**

Vorteile / Anwendungsgebiet: für Wissensabfrage gut geeignet; zur Prüfungsvorbereitung redundant, da GK-Fragen vorhanden; nützlich nur wenn GK-relevante Fragen Verlinkung mit Lerninhalten erwünscht

f) **Fallquiz:**

Vorteile / Anwendungsgebiet: zur Vorbereitung auf Bettentestat;

Nachteile / Änderungswünsche: nicht prüfungsrelevant; Feedback nur auf Wunsch; viele Bilder erwünscht; nur für große Krankheitsbilder /wichtigste Hauterkrankungen

Anhang D - Lernobjekt – Datenbank

Das folgende Entity-Relationship-Diagramm (Abbildung 56) beschreibt die Datenbankstruktur. Zur besseren Übersichtlichkeit wurden nicht alle Tabellen (Quizelemente, Moulagen, etc) aufgenommen.

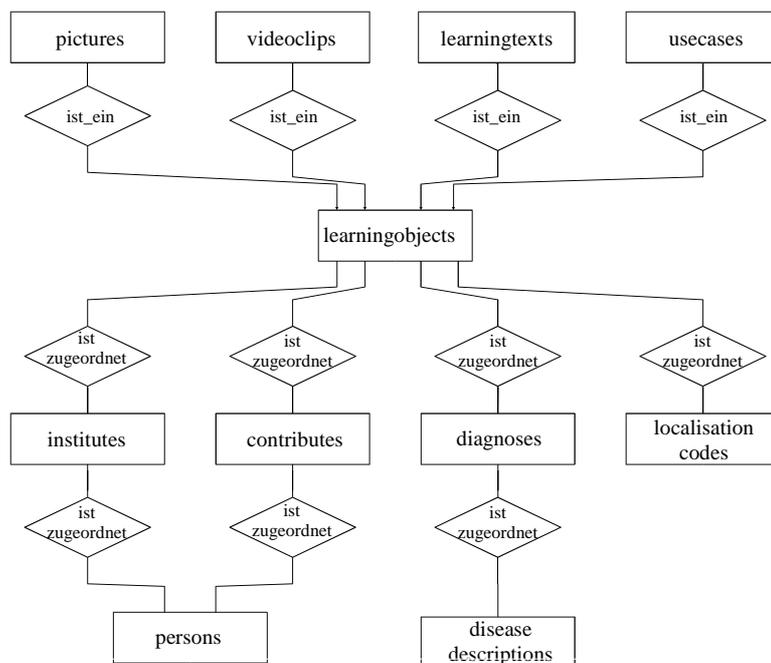


Abbildung 56: Entity-Relationship Diagramm

Im folgenden werden die einzelnen Datenbanktabellen aufgeführt. In der ersten Zeile steht jeweils der Name der Tabelle und eine kurze Erläuterung. Die darauf folgenden Zeilen repräsentieren die Spalten der Datenbanktabelle. In den Tabellen werden jeweils die verwendeten MySQL-Datentypen angegeben, die die folgende Tabelle kurz beschreibt:

tinyint	Integer, -128 bis 127
smallint	Integer, -32768 bis 32767
mediumint	Integer, -8388608 bis 8388607
float	Fließkommazahl
char	1 Zeichen
varchar(M)	Text, max M Zeichen
tinytext	Text, max 255 Zeichen
text	Text, max 65535 Zeichen
timestamp	Das Datum von Insert AND Update-Aktionen
enum('w1','w2',...)	Aufzählung (erlaubte Werte w1,w2,...)

D.1 Allgemeine Tabellen

versions: Aktuelle Version der Datenbankversion sowie des Powerpoint-AddIns.		
Base	float	Versionsnummer der Datenbank
PowerpointAddIn	float	Version (z.B 2.3) Das AddIn fragt diese Version bei der Datenbank ab. Ist die hier gespeicherte Versionsnummer höher als die des AddIns, wird der Nutzer darauf aufmerksam gemacht. Hat sich auch die Vorkommastelle geändert, arbeitet das AddIn mit dieser Datenbankversion nicht mehr zusammen! Beispiel: hier: 2.4, AddIn: 2.3 => AddIn ist nicht aktuell, arbeitet aber mit dieser DB hier: 3.0, AddIn: 2.4 => AddIn arbeitet nicht mit dieser DB
AddInLocation	text	An dem hier angegebenen Pfad befindet sich das AddIn mit (Version → <i>version.PowerpointAddIn</i>). So besteht die Möglichkeit, dem Nutzer anzuzeigen, woher er am besten die aktuellste Version des AddIns bekommen kann.

specialtables: Liste der vorhandenen Lernobjekt-Typen mit Angabe der jeweiligen Haupttabelle		
Special_ID	tinyint	auto_increment
TableName	varchar(30)	Name der Haupttabelle
JavaClass	varchar(50)	Für diesen Lernobjekt-Typ implementierte Java-Klasse
Description_de	tinytext	deutsche Bezeichnung des Lernobjekt-Typs
Description_en	tinytext	englische Bezeichnung des Lernobjekt-Typs

institutes: Institute, von denen Lernobjekte in der Datenbank enthalten sind		
Institute_ID	tinyint	auto_increment
Name	varchar(250)	Bezeichnung des Instituts

learningobjects: zentrale Tabelle, in der alle Lernobjekte gelistet werden.		
LO_ID	mediumint	auto_increment
Title_de	text	deutscher Titel
Title_en	text	englischer Titel
Filename	tinytext	der Filename (ohne Pfad!) des Lernobjekts
Version	varchar(50)	Version dieses Lernobjekts
Status	tinyint	unbenutzt
Special_ID	tinyint	Angabe des Lernobjekt-Typs (<i>specialtables.Special_ID</i>)
Institute_ID	tinyint	Welchem Institut gehört das Objekt (\rightarrow <i>institutes.Institute_ID</i>)?
Context	tinyint	Zielgruppe (Ärzte/Studenten/...)

matching: Wurde die (vorliegende) Datenbank aus anderen Datenbanken gemergt, wird hier die Zuordnung zwischen LO_ID in dieser Datenbank und der Original-LO_ID abgelegt.		
Local_LO_ID	mediumint	\rightarrow <i>learningobjects.LO_ID</i>
Original_LO_ID	mediumint	ID in der Herkunftsdatenbank

locations: Verwaltung aller Pfade, in denen Lernobjekte (relativ zum Server-Lernobjektpfad, ohne Filename) liegen. Sind in einem Verzeichnis mehrere Lernobjekte, muß der Pfad nur einmal abgelegt werden!		
Location_ID	smallint	auto_increment
Location	varchar(250)	Verzeichnispfad

lo_locations: Zuordnung Lernobjekt zu Speicherort		
LO_ID	mediumint	→ <i>learningobjects.LO_ID</i>
Location_ID	smallint	→ <i>locations.Location_ID</i>
NR	tinyint	unbenutzt

persons: Verwaltung der Personen, die mit den Lernobjekten in irgendeiner Weise zu tun haben		
Person_ID	smallint	auto_increment
Name	varchar(250)	Name der Person
Institute_ID	tinyint	Welchem Institut gehört die Person an (→ <i>institutes.Institute_ID</i>)?

annotations: Bemerkungen zu einem Lernobjekt		
LO_ID	mediumint	→ <i>learningobjects.LO_ID</i>
NR	tinyint	lfd. Nummer für Bemerkung zu diesem Objekt
Person_ID	smallint	→ <i>persons.Person_ID</i>
Date	timestamp	Datum, an dem die Bemerkung gemacht wurde
Description	text	Text der Bemerkung
Language	enum	In welcher Sprache liegt die Bemerkung vor?

roles: Liste aller möglichen Rollen, die eine Person bei einem Lernobjekt spielen kann.		
Role_ID	tinyint	auto_increment
Description_de	tinytext	deutsche Beschreibung
Description_en	tinytext	englische Beschreibung

contributes: Zuordnung der Beteiligung einer Person an einem Lernobjekt inkl. Rolle		
LO_ID	mediumint	→ <i>learningobjects.LO_ID</i>
NR	tinyint	lfd. Nummer
Role_ID	tinyint	→ <i>roles.Role_ID</i>
Person_ID	smallint	→ <i>persons.Person_ID</i>
Date	date	Datum, an dem eine Person eine Rolle übernommen hat.

diseases: Liste aller Diagnosen (aus dem DDK, angereichert durch „eigene“ Diagnosen)		
Disease_ID	smallint	auto_increment
ICD9	char	ICD-Code der 9. Revision
ICD10	char	ICD-Code der 10. Revision
Xtended	enum	Wurde die Diagnose selbst hinzugefügt (also nicht aus dem DDK)?
Used	mediumint	Anzahl der Lernobjekte, denen die Diagnose zugeordnet ist

diseasedescriptions: Verwaltung der Bezeichnung von Diagnosen		
Disease_ID	smallint	→ <i>diseases.Disease_ID</i>
NR	tinyint	laufende Nummer (für die gleiche Diagnose gibt es mehrere Bezeichnungen), die Reihenfolge ist wichtig, NR=0 bestimmt die gebräuchlichste Bezeichnung
Description	varchar(250)	Bezeichnung der Diagnose
Language	enum	Sprache

diseasegroups: Die Abbildung der Diagnosenhierarchie		
Group_ID	tinyint	auto_increment
Description	varchar(255)	Bezeichnung der Hierarchie-Ebene
Intervall	varchar(128)	Intervall der ICD10-Codes, die dieser Hierarchie-Ebene zugeordnet sind

diffdiseases: Verwaltung der Differentialdiagnosen		
Disease_ID	mediumint	→ <i>diseases.Disease_ID</i>
DiffDisease_ID	mediumint	ID der Differentialdiagnose (→ <i>diseases.Disease_ID</i>)

diagnoses: Zuordnung von Diagnosen zu Lernobjekten		
LO_ID	mediumint	→ <i>learningobjects.LO_ID</i>
Disease_ID	smallint	→ <i>diseases.Disease_ID</i>
NR	tinyint	lfd. Nummer (mehrere Diagnosen sind pro Lernobjekt möglich), dadurch ist die „Gewichtung“ von Diagnosen möglich (zum Beispiel Hauptdiagnose: NR=0)

pe_codes/se_codes/oe_codes/dispersioncodes/localisationcodes/shapecodes/colorcodes: Die Codes für die verschiedenen Effloreszenzen, Verteilung, Lokalisationen, Formen und Farben.

PE_ID/SE_ID/OE_ID/ Dispersion_ID/ Localisation_ID/Shape_ID/ Color_ID	tinyint	auto_increment
Code	char	Code
Description_de	tinytext	deutsche Bezeichnung
Description_en	tinytext	englische Bezeichnung
Used	mediumint	Anzahl der Lernobjekte, denen der Code zugeordnet wurde

primeefflorescences/secondaryefflorescences/otherefflorescences/dispersions/localisations/shapes/colors: Zuordnung der Effloreszenzen / Verteilung / Lokalisationen / Formen und Farben zu Lernobjekten

LO_ID	mediumint	→ <i>learningobjects.LO_ID</i>
PE_ID/SE_ID/OE_ID/ Dispersion_ID/Localisation_ID/ Shape_ID/Color_ID	tinyint	z.B.: → <i>pe_codes.PE_ID</i>

relationtypes: Typen von Beziehungen zwischen Lernobjekten

Relation_ID	tinyint	auto_increment
Description_de	tinytext	deutsche Beschreibung der Beziehung (zum Beispiel: <i>ist_Teil_von</i>)
Description_en	tinytext	englische Beschreibung

relations: Zuordnung von Beziehungen zwischen Lernobjekten

LO_ID	mediumint	→ <i>learningobjects.LO_ID</i>
Resource_ID	mediumint	ID des „Ziellernobjekts“ (→ <i>learningobjects.LO_ID</i>)
Relation_ID	tinyint	→ <i>relationtypes.Relation_ID</i>

D.2 Einzelnen Lernobjekt-Typen zugeordnete Tabellen

Wie bereits im vorhergehenden Abschnitt werden hier die einzelnen Datenbank-Tabellen mit den jeweiligen Spaltenbezeichnungen, -typen und und einer evtl. Erläuterung der Spalte gelistet:

a) Bildobjekte

pictures: Haupttabelle für den Lernobjekt-Typ <i>Bild</i> .		
LO_ID	mediumint	→ <i>learningobjects.LO_ID</i>
Rotation	tinyint	Angabe des Rotationswinkels für „falsch“ eingescannte Bilder
Mirror	enum	Angabe, ob ein eingescanntes Bild gespiegelt werden muß
Flags	tinyint	Status (ist das Bild neu/hochgeladen?)
X	mediumint	Breite des Original-Bildes in Pixeln
Y	mediumint	Höhe des Original-Bildes in Pixeln
Quality	tinyint	Qualität des Bildes (vom Verschlagwörter in Stufen bewertet)
Agreement	smallint	Unterscheidung, ob eine Anonymisierung notwendig ist, ggf. Angabe, ob ein Einverständnis der abgebildeten Person vorliegt.

blackboxes: Verwaltung der Rechtecke zur Anonymsierung von Bildern.		
LO_ID	mediumint	→ <i>learningobjects.LO_ID</i>
NR	tinyint	Numerierung
Width	smallint	Breite des Rechtecks in Pixeln (im Originalbild)
Height	smallint	Höhe des Rechtecks in Pixeln
X	smallint	X-Pos der linken oberen Ecke des Rechtecks
Y	smallint	Y-Pos der linken oberen Ecke des Rechtecks

resourcetypes: Liste der verschiedenen vorhandenen Bildtypen		
ResourceType_ID	tinyint	auto_increment
Description_de	tinytext	deutsche Beschreibung (zum Beispiel <i>klinisch</i> oder <i>histologisch</i>)
Description_en	tinytext	englische Beschreibung

learningresources: Zuordnung zwischen Bildobjekt und Bildtyp		
LO_ID	mediumint	→ <i>learningobjects.LO_ID</i>
ResourceType_ID	tinyint	→ <i>resourcetypes.ResourceType_ID</i>
NR	tinyint	Numerierung

b) Videoobjekte

videoclips: Haupttabelle für den Lernobjekt-Typ <i>Video</i> .		
LO_ID	mediumint	→ <i>learningobjects.LO_ID</i>
Framerate	tinyint	Framerate des Videos
FrameNumber	mediumint	Anzahl der Frames
Width	smallint	Breite des Videos in Pixeln
Height	smallint	Höhe des Videos in Pixeln
Flags	tinyint	Status (ist das Video neu/hochgeladen?)

c) Lerntexte

learningtexts: Haupttabelle für den Lernobjekt-Typ <i>Lerntext</i> .		
LO_ID	mediumint	→ <i>learningobjects.LO_ID</i>
Flags	tinyint	Status (neu/hochgeladen?)

d) Fallbeispiele

usecases: Haupttabelle für den Lernobjekt-Typ <i>Fallbeispiel</i> .		
LO_ID	mediumint	→ <i>learningobjects.LO_ID</i>
Flags	tinyint	Status (neu/hochgeladen?)

e) Quizelemente

Fallquiz:

fallquiz_quizliste: Liste der bereits erstellten Fallquiz.		
QuizID	int	auto_increment
DatumErstellung	date	Datum der Erstellung des Quiz
Erstellender	varchar(150)	Autor
Vorinformationen	text	Kurzanamnese
Bemerkungen	text	Freitext - Bemerkungen
Titel	varchar(200)	Titel des Quiz

fallquiz_zuordnungsbilder: Verwaltung der Zuordnung von Bildobjekten zu einem Quiz		
ID	int	auto_increment
QuizID	int	→fallquiz_quizliste.QuizID
LO_ID	int	LO_ID des Bildobjekts (→learningobjects.LO_ID)!

fallquiz_anamnesefragen: Pool aller Anamnesefragen.		
AnamneseFrageID	int	auto_increment
AnamneseFrageText	text	Text der Frage

fallquiz_zuordnungsanamnese: Verwaltung der Zuordnung einer Anamnesefrage zu einem Quiz		
ID	int	auto_increment
QuizID	int	→fallquiz_quizliste.QuizID
AnamneseFrageID	int	→fallquiz_anamnesefragen.AnamneseFrageID
Sinnvoll	tinyint	Angabe, ob die Frage in diesem Fall sinnvoll ist oder nicht
Antwort	text	Antwort, die der virtuelle Patient auf die gestellte Frage gibt
Erlaeuterung	text	Erläuterung, die dem Lernenden zusätzlich angezeigt wird

fallquiz_diagnostischeverfahren: Liste aller diagnostischen Verfahren		
DiagnostischesVerfahrenID	int	auto_increment
Verfahren	text	Bezeichnung / Beschreibung des Verf.
SkriptLink	varchar(200)	

fallquiz_zuordnungpe / fallquiz_zuordnungse / fallquiz_zuordnungoe: Zuordnung zwischen Quiz und Effloreszenzen (primäre, sekundäre und andere)		
ID	int	auto_increment
QuizID	int	→fallquiz_quizliste.QuizID
PE_ID/SE_ID/OE_ID	int	entsprechende Ids der Effloreszenzen
Richtig	tinyint	Ist die Effloreszenz im vorliegenden Fall richtig?

fallquiz_zuordnungdiagnostik: Zuordnung zwischen einem Quiz und den diagnostischen Verfahren		
ID	int	auto_increment
QuizID	int	→fallquiz_quizliste.QuizID
DiagnostischesVerfahrenID	int	→fallquiz_diagnostischeverfahren.DiagnostischesVerfahrenID
Sinnvoll	tinyint	Ist es sinnvoll, das Verfahren im vorliegenden Fall anzuwenden?
Ergebnis	text	Welches Ergebnis hat das Verfahren bei diesem virtuellen Patienten geliefert?
Begründung	text	Warum ist es (nicht) sinnvoll, das Verfahren hier anzuwenden?

fallquiz_zuordnungdiagnose: Zuordnung zwischen einem Quiz und den Diagnosen		
ID	int	auto_increment
QuizID	int	→fallquiz_quizliste.QuizID
Disease_ID	int	entsprechende ID der Diagnose
Erläuterung	text	zusätzliche Erläuterung, die dem Lernenden angezeigt wird
Richtig	tinyint	Ist die Diagnose hier richtig?

Übungsfragen

uebauf_themen: Übersicht über die vorhandenen Übungsfragen-Themen		
ex_description	text	Bezeichnung des Themengebiets
ex_thema_id	mediumint	auto_increment

uebauf_fragen: Liste aller vorhandener Fragen		
ex_id	mediumint	auto_increment
ex_frage	text	der Fragentext
ex_thema	mediumint	Id des Themas, zu dem die Frage gehört
ex_sort	mediumint	Reihenfolge der Fragen
ex_erl	text	Erläuterung (warum ist welche Antwort richtig?)
ex_images	text	Bilder, die zu dieser Frage gehören

uebauf_antworten: Liste aller vorhandenen Antworten		
ex_id	mediumint	Angabe der Id der Frage, zu der die Antwort gehört
ex_antwort	text	der Antworttext
ex_korrekt	tinyint	Ist die Antwort falsch oder richtig?
ex_number	smallint	Reihenfolge der Antworten
ex_ant_id	int	auto_increment

D.3 WWW-Suche in der Datenbank

Entwicklungs- umgebung	WindowsME/2000 mit: Apache-WWW-Server (http://www.apache.org/) ActivePerl 5.6.0, build 618 (http://www.activestate.com/) MySQL 3.23.22 (http://www.mysql.com)
Siehe auch	Abschnitt 6.4.6, Seite 101

Zur Suche in der Lernobjekt-Datenbank wurden die folgenden Perl-CGI-Skripte implementiert:

- *SearchForm.cgi*
generiert dynamisch ein HTML-Formular, das aus der Datenbank die verschiedenen Lernobjekt-Typen bzw. die erlaubten Werte für Datenfelder ausliest und in DropDown-Boxen einfügt.
- *getList.cgi*
wird bei jedem Tastendruck im Diagnosen- bzw. Lokalisations-Eingabefeld durch Javascript aufgerufen, generiert eine Liste von Lokalisations- bzw. Diagnosen-Einträgen, die zum übergebenen Präfix passen.
- *SearchStart.cgi*
setzt die Suchanfrage aus den vom HTML-Formular übergebenen Daten zusammen, führt die Abfrage durch und stellt die Ergebnisse in einer HTML-Tabelle dar.

Typische Beispiele von generierten SQL-Statements:

1. Suchen aller Lernobjekt-IDs mit der Diagnose *Alopecia areata* und gleichzeitiger Sekundäreffloreszenz *Schuppe* (es ist bereits bekannt, daß die interne ID für *Schuppe* 8 ist):

```

SELECT LearningObjects.LO_ID
FROM LearningObjects, SecondaryEfflorescences, Diagnoses,
DiseaseDescriptions
WHERE ( (LearningObjects.LO_ID=Diagnoses.LO_ID
AND Diagnoses.Disease_ID=DiseaseDescriptions.Disease_ID
AND DiseaseDescriptions.Description LIKE 'Alopecia areata' ) )
AND LearningObjects.LO_ID=SecondaryEfflorescences.LO_ID
AND SecondaryEfflorescences.SE_ID=8

```

2. IDs aller Lernobjekte vom Typ *Video*

```

SELECT LearningObjects.LO_ID
FROM LearningObjects, SpecialTables
WHERE (LearningObjects.Special_ID=SpecialTables.Special_ID
AND SpecialTables.Description_de ='Video')

```

D.4 Mergen mehrerer Datenbanken

Entwicklungs- umgebung	WindowsME/2000 mit: ActivePerl 5.6.0, build 618 (http://www.activestate.com/) MySQL 3.23.22 (http://www.mysql.com) XML::Simple 1.05
Siehe auch	Abschnitt 6.4.5, Seite 100

Zum Einfügen einer (lokalen) Datenbank in eine andere (zum Beispiel die zentrale) dient das Perl-Skript *dejavumerge.pl*. Es ist durch die XML-Datei *dejavumerge.xml* konfigurierbar. Am wichtigsten ist der Abschnitt *database*:

```

<section name="databases">
<db name="db" id="-1" host="server1" user="deja"/>
<db name="db_wue" id="0" host="server2" user="wue"/>
<db name="db_jena" id="1" host="server3" user="jen"/>
</section>

```

Hier wird festgelegt, in welche Datenbank eingefügt wird (*id="-1"*) und welche Datenbanken eingefügt werden sollen (*id="0"*, *id="1"*, ...). Für jede einzufügende Datenbank muß das Skript gestartet werden. Als Parameter werden dabei Datenbankname und Paßwort der einzufügenden Datenbank und das Paßwort der Datenbank, in die eingefügt wird, angegeben:

```
perl dejavumerge.pl database=db_jena mergepwd=pw1 addpwd=pw2
```

Anhang E - Lehr- und Lernumgebung

Die folgenden Software-Pakete wurden im Rahmen der Projekte DEJAVU [48] und HagerROM2002 [3] am Lehrstuhl für Informatik II der Universität Würzburg zusammen mit den jeweiligen Mitarbeitern entwickelt:

E.1 Installations – CD „Lehr- und Lernumgebung“

Mit der CD „Lehr- und Lernumgebung“ können unter anderem folgende Module auf den Betriebssystemen Windows95/98/ME/ NT/2000 installiert werden:

- Lernobjekt – Datenbank inkl. MySQL-Server (Abschnitt 6.4)
- *DEJAVUCapt* (Abschnitt 6.5.1)
- Powerpoint-AddIn (Abschnitt 6.5.5)
- *DEJAVULearn* (Abschnitt 6.6)

Zunächst sollte die Datenbank installiert werden. Dazu ist auf der CD *SetupServer.exe* auszuführen, danach können die anderen Tools auf den Clients durch Ausführen von *SetupStudent.exe* (DEJAVULearn) bzw. *SetupAutor.exe* (Autorenwerkzeuge) installiert werden.

E.2 Autorenwerkzeuge

E.2.1 DEJAVUCapt

Entwicklungs- umgebung	IBM Visual Age for Java (JDK 1.2.2) unter Windows ME/2000
Start	Aufruf der main-Methode der Klasse <i>dejavu.tool.AdminGUI</i>
Parameter	keine
Klassenpfad	Pakete dejavu, dejavu.tool, dejavu.maintenance, info2xml (s. u.), Apache XML-Parser xerces (in <i>xerces.jar</i>) XSLT-Transformator xalan (in <i>xalan.jar</i>) JDBC-Treiber
Siehe auch	Abschnitt 6.5.1, Seite 102

Die Java-Applikation *DEJAVUCapt* liest/schreibt seine Grundeinstellungen aus der/in die Datei *dejavu.ini* aus dem jeweiligen Arbeitsverzeichnis. Alle angegebenen Daten sind die Einstellungen, die beim Programmstart verwendet werden. Sie können innerhalb des Programms aber geändert werden und werden dann in *dejavu.ini* gespeichert:

```
Aufbau von dejavu.ini:
[Login]
User=root,deja      # Datenbank - User
Database=dejavu    # Name der Datenbank
Server =server1, server2,...      # Name bzw. IP-Adresse DB-Server

[Main]
ServerImageBase =c:/server/ # Server-Objektpfad
UserImageBase =c:/lokal/    # User-Objektpfad

[Upload]
Width =800 # Größe und Qualität der Bilder, die
Height =600 # beim Upload im Server-Objektpfad generiert
Quality=75 # werden

[Look] # Welche Karteikarten sollen angezeigt werden?
Dispersion=no
PrimeEfflo=yes
```

Insgesamt besteht *DEJAVUCapt* aus ca. 90 Klassen. Im folgenden wird nur kurz auf die wichtigsten Klassen der einzelnen Pakete eingegangen.

- Paket *dejavu*: Klassen, die Lernobjekte repräsentieren, zum Beispiel:
 - *DejavuObject*: Oberklasse aller Lernobjekt-Klassen, enthält die LO_ID sowie die Datenbank-Daten.
 - *LearningObject*: allgemeines Lernobjekt.

- *SpecialDejavuObject*: Oberklasse für alle Lernobjekt-Typen, von dieser Klasse erben zum Beispiel *Picture*, *Learningtext* oder *UseCase*
- Paket *dejavu.maintenance*: Klassen zur Wartung der Datenbank, zum Beispiel:
 - *DejavuGenerator*: generiert eine neue Datenbank, mit Hilfe dieser Klasse kann die Datenbank bereits mit Daten aus den textorientierten Bildarchiven gefüllt werden.
 - *DejavuExport*, *DejavuImport*: exportieren, importieren die Datenbankdaten in ein / aus einem Textfile. Die Klassen dienen auch dazu, Datenbank-Updates durchzuführen, wenn sich die Datenbankstruktur ändert.
- Paket *dejavu.tool*: Klassen zur Darstellung der Benutzeroberfläche von *DEJAVUCapt*:
 - *AdminGui*: Hauptframe der Nutzeroberfläche
 - *PictureTabbedPane*: *TabbedPane*, das die einzelnen Karteikarten zur Erfassung der Datenfelder enthält, zum Beispiel *AnnotationTab* oder *ErscheinungTab*
 - *AnonymDialog*: *JFrame* zur Bildanzeige mit Möglichkeit zum Einzeichnen von Anonymisierungs-Rechtecken
 - *VideoViewer*: *JFrame* zur Anzeige von Videos und Auswahl von Thumbnails für Videos
- Paket *info2xml*: Klassen zur Anzeige von XML-basierten Dateien (Lerntexte, Fallbeispiele), z.B.:
 - *XMLEditorPane*: *JFrame* zur Anzeige von XML-Dateien. Die Klasse konvertiert eine XML-Datei mit Hilfe des Apache-Xalan XSL-Transformators nach HTML und zeigt sie in einem *javax.swing.JEditorPane* an.
- Paket *mysql*: Klassen für den Datenbankzugriff
 - *MySQL*: enthält diverse Methoden für den Datenbankzugriff (Erstellen von Tabellen, Ausführen von select-Anweisungen,)

Zur Konvertierung der Bilder und Einfügen der Anonymisierungsrechtecke beim Upload wird das Programm *convert* der *Image Magick Tools* verwendet.

Es folgen noch einige typische SQL-Statements, die von der Applikation generiert werden:
Setzen der Grundattribute für ein Bildobjekt (*Special_ID = 1*):

```
UPDATE LearningObjects
  SET Title_de=null,Title_en=null,Filename="f.jpg",Version=null,
      Status=null,Special_ID=1,Institute_ID=2,Context=0
  WHERE LO_ID=4450;
```

Setzen der Attribute für den speziellen Lernobjekt-Typ Bild:

```
UPDATE Pictures
  SET Rotation=null,Mirror=null,Flags=2,X=800,Y=600,
      Quality=null,Agreement=0
  WHERE LO_ID=4450;
```

Zuweisen einer Diagnose zu einem Lernobjekt (inklusive Update des Diagnosenzählers)

```
INSERT INTO Diagnoses
  VALUES (4450,2385,0);
UPDATE Diseases
  SET Used=39
  WHERE Disease_ID=2385;
```

Eintrag des Datums der Bearbeitung (0 = erste Bearbeitung des Objekts, 17 = *Letzter Bearbeiter*, 1 = ID des Bearbeiters):

```
INSERT INTO Contributes
  VALUES (4450,0,17,1,"2002-03-29");
```

E.2.2 Nachverschlagwortung der Attribute

Entwicklungs- umgebung	WindowsME/2000 mit: Apache-WWW-Server (http://www.apache.org/) ActivePerl 5.6.0, build 618 (http://www.activestate.com/) MySQL 3.23.22 (http://www.mysql.com)
Siehe auch	Abschnitt 6.5.2, Seite 106

Sowohl die Nachverschlagwortung als auch die WWW-Suche in der Datenbank lesen die Einstellungen aus der Datei *dejavu.ini* im Verzeichnis, in dem die Perl-Skripte liegen.

Achtung: Die Datei hat trotz des gleichen Namens eine andere Form wie die Konfigurationsdatei von *DEJAVUCapt*.

```
Aufbau von dejavu.ini:
dejavu # Name der Datenbank
localhost # Name bzw. IP-Adresse des Datenbank-Servers
deja # Datenbank - User
password # Passwort
6 # Anzahl Bilder pro Zeile
12 # Anzahl Bilder pro Seite
../present/scans/ # Server-Objektpfad
```

Die Nachverschlagwortung wurde jeweils für die Datenfelder *Bildqualität*, *Einverständnis*, *Lokalisationen* und *Zielgruppe* implementiert. Für jedes dieser Felder wurden zwei Perl-CGI Skripte *{feldkurzbez}index.cgi* und *{feldkurzbez}.cgi* erstellt, die über Perl DBI mit `DBD::mysql` mit der Datenbank kommunizieren:

`{feldkurzbez}index.cgi` erstellt eine HTML-Übersicht über alle Lernobjekte mit Links zu den einzelnen von `{feldkurzbez}.cgi` erstellten Thumbnail-Übersichtsseiten. Im Link wird jeweils eine Lernobjekt-Id `loid` übergeben. Skript 2 zeigt dann die Lernobjekte

`loid` bis $(loid + \text{Anzahl Objekte/Zeile} \times \text{Anzahl Objekte/Seite}) - 1$ mit den entsprechen Eingabefeldern bzw. Checkboxes in einem HTML-Formular an (siehe Abbildung 37, Seite 106). Der Submit – Button ruft Skript 2 selbst auf, das dann die gemachten Änderungen in die Datenbank überträgt und dann die nächste Übersichtsseite anzeigt. Die Nachverschlagwortung anderer Datenfelder kann durch Anpassung der beiden Skripte vorgenommen werden.

E.2.3 Powerpoint AddIn

Entwicklungs- umgebung	WindowsME/2000 mit: Microsoft VisualBasic 6.0 MySQL 3.23.22 (http://www.mysql.com)
Siehe auch	Abschnitt 6.5.5, S. 108

Das PowerpointAddIn baut die Datenbankverbindung über eine ODBC-Datenquelle auf. Es liest die Einstellungen aus der Datei *Dejapower.set*, die im Windows-Nutzer-Verzeichnis (Environment-Variable *USERPROFILE*) abgelegt wird. Sie enthält nur eine Zeile folgender Form:

```
{Name ODBC-Datenquelle}, "", {Server-Objektpfad}, {Anzahl Bilder/Zeile},
{Anzahl Bilder /Spalte}, {max. Anzahl Bilder},
#TRUE#, #TRUE#, #TRUE#, #TRUE#
```

Die wichtigsten VisualBasic-Formulare sind:

- *StandardSearchForm*
Oberfläche für die Standardsuche
- *ExpertSearchForm*
Expertensuche inklusive der Karteikarten für Suche in Lokalisationen und sonstigen Datenfeldern
- *ThumbForm*
Anzeige der Trefferübersicht inklusive Buttons
- *BigForm*
Ansicht eines Bildes in Großdarstellung
- *VideoForm*
Anzeige eines Videos in Großdarstellung inklusive Buttons zum Abspielen

Der Aufbau des SQL-Statements und die Datenbankabfrage geschieht im Modul *DEJAVU* in der Methode *doPreQuery* bei einer Standardsuche bzw. in *doExpertQuery* bei einer erweiterten Suche.

E.2.4 HTML2XML Konvertierung Lerntexte / Fallbeispiele

Entwicklungs- umgebung	WindowsME/2000 mit: ActivePerl 5.6.0, build 618 (http://www.activestate.com/) Tidy (V. 30th April 2000) [66]
Siehe auch	Abschnitt 6.5.6, 6.5.7, S. 111f

Die Schritte (2) – (4) der Konvertierung von Lerntexten und Fallbeispielen werden vom Skript *convertLT2XMLTags.pl* geleistet.

(1) → Word-Dok. (2) → HTML (3) → XHTML (4) → XML (5) → Datenbank

Dabei wird folgendermaßen vorgegangen:

Auslesen aller HTML-Dateien aus dem Unterverzeichnis *htmlfiles*, dann für jede Datei:

1. Extraktion der durch die Datei *bereiche.txt* vorgegebenen Textbereiche. Dabei wird getestet, ob in den jeweiligen Bereich der Vorlage überhaupt etwas eingegeben wurde. Ist das nicht der Fall, wird der Bereich gelöscht. Aus allen anderen Bereichen werden überflüssige Tags gelöscht.
2. Die so bearbeitete HTML-Datei wird mit dem Tool *Tidy* nach XHTML gewandelt, wodurch das Dokument wohlgeformt wird.
3. Die durch *bereiche.txt* bestimmten Textbereiche werden durch XML-Tags geklammert, der jeweilige Inhalt als CDATA in diese Tags geschrieben. Auf diese Weise entsteht eine zur DTD *lerntextsyst.dtd* gültige XML-Datei.

Aufbau der Datei *bereiche.txt*: Jeweils eine Zeile definiert einen Textbereich durch die folgenden Tabulator-getrennten Einträge:

```
{tag}\t{secNo}\t{desc}\t{delIf}\t{noChildren}
```

wobei:

- `{tag}`: Der Name des entsprechenden Tags in XML
- `{secNo}`: Beim Export von Windows nach Compact-HTML werden die einzelnen Bereiche in der Form `<div class=Section2>` exportiert. Die Zahl gibt die Nummer des Bereichs an.
- `{desc}`: Beschreibung des Bereichs, dient der Lesbarkeit
- `{delIf}`: Entspricht der Abschnitt genau diesem Text, kann der Bereich gelöscht werden, da der Autor keine Eingaben gemacht hat.

{noChildren}: Gibt an, wieviele der folgenden Zeilen in *bereiche.txt* hierarchisch unter dem aktuellen Bereich eingegliedert werden müssen

Beispiel:			
kompli	18	Komplikationen	z.B. Impetiginisierung 0
diagnostik	20	Diagnostik	(z.B. Anamnese) 2
histo	22	Histologie	0
labor	26	Labor	<p class=MsoNormal>(z.B. IgE) 0

E.2.5 Übungsfragen

Entwicklungs- umgebung	WindowsME/2000 mit: Apache-WWW-Server (http://www.apache.org/) ActivePerl 5.6.0, build 618 (http://www.activestate.com/) MySQL 3.23.22 (http://www.mysql.com)
Siehe auch	Abschnitt 6.5.8, S. 115

Die folgenden Skripte zur Erfassung von Übungsfragen lesen die Grundeinstellungen aus der Datei *ex_dejavu.ini*, die im gleichen Verzeichnis wie die Skripte liegt und genauso aufgebaut ist wie die Konfigurationsdatei bei der Nachverschlagwortung.

- *examThema.htm*
HTML – Formular zur Eingabe eines neuen Themas
- *examAddThema.cgi*
fügt ein neues Fragenthema in die Datenbank ein.
- *examAdd.cgi*
 - generiert ein HTML-Formular für eine bestehende Frage (siehe Abbildung 44, Seite 116)
 - fügt eine Antwort zu einer bestehenden Frage hinzu (einschließlich Flag richtig/falsch)
 - speichert Änderungen des Fragentexts, zu einzelnen Antworten, usw. in der Datenbank.
- *examNewQuestion.cgi*
generiert ein HTML-Formular zur Eingabe einer neuen Frage.
- *examShowQuestions.cgi*
zeigt eine Übersicht über alle Fragen zu einem Themenkomplex inklusive Links zum Editieren an.

- *examShowQuestionsAnswers.cgi*

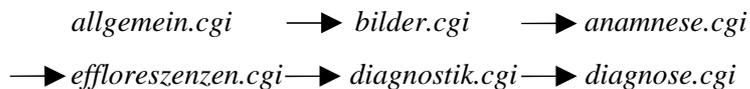
zeigt eine vollständige Übersicht über aller Fragen zu einem Themenkomplex.

E.2.6 Fallquiz

Entwicklungs- umgebung	WindowsME/2000 mit: Apache-WWW-Server (http://www.apache.org/) ActivePerl 5.6.0, build 618 (http://www.activestate.com/) MySQL 3.23.22 (http://www.mysql.com)
Siehe auch	Abschnitt 6.5.8, S. 115

Alle Skripte verwenden *config.cgi* zum Einlesen der Datenbankkonfiguration. Einstiegspunkt ist das Skript *first.cgi*. Es liest zunächst aus der Datenbank die Titel der vorhandenen Fälle ein und zeigt diese in einem Formular an. Es kann ein Fall bearbeitet oder ein neuer Fall angelegt werden. Wird ein Fall bearbeitet, werden in allen folgenden Skripten die entsprechenden Eingabefelder und Checkboxen mit den in der Datenbank gespeicherten Daten vorbelegt.

Der Workflow eines Autors bei der Erstellung eines Fallquiz spiegelt sich in der Reihenfolge des Aufrufs der Hauptskripte wieder:



- *allgemein.cgi*

zeigt ein Formular mit Texteingabefeldern zur Eingabe des Titels, Autors und der Kurznamnese.

- *bilder.cgi*

generiert ein Formular zur Verwaltung (Hinzufügen/Löschen) der zu einem Fall zugeordneten Bilder. Zum Hinzufügen wird das Skript *bildersuche.cgi* (s.u.) gestartet.

- *anamnese.cgi*

- verwaltet die dem Fall zugeordneten Anamnesefragen. Es werden die bereits ausgewählten Fragen angezeigt.
- zeigt eine Liste aller in der Datenbank vorhandenen Fragen an, aus der der Autor dem Fall Fragen zuordnen kann.
- zu jeder Frage kann in einem Textfeld eine Erläuterung angegeben werden und ob sie in diesem Fall sinnvoll ist.
- generiert Buttons zum Einfügen einer neuen Anamnesefrage in die Datenbank

(*editanamnese.cgi*) und zum Bearbeiten der bereits ausgewählten Fragen (*anamnesefragen.cgi*)

- *effloreszenzen.cgi*
zeigt eine Liste aller Effloreszenzen an. Der Autor entscheidet durch Checkboxes, welche Effloreszenzen dem Lerner angezeigt werden und ob sie richtig oder falsch ist.
- *diagnostik.cgi*
 - zeigt eine Liste der in der Datenbank gespeicherten diagnostischen Verfahren in einem HTML-Formular an. Der Autor entscheidet durch Checkboxes, welche diagnostische Verfahren dem Lerner angezeigt werden und ob die Durchführung des Verfahrens in diesem Fall sinnvoll ist oder nicht.
 - generiert einen Button zum Einfügen eines neuen Verfahrens in die Datenbank (siehe *editverfahren.cgi*)
- *diagnose.cgi*
 - generiert in einem Formular eine Liste aller Diagnosen. Der Autor wählt jeweils eine Diagnose, die dem Lernenden angezeigt wird, aus. Er gibt eine Erläuterung dazu an und ob die Diagnose richtig oder falsch ist.
 - generiert einen Button zum Bearbeiten von bereits ausgewählten Diagnosen (siehe *diagnosenauswahl.cgi*)

Neben den Hauptskripten werden folgende Hilfsskripte benutzt:

- *methods.cgi*
ist eine Sammlung von Methoden für den Datenbankzugriff, z.B.:
 - *bilder_dazu*: speichert einem Fall zugeordnete Bilder in der Datenbank
 - *effloreszenzen_sichern*: speichert die zugeordneten Effloreszenzen
- *bildersuche.cgi*
führt eine Suche nach Bildobjekten mit einem übergebenem Diagnosenpräfix in der Datenbank durch. Die gefundenen Objekte werden in einer Thumbnailübersicht gezeigt, in der Bilder durch Checkboxes markiert werden können.
- *editanamnese.cgi*
generiert ein Formular zur Eingabe einer neuen Anamnesefrage
- *anamnesefragen.cgi*
generiert ein Formular zur Bearbeitung bereits ausgewählter Anamnesefragen. In Textfeldern kann zum Beispiel die Erläuterung und die Patientenantwort editiert werden.

- *editverfahren.cgi*
generiert ein Formular zur Eingabe eines neuen Verfahrens
- *diagnosenauswahl.cgi*
generiert ein Formular zur Bearbeitung bereits ausgewählter Diagnosen. In einem Textfeld kann zum Beispiel die Erläuterung editiert werden.

E.3 Lernumgebung

E.3.1 DEJAVULearn

Entwicklungs- umgebung	IBM Visual Age Java (JDK 1.2.2) unter Windows ME/2000
Start	Aufruf der main-Methode von <i>dejavu.learn.nena.NENa</i>
Parameter	{ <i>konf-Datei</i> }
Klassenpfad	Jar-Datei <i>dejavulearn.jar</i>
siehe auch	Abschnitt 6.6, Seite 119

Die Java-Applikation *DEJAVULearn* liest die Grundeinstellungen aus der Konfigurationsdatei *konf-Datei* des Arbeitsverzeichnisses, persönliche Einstellungen von Nutzern (z.B. Fenstergrößen) werden aber im Verzeichnis *System.getProperty("user.home")/konf-Datei* gespeichert.

```
Beispiel einer konf-Datei (Arbeitsverzeichnis):
treeFiles=Lerntexte.tre;Fallbeispiele.tre;Quiz.tre # Bäume
templatePath=notes/ # templateFile- für Bemerkungen
htmlPath=http://127.0.0.1/ # Basispfad
```

Die Navigationsbäume werden durch ein Perl-Skript *convertLernpfad.pl* generiert. Input für das Skript sind dat-Files im Unterverzeichnis *dat*. Jedes dat-File in diesem Verzeichnis repräsentiert einen Navigationsbaum. Dabei hat jede Zeile die Form:

```
{tiefe} - {objekt} - {titel} -- -- {baum} -- {suche}
```

wobei:

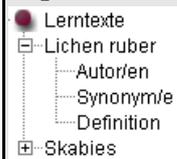
- {tiefe}: die Baumtiefe dieses Eintrags
- {objekt}: Referenz auf das Objekt, das bei Auswahl des Eintrags angezeigt werden soll (zum Beispiel Lerntext-Datei)
- {titel}: Titel des Eintrags im Baum
- {baum}: 0: Eintrag wird nicht im Baum angezeigt
1: Eintrag wird angezeigt
- {suche}: Dieser Eintrag ist reserviert für die Steuerung einer Volltextsuche
0: Eintrag wird nicht in die Suche integriert
1: Eintrag wird in die Suche integriert

Beispiel:

Das dat-File Lerntexte.dat:

```
1 -- -- Lichen ruber -- -- 1 -- 0
2 -- html/lichenruber/gesamt.htm#autor -- Autor/en -- -- 1 -- 0
2 -- html/lichenruber/gesamt.htm#synonym -- Synonym/e -- -- 1 -- 0
2 -- html/lichenruber/gesamt.htm#definition -- Definition -- -- 1 -- 0
1 -- -- Skabies -- -- 1 -- 0
2 -- html/skabies/gesamt.htm#autor -- Autor/en -- -- 1 -- 0
2 -- html/skabies/gesamt.htm#synonym - Synonym/e -- -- 1 -- 0
```

repräsentiert folgenden Baum:



Das Skript liest aus dem Unterverzeichnis *dat* dat-Files ein und generiert daraus tre-Files, die in der *konf-Datei* angegebenen werden sowie eine Datei *namelistnena.txt*, die eine Auflistung aller vorhandenen Objekte darstellt. Die tre-Files sind baumförmig strukturierte Links auf die Einträge in *namelistnena.txt*. Einträge können in den verschiedenen tre-Files beliebig oft referenziert werden.

Außerdem wird die Datei *hidetags.ini* eingelesen. Sie dient der Verwaltung von Textbereichen, die auf HTML-Seiten ein-/ausgeblendet werden können. Jede Zeile definiert einen klappbaren Textbereich, indem durch Tabulator getrennt der Tagname und die Bezeichnung des Bereichs angegeben werden:

```
{tag} \t {bez}
```

In HTML-Dateien können dann Bereiche ein-/und ausgeklappt werden, wenn sie folgendermaßen getagt werden:

```
<div feldname="{tag}" style="padding:0pt" invert="true">
  <!-- hier Text einfügen, der angezeigt wird,
        wenn Bereich eingeklappt ist -->
</div>
<div feldname="{tag}" style="padding:0pt">
  <!-- hier Text einfügen, der angezeigt wird,
        wenn Bereich ausgeklappt ist -->
</div>
```

```

Beispiel:
hidetags.ini:
b_syn\tSynonyme

HTML-Datei:
<div feldname="b_syn" invert="true">
  <b>Synonym/e sind eingeklappt </b>
</div>
<div feldname="b_syn">
  <b>Synonym/e: </b><br>Hier stehen jetzt<br>Synonyme
</div>

Anzeige eingeklappt:           Anzeige ausgeklappt:
 Synonym/e sind eingeklappt
 Synonym/e:
                               Hier stehen jetzt
                               Synonyme

```

DEJAVULearn besteht aus ca. 180 Klassen. Im folgenden wird nur kurz auf die wichtigsten Klassen ausgesuchter Pakete eingegangen:

- Paket *dejavu.learn.nena*: Hauptklassen, zum Beispiel:
 - *NENA*: Hauptklasse, verwaltet alle anderen Komponenten
 - *Book*: verwaltet den aktuellen Status: welches Objekt wird im Moment angezeigt, enthält die zentrale Methode *go(int, String, Object)*, die ein Lernobjekt anzeigt. Die einzelnen Einträge werden dabei über interne IDs verwaltet.
 - *Page_Mapper*: verwaltet die (Baum-)Einträge. Enthält zum Beispiel eine HashTable mit Filenamen als Schlüssel und interner ID als Wert. Soll ein Objekt zum Beispiel über *Book.go* angezeigt werden, wird im *Page_Mapper* nachgefragt, ob es für das Objekt eine ID gibt.
 - *Nena_Collection*; verwaltet die Gesamtheit der Notizen (Bemerkungen, Markierungen und Lesezeichen)
- Paket *dejavu.learn.nena.editor*: Klassen zur Darstellung von HTML-Seiten
 - *NenaEditorPane*: Unterklasse von *javax.swing.JEditorPane*, zeigt HTML-Text mit erweiterten Möglichkeiten an (Anzeige von Bemerkungen, Markierungen, Einklappen von Textbereichen).
- Paket *dejavu.learn.nena.event*: Klassen zur Event-Behandlung, z.B.:
 - Interface *PageChangeListener*: Interface mit der Methode *pageChange(PageChangeEvent)*. Klassen, die wissen müssen, wann sich der angezeigte Eintrag ändert, implementieren das Interface (zum Beispiel muß der Baum wissen, wann sich der

Eintrag ändert, um dann den dem neuen Eintrag entsprechenden Baumeintrag zu selektieren)

- Paket *dejavu.learn.viewer*: Klassen des integrierten Bildbetrachters, z.B.:
 - *HagerView*: Hauptklasse des Viewers

E.3.2 Bilderquiz

Entwicklungs- umgebung	IBM Visual Age for Java (JDK 1.2.2) unter Windows ME/2000
Start	Aufruf der main-Methode von <i>dejavu.quiz.Quiz</i> oder Start aus <i>DEJAVULearn</i>
Parameter	keine
Klassenpfad	Jar-Datei <i>quiz.jar</i>
siehe auch	Abschnitt 6.6.8, Seite 126

Das Bilderquiz liest und schreibt die Grundeinstellungen in die Datei *dejavu.ini* im Arbeitsverzeichnis. Dort wird die Section *[Quiz]* verwendet:

```
[Quiz]
User=deja # Datenbank-User
Password=password # entsprechendes Passwort
Database=dejavu # Datenbankname
Server=localhost # Name/IP des Datenbankservers
Port=3306
```

Das Bilderquiz enthält 27 Klassen im Paket *dejavu.quiz*, zum Beispiel:

- *MainPanel*: Jpanel, das alle anderen visuellen Komponenten enthält.
- *QuestionGenerator*: Klasse zur Generierung eines Satzes von Fragen.
- *AnswerPanel*: Anzeige des Panels mit der Frage und den Antwort-Alternativen

E.3.3 Fallquiz

Entwicklungs- umgebung	WindowsME/2000 mit: Apache-WWW-Server (http://www.apache.org/) ActivePerl 5.6.0, build 618 (http://www.activestate.com/) MySQL 3.23.22 (http://www.mysql.com)
Siehe auch	Abschnitt 6.6.8, Seite 126

Auf Lerner-Seite wurden zur Durchführung des Fallquiz über DBI mit der Datenbank verbundene Perl-CGI-Skripte implementiert.

Die Haupt-Skripte sind *anamnese.cgi*, *effloreszenzen.cgi*, *diagnostik.cgi*, *diagnose.cgi*. Sie generieren jeweils HTML-Formulare, testen, ob die vom Lernenden gemachten Angaben korrekt sind und liefern die entsprechenden Rückmeldungen.

Daneben werden folgende Skripte verwendet:

- *erlaeuterung.cgi*
gibt die Erläuterung, die der Autor des Fallquiz zu einer Anamnesefrage angegeben hat, aus.
- *ergebnis.cgi*
zeigt das Ergebnis eines diagnostischen Verfahrens sowie die vom Autor dazu angegebene Erläuterung an.

E.3.4 Übungsfragen

Entwicklungs- umgebung	WindowsME/2000 mit: Apache-WWW-Server (http://www.apache.org/) ActivePerl 5.6.0, build 618 (http://www.activestate.com/) MySQL 3.23.22 (http://www.mysql.com)
Siehe auch	Abschnitt 6.6.8, Seite 126

Die folgenden über DBI mit der Datenbank verbundenen Skripte lesen die Grundeinstellungen aus der Datei *ex_dejavu.ini*, die im gleichen Verzeichnis wie die Skripte liegt und genauso aufgebaut ist, wie die Konfigurationsdatei bei der Querverschlagwortung.

- *examQuizCreate.cgi*:
Generiert für einen Themenkomplex eine zufällige Auswahl von Fragen.
- *examQuizQuestion.cgi*
Anzeige einer Frage inklusive Antwortalternativen in einem HTML-Formular. Das Skript liest *question.template* als Vorlage für das Layout ein. Frage und Antworten werden an den entsprechenden Platzhaltern eingefügt.
- *examQuizCheckQuestion.cgi*
 - erhält als Parameter den ÜbergabeString des von *examQuizQuestion.cgi* generierten HTML-Formulars
 - prüft, ob die Frage korrekt beantwortet wurde

- zeigt die Erläuterungen zur Frage an, wenn der Nutzer dies wünscht oder die Frage falsch beantwortet wurde. Dazu wird *explanation.template* eingelesen und die Platzhalter ersetzt.
- *examQuizOverview.cgi*
zeigt nach der letzten Frage eine Übersicht (optional mit Erläuterungen) aller gestellten Fragen an. Die vom Nutzer gewählten bzw. richtigen Antworten werden markiert. Das Skript liest *overview.template*, *over_question.template* und *over_question_small.template* als Vorlage für das Layout ein und ersetzt die Platzhalter.

Anhang F - Sonstige Skripte

Export von Lernobjekten aus der Datenbank als XML-Datei konform zum IMS-LOM-Standard	
Skript	createXMLIMSMetadataRecord.pl
Entwicklungs- umgebung	WindowsME/2000 mit: ActivePerl 5.6.0, build 618 (http://www.activestate.com/)
Siehe auch	Abschnitt 6.7, S. 129

createXMLIMSMetadataRecord.pl bekommt als Übergabe-Parameter die Lernobjekt-ID (LO_ID) des zu exportierenden Lernobjekts. Beim Start liest das Skript aus der Datei *dejavu.ini* die Datenbankdaten (siehe E.2.2, S. 184). Ausgabe ist eine Datei *LO_ID.xml*, die gültig zur DTD der Version 1.2, Final Specification, der *IMS Meta-data Specification* vom Juni 2001 (*imsmd_rootv1p2.dtd*), ist.

Export von Lernpfaden konform zum IMS Content Packaging Model	
Skript	convertDatToImsmanifest.pl
Entwicklungs- umgebung	WindowsME/2000 mit: ActivePerl 5.6.0, build 618 (http://www.activestate.com/)
Siehe auch	Abschnitt 6.7, S. 129

Zur Konvertierung der Lernpfade (siehe dat-Files in Anhang E.3.1, Seite 190) in ein zum IMS Content Packaging Model konformes Manifest-File *imsmanifest.xml* dient das Skript *convertDatToImsmanifest.pl*. Es liest beim Starten die Datei *convertDatToImsmanifest.ini*, die folgendermaßen aussieht:

```
f:/nenalearn_prepare/skripte/dat/ # hier liegen die dat-Files
Lernobjekte in der Dermatologie # Beschreibung
Dermatologie # Titel
dermatology # keywords
http://127.0.0.1:8090/ # Basis-Url (=htmlPath in E.3.1, S.190)
01Lerntexte.dat;02Fallbeispiele.dat;03Quiz.dat # Filenamen der Dat-Files
Lerntexte;Fallbeispiele;Quizelemente # Titel der obersten Hierarchie für
jedes dat-File
```

Ausgabe ist eine Datei *imsmanifest.xml*, die dann zum Beispiel mit dem Microsoft LRN-Toolkit weiterverarbeitet werden kann.

Import von Lernpfaden des MS LRN Editor	
Skript	convertImsmanifestToDat.pl
Entwicklungs- umgebung	WindowsME/2000 mit: ActivePerl 5.6.0, build 618 (http://www.activestate.com/)
Siehe auch	Abschnitt 6.7, S. 129

Das Skript liest eine vom *LRN Editor* des *MS LRN Toolkit* erzeugte *imsmanifest.xml* Datei ein und generiert als Ausgabe entsprechende dat-Dateien (siehe Anhang E.3.1, Seite 190). Es wird für jeden Eintrag, der im Inhaltsverzeichnis des *imsmanifest.xml* auf oberster Ebene liegt ein eigenes dat-File erzeugt. Auf diese Weise kann der im *LRN Editor* erzeugte Navigationsbaum in *DEAJAVULearn* mit dem Skript *convertLernpfad.pl* (Anhang E.3.1) importiert werden.

Literaturverzeichnis

- [1] Baumann R. P. et al.: **Tumor-Histologie-Schlüssel ICD-O-DA**, Springer-Verlag, Heidelberg, New York, 1978
- [2] Blaschek W., Ebel S., Hackenthal E., Holzgrabe U., Keller K., Reichling J.: **HagerROM 2001**, Springer-Verlag, Heidelberg, 2000
- [3] Blaschek W., Ebel S., Hackenthal E., Holzgrabe U., Keller K., Reichling J.: **HagerROM 2002**, Springer-Verlag, Heidelberg, 2002
- [4] Enderle, J.: **XML in relationalen Datenbanken**, Informatik Spektrum, 24/6, S.357-368, 2001
- [5] Henseler T., Kreusch J., Tausch I. (Hrsg.): **Dermatologischer Diagnosenkatalog**, Thieme, Stuttgart, 1996
- [6] Höhn H.: **Combining Image Archives, Audio- and Video-Sequences for Teleteaching Lectures On Dermatology**, 2nd European Symposium on Teledermatology, Zürich, Switzerland, 2000
- [7] Höhn H.: **Digital aufbereitete Vorlesungen in der Dermatologie**, Biomedical Journal, Heft 56 / Oktober 2000, S.27-29, 2000
- [8] Höhn H., Esser W., Hamm H., Albert J.: **Image archives, audio- and video-sequences for teleteaching**. In: Burg G. et al (Hrsg.): Current Problems Series: Telemedicine & Teledermatology, Karger-Publisher, Basel (CH), erscheint 2002
- [9] Höhn H. et al.: **Intranet-/Internet Software for Preparation and Presentation of Lectures in Dermatology**. In: Arvanitis T., Eysenbach G., Woodall J. (Hrsg.): Towards the Millenium of Cybermedicine, University of Birmingham, Birmingham/Heidelberg, S. 40-41, 1999
- [10] Hürst W., Müller R., Mayer C.: **Multimedia Information Retrieval from Recorded Presentations**. In: Belkin N.J., Ingwersen P., Leong M. (Hrsg.): SIGIR 2000, ACM Press, New York, S. 339-341, 2000
- [11] Karl S., Höhn H., Hamm H., Albert J.: **Semantic Net Toolbox for Images in Medical Education: ein multimediales Werkzeug für die Dermatologie-**

- Vorlesung**, Hautarzt, 50 Suppl. 1, S.164, 1999
- [12] Körner, W.: **Videsequenzen und Animationen in Hypertext-Tutorials**, Diplomarbeit, Lehrstuhl für Informatik II der Universität Würzburg, 1998
- [13] Krippner, S: **Semantische Netze und Markierungssprachen mit Anwendungen in der klinischen Dermatologie**, Diplomarbeit, Lehrstuhl Informatik II der Universität Würzburg, 1999
- [14] Ringland G.A., Duce D.A. (eds): **Approaches to Knowledge Representation - An Introduction**, Research Study Press LTD., Letchworth, England, 1988
- [15] Said A., Pearlman W.A.: **An Image Multiresolution Representaion for Lossless and Lossy Image Compression**, IEEE Trans. on Image Processing, vol. 5, S.1303--1310, 1996
- [16] Schmoeckel, C.: **CD Klinische Dermatologie**, Georg Thieme Verlag, Stuttgart, 1997
- [17] Schulmeister Prof. Dr. R.: **Selektions- und Entscheidungskriterien für die Auswahl von Lernplattformen und Autorenwerkzeugen**, <http://serverprojekt.fh-joanneum.at/noflash/thema/lernpl/material/Plattformen.pdf>, 2000, Stand: 04/2002
- [18] Sowa J.F.: **Principles of Semantic Networks - Explorations in the Representation of Knowledge**, Morgan Kaufmann Publishers, Inc., San Mateo, Kalifornien, 1991
- [19] Süß C., Freitag B.: **Wiederverwendung von Inhalten und Strukturen offener Lernsysteme**, Fakultät für Mathematik und Informatik, Universität Passau, 1999
- [20] Süß C., Freitag B.: **Meta-Daten als Grundlage für Adaption in Lernsystemen**, Fakultät für Mathematik und Informatik, Universität Passau, 1999
- [21] Toppe, E.: **GK 3 Dermatologie, Originalprüfungsfragen mit Kommentar, Schwarze Reihe**, Georg Thieme Verlag, Stuttgart, 2001
- [22] Ziegler, C.: **Deus ex Machina - Das Web soll lernen, sich und uns zu verstehen**, c't, 6/2002, S.132-137, 2002
- [23] Messe **MEDICA1998**, Düsseldorf: *SENTIMED-Projekt*, MedicaMedia, Medienstraße, 1998
- [24] Messe **MEDICA2001**, Düsseldorf: *DEJAVU*, MedicaMedia, Medienstraße, 2001

-
- [25] Messe **SYSTEMS 97**, München: *Dermatologie Online Lexikon*, "Telekooperation in Forschung und Lehre", 1997
- [26] Albert-Ludwigs Universität Freiburg: **Authoring on the Fly**, <http://ad.informatik.uni-freiburg.de/aof/>, Stand: 04/2002
- [27] Anderson T, Wason T.: **IMS Meta-Data Best Practice and Implementation Guide**, <http://www.imsproject.org/metadata/mdbestv1p1.html>, Stand: 04/2000
- [28] **ARIADNE Foundation**, <http://www.ariadne-eu.org/>, Stand: 04/2002
- [29] **Avisynth**, <http://math.berkeley.edu/~benrg/avisynth.html>, Stand: 2002
- [30] Berners-Lee T. : **Metadata Architecture**, <http://www.w3.org/DesignIssues/Metadata>, Stand: 04/2002
- [31] **Blackboard Inc.**, <http://www.blackboard.com/>, Stand: 04/2002
- [32] **Bundesministerium für Bildung und Forschung (bmb+f)**, <http://www.bmbf.de>, Stand: 04/2002
- [33] **Dermatologie Praktikum 2000**, <http://www.derma2000.de/>, Stand: 04/2002
- [34] **Dermatology Image Bank**, <http://medstat.med.utah.edu/kw/derm/>, Stand: 2002
- [35] **Dermatology Online Atlas**, <http://dermis.multimedica.de/doia/>, Stand: 04/2002
- [36] **Deutsches Institut für Medizinische Dokumentation und Information (DIMDI)**, <http://www.dimdi.de/>, Stand: 04/2002
- [37] **Dublin Core Metadata Initiative**, <http://dublincore.org/>, Stand: 04/2002
- [38] EDBI: **META-LIB**, <http://www.dbi-berlin.de/projekte/einzproj/meta/meta00.htm>, Stand: 04/2002
- [39] **e-Teach GmbH**, <http://www.e-teach.de/>, Stand: 04/2002
- [40] Fakultät für Mathematik und Informatik: **Learning Material Markup Language 1.1**, <http://daisy.fmi.uni-passau.de/pakmas/lmml/11/doc/de/html/all.xml>, Stand: 04/2002
- [41] Forum Bildung: **Neue Lern- und Lehrkultur - Bericht der Expertengruppe**, http://www.forum-bildung.de/bib/material/experten10_neu.pdf, 04/2002, Stand: 2001
- [42] **FORUM BILDUNG**, <http://www.forum-bildung.de/>, Stand: 04/2002

- [43] **Freedom VR**, <http://www.honeylocust.com/vr/>, Stand: 2002
- [44] IEEE Standards Department: **Draft Standard for Learning Object Metadata (WD 6.1)**, http://ltsc.ieee.org/doc/wg12/LOM_WD6-1_1_without_tracking.pdf, Stand: 04/2002
- [45] IMS Global Learning Consortium: **IMS Question & Test Interoperability**, <http://www.imsproject.org/question/index.html>, Stand: 04/2002
- [46] IMS Global Learning Consortium: **IMS Content Packaging Information Model**, 2001
- [47] **IMS Global Learning Consortium, Inc. (IMS)**, <http://www.imsproject.org/>, Stand: 04/2002
- [48] Klinikum der Friedrich-Schiller-Universität Jena, Julius-Maximilians Universität Würzburg, Ruprecht-Karls-Universität Heidelberg: **DEJAVU - Dermatological Education as Joint Accomplishment of Virtual Universities**, <http://www.projekt-dejavu.de/>, Stand: 04/2002
- [49] Koubek A., Lo S., Posch R., Rehatschek H., Rußegger S., Wagner T.: **Multimediale Lernobjekte - Bausteine für einen individuellen Unterricht**, <http://zml.fh.joanneum.at/publikationen/publ97/me97.htm>, 1997, Stand: 04/2002
- [50] Lehrstuhl für Informatik II der Universität Würzburg: **SENTIMED - Semantic Net Toolbox for Images in Medical Education**, <http://www2.informatik.uni-wuerzburg.de/sentimed/>, Stand: 04/2002
- [51] Melton J.: **Information technology-Database languages - SQL- Part 14: XML-Related Specifications (SQL/XML)**, http://www.wiscorp.com/sql/SQL-XML_WorkingDraft_2001_12.pdf, 2001, Stand: 04/2002
- [52] Microsoft: **Office 2000 HTML Filter 2.0**, <http://office.microsoft.com/downloads/2000/Msohtmf2.aspx>, Stand: 04/2002
- [53] Microsoft: **Microsoft Learning Resource iNterchange (LRN)**, <http://www.microsoft.com/eLearn/>, Stand: 04/2002
- [54] **MURMEL - Multimediales Ausbildungssystem für die Medizinische Lehre**, <http://www.medizin.uni-tuebingen.de/nrad/murmel/>, Stand: 04/2002

-
- [55] National Library Of Medicine: **PubMed**, <http://www4.ncbi.nlm.nih.gov/PubMed/>, Stand: 04/2002
- [56] National Library of Medicine (NLM): **Medical Subject Headings**, <http://www.nlm.nih.gov/mesh/meshhome.html>, Stand: 04/2002
- [57] National Library of Medicine (NLM): **Unified Medical Language System (UMLS)**, <http://umlsinfo.nlm.nih.gov/>, Stand: 2001
- [58] **National Library of Medicine (NLM)**, <http://www.nlm.nih.gov/>, Stand: 2002
- [59] Piette W.: **An Introduction to Basic Dermatology**, <http://www.vh.org/Providers/Lectures/PietteDermatology/BasicDermatology.html>, 1995, Stand: 04/2002
- [60] Projektträger Neue Medien in der Bildung: **Spezialthema - Lernplattformen: Standardisierungs-Initiativen** , http://www.medienbildung.net/lernplattformen/lernplattformen_5.php/spezialthemen/lernplattformen, Stand: 04/2002
- [61] Rosendahl J.: **Medizinische Lernprogramme im Internet**, <http://www.projektdejavu.de/pages/deutsch/docs/lernprogramme.pdf>, 2002, Stand: 04/2002
- [62] Virtuelle Hochschule Oberrhein: **CAMPUS Lehr- und Lernsystem**, <http://campus.fh-heilbronn.de/>, Stand: 04/2002
- [63] Wason T. D. : **Dr. Tom's Meta Guide**, <http://www.imsproject.org/drtommeta.html>, Stand: 2001
- [64] **World Health Organization (WHO)**, <http://www.who.int/home-page/>, Stand: 2002
- [65] World Wide Web Consortium (W3C): **XSL Transformations (XSLT) Version 1.0**, <http://www.w3.org/TR/xslt/>, Stand: 04/2002
- [66] World Wide Web Consortium (W3C): **HTML-Tidy**, <http://www.w3.org/People/Raggett/tidy/>, Stand: 04/2002
- [67] World Wide Web Consortium (W3C): **Semantic Web**, <http://www.w3.org/2001/sw/>, Stand: 04/2002
- [68] World Wide Web Consortium (W3C): **Resource Description Framework (RDF)**, <http://www.w3c.org/rdf>, Stand: 2002

- [69] World Wide Web Consortium (W3C): **Extensible Markup Language (XML)**,
<http://www.w3c.org/xml/>, Stand: 2002

Abbildungsverzeichnis

Abbildung 1: Modell des IMS Content Packaging Information Model aus [46].	30
Abbildung 2: Microsoft LRN Editor.	37
Abbildung 3: Microsoft LRN-Viewer.	38
Abbildung 4: e-Author: Navigationsbaum und Vorlagen.	39
Abbildung 5: e-Student. Lernumgebung der e-Teach GmbH	40
Abbildung 6: Blackboard 5. Lernumgebung.	41
Abbildung 7: AOF. Erstellung und Präsentation von Vorträgen.	41
Abbildung 8: CAMPUS Lehr- und Lernsystem	43
Abbildung 9: MURMEL Ausbildungssystem für die medizinische Lehre.	43
Abbildung 10: Client-Server-Architektur	56
Abbildung 11: Ablauf der Datenerfassung bei den Bildobjekten	58
Abbildung 12: Datenerfassung. Übersicht über alle Scan-Verzeichnisse.	61
Abbildung 13: Datenerfassung: Eingabe der Grunddaten für ein Bild	61
Abbildung 14: Datenerfassung: Übersicht über ein Scan-Verzeichnis.	62
Abbildung 15: CGI-Suche im Bildarchiv	63
Abbildung 16: Datenerfassung: Java-Applikation zur Eingabe der übrigen Daten.	64
Abbildung 17: Generierung des semantischen Netzes.	68
Abbildung 18: Aufbau der Knoten im semantischen Netz	69
Abbildung 19: Semantisches Netz im Browser mit interner Zwischenablage	70
Abbildung 20: Applet zur Bildbetrachtung.	72
Abbildung 21: Applet zum Editieren einer PubMed-Suchanfrage.	73
Abbildung 22: Ergebnis der PubMed-Suchanfrage aus Abbildung 21	74
Abbildung 23: Vorlesungsgenerierung. Vorauswahl.	75
Abbildung 24: Vorlesungserstellung. Suchdialog mit Picklisten.	76
Abbildung 25: Vorlesungserstellung.	78
Abbildung 26: Vorlesungspräsentation: Übersicht	80
Abbildung 27: Module der Lernumgebung, lokale und zentrale Datenbanken.	86
Abbildung 28: Lernobjekt-Hierarchie.	94
Abbildung 29: Beispiel für einen Lernpfad	94

Abbildung 30: Verschiedene Arten von Tabellen in der Datenbank	96
Abbildung 31: Client/Server-Architektur mit Zugriff auf ein Netzlaufwerk	97
Abbildung 32: Suche in der Lernobjekt-Datenbank.	101
Abbildung 33: Autorenwerkzeuge. Übersicht.	102
Abbildung 34: Festlegung eines Bildes als Vorschaubild für ein Video	103
Abbildung 35: DEJAVUCapt. Verschlagwortung/Anonymisierung.	104
Abbildung 36: Konfiguration von DEJAVUCapt	105
Abbildung 37: Nachverschlagwortung, hier am Beispiel Zielgruppe	106
Abbildung 38: Aufnahme von Moulagen zur 3D-Darstellung	108
Abbildung 39: Powerpoint-AddIn: Standardsuche und erweiterte Suche	109
Abbildung 40: Powerpoint-AddIn: Suchergebnis und Einfügen	110
Abbildung 41: Vorlesung zur Nachbereitung	111
Abbildung 42: World-Vorlage zur Erfassung von Lerntexten.	113
Abbildung 43: Erstellung von Fallbeispielen.	114
Abbildung 44: Eingabe von Übungsfragen in einem WWW-Browser	116
Abbildung 45: Fallquiz. Eingabe der Kurzanamnese und Auswahl der Bilder.	116
Abbildung 46: Fallquiz. Eingabe der Anamnese.	117
Abbildung 47: Fallquiz. Eingabe der Effloreszenzen.	118
Abbildung 48: Fallquiz. Auswahl der Diagnostik und der Diagnose.	118
Abbildung 49: DEJAVULearn - Umgebung	120
Abbildung 50: Moulagen. 3D-Darstellung.	123
Abbildung 51: Vorlesungen zur Nachbereitung mit Bild, Text, Ton.	125
Abbildung 52: Ablauf des Fallquiz.	127
Abbildung 53: DEJAVULearn. Bilderquiz.	127
Abbildung 54: DEJAVULearn. Übungsaufgaben.	129
Abbildung 55: Lernpfade und Lernobjekte im MS LRN-Viewer	131
Abbildung 56: Entity-Relationship Diagramm	169