



# **Propädeutik metadatenbasierter Publikationsserver**

## **Ontologie und konkordante Implementierung**

Wolfgang Eiden  
*Oktober 1999*

### **Projektarbeit**

AG Künstliche Intelligenz und wissensbasierte Systeme  
Fachbereich Informatik, Betreuung durch Dr. Wolfgang Lenski

**1991 Mathematics Subject Classification:**  
68M05, 68P20, 68T30, 68T35

*Das Verhältnis zwischen dem Informationsträger (Substanz oder Energie) und der Information gleicht dem Zusammenhang zwischen den Gegenständen und ihren Eigenschaften: Ein Gegenstand ist nicht dasselbe wie seine Eigenschaften, sondern er besitzt Eigenschaften; die Eigenschaften ihrerseits brauchen den Gegenstand, um existieren zu können.*

BERNHARD HASSENSTEIN

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>5</b>
<b>2. Anforderungen</b>	<b>6</b>
2.1. Forderungen an den Server.....	6
2.2. Forderungen an das Retrieval.....	7
2.3. Forderungen an die Ressourcen.....	7
2.4. Sonstige Anforderungen.....	7
<b>3. Metadaten</b>	<b>9</b>
3.1. Metadatenbasierte Suchmaschinen.....	9
3.2. Dublin Core Metadata Set.....	10
3.3. Resource Description Framework.....	17
3.4. Resümee.....	23
<b>4. Das Resource-Discovery-System Harvest</b>	<b>24</b>
4.1. Systemübersicht.....	24
4.2. Gatherer.....	25
4.3. Broker.....	30
4.4. Replicator.....	32
<b>5. Der Dokumentenserver KLUEDO</b>	<b>33</b>
5.1. Der Dokumentenserver aus Sicht des Benutzers .....	35
5.2. Systemumgebung.....	43
5.3. Struktureller Aufbau.....	46
5.4. Konfigurationsdateien und Bibliotheken.....	49
5.5. Metadatenmanagement.....	57
5.6. Assimilationsmöglichkeiten.....	60
5.7. Quelltexte.....	62
<b>6. Literatur</b>	<b>111</b>

# Abbildungsverzeichnis

<b>2. Anforderungen</b>	
2.4.a	Datenfluss eines metadatenbasierten Publikationsservers..... 8
<b>3. Metadaten</b>	
3.2.a	Dublin Core Beschreibung..... 11
3.3.a	Modell einer einfachen Relation (RDF) ..... 17
3.3.b	Beispiel einer einfachen Relation (RDF) ..... 17
3.3.c	Modell einer erweiterten Relation (RDF) ..... 18
3.3.d	Beispiel einer erweiterten Relation (RDF) ..... 18
3.3.e	Beispiel eines einfachen Bag-Containers ..... 19
3.3.f	RDF-Einbettung (Serialisationssyntax) in HTML-Code (ältere Browser)..... 22
3.3.g	RDF-Einbettung (Abbreviated Syntax) in HTML-Code..... 23
<b>4. Das Resource-Discovery-System Harvest</b>	
4.1.a	Softwarekomponenten von Harvest..... 24
4.2.a	Schritte zur Informationsextrahierung (Essence) ..... 26
4.3.a	Interface des Broker-Subsystems..... 30
4.3.b	Verteilbarkeit des Gathering- und Brokering-Prozesses..... 32
<b>5. Der Dokumentenserver KLUEDO</b>	
5.0.a	Systemarchitektur..... 34
5.1.a	Startseite..... 35
5.1.b	Deutsche Indexseite des Fachbereiches Mathematik..... 36
5.1.c	Suchmaske..... 37
5.1.d	Ergebnis einer Suche..... 37
5.1.e	Auszug aus einer Veröffentlichungsliste..... 38
5.1.f	Navigationsschaltflächen..... 38
5.1.g	Informationen zu einer Veröffentlichung..... 39
5.1.h	Eingabemaske (Mathematik)..... 40
5.1.i	Testausgabe von Metadaten (mit unzulässigen Angaben) ..... 41
5.1.j	Testausgabe von Metadaten..... 41
5.1.k	Speicherungsbestätigung und Fileuploadmöglichkeit..... 42
5.2.a	Beispielsseite..... 45
5.2.b	Integration von PHP..... 45
5.3.a	Struktur eines Fachbereiches..... 47
5.4.a	Zentrale Konfigurationsdateien..... 49
5.4.b	Bibliotheken..... 50
5.5.a	Listenerzeugung..... 57
5.5.b	Hilfsskripte..... 58
5.5.c	Zentraler Metamaker..... 59
5.5.d	Selbstreferenzierung..... 60

# Abkürzungsverzeichnis

<i>ASCII</i>	American Standard Code for Information Interchange
<i>BMBF</i>	Bundesministerium für Bildung und Forschung
<i>DC</i>	Dublin Core
<i>DFG</i>	Deutsche Forschungsgemeinschaft
<i>DFN</i>	Deutsches Forschungsnetz
<i>DVI</i>	Device Independent Format
<i>FTP</i>	File Transfer Protocol
<i>HTML</i>	HyperText Markup Language
<i>HTTP</i>	HyperText Transfer Protocol
<i>IRTF-RD</i>	Internet Research Task Force Research Group on Resource Discovery
<i>ISO</i>	International Standards Organization
<i>KLUEDO</i>	Kaiserslauterer uniweiter elektronischer Dokumentenserver
<i>MIME</i>	Multipurpose Internet Mail Extensions
<i>MSC</i>	Mathematics Subject Classification
<i>NISO</i>	National Information Standards Organization
<i>PDF</i>	Portable Document Format
<i>PS</i>	PostScript
<i>RDF</i>	Resource Description Framework
<i>RTF</i>	Rich Text Format
<i>SGML</i>	Standard Generalized Markup Language
<i>SOIF</i>	Summary Object Interchange Format
<i>URL</i>	Uniform Resource Locator
<i>W3C</i>	World Wide Web Consortium
<i>XML</i>	Extensible Markup Language

# 1. Einleitung

Gerade in einer von Globalisierungstendenzen geprägten Zeit wie heute - in der sowohl das Angebot als auch die Nachfrage an Informationen exponentiell wächst - gewinnen Publikationsserver an zunehmender Bedeutung: Die weltweite Verfügbarkeit und Recherchierbarkeit von elektronisch gespeicherten Ressourcen (bei gleichzeitig minimalen Kosten) sind zentrale Forderungen der Informationsgesellschaft und Antriebsmotor für internationale Informationsstrukturen. Bei der Konzeption von modernen Publikationsservern sollte daher a priori globales Denken und international anerkannte Konzepte im Vordergrund stehen.

Ziel der vorliegenden Arbeit war die Unterstützung des Aufbaus eines Dokumentenservers für die Universität Kaiserslautern, der diesen Ansprüchen genügt. Im Rahmen dieser von Herrn Dr. Wolfgang Lenski (Fachbereich Informatik) betreuten Arbeit habe ich daher im ersten Teil als Orientierungsmöglichkeit eine Studie über aktuelle Bedürfnisse und Lösungsansätze angefertigt und die wesentlichsten Erkenntnisse niedergeschrieben: Nach der Formulierung eines Anforderungskataloges für Server, Ressourcen und Retrieval wird das international verbreitete Konzept der Metadaten als Ressourcenbeschreibung vorgestellt. Dieses dient zur Unterstützung effizienter und qualifizierter Retrievalverfahren die ihre Anwendung in komplexeren Systemen, wie beispielsweise dem, an der University of Colorado entwickelten, Resource-Discovery-System Harvest finden.

Aufbauend auf diesen Erkenntnissen zeigt der zweite Teil in exemplarischen Zügen die Applizierung der vorgestellten Konzepte anhand der konkreten Realisation des Dokumentenservers *KLUEDO* der Universität Kaiserslautern. Ich beschränke mich hierbei im Wesentlichen auf den von mir entworfenen Teil der Systemarchitektur und das entwickelte Metadatenmanagementsystem.

## 2. Anforderungen

Für die Akzeptanz und Nutzung eines Publikationsservers durch Dritte ist neben der Bereitstellung der elektronisch gespeicherten Ressourcen eine Reihe von weiteren Anforderungen an den Server zu stellen: Zunächst sollte das Angebot von möglichst vielen Benutzern ohne großen Aufwand nutzbar sein. Dabei sind beschreibende Informationen über die erfassten Ressourcen in strukturierter Art und Weise anzubieten. Desweiteren sollte die Möglichkeit einer effizienten und restriktiven Suche nach Ressourcen zur Verfügung gestellt und die (kontrollierte) Erfassung weiterer Ressourcen ermöglicht werden. Damit der Benutzer die Ressourcen als Informationsquelle verwenden kann, ist es zwingend notwendig, dass das Publikationsformat weit verbreitet und plattformunabhängig ist. Dies sind Ausgangspunkte für viele weitere Anforderungen an den Publikationsserver, das Retrieval und die Ressourcen.

### 2.1. Forderungen an den Server

Die Systemarchitektur sollte auf internationalen Standards und offenen Schnittstellen basieren. Daher scheint aus heutiger Sicht die Nutzung des Internets mit seinen mittlerweile weltweit anerkannten Informationsdiensten (wie beispielsweise *World Wide Web* oder *electronic mail*) mit implizierter Plattformunabhängigkeit die logische Konsequenz. Ein weiterer Vorteil ist hierbei die Möglichkeit der Nutzung des Angebotes ohne die Installation spezieller Software.

Aufgrund der jedoch rasanten Entwicklung des Internets und der sich damit ändernden Standards, sollte der Publikationsserver auf einem zentralen System basieren. Nur so ist es mit einem vertretbarem Aufwand möglich, mit den aktuellen Entwicklungen Schritt zu halten und den sich ändernden Bedürfnissen zu genügen. So erfordert beispielsweise auch der Datenaustausch mit Bibliothekskatalogen und fachbezogenen Servern zentrale Anknüpfungspunkte zur Realisierung eines Systems mit offenen Schnittstellen. Diese sind für die Akzeptanz eines Publikationsservers von zentraler Bedeutung, da nur durch Einbettung in bestehende Infrastrukturen die Informationen einem Spektrum von möglichst vielen Benutzern zur Verfügung gestellt werden. Entscheidend ist desweiteren die Art und Weise des Angebotes:

Das Web-User-Interface (als Verbindung zwischen Internet und dem verwendeten Datenbankmanagementsystem) sollte eine benutzerfreundliche Nutzung erlauben, wobei eine multilinguale Ausrichtung, eine klare Struktur, die Konzentration auf das Wesentliche und die Beschränkung auf ein qualitativ hochwertiges Informationsangebot von oberster Bedeutung sind. Die fachspezifische inhaltliche Erschließung von Ressourcen (auf verbale und klassifikatorische Art), typspezifische Veröffentlichungslisten und übersichtliche Suchergebnisse leisten hierbei einen nicht unwesentlichen Beitrag (vgl. [19]).

## 2.2. Forderungen an das Retrieval

Die Informationsangebote eines Publikationsservers sollten selbstverständlicherweise durch geeignete Indexierungs- und Suchwerkzeuge durchsuchbar sein. Für eine effiziente Suche sind nähere Informationen über die publizierten Ressourcen wie Autor, Titel oder Abstrakt (Metainformationen) hilfreich. Als Beispiel für Metadatenerfassungssysteme seien hier das *Dublin Core Metadata Set* (DC) und das *Resource Description Framework* (RDF) genannt, die im nächsten Kapitel näher untersucht werden. Bei Nutzung von Metadaten kann eine effektive, spezifizierte und strukturierte Suche durch Verwendung eines möglichst kontrollierten Vokabulars (unter Einbeziehung von Normdaten) unterstützt werden. Die Anzeige entsprechender Treffermengen sollte darüber hinaus übersichtlich und informativ gestaltet sein, da dies der Entscheidungspunkt zur Verwendung einer Ressource ist (vgl. [10],[19]).

## 2.3. Forderungen an die Ressourcen

Neben den Metainformationen über eine Ressource ist diese kanonischerweise selbst von elementarer Bedeutung. Das Publikationsformat sollte daher so gewählt sein, dass es für eine möglichst breite Benutzerschicht geeignet ist. Prädestiniert hierfür sind Formate für die frei verfügbare Viewer (für die gängigsten Plattformen) angeboten werden, dem Druckformat weitgehendst entsprechen und die Integration in den benutzten Browser ermöglichen. Aus diesen Gründen ergeben sich Anforderungen an das Anlieferungsformat (welches nicht notwendigerweise dem Publikationsformat entsprechen muss): Es sollte ein weit verbreitetes Standardformat sein, von möglichst vielen gängigen Datenverarbeitungssystemen erzeugt werden können, und vorallem die Konvertierbarkeit zum Präsentations- und Publikationsformat gewährleisten (vgl. [28]). Nähere Informationen hierzu finden sich in der Studienarbeit von Herrn Ohst [37] und in den Ausführungen von Herrn Martin [31].

## 2.4. Sonstige Anforderungen

Additiv zu allen bisher genannten Aspekten sollte die Qualität der Ressourcen gesichert sein: Neben der verlangten hohen Qualität der Ressource selbst, sollten die zugehörigen Metainformationen die Ressource möglichst treffend und detailliert beschreiben. Dies ist in der Regel allerdings nur durch die Autoren einer Publikation möglich. Daher sollte der Publikationsserver Outsourcing unterstützen, d.h. die Autoren sollten zusätzlich zur Ressource noch die Metainformationen liefern.

Da die Autoren im allgemeinen jedoch mit dieser Materie nicht vertraut sind, ist ihnen eine maximale Unterstützung im Bereich der Dokumenterstellung und Metadatenlieferung zu gewähren. Einweisungen von Seite der serverbetreibenden Institution wären empfehlenswert, da sonst unter Umständen aufwendige Nachbearbeitungen notwendig werden und somit einen eventuellen Massendurchsatz hemmen. Da das dafür benötigte Personal jedoch sehr kostenintensiv ist und die Personalressourcen sehr knapp sind, scheint die Anbieterung von entsprechend hochwertigen Hilfetexten ein passabler Weg. Neben Empfehlungen für die zu verwendenden Werkzeuge sollten den Autoren auch Vorschläge zur inhaltlichen Strukturierung der Dokumente unterbreitet werden um eine leichtere Projektion auf die Metadaten (wie beispielsweise das Abstrakt und die Klassifikation) zu ermöglichen. Für die Aufnahme der Dokumente in den Publikationsserver sollten den Autoren über das *World Wide Web* intuitiv zu bedienende Eingabemasken bereitstehen, in der diese die Metadaten eintragen können, ohne dass eine Kenntnis der internen Struktur vorausgesetzt wird. Diese Angaben sollten nach dem Eingang bei der serverbetreibenden Institution im Sinne der Qualitätssicherung und der Dokumentkontrolle erneut geprüft werden und erst dann gespeichert und öffentlich zugänglich gemacht werden (vgl. [28]):

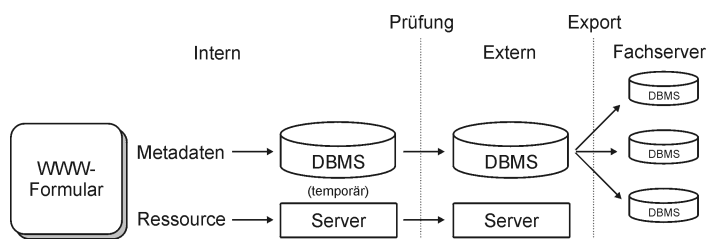


Abbildung 2.4.a : Datenfluss eines metadatenbasierten Publikationsservers



## 3. Metadaten

Dieses Kapitel widmet sich einer Studie über die Verwendung möglicher Metadatenbeschreibungssysteme. Dabei wollen wir unter Metadaten strukturierte Daten verstehen mit deren Hilfe Ressourcen beschrieben und charakterisiert werden können. Sie liefern Basisinformationen über ein Dokument, wie beispielsweise Titel, Autor oder Zeitpunkt der Veröffentlichung. Durch Orientierung an internationalen Standards und Konventionen können durch Nutzung von Metadaten effizientere Retrieval-Mechanismen realisiert werden, so wie dies in modernen Suchmaschinen wie beispielsweise *Altavista* [1], *Fireball* [17] oder *Infoseek* [23] geschieht.

### 3.1. Metadatenbasierte Suchmaschinen

Suchmaschinen erhalten ihre Ergebnisse durch entsprechendes Abfragen ihrer Datenbanken. Die Datenbankinhalte werden im allgemeinen durch sogenannte Robots automatisch generiert. Dazu werden Server (beispielsweise durch Linkverfolgung) rekursiv abgesucht und Informationen zu bekannten Dokumenttypen gespeichert.

Für Suchmaschinen, die nicht auf Metadaten basieren, sind diese Daten zunächst unstrukturiert, weshalb keine exakte Analyse der zu indexierenden Daten möglich ist. Hieraus ergeben sich beim Retrievalprozeß zahlreiche negative Effekte und unerfüllte Qualitätswünsche wie beispielsweise eine zu hohe Trefferzahl, eine zu geringe Präzision, unvollständige Resultate oder Schwierigkeiten bei der Bewertung bzw. Selektion von Ergebnissen.

Basiert die Dokumentindexierung allerdings auf Metadaten, so ist aufgrund der strukturierten und exakt analysierbaren Daten eine bereichsspezifische Selektion zur Unterstützung hochwertiger Recherchen möglich. Die Struktur der Metadaten sollte hierbei so angelegt sein, dass möglichst viele relevanten Informationen erfaßt werden können, die Anzahl der verwendbaren Basiselemente aber möglichst gering ist. Beispielsweise genügt das Ende 1996 eingeführte und mittlerweile international verbreitete *Dublin Core Metadata Set* [15] diesen Forderungen. Während der Standard selbst international zunehmend an Bedeutung gewinnt (vgl. [29]), unterstützten zur Zeit leider nur wenige Suchmaschinen das Metadata Set in seiner gesamten strukturellen Differenzierung. Von den drei oben genannten Suchmaschinen ist dies beispielsweise nur bei *Fireball* [17] der Fall: Die allgemeine Syntax für eine Metadatenuche in *Fireball* lautet:

```
metaname:suchwort
```

Mehrere Suchausdrücke können hier durch logische Operatoren (wie beispielsweise „UND“, „ODER“ oder „NICHT“) verknüpft werden. Näheres hierzu entnehme man bitte der Online-Hilfe von *Fireball*.

## 3.2. Dublin Core Metadata Set

Ein Metadatenstandard sollte mehreren Anforderungen genügen: Zunächst sollten die Metadaten maschinell verarbeitet, aber trotzdem auch von einem Menschen gelesen werden können, also textuell gespeichert werden. Desweiteren sollten sie unabhängig vom Datenformat der Ressource sein und die Austauschbarkeit mit anderen Systemen gewährleisten. Desweiteren sollten Klassifizierungsschemata unterstützt werden und Erweiterungen problemlos möglich sein (vgl. [40]). Das, mit diesen Richtlinien konforme, *Dublin Core Metadata Set* konzentriert sich auf lediglich 15 wichtige Hauptelemente und bietet die Möglichkeit einer fachspezifischen Ausprägung. Es wurde im Dezember 1996 als DC Version 1.0 von der *Dublin Core Metadata community* veröffentlicht (vgl. [39],[16]):

Nr.	Basiselement	Definition
1	<i>Title</i>	Der vom Verfasser, Urheber oder Verleger vorgegebene Name der Ressource.
2	<i>Creator</i>	Die Person(en) oder Organisation(en), die den intellektuellen Inhalt verantworten. Im Falle mehrerer Autoren ist jeder in einem eigenen Meta-Element zu erfassen.
3	<i>Subject</i>	Schlagwörter, Stichwörter oder Phrasen, die das Thema oder den Inhalt beschreiben. Das Element kann sowohl systematische Daten nach einer Klassifikation oder Begriffe aus anerkannten Thesauri enthalten.
4	<i>Description</i>	Eine textuelle Beschreibung des Ressourceninhalts inklusive eines Abstrakts (bei dokumentähnlichen Ressourcen) oder einer Inhaltsbeschreibung (bei graphischen Ressourcen).
5	<i>Publisher</i>	Die Einrichtung, die verantwortet, dass diese Ressource in dieser Form zur Verfügung steht, wie beispielsweise ein Verleger oder eine Universität.
6	<i>Contributors</i>	Zusätzliche Person(en) und Organisation(en) zu jenen, die im Element <i>Creator</i> genannt wurden, die einen bedeutsamen intellektuellen Beitrag zur Ressource geleistet haben, deren Beitrag aber sekundär im Verhältnis zu denen im Element <i>Creator</i> zu betrachten ist (z.B. Herausgeber, Übersetzer, Illustratoren, Konferenzleiter, Moderatoren).
7	<i>Date</i>	Datumsangaben der Ressource, wie beispielsweise das Datum der letzten Änderung oder das Veröffentlichungsdatum.
8	<i>Type</i>	Die Art der Ressource (z.B. Dissertation, Diplomarbeit).
9	<i>Format</i>	Das datentechnische Format der Ressource. Die Angabe in diesem Feld gibt die erforderlichen Informationen über die Verarbeitungsmöglichkeiten der kodierten Daten.
10	<i>Identifier</i>	Zeichenkette oder Zahl, die eine eindeutige Identifikation des Dokumentes ermöglicht. Bei vernetzten Ressourcen sind URLs vorgesehen.
11	<i>Source</i>	In diesem Element wird das gedruckte oder elektronische Werk, aus dem diese Ressource stammt, eingetragen.
12	<i>Language</i>	Sprachcode der Sprache(n) des intellektuellen Inhalts der Ressource.
13	<i>Relation</i>	Verhältnis zu anderen Ressourcen, die einen formalen Bezug zu dieser Ressource haben, aber als eigenständige Ressourcen existieren (z.B. Kapitel eines Buches oder Bilder eines Dokuments).
14	<i>Coverage</i>	Zeitliche, örtliche, flächenhafte oder ähnliche Aspekte die zur Charakterisierung des Objektes sinnvolle Ergänzungen geben.
15	<i>Rights</i>	Urhebervermerk und rechtliche Bedingungen zur Nutzung dieser Ressource.

Die Vorteile des *Dublin Core Metadata Sets* liegen im internationalen Ansatz, der internationalen Verbreitung, der Erweiterbarkeit und der Wiederholbarkeit der Elemente (wobei keine Pflichtelemente vorgeschrieben sind).

### 3.2.1. Dublin Core Qualifiers

Bei nur fünfzehn Elementen sind Mehrfachbelegungen mit unterschiedlichem Inhalt im allgemeinen unumgänglich. Zur Differenzierung und Interpretation ist die Nutzung der Dublin Core Qualifiers *SCHEME* und *TYPE* vorgesehen (vgl. [42]). Diese werden zur ergänzenden Definition des semantischen Inhaltes der Dublin Core Elemente verwendet:

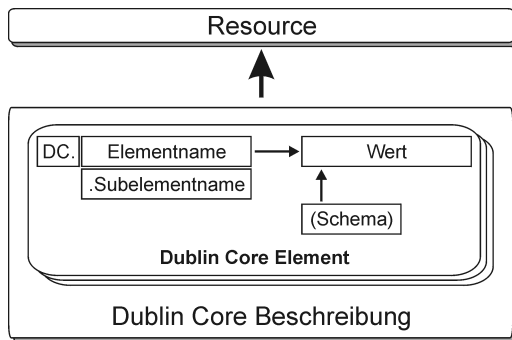


Abbildung 3.2.a : Dublin Core Beschreibung

Das Qualifizierungselement *TYPE* wird zur Spezifizierung des Basiselements benutzt. Hierbei wird die Syntax

```
DC.elementname.typeidentifizier
```

verwendet. Beispiele hierfür sind:

```
DC.Creator.PersonalName
DC.Creator.PersonalName.Address
DC.Date.Created
DC.Description.Abstract
DC.Subject.Topic
```

Das Qualifizierungselement *SCHEME* dient zur Interpretation des Inhalts eines Elements im angegebenen Codesystem. Im Gegensatz zu Werten der Form

```
Wert
```

in denen kein Interpretationsschema angegeben ist, wird hier die Syntax

```
(SCHEME=identifizier) Wert
```

verwendet. Als Beispiele seien genannt:

```
(SCHEME=IMT) application/postscript
(SCHEME=ISBN) 9-87654-321-9
(SCHEME=MSC91) 00A72 General methods of simulation
(SCHEME=URL) http://www.w3.org
(SCHEME=Z39.53) GER
```

### 3.2.2. Spezifizierungstypen

Für die jeweiligen Basiselemente sind im *Dublin Core Metadata Set* unterschiedliche Spezifizierungen vorgesehen. Als Beispiel folgt ein Auszug aus den Spezifikationen des *Math-Net-Projekts* [34] für mathematische Publikationen (vgl. [3]) :

<i>DC Element</i>	Schema / Inhalt	Bedeutung (im Kontext)
<i>DC.Title</i>	...	Titel der Arbeit
<i>DC.Creator.PersonalName</i>	Nachname, Vorname	Name der Person
<i>DC.Creator.PersonalName.Address</i>	(Affiliation)...	Institutname
<i>DC.Creator.PersonalName.Address</i>	(Email)...	email-Adresse
<i>DC.Creator.PersonalName.Address</i>	(Fax)...	Faxnummer
<i>DC.Creator.PersonalName.Address</i>	(Homepage)...	Homepage
<i>DC.Creator.PersonalName.Address</i>	(Keywords)...	Schlagwörter
<i>DC.Creator.PersonalName.Address</i>	(Phone)...	Telefonnummer
<i>DC.Creator.PersonalName.Address</i>	(Postal)...	Anschrift
<i>DC.Publisher.Name</i>	...	Name der Organisation
<i>DC.Publisher.Email</i>	...	email-Adresse der Organisation
<i>DC.Publisher.PersonalName</i>	Nachname, Vorname	Name der Person
<i>DC.Publisher.PersonalName.Address</i>	(Affiliation)...	Institutname
<i>DC.Publisher.PersonalName.Address</i>	(Email)...	email-Adresse
<i>DC.Publisher.PersonalName.Address</i>	(Fax)...	Faxnummer
<i>DC.Publisher.PersonalName.Address</i>	(Homepage)...	Homepage
<i>DC.Publisher.PersonalName.Address</i>	(Keywords)...	Schlagwörter
<i>DC.Publisher.PersonalName.Address</i>	(Phone)...	Telefonnummer
<i>DC.Publisher.PersonalName.Address</i>	(Postal)...	Anschrift
<i>DC.Subject</i>	...	Schlagwörter
<i>DC.Subject.Topic</i>	...	allgemeines Thema
<i>DC.Subject.Msc</i>	(SCHEME=msc91) ...	MSC-Klassifikation
<i>DC.Description.Abstract</i>	(SCHEME=url) #url	Verzweigung auf das Abstrakt
<i>DC.Description.Abstract</i>	...	Abstrakt
<i>DC.Description.Notes</i>	(SCHEME=url) #url	Verzweigung auf die Notizen
<i>DC.Description.Notes</i>	...	Notizen
<i>DC.Date.Created</i>	ISO8601-Datum	Publikationsdatum
<i>DC.Date</i>	ISO8601-Datum	Datum der letzten Änderung
<i>DC.Type</i>	...	Art der Ressource
<i>DC.Format</i>	(SCHEME=IMT) ...	Format der Ressource
<i>DC.Identifier</i>	(SCHEME=url) ...	Url der Ressource
<i>DC.Source</i>	...	Name der Veröffentlichungsserie
<i>DC.Language</i>	(SCHEME=Z39.53) ...	Sprache
<i>DC.Rights</i>	...	rechtliche Bedingungen

### 3.2.3. Interpretationsschemata

Da nur (international-) standardisierte Daten einen Informationsverlust minimieren können, gewinnt die Verwendung von Interpretationsschemata eine besondere Bedeutung. Die gebräuchlichsten Interpretationsschemata werden hier kurz skizziert.

#### 3.2.3.1. Dateitypangaben nach dem MIME-Typ-Standard (Auszug, vgl. [36])

Mime-Typ	Standarderweiterung(en)	Format
<i>application/msword</i>	doc dot rtf wiz	Microsoft Word
<i>application/mspowerpoint</i>	ppt ppz pps pot	Microsoft Powerpoint
<i>application/msexcel</i>	xls xla xlt	Microsoft Excel
<i>application/pdf</i>	pdf	Portable Document Format
<i>application/postscript</i>	ai eps ps	Adobe Postscript
<i>application/rtf</i>	rtf	Rich Text Format
<i>application/x-compress</i>	Z	Compressed Data
<i>application/x-dvi</i>	dvi	TeX DVI Data
<i>application/x-gtar</i>	gtar	GNU Tape Archive
<i>application/x-latex</i>	latex tex	LaTeX Document
<i>application/x-tar</i>	tar	Unix Tape Archive
<i>application/x-tex</i>	tex	TeX Document
<i>application/x-zip-compressed</i>	zip	Zip Compressed Data
<i>text/html</i>	html htm shtml	Hypertext Markup Language
<i>text/plain</i>	asc txt c h	Plain Text
<i>text/x-sgml</i>	sgm sgml	SGML-Datei

#### 3.2.3.2. Datumsangaben nach ISO8601 (Auszug, vgl. [24])

Nach des ISO8601 - Norm wird ein Tagesdatum in der Form

YYYY-MM-TT

angegeben, wobei die Symbole folgende Bedeutung besitzen:

Symbol	Bedeutung	Stelligkeit
YYYY	Jahreszahl	4-stellig
MM	Monat	2-stellig
TT	Tag	2-stellig

Beispiel:

1999-12-31

### 3.2.3.3. Klassifikationen nach der Mathematics Subject Classification 1991 (MSC91)

Die *Mathematics Subject Classification* erlaubt die inhaltliche Klassifikation einer Ressource durch Angabe von jeweils fünfstelligen Codes (vgl. [33]). Insgesamt stehen durch die Definition der Hauptgruppen ( $nn-XX$ ), der Subgruppen ( $nn-cxx$ ) und der Subsubgruppen ( $nn-cnn$ ) drei Klassifikationsebenen zur Verfügung:

Klassifikationsebene 1 (Hauptgruppe):

Gruppe	Beschreibung
00-XX	General
01-XX	History and biography
03-XX	Mathematical logic and foundations
04-XX	Set theory
05-XX	Combinatorics
06-XX	Order, lattices, ordered algebraic structures
08-XX	General algebraic systems
11-XX	Number theory
12-XX	Field theory and polynomials
13-XX	Commutative rings and algebras
14-XX	Algebraic geometry
15-XX	Linear and multilinear algebra; matrix theory
16-XX	Associative rings and algebras
17-XX	Nonassociative rings and algebras
18-XX	Category theory, homological algebra
19-XX	K-theory
20-XX	Group theory and generalizations
22-XX	Topological groups, Lie groups
26-XX	Real functions
28-XX	Measure and integration
30-XX	Functions of a complex variable
31-XX	Potential theory
32-XX	Several complex variables and analytic spaces
33-XX	Special functions
34-XX	Ordinary differential equations
35-XX	Partial differential equations
39-XX	Finite differences and functional equations
40-XX	Sequences, series, summability
41-XX	Approximations and expansions
42-XX	Fourier analysis
43-XX	Abstract harmonic analysis
44-XX	Integral transforms, operational calculus
45-XX	Integral equations
46-XX	Functional analysis

Gruppe	Beschreibung
47-XX	Operator theory
49-XX	Calculus of variations and optimal control
51-XX	Geometry
52-XX	Convex and discrete geometry
53-XX	Differential geometry
54-XX	General topology
55-XX	Algebraic topology
57-XX	Manifolds and cell complexes
58-XX	Global analysis, analysis on manifolds
60-XX	Probability theory and stochastic processes
62-XX	Statistics
65-XX	Numerical analysis
68-XX	Computer science
70-XX	Mechanics of particles and systems
73-XX	Mechanics of solids
76-XX	Fluid mechanics
78-XX	Optics, electromagnetic theory
80-XX	Classical thermodynamics, heat transfer
81-XX	Quantum Theory
82-XX	Statistical mechanics, structure of matter
83-XX	Relativity and gravitational theory
85-XX	Astronomy and astrophysics
86-XX	Geophysics
90-XX	Economics, operations research, programming, games
92-XX	Biology and other natural sciences, behavioral sciences
93-XX	Systems theory; control
94-XX	Information and communication, circuits

Klassifizierungsebene 2 (am Beispiel der Informatik):

Gruppe	Beschreibung
68-00	General reference works (handbooks, dictionaries, bibliographies, etc.)
68-01	Instructional exposition (textbooks, tutorial papers, etc.)
68-02	Research exposition (monographs, survey articles)
68-03	Historical
68-04	Explicit machine computation and programs (not the theory of computation or programming)
68-06	Proceedings, conferences, collections, etc.
68Mxx	Computer system organization
68Nxx	Software
68Pxx	Theory of data
68Rxx	Theory of computing
68Rxx	Discrete mathematics in relation to computer science
68S05	Mathematical linguistics
68Txx	Artificial intelligence
68Uxx	Computing methodologies

Klassifizierungsebene 3 (am Beispiel der Informatik / Systemorganisation):

Gruppe	Beschreibung
68M05	General
68M07	Mathematical problems of computer architecture
68M10	Computer networks
68M15	Reliability and testing
68M20	Performance evaluation; queueing; scheduling
68M99	None of the above but in this section

### 3.2.3.4. Sprachcodes nach NISO Z39.53 (Auszug, vgl. [43])

Code	Sprache
<i>ang</i>	English, Old
<i>ara</i>	Arabic
<i>cze</i>	Czech
<i>dan</i>	Danish
<i>dut</i>	Dutch
<i>eng</i>	English
<i>enm</i>	English, Middle (1100-1500)
<i>fin</i>	Finnish
<i>fre</i>	French
<i>frm</i>	French, Middle (ca. 1400-1600)
<i>fro</i>	French, Old (ca. 842-1400)
<i>ger</i>	German
<i>gmh</i>	German, Middle High (ca. 1050-1500)
<i>goh</i>	German, Old High (ca. 750-1050)

Code	Sprache
<i>ind</i>	Indonesian
<i>ira</i>	Iranian
<i>iri</i>	Irish
<i>ita</i>	Italian
<i>jpn</i>	Japanese
<i>lat</i>	Latin
<i>peo</i>	Old Persian (ca. 600-400 v.Chr.)
<i>per</i>	Persian
<i>por</i>	Portuguese
<i>rus</i>	Russian
<i>slo</i>	Slovak
<i>spa</i>	Spanish
<i>tur</i>	Turkish
<i>yap</i>	Yapese

### 3.2.4. Beispiel

Für die Speicherung der Metadaten existieren grundsätzlich zwei Möglichkeiten: Die Daten können in einer Datenbank oder in einer separaten Datei (beispielsweise im Header eines HTML-Dokuments) gespeichert werden. Der zur Zeit aktuelle HTML-Standard 4.0 schreibt keine konkreten Meta-Angaben mehr vor, sondern definiert lediglich den grundsätzlichen Aufbau einer Meta-Angabe [36]. Zur Standardisierung von Meta-Angaben arbeitet das W3-Konsortium stattdessen an einer Sprache namens *Resource Description Framework* (siehe nächster Paragraph). Die Meta-Angaben in der Form

```
<META name="metaname" content="Wert">
```

sind zur Zeit am verbreitetsten und werden von den gängigsten Suchmaschinen in der Regel erkannt. Als optionale Angabe kann zusätzlich der `<LINK>`-Tag eingetragen werden, der definiert, welche Interpretationsschemata verwendet wurden und der die URL zur Referenz-Dokumentation enthält:

```
<META name="DC.Title" content="Modellierung und Simulation dynamischer Systeme">
<META name="DC.Creator.PersonalName" content="Eiden, Wolfgang">
<META name="DC.Creator.PersonalName" content="Heidenreich, Markus">
<META name="DC.Creator.PersonalName.Address" content="(Email)eiden@ub.uni-kl.de">
<META name="DC.Creator.PersonalName.Address" content="(Email)mheiden@kit.uni-kl.de">
<META name="DC.Creator.PersonalName.Address" content="(Affiliation)Universit&auml;t
  Kaiserslautern">
<META name="DC.Subject" content="Dynasys">
<META name="DC.Subject" content="Modellbildung">
<META name="DC.Subject" content="Modellierung">
<META name="DC.Subject" content="dynamische Systeme">
<META name="DC.Subject" content="Simulation">
<META name="DC.Subject.Msc" content="(SCHEME=msc91) 00A72 General methods of simulation">
<META name="DC.Subject.Msc" content="(SCHEME=msc91) 00A35 Methodology of mathematics,
  didactics">
<META name="DC.Description.Abstract" content="
  Es wird eine Einf&uuml;hrung in Modellierungsprozesse, Wachstumsarten und
  Simulationsverfahren am Beispiel der Software Dynasys gegeben. Anhand konkreter
  Problemstellungen wird auf Probleme der verschiedenen Rechenverfahren im
  Mathematikunterricht und Chancen und Gefahren der Modellbildung hingewiesen.">
<META name="DC.Date.Created" content="1999-02-01">
<META name="DC.Date" content="1999-02-01">
<META name="DC.Type" content="script">
<META name="DC.Identifizier" content="(SCHEME=url) http://kluedo.ub.uni-kl.de/Mathematik/
  seminar_1.pdf">
<META name="DC.Format" content="(SCHEME=IMT) application/pdf">
<META name="DC.Identifizier" content="(SCHEME=url) http://kluedo.ub.uni-kl.de/Mathematik/
  seminar_1.ps.gz">
<META name="DC.Format" content="(SCHEME=IMT) application/postscript">
<META name="DC.Source" content="Seminarvortrag, Universit&auml;t Kaiserslautern, Fachbereich
  Mathematik">
<META name="DC.Language" content="(SCHEME=Z39.53) GER">
<LINK REL="SCHEMA.dc" HREF="http://purl.org/metadata/dublin_core_elements">
<LINK REL="SCHEMA.msc91" HREF="http://www.ams.org/msc/">
<LINK REL="SCHEMA.Z39.53" HREF="http://www.sil.org/sgml/nisoLang3-1994.html">
```

*Beispiel:* Meta-Angaben einer Publikation an der Universität Kaiserslautern



### 3.3. Resource Description Framework

Die Beschreibung von Ressourcen über `<Meta>`-Tags erlaubt leider nicht die Behandlung komplexerer Zusammenhänge in den Metadaten, da Relationen zwischen den Elementen nicht auf vernünftige Art und Weise ausgedrückt werden können. Das *Resource Description Framework* (RDF) soll diesen Umstand beheben [11]. Da das Datenmodell zunächst syntaxunabhängig ist, handelt es sich bei der, in den nachfolgenden Beispielen verwendeten, Syntax nur um eine exemplarische Applizierung dieses Modelles auf Basis der *Extensible Markup Language* (XML) [6]. Die Verwendung von XML als Sprache wird vom World Wide Web Consortium aufgrund der zu erwarteten hohen Akzeptanz und der Erweiterungsmöglichkeit (ohne dass Spezifikationsänderungen nötig werden) vorgeschlagen.

#### 3.3.1. Properties

Allgemein versteht das RDF unter Ressourcen Objekte, die Informationen enthalten. Die Ressourcen werden durch ihren *Resource Identifier* identifiziert. Sie haben benannte Eigenschaften mit Werten. Zur eindeutigen Beschreibung der Eigenschaft einer Ressource weist im RDF-Datenmodell ein Pfeil von der Ressource zum Wert der Eigenschaft:

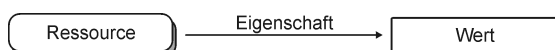


Abbildung 3.3.a : Modell einer einfachen Relation

#### Beispiel:

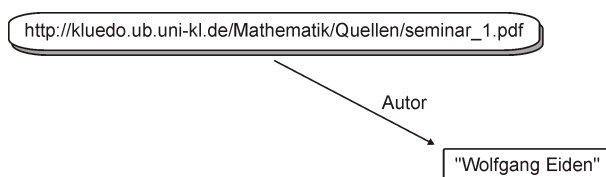


Abbildung 3.3.b : Beispiel einer einfachen Relation

```
<rdf:rdf>
  <rdf:Description about="http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.pdf">
    <s:Autor>"Wolfgang Eiden"</s:Autor>
  </rdf:Description>
</rdf:rdf>
```

Serialisationssyntax

Mehrere Eigenschaften (Properties) einer Ressource werden zu einer Description zusammengefaßt. Oftmals sind die Zusammenhänge von komplexerer Ordnung. So ist beispielsweise der Wert der Eigenschaft *Autor* im allgemeinen nicht nur eine Zeichenkette, sondern selbst wieder ein attribuiertes Objekt, das damit bezüglich dieser Eigenschaft die Rolle einer Ressource übernimmt. Also wird zusätzlich ein Identifier eingeführt, der die Ressource eindeutig beschreibt:

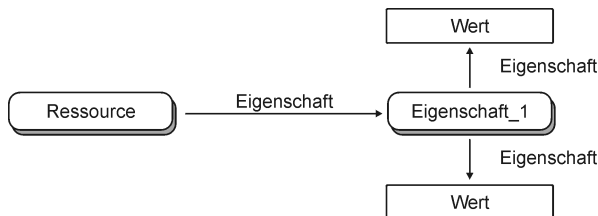


Abbildung 3.3.c : Modell einer erweiterten Relation

### Beispiel:

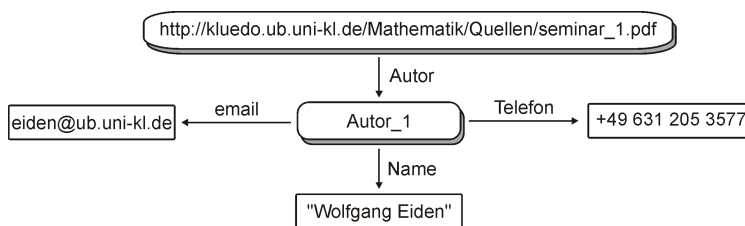


Abbildung 3.3.d : Beispiel einer erweiterten Relation

```
<rdf:rdf>
  <rdf:Description about="http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.pdf">
    <s:Autor rdf:resource="Autor_1"/>
  </rdf:Description>
  <rdf:Description about="Autor_1"/>
    <s:Name>"Wolfgang Eiden"</s:Name>
    <s:email>eiden@ub.uni-kl.de</s:email>
    <s:Telefon>+49 631 205 3577</s:Telefon>
  </rdf:Description>
</rdf:rdf>
```

*Serialisationssyntax*

Hier ist `Autor_1` ein Resource Identifier, der die Person des Autors eindeutig beschreibt.

### 3.3.2. Containers

Oftmals ist auch die Referenzierung einer Kollektion von Ressourcen notwendig. Dies ist beispielsweise dann der Fall wenn eine Arbeit von mehreren Autoren geschrieben wurde. Das Resource Description Framework definiert drei Typen von Container-Objekten um solche Zusammenhänge zu modellieren (vgl. [27]):

Container	Beschreibung
<i>Bag</i>	Eine ungeordnete Liste von Ressourcen oder Literalen. <i>Bags</i> werden zur Deklaration einer Eigenschaft mit mehreren Werten benutzt, wobei diese in keiner signifikanten Ordnung vorliegen. Duplikate sind erlaubt.
<i>Sequence</i>	Eine geordnete Liste von Ressourcen oder Literalen. <i>Sequences</i> werden zur Deklaration einer Eigenschaft mit mehreren Werten benutzt, wobei die Ordnung von Relevanz ist. Auch hier sind Duplikate erlaubt.
<i>Alternative</i>	Eine Liste von Ressourcen oder Literalen die Alternativen für einen (einzelnen) Wert darstellen.

Um eine Kollektion von Ressourcen zu repräsentieren wird mit der Eigenschaft *type* eine zusätzliche Ressource gemäß der obigen Liste zur Spezifikation verwendet. Das folgende Beispiel soll dies exemplarisch anhand des Bag-Containers verdeutlichen :

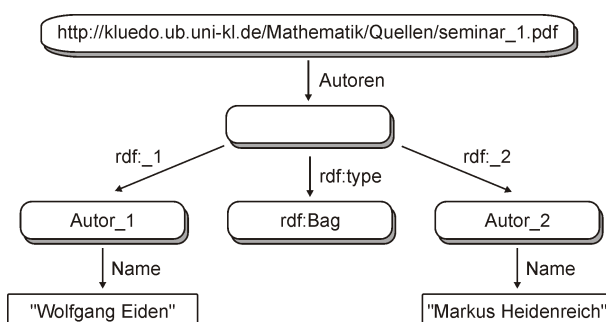


Abbildung 3.3.e : Beispiel eines einfachen Bag-Containers

```

<rdf:rdf>
  <rdf:Description about="http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.pdf">
    <s:Autoren>
      <rdf:Bag>
        <rdf:li ressource="Autor_1"/>
        <rdf:li ressource="Autor_2"/>
      </rdf:Bag>
    </s:Autoren>
    <rdf:Description about="Autor_1"/>
      <s:Name>Wolfgang Eiden</s:Name>
    </rdf:Description>
    <rdf:Description about="Autor_2"/>
      <s:Name>Markus Heidenreich</s:Name>
    </rdf:Description>
  </rdf:Description>
</rdf:rdf>
  
```

Serialisationssyntax

Das Resource Description Framework bietet aufgrund seines klassenorientierten Konzeptes noch viele andere Modellierungsmöglichkeiten. Die bisher vorgestellten Modellierungen sollten dem geneigten Leser nur einen ersten Eindruck von der Mächtigkeit des Resource Description Framework Modelles geben. Für weitere Details sei an dieser Stelle auf die Spezifikationen des W3-Consortiums verwiesen ([27],[7]).

### 3.3.3. Interpretationsschematas

Beim Resource Description Framework wird - im Gegensatz zur Metadatendefinition via <META>-Tags - die Angabe eines Interpretationsschematas erzwungen: Durch Qualifizierung des Elementnamens mit einem Namespace-Präfix [5] wird eine eindeutige Zuordnung des jeweiligen Elements mit dem korrespondierenden RDF-Schema erreicht (vgl. [27]). Im Beispiel

```
<rdf:rdf
  <rdf:Description about="http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.pdf">
    <s>Title>Modellierung und Simulation dynamischer Systeme</s>Title>
  </rdf:Description>
</rdf:rdf>
```

ist der Namespace-Präfix "s" eine Referenz auf den verwendeten spezifischen Namespace-Präfix, der in einer XML Namespace-Deklaration [6] wie beispielsweise

```
xmlns: s="http://description.org/schema/"
```

zu definieren ist. Bei Verwendung des Dublin Core Metadata Sets kann eine Ressourcenbeschreibung beispielsweise folgende Gestalt haben:

```
<rdf:rdf
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/metadata/dublin_core#"
  xmlns:dcq="http://purl.org/metadata/dublin_core_qualifiers#"
  <rdf:Description about="http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.pdf">
    <dc>Title>Modellierung und Simulation dynamischer Systeme</dc>Title>
    <dc:Description rdf:parseType="Ressource">
      <dcq:Abstract>
        Es wird eine Einf&uuml;hrung in Modellierungsprozesse, Wachstumsarten und
        Simulationsverfahren am Beispiel der Software Dynasys gegeben.
      </dcq:Abstract>
    </dc:Description>
    <dc:Identifizier dc:Scheme="URL">
      http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.pdf
    </dc:Identifizier>
    <dc:Subject>
      <rdf:Bag>
        <rdf:li>Dynasys</rdf:li>
        <rdf:li>Modellierung</rdf:li>
        <rdf:li>dynamische Systeme</rdf:li>
      </rdf:Bag>
    </dc:Subject>
  </rdf:Description>
</rdf:rdf>
```

### 3.3.4. Abbreviated Syntax

Die Einbettung der RDF-Beschreibung in HTML-Code liefert bei Verwendung der vorgestellten Serialisationssyntax in älteren Browsern leider unerwünschte Ausgaben: Die neuen und somit unbekanntenen Tags werden zwar (erwünschterweise) nicht dargestellt, jedoch aber ihre jeweiligen Inhalte. Beispielsweise wird bei der Einbettung

```
<html>
<head>
  <rdf:rdf
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/metadata/dublin_core#"
    xmlns:dcq="http://purl.org/metadata/dublin_core_qualifiers#"
    <rdf:Description about="http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.pdf">
      <dc:Title>Modellierung und Simulation dynamischer Systeme</dc:Title>
      <dc:Identifizier dc:Scheme="URL">
        http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.pdf
      </dc:Identifizier>
      <dc:Creator>
        <dcq:Personalname>
          <rdf:Bag>
            <rdf:li ressource="_1"/>
            <rdf:li ressource="_2"/>
          </rdf:Bag>
        </dcq:Personalname>
      </dc:Creator>
    </rdf:Description>
    <rdf:Description about="_1"/>
      <rdf:Name>Wolfgang Eiden</rdf:Name>
    </rdf:Description>
    <rdf:Description about="_2"/>
      <rdf:Name>Markus Heidenreich</rdf:Name>
    </rdf:Description>
  </rdf:rdf>
</head>
<body>
  <h1>
    Modellierung und Simulation dynamischer Systeme
  </h1>
  <a href="mailto:eiden@ub.uni-kl.de">
    Wolfgang Eiden
  </a>
  <br>
  <a href="mailto:mheiden@kit.uni-kl.de">
    Markus Heidenreich
  </a>
  <br><br>
  Seminarvortrag, Fachbereich Mathematik, Universit&auml;t Kaiserslautern
  <br><br>
  <i><b>Datum der Ver&ouml;ffentlichung:</b></i> 1999-02-01
  <br><br>
  <i><b>Verf&uuml;gbare Formate:</b></i><br>
  <a href="http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.pdf">
    application/pdf
  </a>
</body>
</html>
```

von älteren Browsern folgende Ausgabe erzeugt:

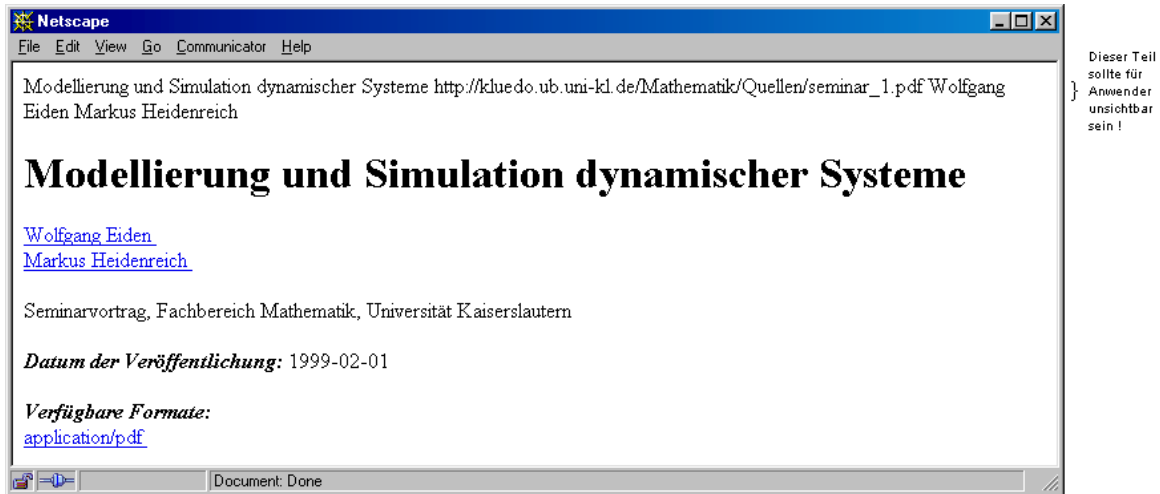


Abbildung 3.3.f : RDF-Einbettung (Serialisationssyntax) in HTML-Code (ältere Browser)

Um dies zu vermeiden schlägt das W3-Consortium parallel zu dieser Syntax die Verwendung der sogenannten *Abbreviated Syntax* vor (vgl. [40]). Das obige Beispiel könnte danach folgende Form annehmen :

```
<html><head>
  <rdf:rdf
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/metadata/dublin_core#"
    xmlns:dcq="http://purl.org/metadata/dublin_core_qualifiers#"
    <rdf:Description about="http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.pdf">
      <dc>Title=Modellierung und Simulation dynamischer Systeme />
      <dc:Identifier="http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.pdf"
        dc:Scheme="URL" />
      <dc:Creator><dcq:Personalname>
        <rdf:Bag><rdf:li ressource="_1"/> <rdf:li ressource="_2"/></rdf:Bag>
      </dcq:Personalname></dc:Creator>
    </rdf:Description>
    <rdf:Description about="_1"/><rdf:Name="Wolfgang Eiden" /></rdf:Description>
    <rdf:Description about="_2"/><rdf:Name="Markus Heidenreich" /></rdf:Description>
  </rdf:rdf>
</head><body>
  <h1>
    Modellierung und Simulation dynamischer Systeme
  </h1>
  <a href="mailto:eiden@ub.uni-kl.de">Wolfgang Eiden</a><br>
  <a href="mailto:mheiden@kit.uni-kl.de">Markus Heidenreich</a><br><br>
  Seminarvortrag, Fachbereich Mathematik, Universit&auml;t Kaiserslautern<br><br>
  <i><b>Datum der Ver&ouml;ffentlichung:</b></i> 1999-02-01
  <br><br>
  <i><b>Verf&uuml;gbare Formate:</b></i><br>
  <a href="http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.pdf">
    application/pdf
  </a>
</body></html>
```

Bei Verwendung der Abbreviated Syntax werden auch in älteren Browserversionen (Versionen vor *Netscape Communicator 4.5* und *Microsoft Internet Explorer 5.0*) keine unerwünschten Teile mehr dargestellt :

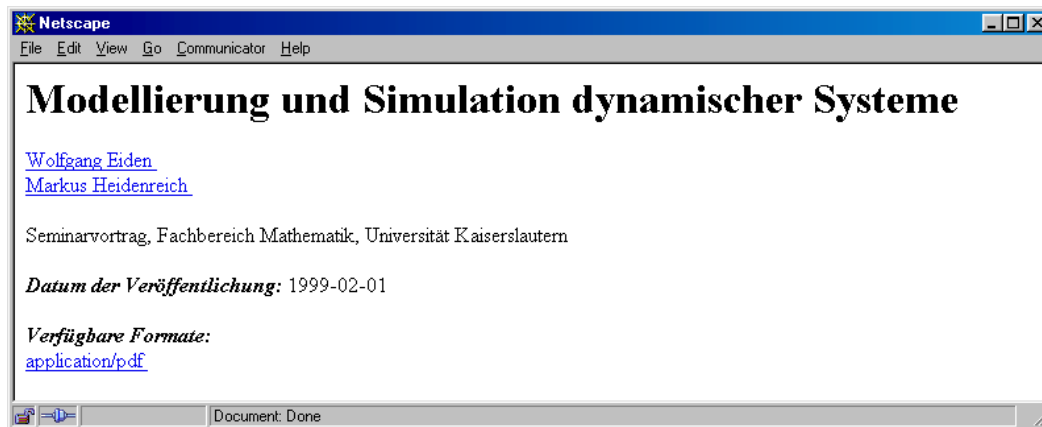


Abbildung 3.3.g : RDF-Einbettung (Abbreviated Syntax) in HTML-Code

### 3.4. Resümee

Das Resource Description Framework erfüllt alle Anforderungen, die zur Zeit an eine qualifizierte Ressourcenbeschreibung gestellt werden. Durch zusätzliche Verwendung des unterstützten Dublin Core Metadata Sets könnten die jeweiligen Vorteile dieser Systeme sogar vereinigt und somit ein optimierteres Ergebnis erreicht werden. Da dieser zukünftige Standard allerdings noch in Entwicklung und Gegenstand einiger aktueller Fachinformationsprojekte (wie beispielsweise CARMEN [9]) ist, scheint mir eine Implementierung zum gegenwärtigen Zeitpunkt noch etwas verfrüht.

## 4. Das Resource-Discovery-System Harvest

Die Beschreibung der einzelnen Ressourcen via Metadaten ist eine elementare Voraussetzung für den Erfolg eines Publikationsservers. Mindestens genauso wichtig ist Bereitstellung effektiver Retrieval-Mechanismen zur Lösung des Verteilerproblems: Die Informationsangebote eines Publikationsservers sollten dabei nicht nur lokal, sondern auch weltweit recherchierbar sein. Dazu sind serverübergreifende Indexierungswerkzeuge, wie das von der IRTF-RD (Internet Research Task Force Research Group on Resource Discovery) entwickelte Resource-Discovery-System Harvest [20], nützlich.

### 4.1. Systemübersicht

Harvest (eng. "Ernte") ist eine Kollektion von Dienstprogrammen zum Sammeln, Extrahieren, Organisieren, Suchen, Cachen und Replizieren von (im Internet verteilten) relevanten Informationen. Ziel von Harvest ist die Bereitstellung eines flexiblen Systems, welches in vielfältiger Art und Weise konfigurierbar ist und eine Adaption an spezielle Bedürfnisse erlaubt (vgl. [20]). Daher , besteht Harvest, wie in *Abbildung 4.1.a* illustriert, aus mehreren Subsystemen:

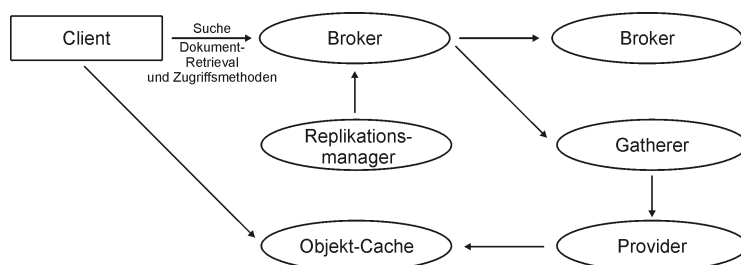


Abbildung 4.1.a : Softwarekomponenten von Harvest

Das *Gatherer*-Subsystem (eng. "Sammler") sammelt Indexierungsinformationen (wie beispielsweise Titel, Autorennamen oder Schlüsselwörter) von den, durch Provider zur Verfügung gestellten, Ressourcen. Das *Broker*-Subsystem (eng. "Makler") erhält Indexierungsinformationen von einem oder mehreren Gatherern (oder Brokern), indexiert die gesammelten Informationen und stellt eine Abfrageschnittstelle zur Verfügung. Das *Replicator*-Subsystem (eng. "Kopierer") dient zur effektiven Replikation von Brokern und somit (durch lokale Datenerhaltung) zu einer günstigeren Lastverteilung. Der *Cache* fungiert als Zwischenspeicher und sorgt für eine effizientere Nutzung von gesammelten Informationen.



## 4.2. Gatherer

Der Gatherer erhält die Informationsressourcen unter Verwendung verschiedener Standardzugriffsmethoden (wie beispielsweise FTP, HTTP und Zugriff auf das lokale Filesystem) und faßt die Informationen zur Generierung von strukturierten Indexierungs-informationen unter Verwendung typspezifischer Methoden zusammen.

### 4.2.1. Informationsbeschaffung

Um Informationen von Providern zu beschaffen, benötigt der Gatherer eine Konfigurationsdatei, in der eine Auflistung der zu besuchenden *Uniform Resource Locators* (URLs) mit der jeweiligen Zugriffsmethode angegeben ist. Wie aus der Beispielskonfigurationsdatei

```
#####
#
# kluedo.cf
# configuration file for the KLUEDO Gatherer
#
# Created by Wolfgang Eiden on Mon Sep 27 19:34:53 MEST 1999
#
#####

Local-Mapping: http://kluedo.ub.uni-kl.de/ /opt/metadata/KLUEDO/Fachbereiche/

<RootNodes>
http://kluedo.ub.uni-kl.de/Mathematik/Metadaten/
http://kluedo.ub.uni-kl.de/Mathematik/Quellen/
</RootNodes>

<LeafNodes>
http://www.uni-kl.de/Bibliothek/eiden/
</LeafNodes>
```

ersichtlich wird, werden die URLs in *RootNode* und *LeafNode* unterteilt. Bei einer *LeafNode* fordert der Gatherer nur die unter dieser URL verfügbare Ressource an und verarbeitet diese. Im Gegensatz dazu werden die *RootNodes* vom Gatherer rekursiv in (für die jeweilige Zugriffsmethode spezifische Art) *LeafNodes* expandiert. So werden beispielsweise FTP-Server durch die erhaltene Verzeichnisliste und HTTP-Server durch Verfolgung der (in die HTML-Seiten eingebetteten) Links rekursiv abgesucht. Hierbei bestehen für die Dokumenterfassung vielfältige Filterungsmöglichkeiten wie beispielsweise nach Zugriffsmethoden, bestimmten Hosts, maximaler Verzeichnistiefe und maximale Anzahl der Dateien pro Host. Werden Server und Gatherer auf ein und demselben Computersystem betrieben, dann kann der Gatherer durch Local-Mapping die Informationen direkt aus dem Dateisystem lesen, anstatt über das Netzwerk auf seine eigenen Ressourcen zugreifen zu müssen.

Dies geschieht beispielsweise durch Deklarationen der Form

```
Local-Mapping: http://kluedo.ub.uni-kl.de/ /opt/metadata/KLUEDO/Fachbereiche/
Local-Mapping: http://www.example.edu/~*/*/ /public/*/pages/
```

innerhalb der Konfigurationsdatei. Auf Basis der eingesammelten Ressourcen werden nun strukturierte Indexinformationen generiert. Dies erfordert jedoch verständlicherweise typspezifische Informationsextrahierungsmethoden.

#### 4.2.2. Typspezifische Informationsextrahierung mit *Essence*

In Umgebungen, die viele unterschiedliche Typen von Daten enthalten, erfordert die Inhaltsindexierung typspezifische Methoden zur effektiven Informationsextrahierung. Für diesen Vorgang verwendet der Gatherer das Teilsystem *Essence* [22] (eng. "inneres Wesen", "Kern"). Dieses Teilsystem besitzt die Fähigkeit Dateitypen zu erkennen und geschachtelte und darstellungsabhängige Formate (wie beispielsweise komprimierte Dateien) durch Auflösung zu elementarisieren (vgl. [21]). Dabei ist eine Festlegung der zu registrierenden Dateien möglich:

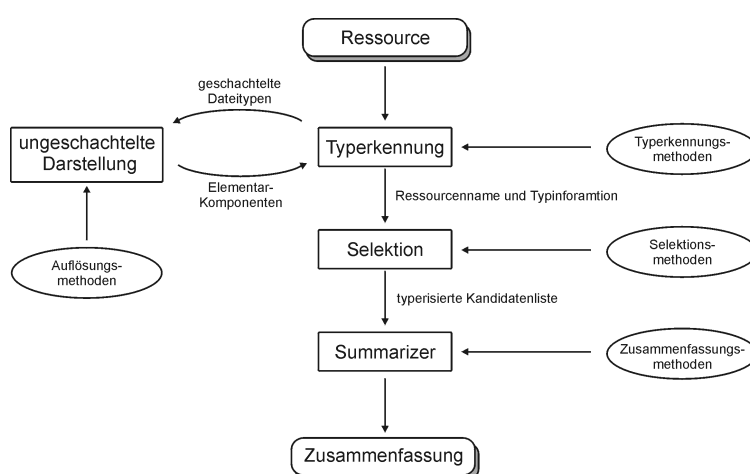


Abbildung 4.2.a : Schritte zur Informationsextrahierung

Im Anschluß wird zur Informationsextrahierung eine für den jeweiligen Dateityp geeignete Methode angewendet, wobei beispielsweise folgende Dateitypen unterstützt werden (Auszug, vgl. [20]):

Dateityp	Informationsextrahierung
<i>Audio</i>	Extrahiert Dateinamen.
<i>Binär-Dateien</i>	Extrahiert lesbare Zeichenkette und Beschreibungen.
<i>C, CHeader</i>	Extrahiert Prozedurnamen, Namen der verwendete Include-Dateien und Kommentare.

Dateityp	Informationsextrahierung
<i>Dvi</i>	Wendet den Textextrahierer für den extrahierten ASCII-Text an.
<i>FAQ, Volltext, README</i>	Extrahiert alle Wörter.
<i>Framemaker</i>	Konvertierung nach SGML und Anwendung des SGML-Extrahierers.
<i>Schrift</i>	Extrahiert Schriftnamen.
<i>HTML</i>	Extrahiert Links und die in einer Konfigurationstabelle angegebenen Felder (siehe später)
<i>LaTex</i>	Extrahiert verwendete LaTex-Felder (Autor, Titel usw.)
<i>Perl</i>	Extrahiert Prozedurnamen und Kommentare.
<i>Postscript</i>	Wendet den Textextrahierer für den extrahierten ASCII-Text an.
<i>RTF</i>	Konvertierung nach SGML und Anwendung des SGML-Extrahierers.
<i>SGML</i>	Extrahiert die in einer Konfigurationstabelle angegebenen Felder
<i>Tex</i>	Wendet den Textextrahierer für den extrahierten ASCII-Text an.
<i>Text</i>	extrahiert die ersten 100 Zeilen und den ersten Satz eines jeden Paragraphen.

Die Erkennung von weiteren Dateitypen und die Manipulation der zu extrahierenden Informationen ist aufgrund der Konfigurationsmöglichkeiten kein Problem. Dabei kann die Festlegung des Dateityps anhand der Dateierdung oder durch Erkennung von markanten Merkmalen innerhalb der Datei (via *Pattern-Matching* und *regular expressions*) erfolgen (Näheres hierzu entnehme man bitte der Abhandlung "*Essence: A Ressource Discovery System Based om Semantic File Indexing*" [22]). Der Summarizer speichert die extrahierten Informationen im Harvest eigenen *Summary Object Interchange Format* (SOIF), wobei die Standarddistribution folgende Attribute vorsieht (vgl. [20]):

Attribut	Beschreibung
<i>Abstract</i>	Brief abstract about the object.
<i>Author</i>	Author(s) of the object.
<i>Description</i>	Brief description about the object.
<i>FileSize</i>	Number of bytes in the object.
<i>FullText</i>	Entire contents of the object.
<i>GathererHost</i>	Host on which the Gatherer ran to extract information from the object.
<i>GathererName</i>	Name of the Gatherer that extracted information from the object. (e.g., FullText, SelectedText, or Terse).
<i>GathererPort</i>	Port number on the GathererHost that serves the Gatherer's information.
<i>GathererVersion</i>	Version number of the Gatherer.
<i>Keywords</i>	Searchable keywords extracted from the object.
<i>LastModificationTime</i>	The time that the object was last modified.
<i>MD5</i>	MD5 16byte checksum of the object.
<i>RefreshRate</i>	The number of seconds after UpdateTime when the summary object is to be regenerated. Defaults to 1 month.
<i>TimeToLive</i>	The number of seconds after UpdateTime when the summary object is no longer valid. Defaults to 6 months.
<i>Title</i>	Title of the object.
<i>Type</i>	The object's type (e.g. HTML)
<i>UpdateTime</i>	The time that the summary object was last updated. REQUIRED field, no default.
<i>URLReferences</i>	Any URL references present within HTML objects.

Beispielsweise verwendet der (auf dem SGML-Parser basierende) HTML-Extraktor standardmäßig folgende Übersetzungsmatrix:

HTML-Element	SOIF-Attribute
<A>	keywords, parent
<A:HREF>	url-references
<ADDRESS>	address
<B>	keywords, parent
<BODY>	body
<CITE>	references
<CODE>	ignore
<EM>	keywords, parent
<H1>	headings
<H2>	headings
<H3>	headings

HTML-Element	SOIF-Attribute
<H4>	headings
<H5>	headings
<H6>	headings
<HEAD>	head
<I>	keywords, parent
<IMG:SRC>	images
<IMG:ALT>	parent
<META:CONTENT>	\$NAME
<STRONG>	keywords, parent
<TITLE>	title
<TT>	keywords, parent

#### 4.2.2.1. Beispiel

Der HTML-Extraktor des Dokumentenservers der Universität Kaiserslautern erzeugt aus der HTML-Datei

```
<html>
  <head>
    <title>
      KLUEDO - Metadaten f&uuml;r: Modellierung und Simulation dynamische Systeme
    </title>
    <link rel=stylesheet type="text/css" href="/php/kluedo.css">
  </head>
  <body>
    <META name="DC.Title" content="Modellierung und Simulation dynamischer Systeme">
    <META name="DC.Creator.PersonalName" content="Eiden, Wolfgang">
    <META name="DC.Creator.PersonalName" content="Heidenreich, Markus">
    <META name="DC.Creator.PersonalName.Address" content="(Email)eiden@ub.uni-kl.de">
    <META name="DC.Creator.PersonalName.Address" content="(Email)heidenre@kit.uni-kl.de">
    <META name="DC.Creator.PersonalName.Address" content="(Affiliation)Universit&auml;t
      Kaiserslautern">
    <META name="DC.Subject" content="Dynasys">
    <META name="DC.Subject" content="Modellbildung">
    <META name="DC.Subject" content="Modellierung">
    <META name="DC.Subject" content="dynamische Systeme">
    <META name="DC.Subject" content="Simulation">
    <META name="DC.Subject.Topic" content="Mathematik">
    <META name="DC.Subject.Msc" content="(SCHEME=msc91) 00A72 General methods of
      simulation">
    <META name="DC.Subject.Msc" content="(SCHEME=msc91) 00A35 Methodology of mathematics,
      didactics">
    <META name="DC.Description.Abstract" content="(SCHEME=url) #abstract">
    <META name="DC.Description.Abstract" content="Es wird eine Einf&uuml;hrung in
      Modellierungsprozesse, Wachstumsarten, Simulationsverfahren am Beispiel der Software
      Dynasys gegeben. Anhand konkreter Problemstellungen wird auf Probleme der
      verschiedenen Rechenverfahren im Mathematikunterricht und Chancen und Gefahren der
      Modellbildung hingewiesen.">
    <META name="DC.Description.Notes" content="Verwendete &lt;a href=&quot;
      http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.example.zip&quot;&gt;
```

```

Beispieldateien</a>>">
<META name="DC.Date.Created" content="1999-02-01">
<META name="DC.Date" content="1999-02-01">
<META name="DC.Type" content="script">
<META name="DC.Identifier" content="(SCHEME=url) http://kluedo.ub.uni-kl.de/Mathematik
/Quellen/seminar_1.ps.gz">
<META name="DC.Format" content="(SCHEME=IMT) application/postscript">
<META name="DC.Identifier" content="(SCHEME=url) http://kluedo.ub.uni-kl.de/Mathematik
/Quellen/seminar_1.pdf">
<META name="DC.Format" content="(SCHEME=IMT) application/pdf">
<META name="DC.Source" content="Seminarvortrag, Fachbereich Mathematik,
Universit&auml;t Kaiserslautern">
<META name="DC.Language" content="(SCHEME=Z39.53) GER">
<LINK REL="SCHEMA.dc" HREF="http://purl.org/metadata/dublin_core_elements">
<LINK REL="SCHEMA.msc91" HREF="http://www.ams.org/msc/">
<LINK REL="SCHEMA.Z39.53" HREF="http://www.sil.org/sgml/nisoLang3-1994.html">
</body>
</html>

```

das folgende Summary:

```

DC.Description.Notes{129}: Verwendete &lt;a href=&quot;http://kluedo.ub.uni-
kl.de/Mathematik/Quellen/seminar_1.example.zip&quot;&gt;Beispieldateien&lt;/a>&gt;
DC.Subject.Topic{10}: Mathematik
DC.Date{10}: 1999-02-01
DC.Subject{76}: Dynasys
| Modellbildung
| Modellierung
| dynamische Systeme
| Simulation
DC.Source{69}: Seminarvortrag, Fachbereich Mathematik, Universitaet Kaiserslautern
DC.Subject.Msc{112}: (SCHEME=msc91) 00A72 General methods of simulation
| (SCHEME=msc91) 00A35 Methodology of mathematics, didactics
DC.Type{6}: script
DC.Title{47}: Modellierung und Simulation dynamischer Systeme
DC.Identifier{148}: (SCHEME=url) http://kluedo.ub.uni-
kl.de/Mathematik/Quellen/seminar_1.ps.gz
| (SCHEME=url) http://kluedo.ub.uni-kl.de/Mathematik/Quellen/seminar_1.pdf
DC.Creator.PersonalName{38}: Eiden, Wolfgang
| Heidenreich, Markus
DC.Language{19}: (SCHEME=Z39.53) GER
DC.Date.Created{10}: 1999-02-01
DC.Description.Abstract{322}: (SCHEME=url) #abstract
| Es wird eine Einfuehrung in Modellierungsprozesse, Wachstumsarten, Simulationsverfahren
am Beispiel der Software Dynasys gegeben. Anhand konkreter Problemstellungen wird auf
Probleme der verschiedenen Rechenverfahren im Mathematikunterricht und Chancen und Gefahren
der Modellbildung hingewiesen.
DC.Creator.PersonalName.Address{102}: (Email)eiden@ub.uni-kl.de
| (Email)heidenre@kit.uni-kl.de
| (Affiliation)Universitaet Kaiserslautern
DC.Format{67}: (SCHEME=IMT) application/postscript
| (SCHEME=IMT) application/pdf

```

## 4.3. Broker

Das Broker-Subsystem besteht, wie in *Abbildung 4.3.a* veranschaulicht, aus mehreren Komponenten:

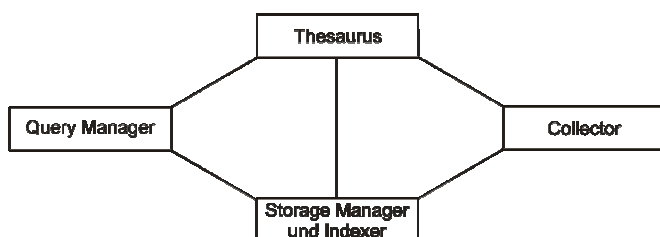


Abbildung 4.3.a : Interface des Broker-Subsystems

Durch das *Collector*-Interface erhält der Broker von anderen Gatherern (oder Brokern) die benötigten Informationen. Diese werden durch den *Storage Manager* (mit integrierter Garbage-Collection) gespeichert und durch den *Indexer* indexiert. Durch das flexible Interface werden eine Vielzahl von Indexierungssystemen, wie beispielsweise *Glimpse* [30], *Wais* [25] und *Nebula* [4] unterstützt. Der *Query Manager* wandelt die ihm übergebenen Abfragen in die interne Repräsentation des verwendeten Indexers um, führt dann die Abfrage durch und formatiert anschließend das Abfrageergebnis.

### 4.3.1. Queries

Standardmäßig wird vom Broker als Indexierungssystem *Glimpse* verwendet. Diese System unterstützt

- case-intensive Suche und case-sensitive Suche.
- die Suche nach Teilworten, ganzen Wörtern oder Phrasen.
- Ähnlichkeitssuche.
- logische Verknüpfungsoperatoren.
- strukturierte Abfragen.
- reguläre Ausdrücke (in eingeschränkter Form).

#### Beispielsabfragen:

Abfrage	Ergebnis
Modellierung	alle Objekte, die das Wort <i>Modellierung</i> enthalten.
Modellierung AND Simulation	alle Objekte, die irgendwo beide Worte enthalten.
"Simulation dynamischer Systeme"	alle Objekte, die <i>Simulation dynamischer Systeme</i> als Phrase enthalten.

Abfrage	Ergebnis
"Simulation dynamischer Systeme" AND Modellierung	alle Objekte, die <i>Simulation dynamischer Systeme</i> als Phrase und das Wort <i>Modellierung</i> enthalten.
Title : Modellierung	alle Objekte, die im Title-Attribut das Wort <i>Modellierung</i> enthalten.
"Simulation dynamischer Systeme" AND (Author : .*Eiden.*)	alle Objekte, die <i>Simulation dynamischer Systeme</i> als Phrase enthalten und bei dem das Author-Attribut desselben Objekts das Teilwort <i>Eiden</i> enthält.

### 4.3.2. Collector

Der Broker erhält durch den Collector Indexierungsinformationen. Hierbei wird in einer Konfigurationsdatei festgelegt, woher und auf welche Art der Broker seine Informationen (im SOIF-Format) erhält:

```
#####
#
# Collection.conf
# Collection points for the KLUEDO Example Broker
#
# Created by Wolfgang Eiden on Mon Sep 27 21:34:53 MEST 1999
#
#####

kluedo.ub.uni-kl.de      8512 2 --
gatherer-host1.exp.com  8500 3 --
gatherer-host2.exp.com  8500 0 --
broker-host1.exp.com    8500 6 --QUERY (URL : Mathematik) AND Modellierung
broker-host2.exp.com    8500 6 --QUERY Harvest --FLAGS #index case sensitive
```

Die Syntax hierfür lautet:

```
host port type query
```

Zur Zeit werden folgende Einsammlungsarten unterstützt:

Art-Nr.	Remote-Prozess	Beschreibung	Komprimiert
0	Gatherer	jederzeit vollständige Einsammlung	Nein
1	Gatherer	inkrementelle Einsammlung	Nein
2	Gatherer	jederzeit vollständige Einsammlung	Ja
3	Gatherer	inkrementelle Einsammlung	Ja
4	Broker	jederzeit vollständige Einsammlung	Nein
5	Broker	inkrementelle Einsammlung	Nein
6	Broker	vollständige Einsammlung basierend auf einer Abfrage	Nein
7	Broker	inkrementelle Einsammlung basierend auf einer Abfrage	Nein

### 4.3.3. Verteilbarkeit des Gathering- und Brokering-Prozesses

Die Gatherer und Broker von Harvest können auf unterschiedliche Art konfiguriert werden und müssen nicht notwendigerweise auf dem gleichen Computersystem betrieben werden. Durch Unterstützung des Remote-Gatherings und der Möglichkeit für Broker auch von anderen Brokern Informationen zu erhalten, ist die Realisation eines verteilten Informationssystems denkbar einfach (vgl. [20]) :

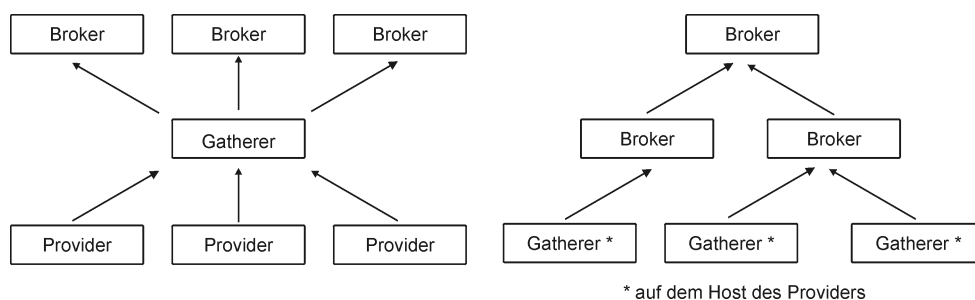


Abbildung 4.3.b : Verteilbarkeit des Gathering- und Brokering-Prozesses.

Auch wenn dies in manchen Fällen aus pragmatischen Gründen gewünscht ist, so bedeutet das Remote-Gathering je nach Systemarchitektur eine erhöhte und gegebenenfalls exzessive Server- und Netzbelastung (vgl. *Abbildung 4.3.b*, links). In diesem Fall ist jedoch bei Assimilation der Systemarchitektur eine Reduktion der Netzlast möglich (vgl. *Abbildung 4.3.b*, rechts). Ungeachtet der Nachteile des Remote-Gatherings ist diese Form jedoch in jedem Fall der Nichtindexierung vorzuziehen.

## 4.4. Replicator

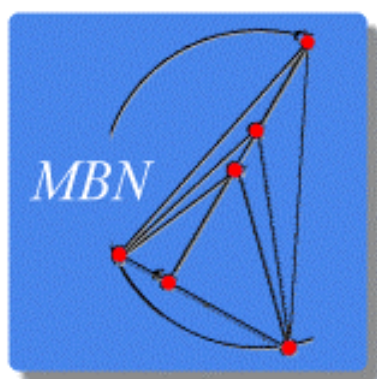
Der Replicator verteilt Kopien der Broker-Datenbank innerhalb der zugehörigen Replica-Verbünde. Hierbei fungiert der Replicator als Master, der (auf Basis des FTP-Mirror-Systems) jede Datenänderung an die anderen Replicas weiterleitet.

Für die Benutzer besteht somit (innerhalb des Replica-Verbunds) die Möglichkeit der Wahl eines (gemäß ihrer jeweiligen Netzinfrastruktur) geeigneten Brokers. Ferner ist durch Nutzung von Replicatoren auch eine Verteilung der Belastung eines Brokers möglich (vgl. *Paragraph 4.3.3*). Da eine Datenkorrektur nur bei Änderungen notwendig ist, hält sich sowohl die Server- als auch die Netzbelastung in einem vertretbaren Rahmen (vgl. [12]).



## 5. Der Dokumentenserver KLUEDO

In diesem Kapitel stelle ich den Dokumentenserver KLUEDO [26] vor und zeige (angefangen von der Benutzerschnittstelle bis hin zur Implementierungsebene) die Applizierung der vorgestellten Konzepte. Der Dokumentenserver entstand auf Initiative der Fachbereiche und der Universitätsbibliothek Kaiserslautern im Rahmen des vom *Bundesministerium für Bildung und Forschung* (BMBF) [8] geförderten *GLOBAL INFO* - Vorprojektes [18]. Hierbei basiert der Dokumentenserver stellenweise auf dem (während des *MathNet*-Projektes [35] des *deutschen Forschungsnetzes* (DFN) [14] und des *MathBibNet*-Projektes [32] der *deutschen Forschungsgemeinschaft* (DFG) [13] aufgebauten) Mathematik-Preprintserver der Universität Kaiserslautern.



MathBibNet:  
*Aufbau nutzerorientierter verteilter Informationsstrukturen in Kooperation von Bibliotheken und Fachbereichen am Beispiel des Faches Mathematik*



MathNet:  
*Informationsdienste für die Mathematik im Internet*



Global-Info:  
*Globale Elektronische und Multimediale Informationssysteme für Naturwissenschaft und Technik*

Auf Basis des (vom erwähnten Mathematikservers verwendeten) Harvestsystemes wurde von unserem Team (d.h. von Herrn Felix Ide, Herrn Thomas Esselen und mir) unter Leitung von Frau Gisela Schmitt der Dokumentenserver KLUEDO entwickelt, der dann den Preprintserver ablöste. Dazu habe ich (aufbauend auf der Analyse der Quelltexte des Preprintservers und der aktuellen Bedürfnisse) in Zusammenarbeit mit Herrn Ide die Systemarchitektur aus *Abbildung 5.0.a* entwickelt. Bei der Implementierung war Herr Ide hauptverantwortlich für die Anpassung von Harvest und die Bereitstellung der Umgebung (Webserver mit PHP-Einbindung, zentrale Navigation und Design). Herr Thomas Esselen kümmerte sich hauptsächlich um die Übernahme der alten Datenbestände und die Beseitigung von Duplikaten. Mir oblag die Adaption des Dublin Core Metadata Sets und die Programmierung des Metadatenmanagementsystemes (MetaMaker, Verwaltungsskripte und Bibliotheken). Im folgenden wird zunächst der Dokumentenserver aus Sicht des Benutzers beschrieben. Anschließend werden sowohl die Systemumgebung als auch die Systemarchitektur in den wesentlichen Punkten genauer vorgestellt und Erweiterungsmöglichkeiten aufgezeigt.

# Systemarchitektur des Dokumentenservers KLUEDO

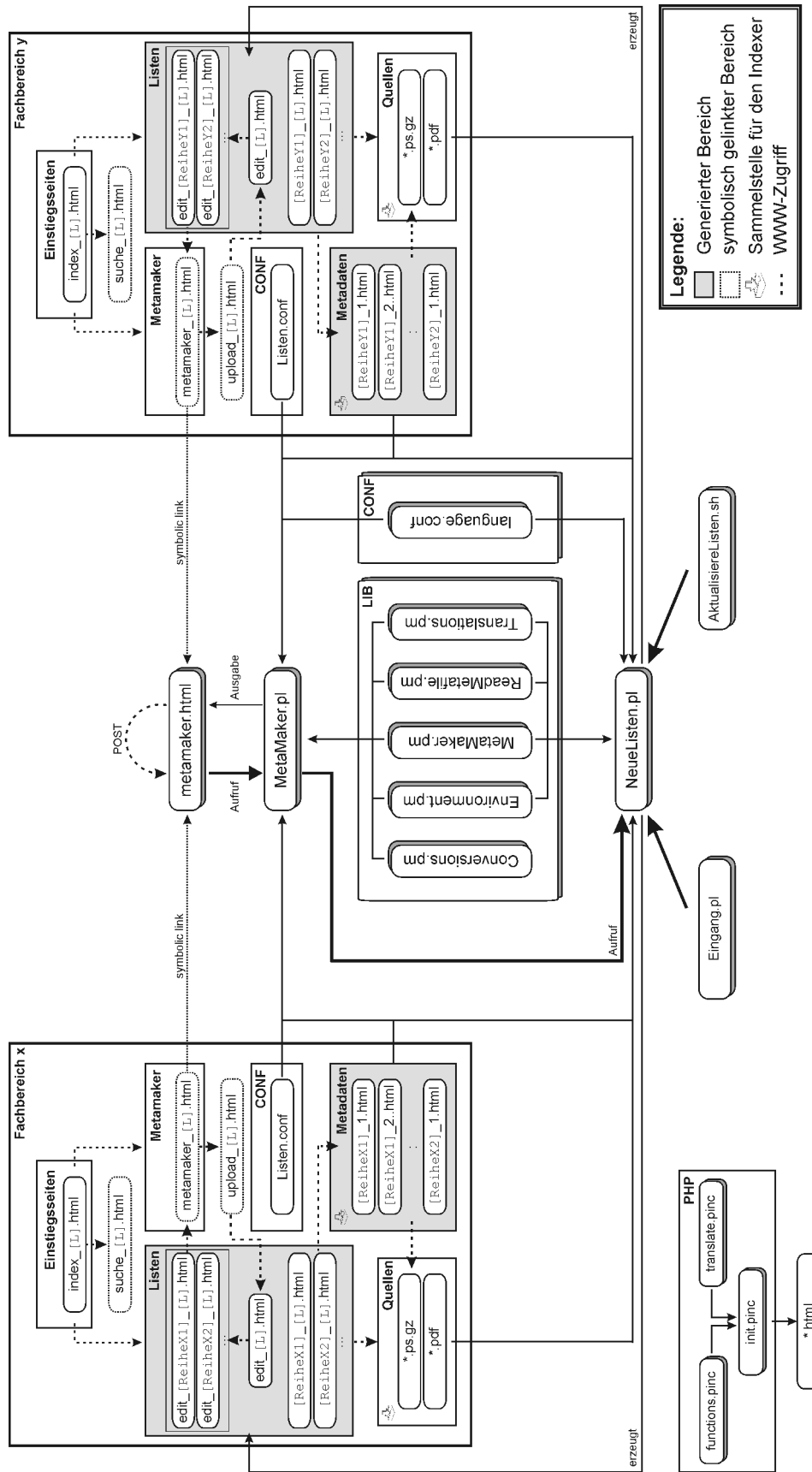


Abbildung 5.0.a : Systemarchitektur

## 5.1. Der Dokumentenserver aus Sicht des Benutzers

Der Dokumentenserver KLUEDO ist unter der Internetadresse <http://kluedo.ub.uni-kl.de> erreichbar. Nach dem Aufruf dieser Seite gelangt der Benutzer zur zentralen Startseite (Abbildung 5.1.a).

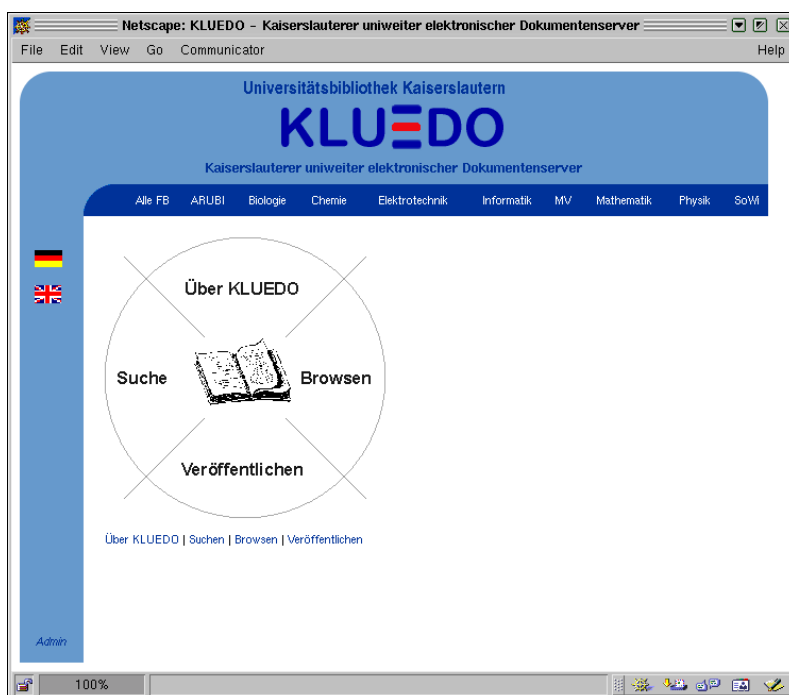


Abbildung 5.1.a : Startseite

Von diesem Punkt aus ist es dem Benutzer möglich sich über KLUEDO und das Veröffentlichen in KLUEDO zu informieren. Desweiteren sind Links zur fachbereichsübergreifenden Suche und eine Liste der eingebunden Fachbereiche vorhanden. Die Liste der Fachbereiche findet sich ebenfalls in der global zur Verfügung stehenden Navigationsleiste. In dieser Leiste kann der Benutzer desweiteren (durch Anklicken einer der abgebildeten Flaggen) die Sprache, in der sich KLUEDO präsentiert, wählen. Die abgebildete Startseite erreicht der Benutzer jederzeit durch Anklicken des KLUEDO-Logos.

### 5.1.1. Fachbereichsspezifische Seiten

Wählt der Benutzer einen Fachbereich aus (entweder direkt über die Navigationsleiste oder über den Punkt "Browsen"), so gelangt dieser zur fachbereichsspezifischen Indexseite (Abbildung 5.1.b).

Der Benutzer erhält hier die Möglichkeit der fachbereichsspezifische Suche und des Browsens in den jeweiligen Listen der verschiedenen Veröffentlichungsreihen beziehungsweise -typen.



Abbildung 5.1.b : Deutsche Indexseite des Fachbereiches Mathematik

Desweiteren stehen hier Links zur Aufnahme von Veröffentlichungen und zum Bearbeiten von Veröffentlichungsinformationen zur Verfügung.

### 5.1.2. Abfrage und Suche von Informationen

Zur Informationssuche steht dem Benutzer eine komfortable Suchmaske zur Verfügung (Abbildung 5.1.c). Gesucht werden kann nach Autor, Titel, Stichwörtern und im Abstrakt und Volltext der Dokumente.

Die fachbereichsspezifische Suchmaske erreicht man durch Auswahl des Punktes "Suchen" von der Hauptseite des Fachbereiches aus und die fachbereichsübergreifende Suche von der KLUEDO-Hauptseite. Von der Suchmaske aus existiert ein direkter Link zur fachbereichsspezifischen Indexseite, sodass der Benutzer immer zwischen Suche und Browsen umschalten kann.

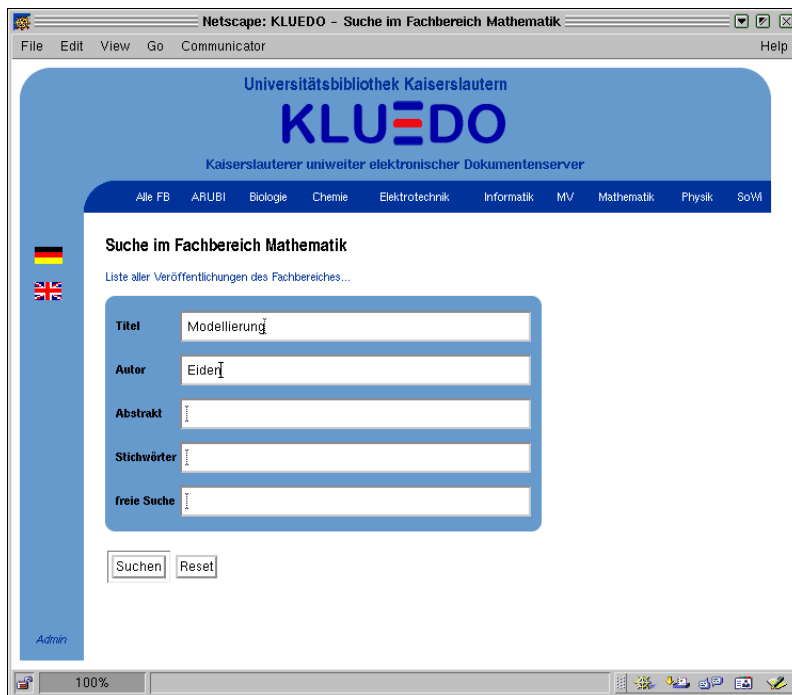


Abbildung 5.1.c : Suchmaske

Als Ergebnis erhält man eine Liste, in der die gefundenen Titel und die Anzahl der Treffer aufgeführt sind (Abbildung 5.1.d). Dort wird zu jedem Suchergebnis unter anderem der Titel, der Autor, ein Link zur kompletten Dokumentbeschreibung (Metadaten) und die Links zum Volltext angeboten.

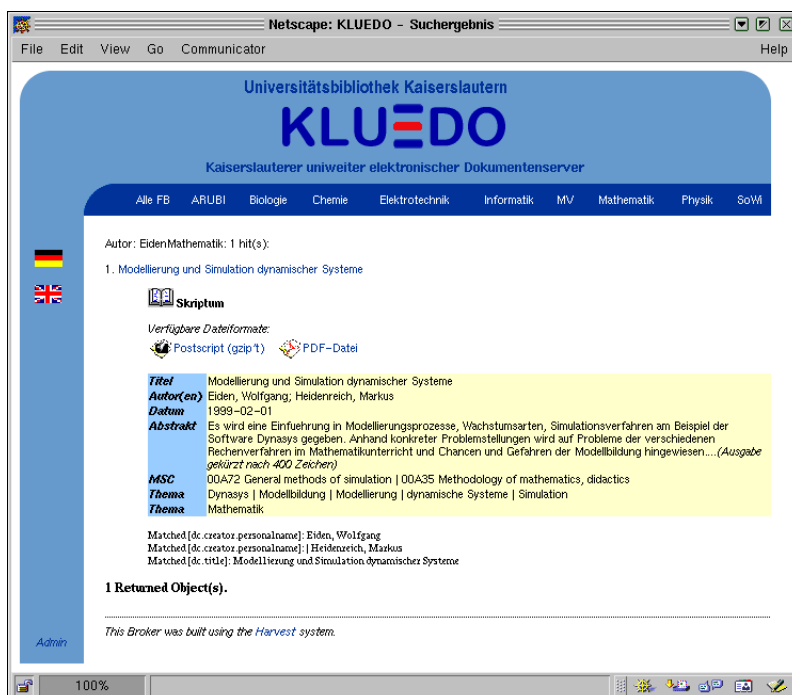


Abbildung 5.1.d : Ergebnis einer Suche

Anhand dieser Übersicht kann der Benutzer eine Vorauswahl treffen, welche Dokumente im konkreten Fall von Interesse ist. Die relevanten Dokumente können nun vom Benutzer im gewünschten Format (in der Regel Postscript oder PDF) direkt auf den eigenen Rechner heruntergeladen werden.

### 5.1.3. Browsen in Veröffentlichungslisten

Auf den fachbereichsspezifischen Seiten findet sich eine Übersicht über alle Veröffentlichungsreihen beziehungsweise -typen des entsprechenden Fachbereiches. Durch Auswahl einer dieser Typen erhält man eine Übersicht über die dem Typ zugehörigen Veröffentlichungen (*Abbildung 5.1.e*).



Abbildung 5.1.e : Auszug aus einer Veröffentlichungsliste

Mit den Navigationsschaltflächen (*Abbildung 5.1.f*) kann der Benutzer bequem durch die Veröffentlichungsliste navigieren und im Eingabefeld den Bereich und damit implizit die Anzahl der angezeigten Informationen bestimmen.



Abbildung 5.1.f : Navigationsschaltflächen

Ausgehend von der Übersicht kann sich der Benutzer dann weitere (Meta-) Informationen zu einer Veröffentlichung ansehen (vgl. *Abbildung 5.1.g*)



Abbildung 5.1.g : Informationen zu einer Veröffentlichung

und / oder die entsprechenden Dokumente herunterladen.

#### 5.1.4. Aufnahme von Veröffentlichungen

Der Dokumentenserver KLUEDO ermöglicht die Aufnahme von Dokumenten durch den Autor selbst. Dazu steht auf den jeweiligen fachbereichsspezifischen Indexseiten unter dem Punkt "Aufnahme von Veröffentlichungen" eine Eingabemaske bereit (Abbildung 5.1.h), in der der Autor die relevanten Informationen eintragen kann um das Dokument mit Zusatzinformationen (Metadaten) zu versehen.

The screenshot shows a Netscape browser window titled "Netscape: KLUEDO - Metamaker - Fachbereich Mathematik". The page is the submission form for a new publication at the Universitätsbibliothek Kaiserslautern. The form is organized into several sections:

- Header:** "Universitätsbibliothek Kaiserslautern KLUEDO Kaiserslauterer uniweiter elektronischer Dokumentenserver". Navigation links include "Alle FB", "ARUBI", "Biologie", "Chemie", "Elektrotechnik", "Informatik", "MV", "Mathematik", "Physik", and "SoWi".
- Aufnahme einer neuen Veröffentlichung:** Includes a "Hinweis" about required fields and a "Bearbeitungshinweise" section with an "Ihre email-Adresse" field and a "Hinweise für die Bearbeitung" text area.
- Bibliographische Beschreibung:** Contains fields for "Autor(en)", "email-Adressen der Autoren", "Institut(e)" (pre-filled with "Universität Kaiserslautern"), "Titel der Veröffentlichung", "Jahr des intellektuellen Entstehens", "Bibliographische Referenz", "Art der Veröffentlichung" (with "Preprint" and "englischer" selected), "Reihe in der die Veröffentlichung erscheinen soll" (pre-filled with "Papers außerhalb einer Reihe"), and "Nummer" or "Kenzeichnung" fields.
- Inhaltliche Beschreibung:** Includes a large "Zusammenfassung" text area, "Schlagwörter", "MSC Klassifikation", and "Notizen" text areas.
- Footer:** "Admin" link and "Ergebnis anzeigen" / "Reset" buttons.

Abbildung 5.1.h : Eingabemaske (Mathematik)



Nach dem Absenden der Daten erhält man eine Ausgabe der vorläufigen Ergebnisse.

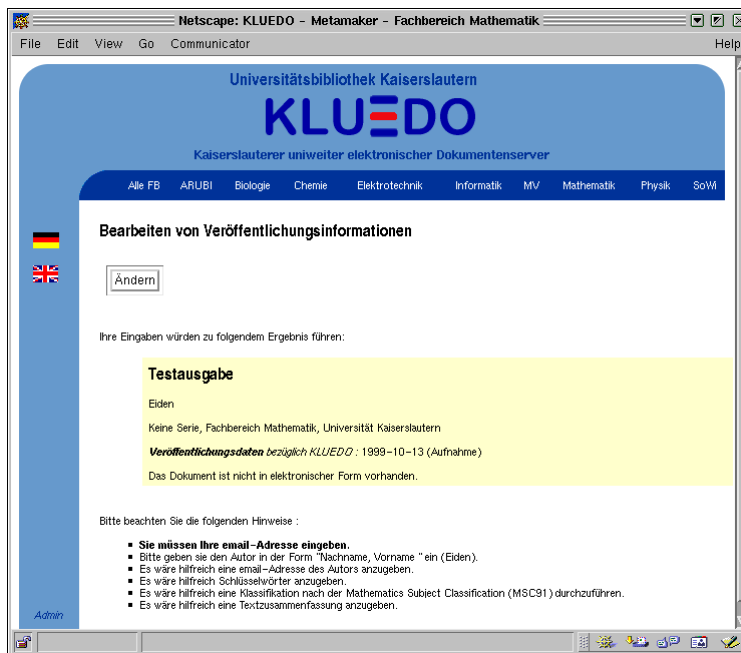


Abbildung 5.1.i : Testausgabe von Metadaten (mit unzulässigen Angaben)

Wurden unzulässige Angaben gemacht (wenn beispielsweise der Benutzer die markierten Pflichtfelder nicht ausgefüllt hat), so können aufgrund der Plausibilitätsprüfung die eingegebenen Daten nur korrigiert werden, ansonsten aber auch (als noch einzubindende Daten) auf den Dokumentenserver gespeichert werden.

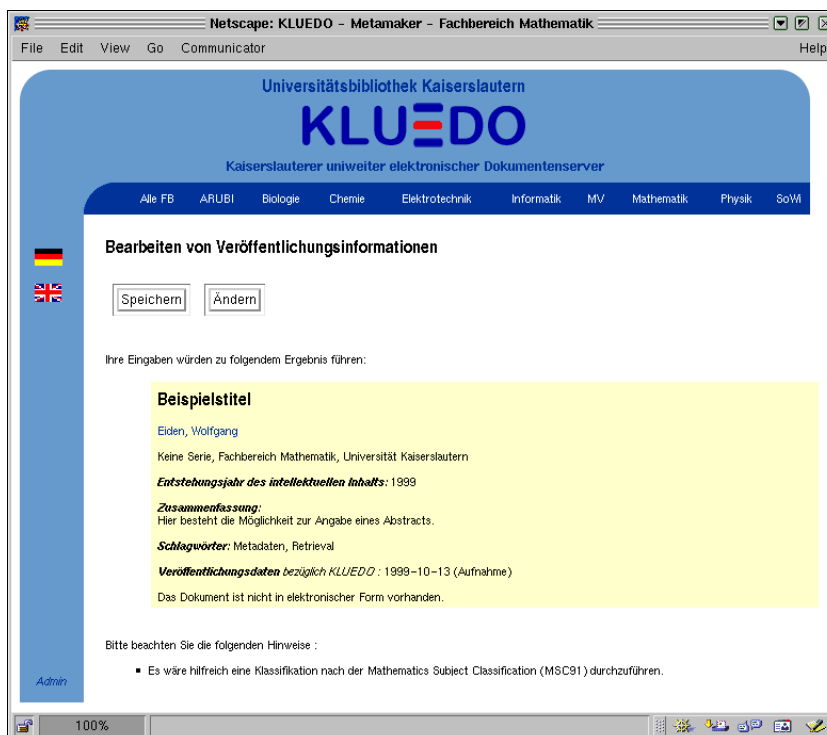


Abbildung 5.1.j : Testausgabe von Metadaten

Der Benutzer erhält nach dem Speichern eine Bestätigung und kann sein Dokument nun via Fileupload auf den Server überspielen (Abbildung 5.1.k).



Abbildung 5.1.k : Speicherungsbestätigung und Fileuploadmöglichkeit

Sowohl die übertragenen Metadaten als auch das eigentliche Dokument werden unter temporären Namen gespeichert und erst nach Prüfung durch das (mit einer automatisch generierten Mail informierte) Administratorenteam übernommen.

### 5.1.5. Editieren von Veröffentlichungsinformationen

Um Veröffentlichungsinformationen zu editieren, kann man von der fachbereichs-spezifischen Indexseite aus den Punkt "Editieren von Veröffentlichungsinformationen" verwenden. Man erhält dann zunächst eine Übersicht über alle Veröffentlichungsreihen (beziehungsweise -typen) des entsprechenden Fachbereiches und schließlich durch Auswahl einer dieser Typen eine Kurzübersicht über alle erschienenen Veröffentlichungen dieses Typuses. Nach Auswahl einer konkreten Veröffentlichung werden dem Benutzer die Metainformationen der entsprechenden Veröffentlichung nochmals angezeigt. Nach der Bestätigung seiner Änderungsabsicht erhält der Benutzer das Änderungsformular mit den Metadaten. Das weitere Procedere ist identisch mit dem der Neuaufnahme von Dokumenten.

## 5.2. Systemumgebung

Der Dokumentenserver KLUEDO setzt einen HTTP-Server mit PHP-Unterstützung und die Installation der Programmiersprache Perl (ab Version 5.0) voraus. Er ist unter dem Betriebssystem *SuSE Linux* [41] auf einem Dual Pentium III mit 512 MB Arbeitsspeicher installiert.

### 5.2.1. HTTP-Server

Als HTTP-Server wird der *Apache Webserver* [2], der derzeit am häufigst verwendete HTTP-Server weltweit, verwendet. Es handelt sich hierbei um einen freien, open-source HTTP-Server mit Distributionen für diverse Server-Betriebssysteme (wie beispielsweise UNIX und Windows NT).

Der Dokumentenserver wurde in der Datei *access.conf* als virtueller Host *kluedo.ub.uni-kl.de* deklariert und verschiedene Verzeichnisse (Fachbereiche und das zentrale Bildverzeichnis) als Aliase eingerichtet:

```
<VirtualHost kluedo.ub.uni-kl.de>
  ServerName          kluedo.ub.uni-kl.de
  DocumentRoot        /usr/local/metadata/KLUEDO/
  ServerAdmin         kluedo@ub.uni-kl.de

  Alias /images/      /usr/local/metadata/KLUEDO/images/
  Alias /ARUBI/       /usr/local/metadata/KLUEDO/Fachbereiche/ARUBI/
  Alias /Biologie/    /usr/local/metadata/KLUEDO/Fachbereiche/Biologie/
  Alias /Chemie/      /usr/local/metadata/KLUEDO/Fachbereiche/Chemie/
  Alias /Elektrotechnik/ /usr/local/metadata/KLUEDO/Fachbereiche/Elektrotechnik/
  Alias /Informatik/  /usr/local/metadata/KLUEDO/Fachbereiche/Informatik/
  Alias /MV/          /usr/local/metadata/KLUEDO/Fachbereiche/MV/
  Alias /Mathematik/  /usr/local/metadata/KLUEDO/Fachbereiche/Mathematik/
  Alias /Physik/      /usr/local/metadata/KLUEDO/Fachbereiche/Physik/
  Alias /SoWi/        /usr/local/metadata/KLUEDO/Fachbereiche/SoWi/

</VirtualHost>
```

Um maximale Sicherheit zu gewährleisten, wurde zunächst der Zugriff für alle Verzeichnisse und Dateien des Servers gesperrt

```
<Directory />
  Options          None
  AllowOverride    None
  Order            deny,allow
  deny            from all
</Directory>
```

und dann im Rahmen der Integration des Hypertext Preprocessors PHP (siehe später) für die Dateien von KLUEDO wieder erlaubt:

```
<Directory /usr/local/metadata/KLUEDO>
  AddType          application/x-httpd-php3 .html
  Options          All
  order            deny,allow
</Directory>
```

Eine Ausnahme bilden die Administrationsverzeichnisse und -dateien der Fachbereiche, auf die nur ein authentizierter Benutzer Zugriff hat:

```
<Directory /usr/local/metadata/KLUEDO/Fachbereiche/*/admin/>
  Deny from all
</Directory>

<IfDefine SSL>
  <VirtualHost kluedo.ub.uni-kl.de:443>

    ServerName kluedo.ub.uni-kl.de
    ServerAdmin kluedo@ub.uni-kl.de
    DocumentRoot /usr/local/metadata/KLUEDO/

    SSLEngine on
    SSLCertificateFile /usr/local/apache/conf/ssl.crt/server.crt
    SSLCertificateKeyFile /usr/local/apache/conf/ssl.key/server_.key

    <Directory /usr/local/metadata/KLUEDO>
      AddType application/x-httpd-php3 .html
      Options All
      order deny,allow
    </Directory>

    Alias /images/ /usr/local/metadata/KLUEDO/images/
    Alias /ARUBI/ /usr/local/metadata/KLUEDO/Fachbereiche/ARUBI/
    Alias /Biologie/ /usr/local/metadata/KLUEDO/Fachbereiche/Biologie/
    Alias /Chemie/ /usr/local/metadata/KLUEDO/Fachbereiche/Chemie/
    Alias /Elektrotechnik/ /usr/local/metadata/KLUEDO/Fachbereiche/Elektrotechnik/
    Alias /Informatik/ /usr/local/metadata/KLUEDO/Fachbereiche/Informatik/
    Alias /MV/ /usr/local/metadata/KLUEDO/Fachbereiche/MV/
    Alias /Mathematik/ /usr/local/metadata/KLUEDO/Fachbereiche/Mathematik/
    Alias /Physik/ /usr/local/metadata/KLUEDO/Fachbereiche/Physik/
    Alias /SoWi/ /usr/local/metadata/KLUEDO/Fachbereiche/SoWi/

    <Directory /usr/local/metadata/KLUEDO/Fachbereiche/*/admin/>
      Options -Indexes FollowSymLinks
      AllowOverride AuthConfig
      Auth_MySQL_Empty_Passwords Off
      Auth_MySQL On
      Auth_MySQL_DB kluedo_sessions
      Auth_MySQL_Password_Table http_auth
      Order allow,deny
      Allow from all
      AuthType Basic
      AuthName "Administration KLUEDO"
      require valid-user
      satisfy all
    </Directory>

  </VirtualHost>
</IfDefine>
```

## 5.2.2. Hypertext Preprocessor PHP

Bei PHP handelt es sich um einen Hypertext-Preprocessor, also um eine serverseitig interpretierte, in HTML eingebettete Skriptsprache. Hauptanwendungsgebiet ist die Erstellung von Webseiten dynamischen Inhaltes. Die Syntax dieser Skriptsprache ist ähnlich zu C, Java und Perl, erweitert durch PHP-eigene Features wie beispielsweise Kommandos zur Integration von Datenbanken. PHP existiert sowohl für Unix als auch für die Windows-Plattformen 95,98 und NT. Weitere Informationen finden sich im PHP-Net [38].

Wir verwenden PHP um ein einheitliches Aussehen aller HTML-Seiten des Dokumentenservers sicherzustellen und ein Session-Management zu ermöglichen. Dazu wurden Routinen entwickelt, die das eigentliche HTML-Dokument als Tabellenzelle in einen Rahmen (Tabelle) mit integriertem Navigationssystem (um beispielsweise Fachbereich- und Sprachwechsel zu ermöglichen) einbettet:

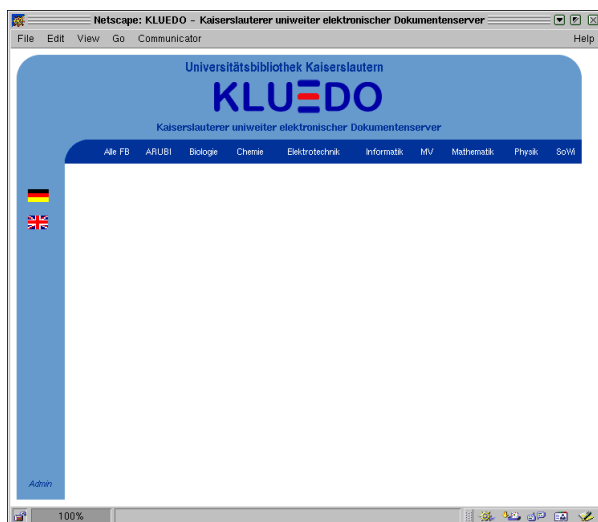


Abbildung 5.2.a : Beispielsseite

Jede HTML-Seite verwendet die PHP-Datei *init.pinc* (die als zentraler Anknüpfungspunkt die angesprochenen Funktionalitäten zur Verfügung stellt)

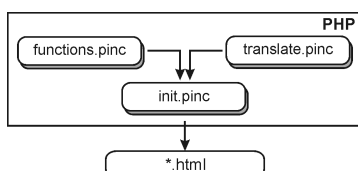


Abbildung 5.2.b : Integration von PHP

und ist von der Struktur

```
<?
$KLUEDO_HOME = "/opt/metadata/KLUEDO";
require("$KLUEDO_HOME/php/init.pinc");
do_header("[Seitentitel] ");
do_body(" ");
?>

...[Standard-HTML-Quelltext]...

<?
do_footer();
?>
```

Beim Aufruf werden die PHP-Seiten serverseitig ausgewertet, sodass der Benutzer reine HTML-Dokumente erhält.

## 5.3. Struktureller Aufbau

Im folgenden wird der strukturelle Aufbau von KLUEDO etwas näher beleuchtet und die Verwendungszwecke einzelner Verzeichnisse und Dateien vorgestellt.

### 5.3.1. Basisverzeichnis

Das Basisverzeichnis von KLUEDO besitzt folgende Form

```
bin
conf
doc
error
Fachbereiche
images
Klassifikationen
lib
php
index -> doc/helpmap.html
index_de.html -> index
index_en.html -> index
suche.html
suche_de.html -> suche.html
suche_en.html -> suchte.html
```

Im Verzeichnis `bin` befinden sich die Administrationskripte, in `doc` die Hilfeseiten zum Dokumentenserver, in `error` die HTML-Seiten zur formatierten Ausgabe von Fehlermeldungen des WWW-Servers und in `images` die von KLUEDO verwendeten Grafiken. Die Bibliotheken von KLUEDO und zentralen Konfigurationsdateien (siehe Paragraph 5.4) befinden sich in den Verzeichnissen `lib` beziehungsweise `conf`. Das Verzeichnis `php` enthält die PHP-Include-Dateien

```
functions.pinc
init.pinc
translate.pinc
navigation.pinc
```

und zentralen HTML-Seiten:

```
index.html
indexadm.html
listen.html
metamaker.html
suche.html
upload.html
```

So ist beispielsweise der Metamaker (Datei `metamaker_[L].html`) der einzelnen

Fachbereiche (die im Verzeichnis `Fachbereiche` jeweils in einem eigenen Unterverzeichnis realisiert sind) ein symbolischer Link auf die Datei `metamaker.html`.

### 5.3.2. Struktur der einzelnen Fachbereiche

Alle Fachbereiche sind bezüglich der Struktur identisch:

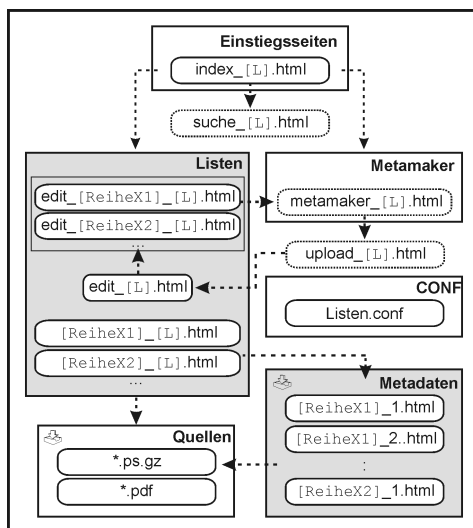


Abbildung 5.3.a : Struktur eines Fachbereiches

Die HTML-Seiten verweisen auf die zentralen PHP-Seiten für den Index, den Metamaker, die Suche und den Dateiapload:

```

index_de.html -> ../../php/index.html
index_en.html -> ../../php/index.html
metamaker_de.html -> ../../php/metamaker.html
metamaker_en.html -> ../../php/metamaker.html
suche_de.html -> ../../php/suche.html
suche_en.html -> ../../php/suche.html
upload_de.html -> ../../php/upload.html
upload_en.html -> ../../php/upload.html

```

Als Unterverzeichnisse existieren die Verzeichnisse *admin*, *conf*, *Listen*, *Metadaten*, *Metadaten\_Eingang*, *Quellen* und *Quellen\_Eingang*. Die Rechte der Verzeichnisse ist so gesetzt, dass die WWW-Benutzer nur in die Verzeichnisse *Metadaten\_Eingang*, *Quellen\_Eingang* und *Listen* schreiben können. Hierbei wird das Schreibrecht im Verzeichnis *Listen* bei Änderungsoperationen via WWW zur automatischen Neuerzeugung der Listen benötigt. Im Verzeichnis *conf* befindet sich zur Zeit nur die Konfigurationsdatei *Listen.conf*, welche die Veröffentlichungsreihen eines Fachbereiches definiert.

Diese Datei besitzt beispielsweise für den Fachbereich Mathematik an der Universität Kaiserslautern zur Zeit folgendes Aussehen:

```
_incoming|Neuzugänge::Neuzugang|incoming::newfile
cagelb|Report on Computeralgebra::ZCA Report|Report on Computeralgebra::ZCA Report
diplom|Diplomarbeiten im Fachbereich Mathematik::Diplomarbeit|Diploma thesis::Diploma
  Thesis
dissertation|Dissertationen im Fachbereich
Mathematik::Dissertation|Dissertation::Dissertation
gelb|Report in Wirtschaftsmathematik::WIMA Report|Report in Wirtschaftsmathematik::WIMA
  Report
gruen|Berichte der Arbeitsgruppe Technomathematik::AGTM Report|Berichte der Arbeitsgruppe
  Technomathematik::AGTM Report
itwm|Berichte des Instituts für Techno- und Wirtschaftsmathematik::ITWM
Report|Berichte des Instituts für Techno- und Wirtschaftsmathematik::ITWM Report
no_series|Papers außerhalb einer Reihe::Keine Serie|Papers not belonging to a
  series::No Series
rot|Preprint::Preprint|Preprint::Preprint
script|Vorlesungsskripte im Fachbereich Mathematik::Vorlesungsskript|Scripts by Department
  of Mathematics::script
seminar|Seminarvorträge im Fachbereich Mathematik::Seminarvortrag|Seminarlectures by
  Department of Mathematics::Seminarlecture
```

Das Unterverzeichnis *admin* besteht ausschließlich aus symbolischen Links:

```
Listen -> ../Listen/
Metadaten -> ../Metadaten
Metadaten_Eingang -> ../Metadaten_Eingang
Quellen -> ../Quellen
Quellen_Eingang -> ../Quellen_Eingang
index_de.html -> ../../../../php/indexadm.html
index_en.html -> ../../../../php/indexadm.html
metamaker_de.html -> ../../../../php/metamaker.html
metamaker_en.html -> ../../../../php/metamaker.html
upload_de.html -> ../../../../php/upload.html
upload_en.html -> ../../../../php/upload.html
```

Die Dateien in diesem Verzeichnis sind geschützt und können nur von einem Systemadministrator (nach Eingabe eines Benutzernamens und Passwortes) über das Internet abgerufen werden. Durch URL-Splitting kann somit die Berechtigungsklasse des Benutzers ermittelt werden.

Die Indexseiten verweisen (in ihrer Funktion als Einstiegsseiten) via URL auf die Suche und die einzelnen Reihen des Fachbereiches. Desweiteren wird dem Benutzer die Möglichkeit der Neuerzeugung und Bearbeitung von Metadaten gegeben (parameterisierter Verweis auf die zentrale PHP-HTML-Seite *Metamaker.html* beziehungsweise direkter Verweis auf die Seite *edit\_[L].html*).



## 5.4. Konfigurationsdateien und Bibliotheken

Als zentrale Konfigurationsdatei ist zur Zeit nur die Datei *language.conf* vorhanden:

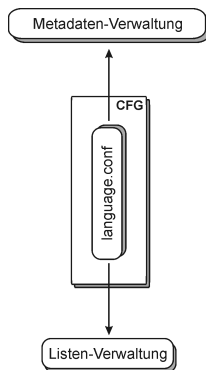


Abbildung 5.4.a : Zentrale Konfigurationsdateien

Die Konfigurationsdatei dient zur Sprachübersetzung einer (mit einem Keyword assoziierten) Phrase. Jede Definitionszeile (Leerzeilen und Kommentarzeilen sind ebenfalls zugelassen) besitzt, wie aus dem Auszug

```

# Fachbereiche
dep_ARUBI|Fachbereich Architektur, Raum- und Umweltplanung, Bauingenieurwesen|Department of
  Architecture, Town and Environmental Planning, Civil Engineering
dep_Biologie|Fachbereich Biologie|Department of Biology
dep_Chemie|Fachbereich Chemie|Department of Chemistry
dep_Elektrotechnik|Fachbereich Elektrotechnik|Department of Electrical Engineering
dep_Informatik| Fachbereich Informatik|Department of Computer Science
dep_MV|Fachbereich Maschinenbau und Verfahrenstechnik|Department of Mechanical and Chemical
  Engineering
dep_Mathematik|Fachbereich Mathematik|Department of Mathematics
dep_Physik|Fachbereich Physik|Department of Physics
dep_SoWi|Fachbereich Sozial- und Wirtschaftswissenschaften|Faculty of Economics, Managment
  and Social Sciences

# Art der Veröffentlichung
Article|Artikel|Article
Preprint|Preprint|Preprint
Script|Skriptum|Script
Phd_thesis|Dissertation|PhD Thesis
Dipl_thesis|Diplomarbeit|Diploma Thesis

# Anzeige von Metadaten
About|Details|About
Authors|Autor(en)|Author(s)
Title|Titel|Title
Year|Jahr|Year
Date|Datum|Date
Document|Dokument|Document
Status|Status|Status
Tip|Hinweis|Tip
  
```

ersichtlich, den Aufbau `keyword|deutsche Phrase|englische Phrase`. Die Übersetzungsroutine und der Dateiaufbau sind so konzipiert, dass eine Sprachunterstützung für weitere Sprachen problemlos möglich ist (siehe später).

Um die, in Perl implementierten, Verwaltungsskripte möglichst einfach zu halten, wurden zentrale Bibliotheken geschaffen, die sowohl bei der Metadaten- als auch bei der Listenverwaltung verwendet werden:

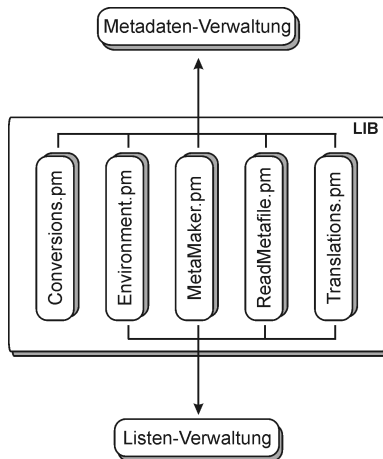


Abbildung 5.4.b : Bibliotheken

### 5.4.1. Die Bibliothek 'Conversions.pm'

Die Bibliothek *Conversions.pm* definiert diverse Konvertierungs- und Verifikationsoperationen für Datentypen, CGI, HTML, Normen (beispielsweise ISO8601 normierte Datumsangaben) und Klassifikationen (beispielsweise MSC1991-Klassifikation).

Routine	<code>checkclassification_MSC</code>
Beschreibung	Überprüft die angegebene MSC und korrigiert sie gegebenenfalls gemäß MSC91-Klassifikation.
Parameter	zu überprüfende MSC
Rückgabe	liefert das Tupel <code>(msc, mscname, changed)</code> , mit: <code>msc</code> : Rückgabe der gewünschten (evtl. korrigierten) MSC <code>mscname</code> : komplette MSC-Bezeichnung (Name) <code>changed</code> : <code>changed = 1</code> gdw. MSC korrigiert

Routine	<code>checkSyntaxOfName</code>
Beschreibung	Syntaxüberprüfung und ggf. (falls möglich) Korrektur auf Form "Nachname, Vorname".
Parameter	Name
Rückgabe	liefert das Tupel <code>(name, error)</code> : <code>name</code> : Rückgabe des ggf. korrigierten Namens <code>error</code> : <code>error = true</code> gdw. falsches Format bzw. keine Korrektur möglich

Routine	<code>code_specialchars</code>
Beschreibung	Kodiert die Sonderzeichen der übergebenen Zeichenkette HTML-Syntax-konform.
Parameter	<code>line</code> : der zu kodierende Ausdruck (Zeichenkette)
Rückgabe	Liefert den kodierten Ausdruck.
Routine	<code>convertDate</code>
Beschreibung	Konvertiert das übergebene Datum gemäß ISO 8601-Norm (YYYY-MM-DD).
Parameter	<code>Tag</code> , <code>Monat</code> , <code>Jahr</code>
Rückgabe	Liefert bei korrektem Datum, das entsprechende ISO 8601 - Format zurück, ansonsten einen leeren String.
Routine	<code>getactualDateInISO8601</code>
Beschreibung	Liefert das aktuelle Datum im ISO 8601-Format (YYYY-MM-DD) zurück.
Parameter	-
Rückgabe	Liefert das gewünschte Datum.
Routine	<code>trim</code>
Beschreibung	Entfernt führende oder anhängende Leerzeichen.
Parameter	Zeichenkette oder Array von Zeichenketten
Rückgabe	Liefert die konvertierte Form.
Routine	<code>unescape</code>
Beschreibung	Dekodiert das Format für Parameterangaben via URL und kodiert die übergebenen Zeichenkette HTML-Syntax-konform.
Parameter	<code>line</code> : Der zu kodierende Ausdruck (Zeichenkette)
Rückgabe	Liefert den kodierten Ausdruck.
Routine	<code>viewSpecialHTMLConstructs</code>
Beschreibung	Realisation von Links, Zeilenumbrüchen, Paragraphen in Abstracts / Notes bei der Anzeige von Metadaten.
Parameter	Zeichenkette
Rückgabe	liefert eine Zeichenkette, bei dem die erkannten HTML-Konstrukte von <code>text</code> in HTML-Syntax notiert sind (beispielsweise wird aus <code>&amp;lt;p&gt;</code> das Konstrukt <code>&lt;p&gt;</code> ).

#### 5.4.2. Die Bibliothek 'Environment.pm'

Die Bibliothek *Environment.pm* stellt Routinen zur Verfügung mit dessen Hilfe die Umgebungsvariablen (wie beispielsweise Basisverzeichnisse und interne Sprachcodes) ermittelt werden können.

Routine	<code>Environment</code>
Beschreibung	Liefert Umgebungsvariablen von KLUEDO. Zur Zeit sind folgende Variablen definiert: <code>CLASS:</code> Zentrales Klassifikationsverzeichnis (Dateiebene) <code>CONF:</code> Zentrales Konfigurationsverzeichnis (Dateiebene) <code>DEPTS:</code> Basisverzeichnis der Fachbereiche (Dateiebene) <code>HOME:</code> Homeverzeichnis (Dateiebene) <code>URL:</code> Basisurl des Dokumentenservers (Internet-URL)
Parameter	-
Rückgabe	HASH-Tabelle
Routine	<code>getLanguageCode</code>
Beschreibung	Liefert zum internen numerischen Code den zweistelligen Sprachcode (nach NISO Z39.53), beispielsweise <code>en</code> (Englisch) für <code>1</code> ).
Parameter	interner numerischer Code
Rückgabe	zugehöriger zweistelliger Sprachcode nach NISO Z39.53
Routine	<code>getLanguageCount</code>
Beschreibung	Liefert die Anzahl der zur Zeit realisierten Sprachen.
Parameter	-
Rückgabe	Anzahl der realisierten Sprachen
Routine	<code>getLanguageID</code>
Beschreibung	Liefert zum zweistelligen Sprachcode (nach NISO Z39.53) den internen numerischen Code (beispielsweise <code>1</code> für Englisch ( <code>en</code> )).
Parameter	zweistelliger Sprachcode nach NISO Z39.53
Rückgabe	Liefert den internen (numerischen) Sprachcode
Routine	<code>getLanguagesInfo</code>
Beschreibung	Liefert die Anzahl der zur Zeit realisierten Sprachen und eine Hash-Tabelle, in der die Sprachcodes auf die internen numerischen Code gemappt werden.
Parameter	-
Rückgabe	liefert das Tupel <code>(numLang, codesLang)</code> , mit: <code>numLang:</code> Anzahl der zur Zeit realisierten Sprachen <code>codesLang:</code> Hashtabelle (siehe oben)

### 5.4.3. Die Bibliothek 'MetaMaker.pm'

Die Bibliothek *Metamaker.pm* dient zum Erzeugen von Metatags gemäß Dublin Core. Hierbei werden die zu erzeugenden Informationen vorher (gemäß Normierung und verwendeten Klassifikationen) geprüft und gegebenenfalls korrigiert.

Routine	<code>checkDates</code>
Beschreibung	korrigiert die von KLUEDO veröffentlichten Datumsangaben (Einbindungsdatum, Datum der letzten Änderung). Das Datum der letzten Änderung wird auf das aktuelle Datum gesetzt. Falls noch kein Einbindungsdatum vorhanden ist, so wird hierfür ebenfalls das aktuelle Datum verwendet.
Parameter	<code>metadata</code> : HASH-Tabelle, die alle Metadaten enthält
Rückgabe	-
Routine	<code>dcmetatag</code>
Beschreibung	Setzt aus den gegebenen Informationen den Dublin Core Metatag zusammen.
Parameter	<code>metaname</code> : Name des DC-Elementes <code>scheme</code> : verwendetes Schema (oder leer) <code>values</code> : Wert / Werte <code>sep</code> : Trennzeichenkette: Falls nicht leer, werden die Werte damit gesplittet und in jeweils einem Metatag erfasst
Rückgabe	Liefert den gewünschten Dublin Core Metatag.
Routine	<code>generateInformationStr</code>
Beschreibung	Generiert einen String zur Anzeige der (für den Benutzer relevanten) Metadaten in einem HTML-Dokument.
Parameter	<code>department</code> : Fachbereich / Fachgebiet <code>metadata</code> : HASH-Tabelle, die alle Metadaten enthält
Rückgabe	Liefert die gewünschte Zeichenkette.
Routine	<code>generateMetadataStr</code>
Beschreibung	Liefert den String für das kompletten Metadatenfile (HTML-Umgebung, Metadaten und die für den Benutzer lesbaren Informationen)
Parameter	<code>department</code> : Fachbereich / Fachgebiet <code>metadata</code> : HASH-Tabelle, die alle Metadaten enthält
Rückgabe	Liefert die gewünschte Zeichenkette.
Routine	<code>generateMetatags</code>
Beschreibung	Liefert die Metadateninformationen für die spezifizierten Dublin Core Elemente als Dublin Core Metatags. Hierbei wird vorher die Existenz von Daten (bei Pflichtfeldern) und die Korrektheit der Angaben geprüft (und ggf. (wo dies möglich ist) korrigiert). Sollten Fehler auftreten, so werden diese in den Felder <code>warning</code> bzw. <code>error</code> der übergebenen HASH-Tabelle aufgenommen.
Parameter	<code>department</code> : Fachbereich / Fachgebiet <code>metadata</code> : HASH-Tabelle, die alle Metadaten enthält
Rückgabe	Liefert die gewünschten Dublin Core Metatags.
Routine	<code>getDCSourceStr</code>
Beschreibung	Liefert den (für das Dublin Core Element <code>DC.Source</code> ) zu verwendenden Wert.
Parameter	<code>department</code> : Fachbereich / Fachgebiet <code>serie</code> : Reihe in der die Ressource veröffentlicht werden soll.
Rückgabe	Liefert den gewünschten Wert.

Routine	<code>getFileformats</code>
Beschreibung	Liefert (anhand der Metadaten) die Angaben zu den vorliegenden Dokumentformaten (wie beispielsweise Format, Größe und URL)
Parameter	<code>metadata</code> : HASH-Tabelle, die alle Metadaten enthält
Rückgabe	Liefert die gewünschte HTML-Zeichenkette.
Routine	<code>metaDCCreatorPersonalname</code>
Beschreibung	Liefert die Metadateninformationen für das Dublin Core Element <i>DC.Creator.Personalname</i> als Dublin Core Metatags. Hierbei wird vorher die Korrektheit der Angaben geprüft (und ggf. (wo dies möglich ist) korrigiert). Sollten Fehler auftreten, so werden diese in den Felder <code>warning</code> bzw. <code>error</code> der übergebenen HASH-Tabelle aufgenommen.
Parameter	<code>department</code> : Fachbereich / Fachgebiet <code>metadata</code> : HASH-Tabelle, die alle Metadaten enthält
Rückgabe	Liefert die gewünschten Dublin Core Metatags.
Routine	<code>metaDCSubjectMSC</code>
Beschreibung	Liefert die Metadateninformationen für das Dublin Core Element <i>DC.Subject.Msc</i> als Dublin Core Metatags. Hierbei wird vorher die Korrektheit der Angaben geprüft (und ggf. (wo dies möglich ist) korrigiert). Sollten Fehler auftreten, so werden diese in den Felder <code>warning</code> bzw. <code>error</code> der übergebenen HASH-Tabelle aufgenommen.
Parameter	<code>department</code> : Fachbereich / Fachgebiet <code>metadata</code> : HASH-Tabelle, die alle Metadaten enthält
Rückgabe	Liefert die gewünschten Dublin Core Metatags.
Routinen	<code>metaDCTitle</code> <code>metaDCCreator</code> <code>metaDCSubject</code> <code>metaDCDescription</code> <code>metaDCPublisher</code> <code>metaDCContributor</code> <code>metaDCDate</code> <code>metaDCType</code> <code>metaDCFormatIdentifier</code> <code>metaDCSource</code> <code>metaDCLanguage</code> <code>metaDCRelation</code> <code>metaDCCoverage</code> <code>metaDCRights</code>
Beschreibung	Liefert die Metadateninformationen für die entsprechende Dublin Core Elemente als Dublin Core Metatags. Hierbei wird vorher die Existenz von Daten (bei Pflichtfeldern) und die Korrektheit der Angaben geprüft (und ggf. (wo dies möglich ist) korrigiert). Sollten Fehler auftreten, so werden diese in den Felder <code>warning</code> bzw. <code>error</code> der übergebenen HASH-Tabelle aufgenommen.
Parameter	<code>department</code> : Fachbereich / Fachgebiet <code>metadata</code> : HASH-Tabelle, die alle Metadaten enthält
Rückgabe	Liefert die gewünschten Dublin Core Metatags.

Routine	<code>metaerror</code>
Beschreibung	Aufnahme von aufgetretenen Fehler- und Warnungsmeldungen.
Parameter	<code>type:</code> Art der Meldung (Warnungsmeldung ("w") oder Fehlermeldung (sonst)) <code>errmsg:</code> Meldungstext <code>metadata:</code> Hashtabelle, deren Einträge "error" oder "warning" um die angegebene Meldung ergänzt wird
Rückgabe	-
Routine	<code>metaSchemas</code>
Beschreibung	Diese Routine ermittelt aus den übergebenen Metatags die verwendeten Schemata und fügt die Referenzlinks hinzu.
Parameter	<code>metadataastr:</code> Zeichenkette mit den erzeugten Dublin Core Metatags
Rückgabe	Liefert den (um die Schemareferenz) ergänzten String zurück.
Routine	<code>testMetadata</code>
Beschreibung	Die Existenz von Daten (bei Pflichtfeldern) und die Korrektheit der Angaben werden geprüft (und ggf. (wo dies möglich ist) korrigiert). Sollten Fehler auftreten, so werden diese in den Felder <code>warning</code> bzw. <code>error</code> der übergebenen HASH-Tabelle aufgenommen. Im Anschluß wird ein String zur Anzeige der (für den Benutzer relevanten) Metadaten in einem HTML-Dokument generiert und dann zurückgegeben.
Parameter	<code>department:</code> Fachbereich / Fachgebiet <code>metadata:</code> HASH-Tabelle, die alle Metadaten enthält
Rückgabe	Liefert die gewünschte Zeichenkette.

#### 5.4.4. Die Bibliothek 'ReadMetafile.pm'

Die Bibliothek *ReadMetafile.pm* stellt eine effiziente Routine zum Einlesen der Metadaten aus einer Datei zur Verfügung.

Routine	<code>AdoptMetadata</code>
Beschreibung	Prüft den übergebenen Metadatenwert auf Korrektheit und nimmt diesen in der HASH-Tabelle auf.
Parameter	<code>metadata:</code> HASH-Tabelle, die die Metadaten enthält <code>hashkey:</code> HASH-Schlüssel <code>value:</code> Wert <code>seperator:</code> Seperator, der bei Mehrfachangabe verwendet werden soll
Rückgabe	-
Routine	<code>CheckMetadataSyntax</code>
Beschreibung	Prüft den übergebenen Metadatenwert auf Korrektheit und liefert die korrigierte Fassung.
Parameter	<code>hashkey:</code> HASH-Schlüssel <code>value:</code> Wert
Rückgabe	Liefert den gewünschten Wert.

Routine	provetag
Beschreibung	Prüft ob der erwartete Elementtyp und das erwartete Schema mit dem übergebenen übereinstimmt und übernimmt diesen dann ggf. in die Hashtabelle.
Parameter	metadata: HASH-Tabelle, die die Metadaten enthält metaname: Elementname des Metatags typ: Elementtyp (Schema oder ergänzende Angabe) value: Wert reqname: Erwarteter Elementname (Groß- /Kleinschreibung ist nicht von Relevanz) reqtyp: Erwarteter Elementtyp (leer, falls kein spezielles Typ verlangt wird) seperator: Seperator, der bei Mehrfachangabe verwendet werden soll hashkey: HASH-Schlüssel (interner Name des Elementnamens)
Rückgabe	Liefert "true", wenn der Wert in die Hashtabelle aufgenommen wurde.

Routine	pushMetatag
Beschreibung	Übernimmt die Daten des Metatags in die Hashtabelle. Hierbei werden die Einträge, die ein bekanntes Schema oder Typ angeben, in die spezifizierten Stellen des HASH-Tabelle aufgenommen. Von unbekannte Schemata und Typs werden nur die Werte gespeichert.
Parameter	metadata: HASH-Tabelle, die die Metadaten enthält metaname: Elementname des Metatags typ: Elementtyp (Schema oder ergänzende Angaben) value: Wert
Rückgabe	-

Routine	readMetadata
Beschreibung	liest die Metadaten aus der übergebenen Datei (im HTML-Format) aus und übergibt diese in einer Hashtabelle.
Parameter	shadowfile:Datei, die die Metatags enthält.
Rückgabe	gewünschte Hashtabelle
Bemerkung:	Es wird vorausgesetzt, dass jeder Metatag in einer einzelnen Zeile steht und von der Form <code>&lt;META name="metaname" content="[(typ) ]value"&gt;</code> ist.

#### 5.4.5. Die Bibliothek 'Translations.pm'

Die Bibliothek *Translations.pm* realisiert die Übersetzung von Begriffen in verschiedene Sprachen.

Routine	Translations
Beschreibung	Liefert eine Hashtabelle mit den Übersetzungen (für die definierten Ausdrücke, Begriffe und Sätze)
Parameter	language: Sprache als zweistelliger Sprachcode (nach NISO Z39.53), in die der Begriff übersetzt werden soll (beispielsweise "de" oder "en") path: Dateipfad der Übersetzungsdatei.
Rückgabe	Liefert die gewünschte Hashtabelle.



## 5.5. Metadatenmanagement

Zur Erreichung eines minimalen Aufwandes zur Administration des Dokumentenservers war die Entwicklung zentraler Module zur Erzeugung und Verarbeitung von Metadaten notwendig. Neben den Bibliotheken, die unter anderem das Grundgerüst zum Metadatenmanagement zur Verfügung stellen, bedeutete dies sowohl die Bereitstellung eines zentralen Web-Interfaces (zur Neuaufnahme und Bearbeitung von Metadaten), als auch die eines zentralen Skriptes zur Erzeugung der verschiedenen Veröffentlichungsübersichten.

### 5.5.1. Listenerzeugung

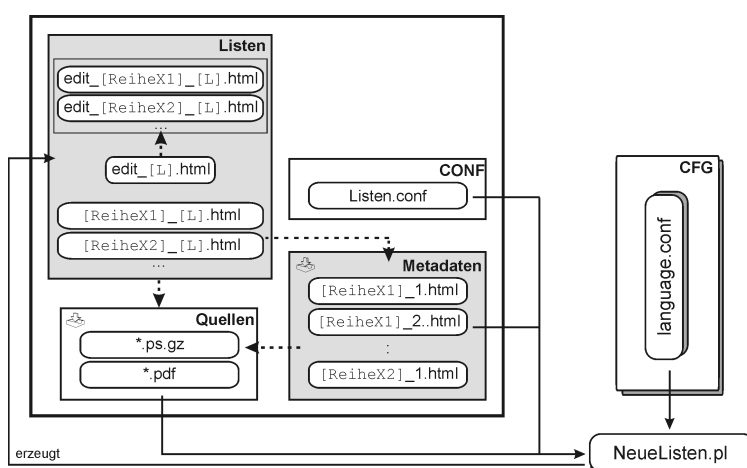


Abbildung 5.5.a : Listenerzeugung

Basierend auf den Metadaten generiert das (zentrale) Perl-Skript `NeueListen.pl` die einzelnen Veröffentlichungslisten der Fachbereiche. Der Aufruf lautet hierbei:

```
NeueListen.pl Fachbereich [Reihe]
```

Wird keine Reihe angegeben, so werden alle Veröffentlichungslisten generiert, die durch die Reihen (in der Konfigurationsdatei `Listen.conf` des übergebenen Fachbereiches) definiert sind (Dies geschieht jeweils für alle in der Bibliothek `Environment.pm` definierten Sprachen). So erzeugt beispielsweise

```
NeueListen.pl Mathematik
```

alle Veröffentlichungslisten des Fachbereiches Mathematik.

Wird eine Reihe angegeben, so geschieht die Listenerzeugung nur für die übergebene Reihe und für die Liste der Neuzugänge. Beim Aufruf des Skripts mit

```
NeueListen.pl Mathematik diplom
```

werden daher nur die Liste der in KLUEDO aufgenommenen Diplomarbeiten und die Liste der Neuzugänge des Fachbereiches Mathematik generiert. Hierbei handelt es sich bei der Liste der Neuzugänge um die Liste derjenigen Dokumente bzw. Dokumentinformationen, die von den Benutzern eingebunden wurden, aber noch nicht durch die Administratoren überprüft und offiziell aufgenommen wurden.

Für jede Veröffentlichungsreihe *Reihe* werden pro Sprache *L* zunächst immer die Datei `Listen/[Reihe]_[L].html` (enthält die Links zu den Metadaten der einzelnen Veröffentlichungen der Reihe *Reihe*) und die Datei `Listen/edit_[Reihe]_[L].html` (enthält die Links zum Bearbeiten der Metadaten der jeweiligen Veröffentlichung) auf Basis der Dateien `Metadaten/[Reihe]_[?].html` (Metadatendateien, auch Shadowfiles genannt) neu generiert (vgl. *Abbildung 5.5.a*). Desweiteren wird beim Aufruf des Skriptes automatisch die Liste `Listen/edit_[L].html` neu erzeugt. Diese Liste enthält die Verweise auf die Listen `edit_[Reihe]_[L].html`.

Auf Basis des Perl-Skriptes `NeueListen.pl` wurden zur Vereinfachung von diversen Operationen (siehe später) das Perl-Skript `Eingang.pl` und das Shell-Skript `AktualisiereListen.sh` geschrieben:



Abbildung 5.5.b : Hilfskripte

Das Skript `Eingang.pl` aktualisiert die Liste der Neuzugänge des Fachbereiches, in dessen Dateibaum sich der Administrator beim Aufruf dieses Befehls gerade befand. Das Skript `AktualisiereListen.sh` aktualisiert die Listen aller Fachbereiche.

### Anmerkung:

Wie wir in Benchmark-Tests festgestellt haben, benötigt der Bildschirmaufbau bei Anzeige sehr umfangreicher Veröffentlichungslisten enorme Zeit. Damit der Benutzer aber auch diese in annehmbarer Zeit betrachten kann, lassen wir den Dokumentenserver standardmäßig nur einen Teil der Veröffentlichungen pro Bildschirm anzeigen und bieten zusätzliche Navigationsleisten (vgl. *Abbildung 5.1.f*) an. Hier kann der Benutzer die Anzahl der Einträge pro Bildschirmseite selbst bestimmen oder sich sogar die komplette Liste ausgeben lassen. Die Navigationsroutinen werden vom PHP-Skript `navigation.pinc` zur Verfügung gestellt.

## 5.5.2. Zentraler Metamaker

Wie bereits in Paragraph 5.3 beschrieben, ist der Metamaker eines jeden Fachbereiches (Datei `metamaker_[L].html`) nur ein symbolischer Link auf die Datei `metamaker.html` im (zentralen) Verzeichnis `php`.

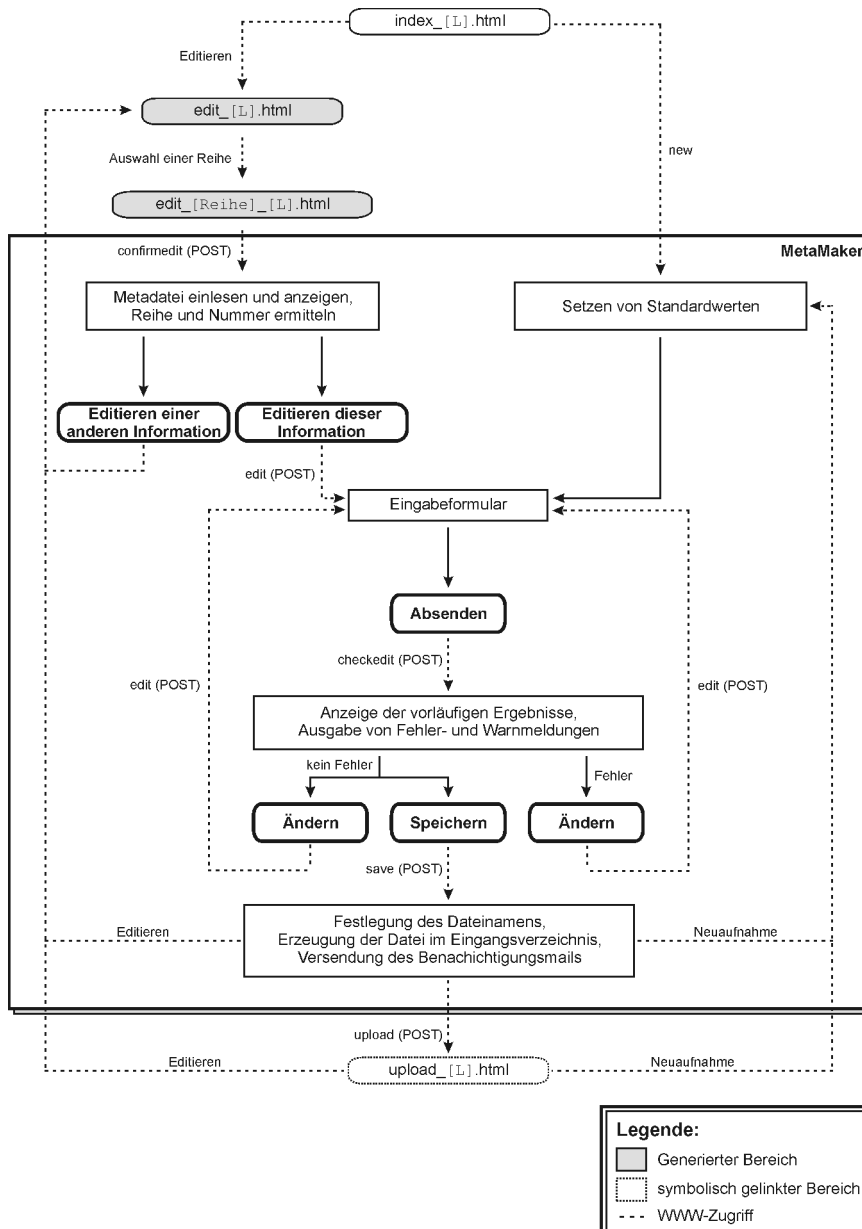


Abbildung 5.5.c: Zentraler Metamaker

Das zugrundeliegenden PHP-Skript kann (aufgrund der in Paragraph 5.2 und Paragraph 5.3 beschriebenen Strukturen) durch URL-Splitting zwischen den verschiedenen Fachbereichen und Sprachen differenzieren. Im Anschluß ruft es das Perl-Skript `MetaMaker.pl` mit den entsprechenden Parametern und dem entsprechenden Status (wie beispielsweise `confirmedit` oder `new`) auf und läßt dann dessen (HTML)-Ausgabe im Browser anzeigen.

Beim Absenden von Formulardaten referenziert sich der Metamaker selbst (d.h. er ist Sender und Empfänger zugleich) und übergibt zusätzliche, den neuen Bearbeitungsmodus charakterisierende, POST-Variablen.

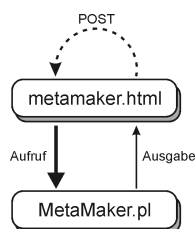


Abbildung 5.5.d : Selbstreferenzierung

Daher gelingt es dem Metamaker aufgrund der Analyse der POST-Variablen den aktuellen Modus festzustellen und entsprechend zu operieren.

### 5.5.3. Einbindung von Dokumenten durch die Administratoren

Nachdem der Benutzer Metadaten auf dem Dokumentenserver gespeichert hat, wird automatisch eine Informationsmail an das KLUEDO-Team versandt. Die einzubindenden Metadaten werden dann durch einen Mitarbeiter zunächst geprüft und gegebenenfalls einer Korrektur unterworfen (beziehungsweise sogar gelöscht, falls es sich um keine sinnvollen Daten handelt). Hat der Benutzer kein Dokument (via Fileupload) auf den Server übertragen, so wird dieses (entsprechend des, vom Benutzer angegebenen, Bearbeitungshinweises) besorgt. Im Anschluß findet eine Konvertierung in die anderen angebotenen Formate statt. Danach werden den Metadaten die URLs und die MIME-Typ-Informationen der zugehörigen Formate hinzugefügt. Schließlich werden die Metadaten und die Dokumente in die Verzeichnisse `Metadaten` und `Quellen` verschoben und dann die Veröffentlichungslisten aktualisiert.

## 5.6. Assimilationsmöglichkeiten

Bei der Entwicklung des Dokumentenservers KLUEDO wurde akribisch darauf geachtet, dass die Konzeption und Implementierung der verschiedenen Komponenten bei gleichzeitigem Streben nach Zentralität und offenen Schnittstellen so allgemein wie möglich erfolgte. Die Assimilation an aktuelle Bedürfnisse sollte daher kein größeres Problem darstellen. Dies möchte ich zum Abschluß noch an ein paar Beispielen demonstrieren.

### 5.6.1. Änderung des Metadatenstandards

Bei Änderung des Metadatenstandards müssen nur die Bibliotheken `MetaMaker.pm` und `ReadMetafile.pm` entsprechend angepasst werden. Dann müssen die alten Metadaten eingelesen und entsprechend modifiziert abgespeichert werden. Im Anschluß müssen dann nur noch die Veröffentlichungslisten durch Aufruf des Skriptes `AktualisiereListen.sh` neu generiert werden.

### 5.6.2. Hinzufügen einer Veröffentlichungsreihe

Um in einem Fachbereich eine weitere Veröffentlichungsreihe hinzuzufügen, muss lediglich die Datei `Listen.conf` des Fachbereichs um eine Zeile der Form

```
interner Name|Langbezeichnung (DE)::Kurzbezeichnung (DE)|....
```

erweitert werden, das Skript `NeueListen.pl` mit

```
NeueListen.pl (Fachbereich) (interner Name)
```

aufgerufen werden und anschließend die erzeugten Seiten in den Indexseiten des Fachbereichs gelinkt werden.

### 5.6.3. Unterstützung weiterer Dateiformate

Zur Unterstützung weiterer Dateiformate muss nur die Routine `GetFileformats` der Bibliothek `MetaMaker.pm` und die Datei `php/upload.html` entsprechend angepasst werden.

### 5.6.4. Unterstützung von weiteren Sprachen

Um eine weitere Sprachen hinzuzufügen muss zunächst die zentrale Konfigurationsdatei `language.conf` um die jeweiligen Keyword-Übersetzungen erweitert werden.

Hierbei ist darauf zu achten, dass jede Definitionszeile den Aufbau

```
keyword|Sprache 1|Sprache 2|...|Sprache n
```

besitzt. Dann ist die Routine `getLanguageID` der Bibliothek `Environment.pm` so zu modifizieren, dass die Anzahl der definierten Sprachen korrekt ist und der zweistellige NISO-Sprachcode der  $i$ -ten Sprache auf den Wert  $(i-1)$  abgebildet wird. Im Anschluß muss dann nur noch für jeden Fachbereich die Konfigurationsdatei `Listen.conf` (wie in Paragraph 5.6.2 beschrieben) erweitert werden und das Skript `AktualisiereListen.sh` aufgerufen werden.

## 5.7. Quelltexte

Im folgenden sind die wichtigsten Quelltexte der von mir geschriebenen Bibliotheken, Skripte und PHP-Seiten abgedruckt, wobei sich einzelne Passagen stellenweise an die Quelltexte des Metamakers der Universität Osnabrück anlehnen:

<i>A</i>	Bibliothek 'Conversions.pm' .....	63
<i>B</i>	Bibliothek 'Environment.pm' .....	67
<i>C</i>	Bibliothek 'MetaMaker.pm' .....	69
<i>D</i>	Bibliothek 'ReadMetafile.pm' .....	82
<i>E</i>	Bibliothek 'Translations.pm' .....	87
<i>F</i>	PHP-HTML-Seite 'metamaker.html' .....	88
<i>G</i>	PHP-HTML-Seite 'upload.html' .....	89
<i>H</i>	PHP-Include 'navigation.pinc' .....	90
<i>I</i>	Skript 'AktualisiereListen.sh' .....	92
<i>J</i>	Skript 'Eingang.pl' .....	93
<i>K</i>	Skript 'MetaMaker.pl' .....	93
<i>L</i>	Skript 'NeueListen.pl' .....	106

## A. Bibliothek 'Conversions.pm'

```
#####
#
# Quelltext:
# (KLUEDO)/lib/KLUEDO/Conversions.pm
#
# Beschreibung:
# Die Bibliothek Conversions.pm definiert diverse Konvertierungs- und
# Verifikationsoperationen für Datentypen, CGI, HTML, Normen
# (beispielsweise ISO8601 normierte Datumsangaben) und Klassifikationen
# (beispielsweise MSC1991-Klassifikation).
#
# Autor:
# Wolfgang Eiden, Universität Kaiserslautern
#
#####

use strict;

use lib('/opt/metadata/KLUEDO/lib');

use IO::File;
use KLUEDO::Environment;

# -----
# - Routine:
# - trim
# - Beschreibung:
# - Entfernt führende oder anhängende Leerzeichen.
# - Parameter:
# - Zeichenkette oder Array von Zeichenketten
# - Rückgabe:
# - Liefert die konvertierte Form.
# -----

sub trim {
    my @field = @_ ;

    for(@field) {
        s/^\s+//;
        s/\s+$//;
    }

    # Ergebnis zurückliefern
    return wantarray ? @field : $field[0];
}

# -----
# - Routine:
# - code_specialchars
# - Beschreibung:
# - Kodiert die Sonderzeichen der übergebenen Zeichenkette
# - HTML-Syntax-konform.
# - Parameter:
# - line: der zu kodierende Ausdruck (Zeichenkette)
# - Rückgabe:
# - Liefert den kodierten Ausdruck.
# -----

sub code_specialchars {
    my ($line) = @_ ;

    # Deutsche Sonderzeichen
    $line =~ s/ä/\&auml;/g; $line =~ s/Ä/\&Auml;/g;
    $line =~ s/ö/\&ouml;/g; $line =~ s/Ö/\&Ouml;/g;
    $line =~ s/ü/\&uuml;/g; $line =~ s/Ü/\&Uuml;/g;
    $line =~ s/ß/\&szlig;/g;

    # Französische Sonderzeichen
    $line =~ s/á/\&aacute;/g; $line =~ s/à/\&agrave;/g;
    $line =~ s/â/\&acirc;/g; $line =~ s/Á/\&Aacute;/g;
    $line =~ s/Ã/\&Agrave;/g; $line =~ s/Â/\&Acirc;/g;
    $line =~ s/é/\&eacute;/g; $line =~ s/è/\&egrave;/g;
    $line =~ s/ê/\&ecirc;/g; $line =~ s/È/\&Eacute;/g;
    $line =~ s/É/\&Egrave;/g; $line =~ s/Ê/\&Ecirc;/g;
    $line =~ s/í/\&iacute;/g; $line =~ s/Î/\&Igrave;/g;
    $line =~ s/ï/\&icirc;/g; $line =~ s/Í/\&Iacute;/g;
    $line =~ s/ì/\&igrave;/g; $line =~ s/î/\&icirc;/g;
    $line =~ s/ó/\&oacute;/g; $line =~ s/ò/\&ograve;/g;
    $line =~ s/ô/\&ocirc;/g; $line =~ s/Ó/\&Oacute;/g;
    $line =~ s/Ô/\&Ograve;/g; $line =~ s/Õ/\&Ocirc;/g;
    $line =~ s/ú/\&uacute;/g; $line =~ s/ù/\&ugrave;/g;
    $line =~ s/û/\&ucirc;/g; $line =~ s/Û/\&Uacute;/g;
    $line =~ s/Û/\&Ugrave;/g; $line =~ s/Û/\&Ucirc;/g;
    $line =~ s/ç/\&ccedil;/g; $line =~ s/Ç/\&Ccedil;/g;
    $line =~ s/«/\&laquo;/g; $line =~ s/»/\&raquo;/g;
}
```

```

# Mathematische Sonderzeichen
$line =~ s/~\/\&not;/g;      $line =~ s/~\/\&macr;/g;
$line =~ s/~\/\&deg;/g;     $line =~ s/~\/\&divide;/g;
$line =~ s/~\/\&plusmn;/g;  $line =~ s/~\/\&ordm;/g;
$line =~ s/~\/\&ordf;/g;    $line =~ s/~\/\&sup1;/g;
$line =~ s/~\/\&sup2;/g;    $line =~ s/~\/\&sup3;/g;
$line =~ s/~\/\&micro;/g;   $line =~ s/~\/\&para;/g;
$line =~ s/~\/\&middot;/g;  $line =~ s/~\/\&frac14;/g;
$line =~ s/~\/\&frac12;/g;  $line =~ s/~\/\&frac34;/g;
$line =~ s/~\/\&excl;/g;    $line =~ s/~\/\&iquest;/g;
$line =~ s/~\/\&aring;/g;   $line =~ s/~\/\&Aring;/g;
$line =~ s/~\/\&aelig;/g;   $line =~ s/~\/\&AElig;/g;
$line =~ s/~\/\&eth;/g;    $line =~ s/~\/\&ETH;/g;
$line =~ s/~\/\&ntilde;/g;  $line =~ s/~\/\&Ntilde;/g;
$line =~ s/~\/\&otilde;/g;  $line =~ s/~\/\&Otilde;/g;
$line =~ s/~\/\&oslash;/g;  $line =~ s/~\/\&Oslash;/g;
$line =~ s/~\/\&yacute;/g;  $line =~ s/~\/\&Yacute;/g;
$line =~ s/~\/\&thorn;/g;   $line =~ s/~\/\&THORN;/g;
$line =~ s/~\/\&times;/g;   $line =~ s/~\/\&brvbar;/g;

# Währungen
$line =~ s/~\/\&pound;/g;   $line =~ s/~\/\&yen;/g;
$line =~ s/~\/\&cent;/g;

# Sonstige
$line =~ s/~\/\&curren;/g;  $line =~ s/~\/\&copy;/g;
$line =~ s/~\/\&reg;/g;

# Ergebnis zurückliefern
return $line;
}

# -----
# - Routine
# - unescape
# - Beschreibung:
# - Dekodiert das Format für Parameterangaben via URL und kodiert die
# - übergebenen Zeichenkette HTML-Syntax-konform.
# -Parameter:
# - line: Der zu kodierende Ausdruck (Zeichenkette)
# - Rückgabe:
# - Liefert den kodierten Ausdruck.
# -----

sub unescape {
    my($line) = @_ ;

    # Zeilenumbrüche entfernen
    $line =~ s/%(13)/ /g;
    $line =~ s/%(10)/ /g;
    $line =~ s/\n/ /g;

    # Angehäufte Leerzeichen entfernen
    $line =~ s/\s+/ /g;

    # HTML-Sonderzeichen
    $line =~ s/</&lt;/g;
    $line =~ s/>/&gt;/g;
    $line =~ s/"/&quot;/g;

    # Sonstige Sonderzeichen
    $line = code_specialchars($line);

    # Leerzeichen am Anfang und Ende entfernen
    $line = trim($line);

    # Ergebnis zurückliefern
    return $line;
}

# -----
# - Routine:
# - convertDate
# - Beschreibung:
# - Konvertiert das übergebene Datum gemäß ISO 8601-Norm (YYYY-MM-DD).
# - Parameter:
# - Tag, Monat, Jahr
# - Rückgabe:
# - Liefert bei korrektem Datum, das entsprechende ISO 8601 - Format zurück,
# - ansonsten einen leeren String.
# -----

sub convertDate {
    my ($day, $month, $year) = @_;
    my ($datestring,$checkeddate,$result);

    # Daten ins richtige Format bringen (YYYY-MM-DD)
    $day = sprintf("%02d", $day);
    $month = sprintf("%02d", $month);
    $year = sprintf ("%04d", $year);
    $datestring = $year."-".$month."-".$day;
}

```



```

# Korrektheit des Datums prüfen (erfolgt über die Unix-Subroutine date)
$checkeddate = `date -d "$datestring" +"%Y-%m-%d"` ;
chomp $checkeddate;
if ($checkeddate ne $datestring) {
    $checkeddate = "";
}

# Ergebnis zurückliefern
return $checkeddate;
}

# -----
# - Routine:
# -   getactualDateInISO8601
# - Beschreibung:
# -   Liefert das aktuelle Datum im ISO 8601-Format (YYYY-MM-DD) zurück.
# - Parameter:
# -
# - Rückgabe:
# -   Liefert das gewünschte Datum.
# -----

sub getactualDateInISO8601
{
    my ($res);

    $res = `date +%Y-%m-%d`;
    chomp $res;

    # Ergebnis zurückliefern
    return $res;
}

# -----
# - Routine:
# -   checkclassification_MSC
# - Beschreibung:
# -   Überprüft die angebene MSC und korrigiert sie gegebenenfalls gemäß
# -   MSC91-Klassifikation.
# - Parameter:
# -   zu überprüfende MSC
# - Rückgabe:
# -   liefert das Tupel (msc,mscname,changed), mit:
# -   msc: Rückgabe der gewünschten (evtl. korrigierten) MSC
# -   mscname: komplette MSC-Bezeichnung (Name)
# -   changed: changed = 1 gdw. MSC korrigiert
# -----

sub checkclassification_MSC
{
    my ($mscin) = @_ ;
    my($msc,$lengthmsc,$mscname,$CFILE,$line,$env,$changed);

    # Environment
    $env = &Environment();

    # Formale Korrektur MSC (Leerzeichen entfernen, Umwandlung in Groß-
    # buchstaben
    $mscin =~ s/\+/ /g;
    $mscin =~ tr/[a-w]/[A-W]/;

    # Feststellen der Länge der MSC
    $lengthmsc = length $mscin;

    # Aufbau der MSC entsprechend der Länge; es werden nur mögliche MSC
    # Nummern aufgebaut (00 braucht nicht unter 00A gesucht zu werden)
    $msc = "";
    if ($lengthmsc ge 5) {
        $msc = substr($mscin,0,5);
    }
    elsif ($lengthmsc ge 3) {
        $msc =substr($mscin,0,3).'xx';
    }
    elsif ($lengthmsc eq 2) {
        $msc =substr($mscin,0,2).'-XX';
    }
}

# MSC-Namen ermitteln nur bei Länge größer gleich 2 !
# Aufbau der Zeilen der MSC-Datei: "XXXXXX Beschreibung"
$mscname="";
if ($msc) {
    $CFILE = IO::File->new("< ". $env{CLASS}."/msc91.txt");
    while (defined($line = $CFILE->getline()) & (not $mscname)) {
        # Handelt es sich um die gewünschte MSC ?
        if ($line) {
            # chomp ($line);
            if ($line =~ /^$msc/) {
                $mscname=$line;
            }
        }
    }
    $CFILE->close();
}
}

```

```

# MSC, MSC-Name, Änderungen MSC zurückgeben
$changed = 0;
if ($lengthmsc lt 5) {
    $changed=1;
}

# Ergebnis zurückliefern
return ($msc,$mscname,$changed);
}

# -----
# - Routine:
# - checkSyntaxOfName
# - Beschreibung:
# - Syntaxüberprüfung und ggf. (falls möglich) Korrektur auf Form
# - "Nachname, Vorname".
# - Parameter:
# - Name
# - Rückgabe:
# - liefert das Tupel (name,error):
# - name: Rückgabe des ggf. korrigierten Namens
# - error: error = true gdw. falsches Format bzw. keine Korrektur möglich
# -----

sub checkSyntaxOfName
{
    my($name)=@_;
    my($res,$error,@array);

    # Initialisierung
    $error="";
    $res=$name;

    # Substitutionen
    $res =~ s/\s*\.\s+\/\./g;
    $res =~ s/\/\./ /g;

    # Anzahl der Kommas prüfen
    @array = split("\.",$res);
    $error="true" if ($#array ne 1);

    # Ergebnis zurückliefern
    return ($res,$error);
}

# -----
# - Routine:
# - viewSpecialHTMLConstructs
# - Beschreibung:
# - Realisation von Links, Zeilenumbrüchen, Paragraphen in Abstracts / Notes
# - bei der Anzeige von Metadaten.
# - Parameter:
# - Zeichenkette
# - Rückgabe
# - liefert eine Zeichenkette, bei dem die erkannten HTML-Konstrukte von
# - text in HTML-Syntax notiert sind (beispielsweise wird aus &lt;p&gt; das
# - Konstrukt <p>).
# - Anmerkung:
# - Globale Substitutionen können bei gruppierenden Substitutionen nicht
# - verwendet werden. Daher ist diese Routine rekursiv geartet.
# -----

sub viewSpecialHTMLConstructs {
    my($string)=@_;
    my($res,$muster);

    # Initialisierung
    $res = $string;

    # Realisation von Links (Form: <a href="...">...</a>)
    $muster = '&lt;a\s+href=&quot;(.*?)&quot;&gt;(.*?)&lt;/a&gt;';
    if ($res =~ /$muster/i) {
        # gruppierende Substitution
        # (vgl. Buch: Programmieren mit Perl, Kapitel 2, S71)
        $res =~ s/$muster/<a href="$1">$2</a>/i;
        $res = viewSpecialHTMLConstructs($res);
    }

    # Realisation von Zeilenumbrüchen, Paragraphen
    $res =~ s/&lt;br&gt;/<br>/ig;
    $res =~ s/&lt;p&gt;/<p>/ig;

    # Ergebnis zurückliefern
    return $res;
}

# -----
1; #return true

```

## B. Bibliothek 'Environment.pm'

```
#####
#
# Quelltext:
# (KLUEDO)/lib/KLUEDO/Environment.pm
#
# Beschreibung:
# Umgebungsvariablen von KLUEDO
#
# Autor:
# Wolfgang Eiden, Universität Kaiserslautern
#
#####

use strict;

# -----
# - Routine
# - Environment
# - Beschreibung:
# - Liefert Umgebungsvariablen von KLUEDO. Zur Zeit sind folgende Variablen
# - definiert:
# - CLASS: Zentrales Klassifikationsverzeichnis (Dateiebene)
# - CONF: Zentrales Konfigurationsverzeichnis (Dateiebene)
# - DEPTS: Basisverzeichnis der Fachbereiche (Dateiebene)
# - HOME: Homeverzeichnis (Dateiebene)
# - URL: Basisurl des Dokumentenservers (Internet-URL)
# - Parameter:
# - -
# - Rückgabe:
# - HASH-Tabelle
# -----

sub Environment {
    my (%names);

    %names =
    (
        URL => "http://\kluedo.ub.uni-kl.de",
        DEPTS => "\opt\metadata\KLUEDO\Fachbereiche",
        HOME => "\opt\metadata\KLUEDO",
        CONF => "\opt\metadata\KLUEDO\conf",
        CLASS => "\opt\metadata\KLUEDO\Klassifikationen"
    );

    # Hashtabelle zurückliefern
    return %names;
}

# -----
# - Routine:
# - getLanguagesInfo
# - Beschreibung:
# - Liefert die Anzahl der zur Zeit realisierten Sprachen und eine
# - Hash-Tabelle, in der die Sprachcodes auf die internen numerischen Code
# - gemappt werden.
# - Parameter:
# - -
# - Rückgabe:
# - liefert das Tupel (numLang,codesLang), mit:
# - numLang: Anzahl der zur Zeit realisierten Sprachen
# - codesLang: Hashtabelle (siehe oben)
# -----

sub getLanguagesInfo {
    my(%codesLang,$numLang);

    %codesLang =
    (
        de => 0,
        en => 1
    );
    $numLang = 2;

    # Ergebnis zurückliefern
    return ($numLang,%codesLang);
}
```

```
# -----
# - Routine:
# -   getLanguageID
# - Beschreibung:
# -   Liefert zum zweistelligen Sprachcode (nach NISO Z39.53) den internen
# -   numerischen Code (beispielsweise 1 für Englisch (en)).
# - Parameter:
# -   zweistelliger Sprachcode nach NISO Z39.53
# - Rückgabe:
# -   Liefert den internen (numerischen) Sprachcode
# -----

sub getLanguageID
{
  my($language) = @_ ;
  my(%codesLang,$languageid,$numLang);

  ($numLang,%codesLang) = getLanguagesInfo();
  $languageid = $codesLang{$language};
  $languageid = 0 if $languageid eq "";
  return $languageid;
}

# -----
# - Routine:
# -   getLanguageCode
# - Beschreibung:
# -   Liefert zum internen numerischen Code den zweistelligen Sprachcode
# -   (nach NISO Z39.53), beispielsweise en (Englisch) für 1).
# - Parameter:
# -   interner numerischer Code
# - Rückgabe:
# -   zugehöriger zweistelliger Sprachcode nach NISO Z39.53
# -----

sub getLanguageCode {
  my($codeLang) = @_ ;
  my(%codesLang,$key,$languageid,$numLang,$value);

  ($numLang,%codesLang) = getLanguagesInfo();

  $languageid="";
  while(($key,$value)=each %codesLang) {
    $languageid=$key if ($value eq $codeLang) && ($languageid eq "");
  }
  $languageid = "de" if ($languageid eq "");

  # Ergebnis zurückgeben
  return $languageid;
}

# -----
# - Routine
# -   getLanguageCount
# - Beschreibung:
# -   Liefert die Anzahl der zur Zeit realisierten Sprachen.
# - Parameter
# -   -
# - Rückgabe:
# -   Anzahl der realisierten Sprachen
# -----

sub getLanguageCount {
  my(%codesLang,$numLang);

  ($numLang,%codesLang) = getLanguagesInfo();
  return $numLang;
}

# -----

1; #return true
```

## C. Bibliothek 'MetaMaker.pm'

```
#####
#
# Quelltext:
# (KLUEDO)/lib/KLUEDO/MetaMaker.pm
#
# Beschreibung:
# Die Bibliothek Metamaker.pm dient zum Erzeugen von Metatags gemäß
# Dublin Core. Hierbei werden die zu erzeugenden Informationen vorher
# (gemäß Normierung und verwendeten Klassifikationen) geprüft und
# gegebenenfalls korrigiert.
#
# Autor:
# Wolfgang Eiden, Universität Kaiserslautern
#
#####

use strict;

use lib('/opt/metadaten/KLUEDO/lib');

use KLUEDO::Conversions;

# -----
# - Routine:
# - metaerror
# - Beschreibung:
# - Aufnahme von aufgetretenen Fehler- und Warnungsmeldungen.
# - Parameter:
# - type: Art der Meldung (Warnungsmeldung ("W") oder Fehlermeldung
# - (sonst))
# - errormsg: Meldungstext
# - metadata: Hashtabelle, deren Einträge "error" oder "warning" um die
# - angegebene Meldung ergänzt wird.
# - Rückgabe:
# - -
# -----

sub metaerror {
    my($type,$errmsg,$metadata)=@_;

    # Initialisierung
    die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");

    # Fehler / Meldung merken
    if ($type =~ /W/i) {
        $metadata{"warning"} .= " " if ($metadata{"warning"});
        $metadata{"warning"} .= unescape($errmsg);
    } else {
        $metadata{"error"} .= " " if ($metadata{"error"});
        $metadata{"error"} .= unescape($errmsg);
    }
}

# -----
# - Routine:
# - composeOneDublinCoreMetatag
# - Beschreibung:
# - Setzt aus den gegebenen Informationen einen Dublin Core Metatag zusammen.
# - Parameter:
# - metaname: Name des DC-Elementes
# - scheme: Verwendetes Schema (oder leer)
# - entry: Wert
# - prefix: Präfix
# -----

sub composeOneDublinCoreMetatag {
    my ($metaname,$scheme,$entry,$prefix) = @_;
    my ($res,@entries);

    # Initialisierung
    $res = "";
    $entry = trim(unescape($entry));

    # Metatag erzeugen, falls Informationsrepräsentation vorhanden
    if ($entry) {
        $res .= "<META name=\"".$metaname."\" content=\"\"";
        $res .= "(SCHEME=".$scheme.") " if ($scheme);
        $res .= $prefix.$entry.">\n";
    }

    # Ergebnis zurückliefern
    return $res;
}

```

```

# -----
# - Routine:
# - dcmetag
# - Beschreibung:
# - Setzt aus den gegebenen Informationen den Dublin Core Metatag zusammen.
# - Parameter:
# - metaname: Name des DC-Elementes
# - scheme: verwendetes Schema (oder leer)
# - values: Wert / Werte
# - sep: Trennzeichenkette: Falls nicht leer, werden die Werte damit
# -         gesplittet und in jeweils einem Metatag erfasst
# - Rückgabe:
# - Liefert den gewünschten Dublin Core Metatag.
# -----

sub dcmetag {
my ($metaname,$scheme,$values,$prefix,$sep) = @_;
my ($res,$entry,@entries);

# Initialisierung
$res = "";

# Metatags erzeugen
if ($values) {
if ($sep) {
$sep = "\\$sep";
@entries = split($sep,$values);
foreach $entry(@entries) {
$res .= composeOneDublinCoreMetatag($metaname,$scheme,$entry,$prefix);
}
} else {
$res .= composeOneDublinCoreMetatag($metaname,$scheme,$values,$prefix);
}
}

# Ergebnis zurueckliefern
return $res;
}

# -----
# - Routine:
# - metaDCSubjectMSC
# - Beschreibung:
# - Liefert die Metadateninformationen für das Dublin Core Element
# - DC.Subject.Msc als Dublin Core Metatags. Hierbei wird vorher die
# - Korrektheit der Angaben geprüft (und ggf. (wo dies möglich ist)
# - korrigiert). Sollten Fehler auftreten, so werden diese in den Felder
# - warning bzw. error der übergebenen HASH-Tabelle aufgenommen.
# - Parameter:
# - department: Fachbereich / Fachgebiet
# - metadata: HASH-Tabelle, die alle Metadaten enthält
# - Rückgabe:
# - Liefert die gewünschten Dublin Core Metatags.
# -----

sub metaDCSubjectMSC {
my ($department,$metadata) = @_;
my ($res,$i,$msc,$text,$changed,@mscs,@mscinfo);

if ($metadata{"subject.msc"}) {

# MSC-Klassifikationsnummen überprüfen und ggf. korrigieren
@mscs = split('+',$metadata{"subject.msc"});
for ($i=0; $i <= $#mscs; $i++) {
@mscinfo = checkClassification_MSC($mscs[$i]);
$changed = $mscinfo[2];

# Wurde der Klassifikationscode geändert ?
if ($changed eq 1) {
$text = "Die eingegebene MSC konnte nicht gefunden werden";
$text .= " ("".$mscs[$i]."").";
if ($mscinfo[1]) {
$text .= " ". "Sie wurde geändert zu : ";
$text .= $mscinfo[1];
} else {
$text .= " ". "Sie wird bei der Speicherung nicht &uuml;bernommen.";
}
metaerror("W",$text,$metadata);
}
$mscs[$i]=$mscinfo[1];
}

# Metatag erzeugen und MSCText aufbereiten
$$metadata{"subject.msc.text"}="";
foreach $msc(@mscs) {
if ($msc) {
$res .= &dcmetag("DC.Subject.Msc","msc91",$msc,"","");
$$metadata{"subject.msc.text"} .= "<li>".$msc."</li>";
}
}
} else {
metaerror("W","Es &uuml;ber hilfreich eine Klassifikation nach der ".
"Mathematics Subject Classification (MSC91) ".
"durchzuf&uuml;hren.",$metadata);
}
}

```

```

# Ergebnis zurückliefern
return $res;
}

# -----
# - Routine:
# - checkDates
# - Beschreibung:
# - korrigiert die von KLUEDO veröffentlichten Datumsangaben
# - (Einbindungsdatum, Datum der letzten Änderung). Das Datum der letzten
# - Änderung wird auf das aktuelle Datum gesetzt. Falls noch kein
# - Einbindungsdatum vorhanden ist, so wird hierfür ebenfalls das aktuelle
# - Datum verwendet.
# - Parameter:
# - metadata: HASH-Tabelle, die alle Metadaten enthält
# - Rückgabe:
# - -
# -----

sub checkDates {
    my ($metadata) = @_;

    # Ist Datum der erstmaligen Einbindung vorhanden ?
    if (!$metadata{"date.created"}) {
        $metadata{"date.created"} = getactualDateInISO8601();
    }

    # Datum der letzten Änderung wird das aktuelle Datum
    $metadata{"date.modified"} = getactualDateInISO8601();
}

# -----
# - Routine:
# - metaDCCreatorPersonalname
# - Beschreibung:
# - Liefert die Metadateninformationen für das Dublin Core Element
# - DC.Creator.Personalname als Dublin Core Metatags. Hierbei wird vorher die
# - Korrektheit der Angaben geprüft (und ggf. (wo dies möglich ist)
# - korrigiert). Sollten Fehler auftreten, so werden diese in den Felder
# - warning bzw. error der übergebenen HASH-Tabelle aufgenommen.
# - Parameter:
# - department: Fachbereich / Fachgebiet
# - metadata: HASH-Tabelle, die alle Metadaten enthält
# - Rückgabe:
# - Liefert die gewünschten Dublin Core Metatags.
# -----

sub metaDCCreatorPersonalname {
    my ($department,$metadata) = @_;
    my ($res,$text,@creators,$i,$error);

    # Initialisierung
    $res = "";

    # Wurden überhaupt Autorennamen angegeben ?
    if ($metadata{"creator.personalname"}) {

        # Einzelname soweit wie möglich in verlangte Form bringen, wo dies nicht
        # möglich ist, wird eine Fehlermeldung ausgeben.
        @creators = split('\+', $metadata{"creator.personalname"});
        for ($i=0; $i <= $#creators ; $i++) {
            ($creators[$i],$error) = checkSyntaxOfName($creators[$i]);
            if ($error) {
                $text = "Bitte geben sie den Autor in der Form ";
                $text .= "\"Nachname, Vorname \" ein ";
                $text .= "(". $creators[$i]. ").";
                metaerror("W", $text, $metadata);
            }
        }

        # Namensauflistung neu zusammensetzen und Metatag übernehmen
        $metadata{"creator.personalname"} = "";
        for ($i=0; $i <= $#creators ; $i++) {
            $metadata{"creator.personalname"} .= "+" if ($i ne 0);
            $metadata{"creator.personalname"} .= $creators[$i];
        }
        $res .= &dcmetag("DC.Creator.PersonalName", "",
            $metadata{"creator.personalname"}, "", "+");
    } else {
        $text = "Sie müssen mindestens einen Autor angeben.";
        metaerror("E", $text, $metadata);
    }

    # Ergebnis zurückliefern
    return $res;
}

```

```

# -----
# - Routinen:
# - metaDCTitle
# - metaDCCreator
# - metaDCSubject
# - metaDCDescription
# - metaDCPublisher
# - metaDCContributor
# - metaDCDate
# - metaDCType
# - metaDCFormatIdentifizier
# - metaDCSource
# - metaDCLanguage
# - metaDCRelation
# - metaDCCoverage
# - metaDCRights
# - Beschreibung:
# - Liefert die Metadateninformationen für die entsprechende Dublin Core
# - Elemente als Dublin Core Metatags. Hierbei wird vorher die Existenz von
# - Daten (bei Pflichtfeldern) und die Korrektheit der Angaben geprüft
# - (und ggf. (wo dies möglich ist) korrigiert). Sollten Fehler auftreten,
# - so werden diese in den Felder warning bzw. error der übergebenen
# - HASH-Tabelle aufgenommen.
# - Parameter:
# - department: Fachbereich / Fachgebiet
# - metadata: HASH-Tabelle, die alle Metadaten enthält
# - Rückgabe:
# - Liefert die gewünschten Dublin Core Metatags.
# -----

sub metaDCTitle {
  my ($department,$metadata) = @_ ;
  my ($res);

  # Initialisierung
  die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
  $res = "";

  # Titel (1)
  $res .= &dcmetatag("DC.Title","",$$metadata{"title"},"","");
  if (!$$metadata{"title"}) {
    metaerror("E","Sie müssen einen Titel eingeben.",$metadata);
  }

  # Ergebnis zurückliefern
  return $res;
}

sub metaDCCreator {
  my ($department,$metadata) = @_ ;
  my ($res,$text);

  # Initialisierung
  die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
  $res = "";

  # Autor (*)
  $res .= metaDCCreatorPersonalname($department,$metadata);

  # Postalische Adresse (n)
  $res .= &dcmetatag("DC.Creator.PersonalName.Address","",
    $$metadata{"creator.postal"},"(Postal)","");

  # Telefonnummer (1)
  $res .= &dcmetatag("DC.Creator.PersonalName.Address","",
    $$metadata{"creator.phone"},"(Phone)","");

  # Faxnummer (1)
  $res .= &dcmetatag("DC.Creator.PersonalName.Address","",
    $$metadata{"creator.fax"},"(Fax)","");

  # email-Adresse (*)
  $res .= &dcmetatag("DC.Creator.PersonalName.Address","",
    $$metadata{"creator.email"},"(Email)","");
  if (!$$metadata{"creator.email"}) {
    metaerror("W","Es wäre hilfreich eine email-Adresse ".
      "des Autors anzugeben.",$metadata);
  }

  # Institut (*)
  $res .= &dcmetatag("DC.Creator.PersonalName.Address","",
    $$metadata{"creator.affiliation"},"(Affiliation)","");

  # Persönliche Schlagwörter (*)
  $res .= &dcmetatag("DC.Creator.PersonalName.Address","",
    $$metadata{"creator.keywords"},"(Keywords)","");

  # Position (1) (Spezialfall von Keywords)
  $res .= &dcmetatag("DC.Creator.PersonalName.Address","",
    $$metadata{"creator.position"},"(Keywords)","");

  # besondere Funktionen (1) (Spezialfall von Keywords)
  $res .= &dcmetatag("DC.Creator.PersonalName.Address","",
    $$metadata{"creator.task"},"(Keywords)","");

  # Homepage (1)
  $res .= &dcmetatag("DC.Creator.PersonalName.Address","",
    $$metadata{"creator.homepage"},"(Homepage)","");
}

```



```

# Ergebnis zurueckliefern
return $res;
}

sub metaDCSubject {
my ($department,$metadata) = @_;
my ($res);

# Initialisierung
die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
$res = "";

# Schlüsselwörter (*)
$res .= &dcmetatag("DC.Subject","",$$metadata{"subject.keywords"},"","");
if (!$$metadata{"subject.keywords"}) {
metaerror("W","Es wäre hilfreich ".
"Schlüsselwörter anzugeben.", $metadata);
}

# MathNet (*)
$res .= &dcmetatag("DC.Subject","MathNet",$$metadata{"subject.mathnet"},
"","","");

# Topic (*)
$res .= &dcmetatag("DC.Subject.Topic","",$$metadata{"subject.topic"},"","");

# Klassifikation: MSC91 (*)
if ($department eq "Mathematik") {
$res .= metaDCSubjectMSC($department,$metadata);
}

# Ergebnis zurückliefern
return $res;
}

sub metaDCDescription {
my ($department,$metadata) = @_;
my ($res);

# Initialisierung
die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
$res = "";

# Abstract (1)
if ($$metadata{"description.abstract"}) {
$res .= &dcmetatag("DC.Description.Abstract", "url",
"#abstract", "", "");
$res .= &dcmetatag("DC.Description.Abstract","",
unescape($$metadata{"description.abstract"},"",""));
}
if (!$$metadata{"description.abstract"}) {
metaerror("W","Es wäre hilfreich eine Textzusammenfassung ".
"anzugeben.", $metadata);
}

# Notes (1)
if ($$metadata{"description.notes"}) {
$res .= &dcmetatag("DC.Description.Notes","",
unescape($$metadata{"description.notes"},"",""));
}

# Ergebnis zurückliefern
return $res;
}

sub metaDCPublisher {
my ($department,$metadata) = @_;
my ($res);

# Initialisierung
die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
$res = "";

# Publisher (1)
$res .= &dcmetatag("DC.Publisher","",
$$metadata{"publisher"},"","");

# CorporateName (1)
$res .= &dcmetatag("DC.Publisher.CorporateName","",
$$metadata{"publisher.corporatename"},"","");

# Email (1)
$res .= &dcmetatag("DC.Publisher.CorporateName.Address","",
$$metadata{"publisher.corporatename.email"},"(Email)","");

# Homepage (1)
$res .= &dcmetatag("DC.Publisher.CorporateName.Address","",
$$metadata{"publisher.corporatename.homepage"},
"(Homepage)","");

# Name (*) (Personenname)
$res .= &dcmetatag("DC.Publisher.PersonalName","",
$$metadata{"publisher.personalname"},"","");
}

```

```

# Post Adresse (*)
$res .= &dcmetatag("DC.Publisher.Address","",
    $$metadata{"publisher.postal"},"(Postal)","");

# Telefonnummer (1)
$res .= &dcmetatag("DC.Publisher.Address","",
    $$metadata{"publisher.phone"},"(Phone)","");

# Faxnummer (1)
$res .= &dcmetatag("DC.Publisher.Address","",
    $$metadata{"publisher.fax"},"(Fax)","");

# Email (*)
$res .= &dcmetatag("DC.Publisher.Address","",
    $$metadata{"publisher.email"},"(Email)","");

# Organisation (*)
$res .= &dcmetatag("DC.Publisher.Address","",
    $$metadata{"publisher.affiliation"},"(Affiliation)","");

# Publisher Keywords (1)
$res .= &dcmetatag("DC.Publisher.Address","",
    $$metadata{"publisher.keywords"},"(Keywords)","");

# Homepage
$res .= &dcmetatag("DC.Publisher.Address","",
    $$metadata{"publisher.homepage"},"(Homepage)","");

# Ergebnis zurückliefern
return $res;
}

sub metaDCContributor {
my ($department,$metadata) = @_;
my ($res);

# Initialisierung
die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
$res = "";

# PersonalName (*)
$res .= &dcmetatag("DC.Contributor.PersonalName","",
    $$metadata{"contributor.personalname"},"","");

# postalische Adresse (*)
$res .= &dcmetatag("DC.Contributor.PersonalName.Adress","",
    $$metadata{"contributor.postal"},"(Postal)","");

# Telefonnummer (1)
$res .= &dcmetatag("DC.Contributor.PersonalName.Adress","",
    $$metadata{"contributor.phone"},"(Phone)","");

# Faxnummer (1)
$res .= &dcmetatag("DC.Contributor.PersonalName.Adress","",
    $$metadata{"contributor.fax"},"(Fax)","");

# Email (*)
$res .= &dcmetatag("DC.Contributor.PersonalName.Adress","",
    $$metadata{"contributor.email"},"(Email)","");

# Organisation (*)
$res .= &dcmetatag("DC.Contributor.PersonalName.Adress","",
    $$metadata{"contributor.affiliation"},
    "(Affiliation)","");

# Contributor Keywords (1)
$res .= &dcmetatag("DC.Contributor.PersonalName.Adress","",
    $$metadata{"contributor.keywords"},"(Keywords)","");

# Homepage
$res .= &dcmetatag("DC.Contributor.PersonalName.Adress","",
    $$metadata{"contributor.homepage"},"(Homepage)","");

# Ergebnis zurückliefern
return $res;
}

sub metaDCDate {
my ($department,$metadata) = @_;
my ($res);

# Initialisierung
die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
$res = "";

# Datum der Aufnahme in Kluedo (1)
$res .= &dcmetatag("DC.Date.Created","", $$metadata{"date.created"},"","");

# Datum der letzten Änderung (1)
$res .= &dcmetatag("DC.Date","", $$metadata{"date.modified"},"","");

# Entstehungsjahr des intellektuellen Inhalts
$res .= &dcmetatag("DC.Date.Creation_of_intellectual_content","",
    $$metadata{"date.intellectualcreation.year"},"","");

```

```
# Ergebnis zurückliefern
return $res;
}

sub metaDCType {
  my ($department,$metadata) = @_;
  my ($res);

  # Initialisierung
  die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
  $res = "";

  # Type übernehmen
  $res .= &dcmetatag("DC.Type","",$$metadata{"type"},"","");

  # Ergebnis zurückliefern
  return $res;
}

sub metaDCFormatIdentifizier {
  my ($department,$metadata) = @_;
  my ($i,$res,@sources,@formats);

  # Initialisierung
  die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
  $res = "";

  # Quellen (URL) und deren Formate
  @sources = split('\+',$$metadata{"identifizier.url"});
  @formats = split('\+',$$metadata{"format"});
  if ($#sources == $#formats) {
    for ($i=0; $i <= $#sources; $i++) {
      if ($i <= $#formats) {
        $res .= &dcmetatag("DC.Identifizier","url",$sources[$i],"","");
        $res .= &dcmetatag("DC.Format","IMT",$formats[$i],"","");
      }
    }
  } else {
    metaerror("E","Die Anzahl der Dateiformate stimmt nicht mit der Anzahl ".
      "der Quellformate überein (".$$metadata{"identifizier.url"}."
      ").",$metadata);
  }

  # Dokumentreferenz
  $res .= &dcmetatag("DC.Identifizier","",$$metadata{"identifizier"},"","");

  # Sonderfeld für Kaiserslautern: Angabe ob Daten an den SWB gemeldet wurden
  # (enthält idn-Nummer des SWB)
  $res .= &dcmetatag("DC.Identifizier","idn",$metadata{"idn"},"","");

  # Ergebnis zurückliefern
  return $res;
}

sub metaDCSource {
  my ($department,$metadata) = @_;
  my ($res);

  # Initialisierung
  die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
  $res = "";

  # Series
  $res .= &dcmetatag("DC.Source","",$$metadata{"source"},"","");

  # Source-Identifikation
  $res .= &dcmetatag("DC.Source.Identifizier","",$$metadata{"source.identifizier"},
    "", "");

  # Ergebnis zurückliefern
  return $res;
}

sub metaDCLanguage {
  my ($department,$metadata) = @_;
  my ($res);

  # Initialisierung
  die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
  $res = "";

  # Sprache
  $res .= &dcmetatag("DC.Language","Z39.53",$$metadata{"language"},"","");

  # Ergebnis zurückliefern
  return $res;
}
```

```

sub metaDCRelation {
  my ($department,$metadata) = @_ ;
  my ($res);

  # Initialisierung
  die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
  $res = "";

  # Ergebnis zurückliefern
  return $res;
}

sub metaDCCoverage {
  my ($department,$metadata) = @_ ;
  my ($res);

  # Initialisierung
  die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
  $res = "";

  # Ergebnis zurückliefern
  return $res;
}

sub metaDCRights {
  my ($department,$metadata) = @_ ;
  my ($res);

  # Initialisierung
  die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
  $res = "";

  $res .= &dcmetatag("DC.Rights","",$$metadata{"rights"},"","");

  # Ergebnis zurückliefern
  return $res;
}

# -----
# - Routine:
# - metaSchemas
# - Beschreibung:
# - Diese Routine ermittelt aus den übergebenen Metatags die verwendeten
# - Schemata und fügt die Referenzlinks hinzu.
# - Parameter:
# - metadatastr: Zeichenkette mit den erzeugten Dublin Core Metatags
# - Rückgabe:
# - Liefert den (um die Schemareferenz) ergänzten String zurück.
# -----

sub metaSchemas {
  my ($metadataastr) = @_ ;
  my ($res);

  # Initialisierung
  $res = "";

  # Dublin Core
  $res .= "<LINK REL=\"SCHEMA.dc\" HREF=\"\"";
  $res .= "http://purl.org/metadata/dublin_core_elements\">\n";

  # MathNet
  if ($metadataastr =~ /(SCHEME=MathNet)/) {
    $res .= "<LINK REL=\"SCHEMA.MathNet\" HREF=\"\"";
    $res .= "http://www.mathematik.uni-osnabrueck.de/ak-technik/";
    $res .= "MathNet/BasicMeta.html\">\n";
  }

  # MSC
  if ($metadataastr =~ /(SCHEME=msc91)/) {
    $res .= "<LINK REL=\"SCHEMA.msc91\" HREF=\"\"";
    $res .= "http://www.ams.org/msc/\">\n";
  }

  # Z39.53
  if ($metadataastr =~ /(SCHEME=Z39.53)/) {
    $res .= "<LINK REL=\"SCHEMA.Z39.53\" HREF=\"\"";
    $res .= "http://www.sil.org/sgml/nisoLang3-1994.html\">\n";
  }

  # Ergebnis zurückliefern
  return $res;
}

```

```

# -----
# - Routine:
# - generateMetatags
# - Beschreibung:
# - Liefert die Metadateninformation für die spezifizierten Dublin Core
# - Elemente als Dublin Core Metatags. Hierbei wird vorher die Existenz von
# - Daten (bei Pflichtfeldern) und die Korrektheit der Angaben geprüft (und
# - ggf. (wo dies möglich ist) korrigiert). Sollten Fehler auftreten, so
# - werden diese in den Felder warning bzw. error der übergebenen
# - HASH-Tabelle aufgenommen.
# - Parameter:
# - department: Fachbereich / Fachgebiet
# - metadata: HASH-Tabelle, die alle Metadaten enthält
# - Rückgabe:
# - Liefert die gewünschten Dublin Core Metatags.
# -----

sub generateMetatags {
  my ($department,$metadata) = @_;
  my ($res,$key,$value);

  # Initialisierung
  die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");

  # Aktuelle Datumswerte setzen
  checkDates($metadata);

  # Alle führenden und nachfolgenden Leerzeichen entfernen
  while (($key,$value)=each %$metadata) {
    $$metadata{$key} = trim($value);
  }

  # Kennzeichen von KLUEDO
  $res = "<!-- Metamaker Version KLUEDO 0.9 -->\n";

  # Beginn der Resourcebeschreibung via Metadaten
  $res .= "<!-- Meta -->\n";

  # Dublin Core Elemente
  $res .= metaDCTitle($department,$metadata);
  $res .= metaDCCreator($department,$metadata);
  $res .= metaDCSubject($department,$metadata);
  $res .= metaDCDescription($department,$metadata);
  $res .= metaDCPublisher($department,$metadata);
  $res .= metaDCContributor($department,$metadata);
  $res .= metaDCDate($department,$metadata);
  $res .= metaDCType($department,$metadata);
  $res .= metaDCFormatIdentifier($department,$metadata);
  $res .= metaDCSource($department,$metadata);
  $res .= metaDCLanguage($department,$metadata);
  $res .= metaDCRelation($department,$metadata);
  $res .= metaDCCoverage($department,$metadata);
  $res .= metaDCRights($department,$metadata);

  # Links zu den verwendeten Schematas
  $res .= metaSchemas($res);

  # Ende der Resourcebeschreibung via Metadaten
  $res .= "<!-- /Meta -->\n";

  # Ergebnis zurückliefern
  return $res;
}

# -----
# - Routine:
# - generateInformationStr
# - Beschreibung:
# - Generiert einen String zur Anzeige der (für den Benutzer relevanten)
# - Metadaten in einem HTML-Dokument.
# - Parameter:
# - department: Fachbereich / Fachgebiet
# - metadata: HASH-Tabelle, die alle Metadaten enthält
# - Rückgabe:
# - Liefert die gewünschte Zeichenkette.
# -----

sub generateInformationStr {
  my ($department, $metadata) = @_;
  my ($i, $res, $temp,$line,$name, @authors, @emails,$formats);

  # Initialisierung
  die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
  $res = "";

  # Kennzeichnung dieses Abschnittes (Beginn)
  $res .= "<!-- Data -->\n";

  ## Ausgabe Titel
  $res .= "<h2>";
  $res .= unescape($$metadata{"title"});
  $res .= "&nbsp;</h2>\n";

  ## Ausgabe der Autoren mit assoziierter EMAIL
  @authors = split('\+',$$metadata{"creator.personalname"});
  @emails = split('\+',$$metadata{"creator.email"});

```

```

for ($i=0; $i <= $#authors; $i++) {
  if ($#authors == $#emails) {
    $res .= "<a href=\"mailto:\"";
    $res .= unescape($emails[$i]).\">";
    $res .= unescape($authors[$i]);
    $res .= "</a><br>\n";
  } else {
    $res .= unescape($authors[$i]).\"<br>\n";
  }
}
$res .= "<br>\n";

## Ausgabe Reihe
if ($$metadata{"source"}) {
  $res .= unescape($$metadata{"source"});
  if ($$metadata{"source.identifizier"}) {
    $res .= "; ".$$metadata{"source.identifizier"}
  }
  $res .= "<br><br>\n";
}

## Ausgabe des Entstehungsjahres
if ($$metadata{"date.intellectualcreation.year"}) {
  $res .= "<i><b>";
  $res .= "Entstehungsjahr des intellektuellen Inhalts";
  $res .= " :</b></i> ";
  $res .= unescape($$metadata{"date.intellectualcreation.year"});
  $res .= "<br><br>\n";
}

## Ausgabe Abstract
if ($$metadata{"description.abstract"}) {
  $res .= "<a name=\"abstract\"></a><i><b>";
  $res .= "Zusammenfassung";
  $res .= " :</b></i><br>\n";
  $res .= viewSpecialHTMLConstructs(
    unescape($$metadata{"description.abstract"})).\"\\n";
  $res .= "<br><br>\n";
}

# Ausgabe keywords
if ($$metadata{"subject.keywords"}) {
  $temp = unescape($$metadata{"subject.keywords"});
  $temp =~ s/\+//ig;
  $res .= "<i><b>";
  $res .= "Schlagw&ouml;rter";
  $res .= " :</b></i> ".$temp."<br><br>\n";
}

## Ausgabe MSC-Klassifikation (Fachbereich Mathematik)
if ($department eq "Mathematik") {
  if ($$metadata{"subject.msc"}) {
    $temp = metaDCSubjectMSC($department,$metadata);
    $res .= "<i><b>";
    $res .= "MSC-Klassifikation";
    $res .= " :</b></i> ";
    $res .= $$metadata{"subject.msc.text"};
    $res .= "<br><br>\n";
  }
}

# Ausgabe Ver&ouml;ffentlichungsreferenz
if ($$metadata{"identifizier"}) {
  $res .= "<i><b>";
  $res .= "Ver&ouml;ffentlichungsreferenz";
  $res .= " :</b></i> ";
  $res .= $$metadata{"identifizier"}."<br><br>\n";
}

# Ausgabe Notes
if ($$metadata{"description.notes"}) {
  $res .= "<!--Notes-->\n";
  $res .= "<a name=\"notes\"></a><i><b>";
  $res .= "Bemerkungen";
  $res .= " :</b></i><br>\n";
  $res .= viewSpecialHTMLConstructs(
    unescape($$metadata{"description.notes"})).\"\\n";
  $res .= "<!--/Notes-->\n";
  $res .= "<br><br>\n";
}

# Ausgabe der Ver&ouml;ffentlichungsdaten bez&uuglich KLUEDO
$res .= "<i><b>";
$res .= "Ver&ouml;ffentlichungsdaten";
$res .= " :</b> ";
$res .= "bez&uuglich KLUEDO";
$res .= " :</i>";
$res .= $$metadata{"date.created"}." (" ;
$res .= "Aufnahme";
$res .= ")<br><br>\n";

# Ausgabe der ver&uugbaren Dateiformate
$formats = getFileformats($metadata);
if ($formats) {
  $res .= "<i><b>";
  $res .= "Verf&uugbare Formate";
  $res .= " :</b></i><br>";
  $res .= $formats;
} else {

```

```

    $res .= "Das Dokument ist nicht in elektronischer Form vorhanden.";
    $res .= "<br>";
}

# Kennzeichnung des Abschnittendes
$res .= "<!-- /Data -->\n";

# Ergebnis zurückliefern
return $res;
}

# -----
# - Routine:
# - testMetadata
# - Beschreibung:
# - Die Existenz von Daten (bei Pflichtfeldern) und die Korrektheit der
# - Angaben werden geprüft (und ggf. (wo dies möglich ist) korrigiert).
# - Sollten Fehler auftreten, so werden diese in den Felder warning bzw.
# - error der übergebenen HASH-Tabelle aufgenommen Im Anschluß wird ein
# - String zur Anzeige der (für den Benutzer relevanten) Metadaten in einem
# - HTML-Dokument generiert und dann zurückgegeben.
# - Parameter:
# - department: Fachbereich / Fachgebiet
# - metadata: HASH-Tabelle, die alle Metadaten enthält
# - Rückgabe:
# - Liefert die gewünschte Zeichenkette.
# -----

sub testMetadata {
    my ($department, $metadata) = @_;
    my ($temp);

    # Initialisierung
    die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");

    # Prüfungen und ggf. Korrekturen durchführen
    $temp = generateMetatags($department,$metadata);

    # Datumskorrekturen
    checkDates($metadata);

    # Ergebnis zurückliefern
    return generateInformationStr($department,$metadata);
}

# -----
# - Routine:
# - generateMetadataStr
# - Beschreibung:
# - Liefert den String für das kompletten Metadatenfile (HTML-Umgebung,
# - Metadaten und die für den Benutzer lesbaren Informationen)
# - Parameter:
# - department: Fachbereich / Fachgebiet
# - metadata: HASH-Tabelle, die alle Metadaten enthält
# - Rückgabe:
# - Liefert die gewünschte Zeichenkette.
# -----

sub generateMetadataStr {
    my ($department, $metadata) = @_;
    my ($res);

    # Initialisierung
    die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
    $res = "";

    # Headpart, PHP-Einbettung
    $res .= "<?php\n";
    $res .= "require(\"/opt/metadaten/KLUEDO/php/init.pinc\");\n";
    $res .= "do_header1(\"Metadaten &uuml;r: \";
    $res .= unescape($$metadata{"title"});
    $res .= "\");\n";
    $res .= "?>\n";

    # Metatags
    $res .= generateMetatags($department,$metadata);

    # MID-Part der PHP-Einbettung
    $res .= "<?php\n";
    $res .= "do_header2();\n";
    $res .= "do_body(\"\\");\n";
    $res .= "?>\n";

    # für den Benutzer lesbare Informationen
    $res .= generateInformationStr($department,$metadata);

    # Tail-Part der PHP-Einbettung
    $res .= "<?php\n";
    $res .= "do_footer();\n";
    $res .= "?>\n";

    # Ergebnis zurückliefern

```

```

return $res;
}

# -----
# - Routine:
# - getDCSourceStr
# - Beschreibung:
# - Liefert den (für das Dublin Core Element DC.Source) zu verwendenden Wert.
# - Parameter:
# - department: Fachbereich / Fachgebiet
# - serie: Reihe in der die Ressource veröffentlicht werden soll.
# - Rückgabe:
# - Liefert den gewünschten Wert.
# -----

sub getDCSourceStr {
my ($department,$serie,) = @_;
my ($res,$listpath,$noerror,$CONF_FILE,$line,$fserie,$parameter);
my (@fields,@fields2,%env);

# Initialisierung
$res = "";
$noerror = "true";
%env = Environment();
$listpath = $env{DEPTS}."/department/conf/Listen.conf";

# Konfigurationsdatei scannen
$CONF_FILE = IO::File->new("$listpath") or $noerror="";
while (($noerror) && (defined($line = $CONF_FILE->getline()))) {
@fields = split('\|', $line);
$fserie = $fields[0];

# Stimmt die Reihe mit der übergebenen Reihe überein ?
if ($serie eq $fserie) {
$parameter = $fields[1];chomp($parameter);
@fields2 = split(":",$parameter);
$res = " $fields2[1], ";
$res .= "Fachbereich $department, Universit&auml;t Kaiserslautern";
}
}
$CONF_FILE->close() if ($noerror);

# Ergebnis zurückliefern
return $res;
}

# -----
# - Routine:
# - getFileformats
# - Beschreibung:
# - Liefert (anhand der Metadaten) die Angaben zu den vorliegenden Dokument-
# - formaten (wie beispielsweise Format, Größe und URL)
# - Parameter:
# - metadata: HASH-Tabelle, die alle Metadaten enthält
# - Rückgabe:
# - Liefert die gewünschte HTML-Zeichenkette.
# -----

sub getFileformats {
my($metadata)=@_;
my($i,$res,$source,$format,$img,$size,$filename,$bytes,$kbytes);
my(@sources,@formats,@statistic,%env);

# Initialisierung
%env = Environment();
$res="";
$source = $$metadata{"identifizier.url"};
$format = $$metadata{"format"};
if ($source) {
@sources = split('\+', $source);
@formats = split('\+', $format);
} else {
undef @sources;
undef @formats;
}

# Ausgabe der Formate
if ($#sources ge 0) {
for ($i=0; $i <= $#sources; $i++) {
if ($i <= $#formats) {

# Grafik bestimmen
$img = "";
$formats[$i]=lc(trim($formats[$i]));
if ($formats[$i] eq "application/postscript") {
$img = "pgz.gif";
$format = "Postscript (gzip't)";
} elsif ($formats[$i] eq "application/pdf") {
$img = "pdf.gif";
$format = "PDF-Datei";
} elsif ($formats[$i] eq "text/html") {
$img = "html.gif";
$format = "HTML";
}
}
}
}

```



```
} else {
    $img = "unknown.gif";
    $format = $formats[$i];
}

# Für lokale Dateien soll die Größe angezeigt werden
$size="";
$filename = trim($sources[$i]);
if ($filename =~ /^http:\/\/\kluedo.ub.uni-kl.de\/i)
{
    $filename =~ s/^http:\/\/\kluedo.ub.uni-kl.de\/$env{DEPTS}\//;
    if (-e $filename) {

        # Datei existiert, also: Informationen abfragen
        $size = "&nbsp;(";
        @statistic = stat $filename;
        $bytes = $statistic[7];
        if ($bytes < 1024) {
            $size .= sprintf("%.0f Byte", $bytes);
        } else {
            $kbytes = $bytes/1024;
            if ($kbytes < 1024) {
                $size .= sprintf("%.0f kB", $kbytes);
            } else {
                $size .= sprintf("%.1f MB", $kbytes/1024);
            }
        }
        $size .= ")";
    }
}

# Link-Tag zusammensetzen
$res .= "<a href=\"". $sources[$i]. "\">";
$res .= "<img border=0 width=16 height=16 ";
$res .= "src=\"/images/filetypes/$img\">";
$res .= "&nbsp; $format";
$res .= "</a>$size<br>";
}
}

# Ergebnis zurückliefern
return $res;
}

# -----
1; #return true
```

## D. Bibliothek 'ReadMetafile.pm'

```
#####
#
# Quelltext:
# (KLUEDO)/lib/KLUEDO/ReadMetafile.pm
#
# Beschreibung:
# Die Bibliothek ReadMetafile.pm stellt eine effiziente Routine zum Einlesen
# der Metadaten aus einer Datei zur Verfügung.
#
# Autor:
# Wolfgang Eiden, Universität Kaiserslautern
#
#####

use strict;

use lib('/opt/metadaten/KLUEDO/lib');

use IO::File;
use KLUEDO::Conversions;

# -----
# - Routine:
# - readMetadaten
# - Beschreibung:
# - liest die Metadaten aus der übergebenen Datei (im HTML-Format) aus und
# - übergibt diese in einer Hashtabelle.
# - Parameter:
# - shadowfile: Datei, die die Metatags enthält.
# - Rückgabe:
# - gewünschte Hashtabelle
# - Bemerkung:
# - Es wird vorausgesetzt, dass jeder Metatag in einer einzelnen Zeile steht
# - und von der Form
# - <META name="metaname" content="[(typ) ]value">
# - ist.
# -----

sub readMetadaten {
    my($shadowfile) = @_ ;
    my($FILE,$line,$metatag,$metaname,$content,$value,$typ,$metadaten);

    # Initialisierung
    $metatag = '\W*<META\s+name="(.*?)"\s+content="(.*?)"\s*\W*';

    # Datei parsen
    $FILE = IO::File->new("< $shadowfile");
    while (defined($line = $FILE->getline())) {
        chomp($line);

        # Handelt es sich überhaupt um ein Metatag ?
        if ($line =~ /$metatag/i) {
            $metaname = trim($1);
            $content = trim($2);

            # Ist eine Typangabe vorhanden ?
            if ($content =~ /^((.*?)\s*)/) {
                $typ = $1;
                $value = trim($');
            } else {
                $typ = "";
                $value = $content;
            }

            # Metadaten in Hashtabelle aufnehmen
            pushMetatag(\%metadaten,$metaname,$typ,$value);
        }
    }
    $FILE->close();

    # Ergebnis zurückliefern
    return \%metadaten;
}

# -----
# - Routine:
# - pushMetatag
# - Beschreibung:
# - Übernimmt die Daten des Metatags in die Hashtabelle. Hierbei werden die
# - Einträge, die ein bekanntes Schema oder Typ angeben, in die
# - spezifizierten Stellen des HASH-Tabelle aufgenommen. Von unbekannte
# - Schemata und Typs werden nur die Werte gespeichert.
# - Parameter:
# - metadaten: HASH-Tabelle, die die Metadaten enthält
# - metaname: Elementname des Metatags
# - typ: Elementtyp (Schema oder ergänzende Angaben)
# - value: Wert
# - Rückgabe:
# - -
# -----
```

```

sub pushMetatag {
  my($metadata,$metaname,$typ,$value) = @_ ;

  # Initialisierung
  die "pushMetatag: Hashreferenz erwartet. " if (ref($metadata) ne "HASH");

  # Werte je nach Inhalt/Schema in das entsprechende Hashfeld speichern
  # Teste dabei zuerst die spezialisierten Felder
  SWITCH: {

    # DC.Title (01)
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.title","lang=de","", "title.de"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.title","lang=en","", "title.en"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.title","", "", "title"))
      {last SWITCH};

    # DC.Creator (02)
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.creator.personalname.address","postal","+",
      "creator.postal"))
      {last SWITCH};

    if (provetag($metadata,$metaname,$typ,$value,
      "dc.creator.personalname.address","phone","", "creator.phone"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.creator.personalname.address","fax","", "creator.fax"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.creator.personalname.address","email","+",
      "creator.email"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.creator.personalname.address","affiliation","+",
      "creator.affiliation"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.creator.personalname.address","keywords","+",
      "creator.keywords"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.creator.personalname.address","homepage","",
      "creator.homepage"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.creator.personalname","", "+", "creator.personalname"))
      {last SWITCH};

    # DC.Subject (03)
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.subject","scheme=mathnet","", "subject.mathnet"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.subject.topic","", "", "subject.topic"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.subject.msc","scheme=msc91","+","subject.msc"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.subject","", "+", "subject.keywords"))
      {last SWITCH};

    # DC.Description (04)
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.description.abstract","scheme=url","", ""))
      {last SWITCH}; # Feld enthält keine benötigte Informationen
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.description.abstract","", "", "description.abstract"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.description.notes","scheme=url","", ""))
      {last SWITCH}; # Feld enthält keine benötigte Informationen
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.description.notes","", "", "description.notes"))
      {last SWITCH};

    # DC.Publisher (05)
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.publisher.address","phone","+","publisher.phone"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.publisher.address","fax","+","publisher.fax"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.publisher.address","email","+","publisher.email"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.publisher.address","affiliation","+",
      "publisher.affiliation"))
      {last SWITCH};
    if (provetag($metadata,$metaname,$typ,$value,
      "dc.publisher.address","keywords","", "publisher.keywords"))
      {last SWITCH};
  }
}

```

```

if (provetag($metadata,$metaname,$typ,$value,
  "dc.publisher.address","homepage","", "publisher.homepage"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.publisher.personalname","", "publisher.personalname"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.publisher.corporatename.address","homepage","",
  "publisher.corporatename.homepage"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.publisher.corporatename.address","email","",
  "publisher.corporatename.email"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.publisher.corporatename","", "publisher.corporatename"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.publisher","", "publisher"))
{last SWITCH};

# DC.Contributor (06)
if (provetag($metadata,$metaname,$typ,$value,
  "dc.contributor.personalname.address","postal","",
  "contributor.postal"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.contributor.personalname.address","phone","",
  "contributor.phone"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.contributor.personalname.address","fax","",
  "contributor.fax"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.contributor.personalname.address","email","",
  "contributor.email"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.contributor.personalname.address","affiliation","",
  "contributor.affiliation"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.contributor.personalname.address","keywords","",
  "contributor.keywords"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.contributor.personalname.address","homepage","",
  "contributor.homepage"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.contributor.personalname.address","", "contributor.personalname"))
{last SWITCH};

# DC.Date (07)
if (provetag($metadata,$metaname,$typ,$value,
  "dc.date.created","", "date.created"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.date","", "date.modified"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.date.creation_of_intellectual_content","",
  "date.intellectualcreation.year"))
{last SWITCH};

# DC.Type (08)
if (provetag($metadata,$metaname,$typ,$value,
  "dc.type","", "type"))
{last SWITCH};

# DC.Format (09)
if (provetag($metadata,$metaname,$typ,$value,
  "dc.format","scheme=imt","", "format"))
{last SWITCH};

# DC.Identifier (10)
if (provetag($metadata,$metaname,$typ,$value,
  "dc.identifier","scheme=url","", "identifier.url"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.identifier","scheme=idn","", "idn"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.identifier","", "identifier"))
{last SWITCH};

# DC.Source (11)
if (provetag($metadata,$metaname,$typ,$value,
  "dc.source.identifier","", "source.identifier"))
{last SWITCH};
if (provetag($metadata,$metaname,$typ,$value,
  "dc.source","", "source"))
{last SWITCH};

```

```

# DC.Language (12)
if (provetag($metadata,$metaname,$typ,$value,
  "dc.language","scheme=z39.53","", "language"))
  {last SWITCH};

# DC.Relation (13)
# wird in Universität Kaiserslautern zur Zeit nicht verwendet

# DC.Coverage (14)
# wird in Universität Kaiserslautern zur Zeit nicht verwendet

# DC.Rights (15)
if (provetag($metadata,$metaname,$typ,$value,
  "dc.rights","", "rights"))
  {last SWITCH};
}
}

# -----
# - Routine:
# - provetag
# - Beschreibung:
# - Prüft ob der erwartete Elementtyp und das erwartete Schema mit dem
# - übergebenen übereinstimmt und übernimmt diesen dann ggf. in die
# - Hashtabelle.
# - Parameter:
# - metadata: HASH-Tabelle, die die Metadaten enthält
# - metaname: Elementname des Metatags
# - typ: Elementtyp (Schema oder ergänzende Angabe)
# - value: Wert
# - reqname: Erwarteter Elementname (Groß- /Kleinschreibung ist nicht von
# - Relevanz)
# - reqtyp: Erwarteter Elementtyp (leer, falls kein spezielles Typ
# - verlangt wird)
# - seperator: Seperator, der bei Mehrfachangabe verwendet werden soll
# - hashkey: HASH-Schlüssel (interner Name des Elementnamens)
# - Rückgabe:
# - Liefert "true", wenn der Wert in die Hashtabelle aufgenommen wurde.
# -----

sub provetag {
my($metadata,$metaname,$typ,$value,$reqname,$reqtyp,$separator,
  $hashkey) = @_;
my($match);

# Initialisierung
die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");
$metaname= lc(trim($metaname));
$typ=lc(trim($typ));
$value=trim($value);
$reqname = lc(trim($reqname));
$reqtyp = lc(trim($reqtyp));
$match = "";

# Handelt es sich überhaupt um den richtigen Metanamen ?
if ($metaname eq $reqname) {

  # Wird ein bestimmter Inhalt / Schema verlangt ?
  if ($reqtyp) {

    # Ja, dann nur bei exakter Übereinstimmung übernehmen !
    $match = "true" if ($typ eq $reqtyp);
  } else {
    $match = "true";
  }
}

# Falls Überprüfung erfolgreich war, Wert übernehmen (falls nicht leer)
if (($match) && ($value) && ($hashkey)) {
  adoptMetadata($metadata,$hashkey,$value,$separator);
}

# Ergebnis zurückgeben
return $match;
}

# -----
# - Routine:
# - adoptMetadata
# - Beschreibung:
# - Prüft den übergebenen Metadatenwert auf Korrektheit und nimmt diesen in
# - der HASH-Tabelle auf.
# - Parameter:
# - metadata: HASH-Tabelle, die die Metadaten enthält
# - hashkey: HASH-Schlüssel
# - value: Wert
# - seperator: Seperator, der bei Mehrfachangabe verwendet werden soll
# - Rückgabe:
# - -
# -----

```

```
sub adoptMetadata {
  my($metadata,$hashkey,$value,$separator) = @_;

  # Initialisierung
  die "Hashreferenz erwartet. " if (ref($metadata) ne "HASH");

  # Vor Übernahme Syntaxprüfungen und ggf. Manipulationen vornehmen ...
  $value=checkMetadataSyntax($hashkey,$value);

  # ... und dann erst übernehmen
  if ($$metadata{"$hashkey"}) {
    $$metadata{"$hashkey"} .= $separator.$value;
  } else {
    $$metadata{"$hashkey"} = $value;
  }
}

# -----
# - Routine:
# -   checkMetadataSyntax
# - Beschreibung
# -   Prüft den übergebenen Metadatenwert auf Korrektheit und liefert die
# -   korrigierte Fassung.
# - Parameter:
# -   hashkey: HASH-Schlüssel
# -   value:   Wert
# - Rückgabe:
# -   Liefert den gewünschten Wert.
# -----

sub checkMetadataSyntax {
  my($hashkey,$value) = @_;
  my($res);

  # Initialisierung
  $res = $value;

  # MSC91-Klassifikation
  if ($hashkey eq "subject.msc") {
    # Primärschlüssel aus MSC-Klassifikation extrahieren
    $res = substr($value,0,5);
  }

  # Ergebnis zurückliefern
  return $res;
};

# -----
1; #return true
```

## E. Bibliothek 'Translations.pm'

```
#####
#
# Quelltext:
# (KLUEDO)/lib/KLUEDO/Translations.pm
#
# Beschreibung:
# Realisiert die Übersetzung von Begriffen in verschiedene Sprachen.
#
# Autor:
# Wolfgang Eiden, Universität Kaiserslautern
#
#####

use strict;

use lib('opt/metadata/KLUEDO/lib');

use IO::File;
use KLUEDO::Environment;

# -----
# - Routine      : Translations
# - Beschreibung : Übersetzt einen Begriff in eine (andere) Sprache.
# - Parameter   : (String) language:
#                 Sprache, in die der Begriff übersetzt werden soll
#                 z.B. "de","en"
# -             (String) path:
#                 Dateipfad der Übersetzungsdatei
# - Rückgabe    : (Hash) translation:
#                 Hashtabelle mit übersetzten Begriffen
# -----

sub Translations {
    my($language, $file) = @_ ;
    my($line, @field, %translation,$FILE);

    # Sprachdatei öffnen
    $FILE = IO::File->new("< $file");

    LINE: while (defined($line = $FILE->getline())) {

        # Kommentar- und Leerzeilen ignorieren
        if (($line =~ /\#\#/ || ($line =~ /\n/)) {
            next LINE;
        }

        # Aufsplitten der Einträge zum Ermitteln der Begriffe
        chop($line);
        @field = split('\|', $line);
        $translation{$field[0]} = $field[getLanguageID($language)+1];
    }

    # Konfigurationsdatei schließen und Übersetzung zurückliefern
    $FILE->close();
    return %translation;
}

# -----
1; #return true
```

## F. PHP-HTML-Seite 'metamaker.html'

```

<?
$KLUEDO_HOME = "/opt/metadaten/KLUEDO";
require("$KLUEDO_HOME/php/init.pinc");

# Initialisierung
$formvars="";
$modeprefix="";
if(isset($POST))
{
    reset($POST);
    while ( list( $key, $val ) = each( $POST ) )
    {
        if ( $key != "modus" )
        {
            $key = str_replace("_","",$key);
            if ( $formvars != "" ) { $formvars .= " & "; }
            $key = str_replace("-", "=", str_replace("&", "\&", $key));
            $val = str_replace("-", "=", str_replace("&", "\&", $val));
            $val = htmlentities($val);
            $formvars .= "$key = $val";
        }
    }
}
$formvars = str_replace("`", "", $formvars);
$formvars = str_replace("`", "", $formvars);
if (preg_match("/\admin\/i", $sess->self_url()))
{
    $modeprefix = "admin=".$REMOTE_USER.":";
}
if (!$modus)
{
    $modus = "new";
}
$department = $FB_dir[$FB];
if ($metadir != "Metadaten")
{
    $metadir = "Metadaten_Eingang";
}
$metamaker = "$KLUEDO_HOME/bin/MetaMaker.pl";

do_header("Metamaker - Fachbereich $department");
do_body("");
if (($modus == "edit") || ($modus == "confirmedit") || ($modus == "checkedit") || ($modus == "save") || ($modus == "new"))
{
    # Session ermitteln und als Formparameter hinzunehmen
    ereg("Kluedo_Session=([:alnum:]+)", $sess->self_url(), $elements);
    $session = $elements[1];

    # Aufruf des Metamakers
    $parameter = "$department $language \"$session\" \"$formvars\" $metadir $metafile";
    $modus = $modeprefix.$modus;$command = "$metamaker $modus $parameter";
    $result = ""; $res = "";
    exec($command, $result);

    # Ausgabe der Benutzerinformationen
    while ( list( $key, $val ) = each( $result ) ) {
        echo "$val\n";
    }
} else {
    print "<h2>Metamaker der Universit&auml;t Kaiserslautern</h2>";
    print "Der Metamaker wurde auf unerlaubte Art und Weise aufgerufen.";
    print "Bitte kontaktieren Sie den Administrator.";
}
do_footer();
?>

```



## G. PHP-HTML-Seite 'upload.html'

```

<?
$KLUEDO_HOME = "/opt/metadata/KLUEDO";
require("$KLUEDO_HOME/php/init.pinc");
do_header("KLUEDO - Kaiserslauterer uniweiter elektronischer Dokumentenserver");
do_body("<h2>Datei&uuml;bertragung</h2>");
?>

Vielen Dank für die Benutzung unseres Angebotes.<br><br>

<?
// Dateinamen bestimmen
$suffix = "";
if ($userfile_type == "application/postscript") {
    $suffix = ".ps";
}
if ($userfile_type == "application/pdf") {
    $suffix = ".pdf";
}
$filename = $KLUEDO_HOME."/Fachbereiche/".$department;
$filename .= "/Quellen_Eingang/".$fileprefix.$suffix;

// Quelle akzeptieren ?
if ($suffix) {
    if (copy("$userfile", "$filename")) {
        echo "Die Datei <i><?echo $userfile_name?></i> wurde erfolgreich übertragen.<br>";
        unlink("$userfile");
    }
    else {
        echo "Die Datei <i><?echo $userfile_name?></i> konnte leider nicht erfolgreich übertragen werden.<br>";
    }
} else {
    echo "<b>Achtung: </b>";
    echo "Das angegebenen Dokument <i>$userfile_name</i> lag nicht im erwarteten Dateiformat vor und wurde daher auch nicht
&uuml;bertragen. ";
    echo "Als Dateiformate akzeptieren wir lediglich <i>Postscript</i> und <i>PDF</i><br>";
    if ($userfile) {
        unlink("$userfile");
    }
}
?>

<br>
Sie haben nun folgende Auswahl&uuml;glichkeiten
<dir>
<li><a href="http://kluedo.ub.uni-kl.de/$department/metamaker_$language.html">Eingeben einer neuen
Ver&uuml;ffentlichung</a></li>
<li><a href="http://kluedo.ub.uni-kl.de/$department/Listen/edit_$language.html">Bearbeiten einer
Ver&uuml;ffentlichung</a></li>
</dir>
<p>

<?
do_footer();
?>

```

## H. PHP-Include 'navigation.pinc'

```

<?
////////////////////////////////////////////////////////////////////
//
// Quelltext:
// (KLUEDO)/php/navigation.pinc
//
// Beschreibung:
// Realisiert die Navigation in Listen. Die Routine do_navigation() wird
// von den PHP-HTML-Seiten aufgerufen
//
// Autor:
// Wolfgang Eiden, Universität Kaiserslautern

// -----
// - Routine:
// - do_navigation_getdata
// - Beschreibung:
// - Ermittelt aus der übergebenen Zeile die Reihennummer und liefert diese
// - und die Zeile (ohne diese Angaben) zurück
// - Voraussetzung hierfür ist, dass die Zeile in der Form
// - "Value[-Value]:Data"
// - vorliegt.
// - Rückgabe:
// - Tupel (number,line), mit
// - number: Nummer der Veröffentlichung
// - line : Veröffentlichungsdaten
// -----

function do_navigation_getdata($argline)
{
    $number = 0;
    $line = "";
    if (preg_match("/^\(w+\)-w*:(.*)/i",$argline,$parts)) {
        $number = $parts[1];
        $line = $parts[2];
    } else {
        if (preg_match("/^\(w+\):(.*)/i",$argline,$parts)) {
            $number = $parts[1];
            $line = $parts[2];
        }
    }
    $res = array($number,$line);
    return $res;
}

// -----
// - Routine:
// - do_navigation
// - Beschreibung:
// - Realisiert die Navigation
// -----

function do_navigation($rows,$docs,$POST)
{
    // Standardanzahl angezeigter Einträge
    $NAV_CONST_STDDIST = 6;
    $NAV_CONST_ENTRIES = $rows;

    // Voraussetzung: sortierte Liste mit Maximalitätseintrag in erster Zeile
    $getdata = do_navigation_getdata($docs[1]);
    $number = $getdata[0];
    $NAV_CONST_MAX = $number;
    $OP="";

    // Führte der Benutzer einen Navigationsschritt aus ?
    if (isset($POST)) {
        // Variablen mappen
        reset($POST);
        while (list($key,$val) = each($POST)) {
            if ($key == "NavRangeStart") {$NavRangeStart = $val;}
            if ($key == "NavRangeWidth") {$NavRangeWidth = $val;}
            if ($key == "Position") {$Position = $val;}
            if ($key == "First_x") {$First = true;}
            if ($key == "Previous_x") {$Previous = true;}
            if ($key == "Next_x") {$Next = true;}
            if ($key == "Last_x") {$Last = true;}
        }
    }
}

```

```

// Alte Werte zunächst übernehmen
$NAV_RANGEWIDTH = $NavRangeWidth;
if ($NavRangeStart < $NAV_CONST_MAX) {
    $NAV_RANGESTART = $NavRangeStart;
} else {
    $NAV_RANGESTART = $NAV_CONST_MAX;
}

// Welche Aktion möchte der Benutzer ausführen ?
if ($First) {
    // Bewegung an den Anfang
    $OP="FIRST";
    $NAV_RANGESTART = 1;
}
if ($Previous) {
    // Bereichsverschiebung (vor)
    $OP="PREV";
    $NAV_RANGESTART = $NAV_RANGESTART - $NAV_RANGEWIDTH;
}
if ($Next) {
    // Bereichsverschiebung (weiter)
    $OP="NEXT";
    $NAV_RANGESTART = $NAV_RANGESTART + $NAV_RANGEWIDTH;
}
if ($Last) {
    // Bewegung an das Ende
    $OP="LAST";
    $NAV_RANGESTART = $NAV_CONST_MAX - $NAV_RANGEWIDTH + 1;
}
if ($Position) {
    // Bereichsänderung
    $OP="POSITION";

    // Werte ermitteln
    if (preg_match("/(\d+)-(\d+)/i",$Position,$parts)) {
        if ($parts[1]>$parts[2]) {
            $tempvar = $parts[1];
            $parts[1] = $parts[2];
            $parts[2] = $tempvar;
        }
        $NAV_RANGESTART = $parts[1];
        $NAV_RANGEWIDTH = Abs($parts[2]-$parts[1]) + 1;
        if ($NAV_RANGEWIDTH > $NAV_CONST_MAX) {
            $NAV_RANGEWIDTH = $NAV_CONST_STDDIST;
        }
    } else {
        if (preg_match("/(\d+)/i",$Position,$parts)) {
            $NAV_RANGESTART = $parts[1];
            $NAV_RANGEWIDTH= $NAV_RANGEWIDTH;
        }
    }
}

// Bereichskorrekturen
if ($NAV_RANGESTART < 1) {
    $NAV_RANGESTART = 1;
}
if ($NAV_RANGESTART > $NAV_CONST_MAX) {
    $NAV_RANGESTART = $NAV_CONST_MAX;
}
$NAV_RANGEEND = $NAV_RANGESTART + $NAV_RANGEWIDTH - 1;
if ($NAV_RANGEEND > $NAV_CONST_MAX) {
    $NAV_RANGEEND = $NAV_CONST_MAX;
}
}

// Falls keine eindeutige Operation ausgeführt werden konnte, zeige die letzten
// Einträge an...
if ($OP == "") {
    $NAV_RANGESTART = $NAV_CONST_MAX - $NAV_CONST_STDDIST + 1;
    $NAV_RANGEWIDTH = $NAV_CONST_STDDIST;
    if ($NAV_RANGESTART < 1) {
        $NAV_RANGESTART = 1;
    }
    $NAV_RANGEEND = $NAV_CONST_MAX;
}

// Bereichstext zusammensetzen
$NAV_RANGETEXT = $NAV_RANGESTART."-".$NAV_RANGEEND." ";
$NAV_RANGETEXT .= "von";
$NAV_RANGETEXT .= " ". $NAV_CONST_MAX;

// Formtags generieren
$IMG_DATA = " border=0 width=\`20\` height=\`20\` align=middle alt=\`\" ";
$NAV_FORMDATA = "<input type=hidden name=\`NavRangeStart\` value=\`\". $NAV_RANGESTART. "\`>";
$NAV_FORMDATA .= "<input type=hidden name=\`NavRangeWidth\` value=\`\". $NAV_RANGEWIDTH. "\`>";
$NAV_FORMDATA .= "\`n";
$BAR = "";
$BAR .= "<table border=0 cellspacing=0 cellpadding=0>";
$BAR .= "<tr valign=center>";
$BAR .= "<td><form method=\`post\`>\". $NAV_FORMDATA;
$BAR .= " <input type=image src=\`/images/navigation/first.gif\`\". $IMG_DATA;
$BAR .= " name=\`First\`></form></td>";
$BAR .= "<td><form method=\`post\`>\". $NAV_FORMDATA;
$BAR .= " <input type=image src=\`/images/navigation/previous.gif\`\". $IMG_DATA;
$BAR .= " name=\`Previous\`></form></td>";
$BAR .= "<td><form method=\`post\`>\". $NAV_FORMDATA;
$BAR .= " <input name=\`Position\` type=\`text\` size=\`18\` border=0";

```

```

$BAR .= " align=middle alt=\"\" value=\"\".$NAV_RANGETEXT.\"\">&nbsp;";
$BAR .= "</form></td>";
$BAR .= "<td><form method=\"post\">".$NAV_FORMDATA;
$BAR .= " <input type=image src=\"/images/navigation/next.gif\".\".$IMG_DATA;
$BAR .= " name=\"Next\"></form></td>";
$BAR .= "<td><form method=\"post\">".$NAV_FORMDATA;
$BAR .= " <input type=image src=\"/images/navigation/last.gif\".\".$IMG_DATA;
$BAR .= " name=\"Last\"></form></td>";
$BAR .= "</tr></table>";

// echo "<hr><h1>Achtung</h1>Gerade in diesem Moment, wird am Benutzerinterface gearbeitet. Bitte haben Sie für evtl.
Nebeneffekte Verständnis<br>Wolfgang Eiden<hr>";
echo $BAR;

?>

<p>
<table cellpadding=1 cellspacing=0 border=0 width=100%>
<?
  $i = $NAV_CONST_ENTRIES;
  $endloop=false;
  while (($i > 0) && (!$endloop)) {
    $getdata = do_navigation_getdata($docs[$i]);
    $number = $getdata[0];
    $line = $getdata[1];

    // Zeile ausgeben, falls innerhalb des gewünschten Bereiches
    if (($number >= $NAV_RANGESTART) && ($number <= $NAV_RANGEEND)) {
      echo $line."<br>";
    }

    // Laufvariablen erniedrigen
    $i--;
  }
?>

</table>
<p>

<?
  echo $BAR;
} // end do_navigation

?>

```

## I. Skript 'AktualisiereListen.sh'

```

#!/bin/sh
# /opt/metadata/KLUEDO/bin/AktualisiereListen.sh
# Erneuert die Listen der verschiedenen Fachbereiche

for Fachbereich in ARUBI Biologie Chemie Elektrotechnik Informatik \
  MV Mathematik Physik SoWi
do
  NeueListen.pl $Fachbereich
done

```

## J. Skript 'Eingang.pl'

```
#!/usr/bin/perl -w
# /opt/metadata/KLUEDO/bin/Eingang.pl
# Erneuert die Eingangsliste des Fachbereichs der im aktuellen Pfad steht
#
# 1999-08-24 Wolfgang Eiden

use strict;

sub main
{
    my($path,$fb,$muster);

    $path = `pwd`;
    $muster= '\metadata\KLUEDO\Fachbereiche\';
    if ($path =~ /$muster/i)
    {
        $fb = $path;
        $fb =~ s/,*$muster//i;
        $fb =~ s/\./\./i;
        chomp($fb);
        system("\opt/metadata\KLUEDO\bin\NeueListen.pl $fb _incoming");
    }
    else
    {
        print "Das aktuellen Verzeichnis befindet sich nicht innerhalb eines Fachbereiches.\n\n";
    }
}

main();
```

## K. Skript 'MetaMaker.pl'

```
#!/usr/bin/perl

#####
#
# Quelltext:
#   (KLUEDO)/bin/Metamaker.pl
#
# Beschreibung:
#
#
# Autor:
#   Wolfgang Eiden, Universität Kaiserslautern
#
#####

use strict;

use lib ('/opt/metadata/KLUEDO/lib');

use KLUEDO::Conversions;
use KLUEDO::Environment;
use KLUEDO::MetaMaker;
use KLUEDO::ReadMetafile;
use KLUEDO::Translations;

# -----
# - Routine:
# -   outputHash
# - Beschreibung:
# -   Debug-Routine: Gibt den Hash visualisiert in einer HTML-Tabelle aus.
# - Parameter:
# -   hash: Der auszugebende Hash
# - Rückgabe:
# -   HTML-Text
# -----

sub outputHash {
    my($hash) = @_;
    my($key,$value,$res);

    $res="<table border=1>";
    $res.= "<tr><td><b>Key</b></td><td><b>Value</b></td></tr>";
    while (($key,$value)=each $hash) {
        $res.= "<tr><td><i>$key &nbsp;</i></td><td>$value &nbsp;</td></tr>";
    }
    $res.= "</table>";
}
```

```

# Ergebnis zurückliefern
return $res;
}

# -----
# - Routine:
# - formvars2hash
# - Beschreibung:
# - Zerlegt die übergebene Zeile des Formates "x1 = y1 & x2 = y2 & ..." und
# - übergibt die Werte in einer Hashtabelle, sodass hash(x1)=y1 (usw.) gilt.
# - Diese Routine dient zur Kommunikation mit der PHP-Seite.
# - Parameter:
# - Zeile im angegebenen Format
# - Rückgabe:
# - HTML-Text
# -----

sub formvars2hash {
    my($formvars) = @_;
    my(%metadata,@pairs,@tupel,$pair,$stupel,$key,$value);

    # In Zuweisungen der Form (x1 = y1) zerlegen
    @pairs = split(/ & /, $formvars);

    # Zuweisungen in Schluessel bzw. Wert zerlegen und aufnehmen
    foreach $pair(@pairs) {
        @tupel = split(/ =/, $pair);
        $key = $tupel[0]; $key =~ s/\\=//g; $key =~ s/\\&/&/g ;
        $value = $tupel[1]; $value =~ s/\\=//g; $value =~ s/\\&/&/g;
        $metadata{"$key"} = trim($value);
    }

    # Ergebnis zurückliefern
    return %metadata
}

# -----
# - Routine:
# - getfilenumber
# - Beschreibung:
# - Ermittelt bezüglich der angegebenen Reihe (im übergebenen Verzeichnis)
# - die höchste Nummer anhand des Dateinamens und liefert diese um 1
# - inkrementiert zurück. Diese Routine wird dazu verwendet, die nächste
# - freie Nummer zu ermitteln.
# - Parameter:
# - directory: Verzeichnis
# - serie : Serie (als Dateinamenpräfix)
# - Rückgabe:
# - nächste freie Nummer
# -----

sub getfilenumber {
    my ($directory,$serie) = @_;
    my ($number,@allfiles,@filelist);

    # Verzeichnis öffnen und sortierte Liste dieser Serie erstellen
    opendir(DIR,$directory/);
    @allfiles = grep /html/, grep /^$serie\\*/,readdir DIR;
    closedir(DIR);
    @filelist = map { $_->[1] } sort { $b->[0] <=> $a->[0] }
    map { [ /(\d+)/, $_ ] } @allfiles;

    # Höchste Nummer auswählen und erhöhen
    $number = $filelist[0];
    $number =~ s/$serie\\//;
    $number =~ s/\\.html//;
    $number += 1;

    # Ergebnis zurückliefern
    return $number;
}

# -----
# - Routine:
# - startSection
# - Beschreibung:
# - Beginnt eine neue Tabelle mit Captionzeile
# - Parameter:
# - othercolor: Angabe, ob die Captionzeile dieser Tabelle besonders
# - gekennzeichnet werden soll (z.B. für Bereiche, die nur dem
# - Administrator zur Verfügung stehen)
# - msg : Text in der Captionzeile
# - Rückgabe:
# - HTML-Text
# -----

```

```

sub startSection {
  my($othercolor,$msg) = @_ ;
  my($res);

  # Aufbau des HTML-Textes
  $res = "";
  $res .= " <table cellpadding=4 cellspacing=0 border=0 width=\"100%\">\n";
  $res .= " <tr valign=center>\n";
  if ($othercolor) {
    $res .= " <td bgcolor=\"FF3399\">\n";
  } else {
    $res .= " <td bgcolor=\"003399\">\n";
  }
  $res .= " <font color=\"ffffff\"><b>.$msg.</b></font>\n";
  $res .= " </td>\n";
  $res .= " </tr>\n";
  $res .= " <tr>\n";
  $res .= " <td bgcolor=\"6699CC\">\n";

  # Ergebnis zurückliefern
  return $res;
}

# -----
# - Routine:
# - endSection
# - Beschreibung:
# - Beendet die mit startSection begonnene Tabelle
# - Parameter:
# - -
# - Rückgabe:
# - HTML-Text
# -----

sub endSection {
  my($res);

  # Aufbau des HTML-Textes
  $res = "";
  $res .= " </td>\n";
  $res .= " </tr>\n";
  $res .= " </table>\n";

  # Ergebnis zurückliefern
  return $res;
}

# -----
# - Routine:
# - getformheader
# - Beschreibung:
# - Erzeugt den Header für ein HTML-Formular des Metamakers
# - Parameter:
# - department: Fachbereich
# - language : Sprache
# - modus : Modus in den der Metamaker wechseln soll
# - metadir : Verzeichnis in denen der Metamaker die Daten lesen/schreiben
# - soll
# - metafilename : Datei im Verzeichnis metadir, die der Metamaker verarbeiten
# - soll
# - session : Session-Variable von KLUEDO
# - Rückgabe:
# - HTML-Text
# -----

sub getformheader {
  my($department,$language,$modus,$metadir,$metafilename,$session) = @_ ;
  my($res);

  # Aufbau des HTML-Textes
  $res = "";
  $res .= "<form method=\"post\"";
  if ($modus ne "abort") {
    $res .= " action =\"metamaker_.$language_.html";
    $res .= "?Kluedo_Session=$session" if $session;
  } else {
    $res .= " action =\"http://kluedo.ub.uni-kl.de/$department/Listen/";
    $res .= "edit_$language.html\"";
  }
  $res .= ">\n";
  $res .= " <input name=\"modus\" type=\"hidden\" value=\"$modus\">\n";
  $res .= " <input name=\"metadir\" type=\"hidden\" value=\"$metadir\">\n";
  $res .= " <input name=\"metafilename\" type=\"hidden\" value=\"$metafilename\">\n";
  $res .= " <input name=\"subject.topic\" type=\"hidden\" value=\"$subject.topic\">\n";
  $res .= " <input name=\"department\" type=\"hidden\" value=\"$department\">\n";

  # Ergebnis zurückliefern
  return $res;
}

```

```

# -----
# - Routine:
# - generateform
# - Beschreibung:
# - Erzeugt das Eingabe / Bearbeitungsformular des Metamakers
# - Parameter:
# - admin      : Name des Administrators (oder leer, falls normaler Benutzer)
# - department: Fachbereich
# - language   : Sprache
# - basicdir   : Fachbereichspfad
# - modus      : Modus in den der Metamaker wechseln soll
# - metadir    : Verzeichnis in denen der Metamaker die Daten lesen/schreiben
# -           : soll
# - metafile   : Datei im Verzeichnis metadir, die der Metamaker verarbeiten
# -           : soll
# - session    : Session-Variable von KLUEDO
# - metadata   : Hash-Tabelle mit den Metadaten
# - Rückgabe:
# - HTML-Text
# -----

sub generateform {
my($admin,$department,$language,$basicdir,$metadir,$metafile,$session,
  %metadata) = @_;
my($CONF_FILE,$res,$slangid,$noerror,$line,$serie,$parameter,@fields);
my(@fields2,%lang,%env);

$res = "";
$noerror = "true";
$slangid = getLanguageID($language);
%env = Environment();
%lang = Translations($language,$env{"CONF"}."/language.conf");

# Formulardefinition beginnen
$res .= getformheader($department,$language,"checkedit",$metadir,
  $metafile,$session);
$res .= "<b>".$lang{"Tip"}."</b> ";
$res .= $lang{"mm_mfieldmark"}." : ";
$res .= "<img src=\"/images/obligation.gif\"><p><br>";

# Bearbeitungshinweise
$res .= startSection("", $lang{"mm_editornote"});
$res .= "<br><p>\n";
$res .= "<b>".$lang{"mm_email"}."</b>: (".$lang{"mm_fordemands"}.)<br>\n";
$res .= "<input name=\"editor\" size=\"80\" value=\"\"";
$res .= $metadata{"editor"};
$res .= "\>&nbsp;<img src=\"/images/obligation.gif\"><p>\n";
$res .= "<b>".$lang{"mm_docnote"}."</b>:";
$res .= " (".$lang{"mm_docnote_add"}.)<br>\n";
$res .= "<textarea name=\"editor.comment\" rows=\"4\" cols=\"80\">";
$res .= $metadata{"editor.comment"};
$res .= "</textarea><p>\n";
$res .= endSection();
$res .= "<br>\n";

# Administrator
if ($admin) {
  $res .= startSection($admin,"Administrator");
  $res .= "<br><p>\n";
  $res .= "<b>".$lang{"mm_files"}."</b>:<br>\n";
  $res .= "<input name=\"identifier.url\" size=\"80\" value=\"\"";
  $res .= $metadata{"identifier.url"}."><p>\n";
  $res .= "<b>".$lang{"mm_formats"}."</b>:<br>\n";
  $res .= "<input name=\"format\" size=\"80\" value=\"\"";
  $res .= $metadata{"format"}.">\n";
  $res .= endSection();
  $res .= "<br>\n";
} else {
  $res .= "<input type=\"hidden\" name=\"identifier.url\" value=\"\"";
  $res .= $metadata{"identifier.url"}.">\n";
  $res .= "<input type=\"hidden\" name=\"format\" value=\"\"";
  $res .= $metadata{"format"}.">\n";
}

# Bibliographische Beschreibung
$res .= startSection("", $lang{"mm_bibref"});
$res .= "<br><p>\n";
$res .= "<b>".$lang{"mm_bib_authors"}."</b>: ";
$res .= $lang{"mm_bib_authors_add"}."<br>\n";
$res .= "<input name=\"creator.personalname\" size=\"80\" value=\"\"";
$res .= $metadata{"creator.personalname"};
$res .= "\>&nbsp;<img src=\"/images/obligation.gif\"><p>\n";
$res .= "<b>".$lang{"mm_bib_email"}."</b>:";
$res .= " ".$lang{"mm_bib_email_add"}."<br>\n";
$res .= "<input name=\"creator.email\" size=\"80\" value=\"\"";
$res .= $metadata{"creator.email"};
$res .= "\><p>\n";
$res .= "<b>".$lang{"mm_bib_institut"}."</b>: ";
$res .= $lang{"mm_bib_institut_add"}."<br>\n";
$res .= "<input name=\"creator.affiliation\" size=\"80\" value=\"\"";
$res .= $metadata{"creator.affiliation"};
$res .= "\><p>\n";

$res .= $lang{"mm_bib_title"}."<br>\n";
$res .= "<input name=\"title\" size=\"80\" VALUE=\"\"";
$res .= $metadata{"title"};
$res .= "\>&nbsp;<img src=\"/images/obligation.gif\"><p>\n";

```



```

$res .= $lang{"mm_bib_year"}.: <br>\n";
$res .= "<input name=\"date.intellectualcreation.year\" ";
$res .= "size=\"4\" maxlength=\"4\" ";
$res .= "value=\"";
$res .= $metadata{"date.intellectualcreation.year"};
$res .= "\"><p>\n";
$res .= "<input type=\"hidden\" name=\"date.created\" value=\"";
$res .= $metadata{"date.created"}."\">\n";
$res .= "<input type=\"hidden\" name=\"date.modified\" value=\"";
$res .= $metadata{"date.modified"}."\">\n";

$res .= $lang{"mm_bib_ref"}.: ";
$res .= "<br><input name=\"identifier\" size=\"80\" value=\"";
$res .= $metadata{"identifier"};
$res .= "\"><p>\n";

$res .= $lang{"mm_bib_type"}.:<br>\n";
$res .= "<select name=\"type\">\n";
$res .= " <option value=\"preprint\"";
$res .= " selected\" if ($metadata{\"type\"} =~ /preprint/i);
$res .= ">". $lang{"Preprint"}."\"";
$res .= " <option value=\"article\"";
$res .= " selected\" if ($metadata{\"type\"} =~ /article/i);
$res .= ">". $lang{"Article"}."\"";
$res .= " <option value=\"Diploma Thesis\"";
$res .= " selected\" if ($metadata{\"type\"} =~ /diploma/i);
$res .= ">". $lang{"Dipl_thesis"}."\"";
$res .= " <option value=\"PhD Thesis\"";
$res .= " selected\" if ($metadata{\"type\"} =~ /phd/i);
$res .= ">". $lang{"Phd_thesis"}."\"";
$res .= " <option value=\"script\"";
$res .= " selected\" if ($metadata{\"type\"} =~ /script/i);
$res .= ">". $lang{"Script"}."\"";
$res .= "</select>&nbsp;\n";

$res .= "in\n";
$res .= "&nbsp;<select name=\"language\">\n";
$res .= " <option value=\"ENG\"";
$res .= " SELECTED\" if ($metadata{\"language\"} =~ /ENG/);
$res .= ">". $lang{"mm_bib_lang_en"}."\"";
$res .= " <option value=\"GER\"";
$res .= " SELECTED\" if ($metadata{\"language\"} =~ /GER/);
$res .= ">". $lang{"mm_bib_lang_de"}."\"";
$res .= " <option value=\"\"";
$res .= $lang{"mm_bib_lang_other"}."\"";
$res .= "</select>&nbsp;\n";
$res .= $lang{"mm_bib_lang"}."<br><p>\n";

$res .= $lang{"mm_bib_serie"}.: <br>\n";
$res .= "<select name=\"source.internal.serie\">\n";
$CONF_FILE = IO::File->new("<$basicdir/conf/Listen.conf\" or $noerror="";
while ((($noerror) && (defined($line = $CONF_FILE->getline())))) {
    @fields = split('\|', $line);
    $serie = $fields[0];
    if ($serie ne "incoming") {
        $parameter = $fields[1];chomp($parameter);
        @fields2 = split(':', $parameter);
        $parameter = $fields[$langid+1];chomp($parameter);
        @fields = split(':', $parameter);
        $res .= " <option value=\"$serie\"";
        $res .= " SELECTED\" if ($metadata{\"source.internal.serie\"} =~ /$serie/i);
        $res .= ">". $fields[0].\"";
    }
}
$CONF_FILE->close() if ($noerror);
$res .= "</select><p>\n";
$res .= "<input type=\"hidden\" name=\"source.internal.nr\" value=\"";
$res .= $metadata{"source.internal.nr"}."\">\n";
$res .= $lang{"mm_bib_serie_count"}.:<br>\n";
$res .= $lang{"mm_bib_serie_nr"}.: ";
$res .= "<input name=\"source.identifier.nr\" size=5 maxlength=5 value=\"";
$res .= $metadata{"source.identifier.nr"}."\">\n";
$res .= "<i>&nbsp; " . $lang{"or"} . "&nbsp;</i><b> ";
$res .= $lang{"mm_bib_serie_tag"};
$res .= "</b> (" . $lang{"else"} . ") &nbsp;";
$res .= "<input name=\"source.identifier.tag\" size=10 maxlength=10 ";
$res .= "value=\"";
$res .= $metadata{"source.identifier.tag"}."\">\n";
$res .= "<p>\n";
$res .= endSection();
$res .= "<br>";

# Inhaltliche Beschreibung
$res .= startSection("", $lang{"mm_content"}).<br><p>\n";
$res .= "<b>". $lang{"mm_abstract"}."</b><br>\n";
$res .= <textarea name="description.abstract\" rows="12\" ";
$res .= "cols="80\">";
$res .= unescape($metadata{"description.abstract"});
$res .= "</textarea><p>\n";
$res .= $lang{"mm_keywords"}.:<br>\n";
$res .= "<input name=\"subject.keywords\" size="80\" value=\"";
$res .= $metadata{"subject.keywords"};
$res .= "\"><p>\n";

```

```

if ($department eq "Mathematik") {
    $res .= "<a href=\"";
    $res .= "http://www.uni-kl.de/Bibliothek/";
    $res .= "mathphybib/Systematik/Mathe/Ams/\"";
    $res .= "target=\"_blank\"><b>";
    $res .= $lang{"mm_msc"};
    $res .= "</b>:</a> (".$lang{"mm_msc_add"}).<br><p>\n";
    $res .= "<input name=\"subject.msc\" size=\"80\"\"";
    $res .= "value=\"\"";
    $res .= $metadata{"subject.msc"};
    $res .= "\&nbsp;<p>\n";
}

$res .= "<b>".$lang{"mm_notes"}.</b><br>\n";
$res .= "<textarea name=\"description.notes\" rows=\"4\" cols=\"80\">";
$res .= unescape($metadata{"description.notes"});
$res .= "</textarea><p>\n";
$res .= endSection().<br>\n";

# Formulardefinition beenden
$res .= "<input type=\"submit\" Value=\"".$lang{"mm_viewresult"}.\">\n";
$res .= "<input type=\"reset\" Value=\"".$lang{"mm_reset"}.\">\n";
$res .= "</form>";

# Ergebnis zurückliefern
return $res;
}

# -----
# - Routine:
# - confirmedit
# - Beschreibung:
# - Liest die angegebene Metadatei ein, und generiert eine HTML-Seite mit
# - den Metainformationen bei der der Benutzer auswählen kann, ob er diese
# - Metainformationen wirklich bearbeiten möchte.
# - Parameter:
# - admin      : Name des Administrators (oder leer, falls normaler Benutzer)
# - department : Fachbereich
# - language   : Sprache
# - basicdir   : Fachbereichspfad
# - metadir    : Verzeichnis in denen der Metamaker die Daten lesen/schreiben
# -            soll
# - metafile   : Datei im Verzeichnis metadir, die der Metamaker verarbeiten
# -            soll
# - session    : Session-Variable von KLUEDO
# - Rückgabe:
# - HTML-Text
# -----

sub confirmedit {
    my($admin,$department,$language,$basicdir,$metadir,$metafile,$session) = @_;
    my($filename,$metadata,$serie,$number,$identifier,$i,$tempstr,@temp);
    my($information,$key,$value,$res,$langid,$env,$lang);

    # Umgebung
    $langid = getLanguageID($language);
    $env = Environment();
    $lang = Translations($language,$env{"CONF"}."/language.conf");

    # Metadatendatei einlesen
    $filename = "$basicdir/$metadir/$metafile";
    $metadata = readMetadata("$filename");
    $metadata{"editor"}=$admin if ($admin);

    # Reihe und Nummer extrahieren
    $serie="no_series";
    $number="";
    if ($metafile =~ /\.(.*?)_(.*?)\/) {
        ($tempstr,) = split('\.html',$metafile);
        @temp = split('_', $tempstr);
        $number=$temp[$#temp];
        $serie = "";
        for ($i=0; $i < $#temp; $i++) {
            $serie .= $temp[$i];
            $serie .= "_" if ($i lt ($#temp-1));
        }
    }
    $metadata{"source.internal.serie"}=$serie;
    $metadata{"source.internal.nr"}=$number;

    # Handelt es sich um eine numerische Zählung oder um eine alphanumerische
    # Zählung ?
    $identifier = $metadata{"source.identifier"};
    if ($identifier eq $number) {
        $metadata{"source.identifier.nr"} = $number;
        $metadata{"source.identifier.tag"} = "";
        $metadata{"source.identifier"} = $number;
    } else {
        $metadata{"source.identifier.nr"} = "";
        $metadata{"source.identifier.tag"} = $identifier;
        $metadata{"source.identifier"} = $identifier;
    }

    # Metadaten-Informationen-string erzeugen
    $information = generateInformationStr($department,\%metadata);
}

```

```

# Absicht des Benutzer überprüfen
$res = "";
$res .= "<h2>". $lang{"mm_mode_edit"}."</h2>\n";
$res .= $lang{"mm_question_editthis"}."<br>\n";
$res .= "<table>\n";
$res .= "  <tr>\n";
$res .= "    <td>\n";
$res .= "      getformheader($department,$language,"edit",$metadir,$metafile,
        $session);
while (($key,$value) = each %metadata) {
  $res .= "    <input name=\"$key\" type=\"hidden\" value=\"$value\"";
  $res .= "  </td>\n";
}
$res .= "  <td>\n";
$res .= "    <input name=\"send\" type=\"submit\" value=\"$value\"";
$res .= "  </td>\n";
$res .= "</tr>\n";
$res .= "</table>\n";
$res .= "  <tr>\n";
$res .= "    <td>\n";
$res .= "      getformheader($department,$language,"abort","",,$session);
$res .= "    <input name=\"send\" type=\"submit\" value=\"$value\"";
$res .= "  </td>\n";
$res .= "</tr>\n";
$res .= "</table>\n";
$res .= "  <tr>\n";
$res .= "    <td>\n";
$res .= "      <br>\n";
$res .= "      <dir>\n";
$res .= "    </td>\n";
$res .= "</tr>\n";
$res .= "</table>\n";
$res .= "</dir>\n";

# Ergebnis zurückliefern
return $res;
}

# -----
# - Routine:
# - edit
# - Beschreibung:
# - Zeigt das Eingabeformular für Metadaten ein und verwendet dabei die (mit
# - der Variablendeklarationszeile) übergebenen Metadaten als Vorgabe
# - Parameter:
# - admin      : Name des Administrators (oder leer, falls normaler Benutzer)
# - department: Fachbereich
# - language   : Sprache
# - basicdir   : Fachbereichspfad
# - metadir    : Verzeichnis in denen der Metamaker die Daten lesen/schreiben
# -            soll
# - metafile   : Datei im Verzeichnis metadir, die der Metamaker verarbeiten
# -            soll
# - formvars   : Variablendeklarationszeile in der Form
# -            "x1 = y1 & x2 = y2 & ..."
# - session    : Session-Variable von KLUEDO
# - Rückgabe:
# - HTML-Text
# -----

sub edit {
  my($admin,$department,$language,$basicdir,$metadir,$metafile,$formvars,
    $session) = @_;
  my(%metadata,$filename,$information,$key,$value,$vars,$res,$langid,$env);
  my($lang);

  # Umgebung
  $langid = getLanguageID($language);
  $env = Environment();
  $lang = Translations($language,$env{"CONF"}."/language.conf");

  # Hash aus den Formvariablen erzeugen
  %metadata = formvars2hash($formvars);

  # Formular zum Editieren anzeigen
  $res = "";
  $res .= "<h2>". $lang{"mm_mode_edit"}."</h2>\n";
  $res .= generateForm($admin,$department,$language,$basicdir,$metadir,
    $metafile,$session,%metadata);

  # Ergebnis zurückliefern
  return $res;
}

```

```

# -----
# - Routine:
# - checkedit
# - Beschreibung:
# - Überprüft die (mit der Variablendeklarationszeile) übergebenen Metadaten
# - auf Korrektheit
# - Parameter:
# - admin      : Name des Administrators (oder leer, falls normaler Benutzer)
# - department : Fachbereich
# - language   : Sprache
# - basicdir   : Fachbereichspfad
# - metadir    : Verzeichnis in denen der Metamaker die Daten lesen/schreiben
# -            soll
# - metafile   : Datei im Verzeichnis metadir, die der Metamaker verarbeiten
# -            soll
# - formvars   : Variablendeklarationszeile in der Form
# -            "x1 = y1 & x2 = y2 & ..."
# - session    : Session-Variable von KLUEDO
# - Rückgabe:
# - HTML-Text
# -----

sub checkedit {
my($admin,$department,$language,$basicdir,$metadir,$metafile,$formvars,
  $session) = @_;
my($metadata,$metaoutput,$res,$key,$value,$nr,$tag,$langid,$env,$lang);

# Umgebung
$langid = getLanguageID($language);
$env = Environment();
$lang = Translations($language,$env{"CONF"}."/language.conf");

# Hash aus den Formvariablen erzeugen
$metadata = formvars2hash($formvars);

# Editor
if (!$metadata{"editor"}) {
  metaerror("E",$lang{"mm_error_email"},\%metadata);
}

# DC.Source-Angaben
$metadata{"source"}= getDCSourceStr($department,
  $metadata{"source.internal.serie"});
$nr = $metadata{"source.identifizier.nr"};
$tag = $metadata{"source.identifizier.tag"};
if (($nr) && (!$tag)) {

  # Lfd. Nummer
  $metadata{"source.identifizier"} = $nr;

} elsif ((!$nr) && ($tag)) {

  # Alphanumerische Zählung
  $metadata{"source.identifizier"} = $tag;

} elsif (($nr) && ($tag)) {

  # Beide Felder wurden ausgefüllt !
  metaerror("E",$lang{"mm_error_TagAndNr"},\%metadata);

}

## Metadaten testen
$metaoutput = testMetadata($department,\%metadata);

# Benutzeranzeige
$res = "";
$res .= "<h2>".$lang{"mm_mode_edit"}."</h2>\n";

# Erklärung bei Dissertationen
if ((!$metadata{"error"}) && ($metadata{"type"} =~ /phd/i)) {
  $res .= $lang{"mm_Phdtthesis_explanation"};
  $res .= "<br>";
}

# Benutzeranzeige
$res .= "<table>\n";
$res .= "  <tr>\n";

# Sind Fehler aufgetreten, sollte der Benutzer das Dokument nicht speichern
# dürfen
if (!$metadata{"error"}) {
  $res .= "    <td>\n";
  $res .= "      getformheader($department,$language,"save",$metadir,$metafile,
        $session);
  $res .= "      <input name=\"send\" type=\"submit\" value=\"\"";
  $res .= "      $lang{\"mm_save\"}.\>\n";
  while (($key,$value) = each %metadata) {
    $res .= "      <input name=\"\$key\" type=\"hidden\" value=\"\"";
    $res .= "      unescape($value).\>\n";
  }
  $res .= "    </form>\n";
  $res .= "    </td>\n";
}
}

```

```

# Erneute Bearbeitung erlauben
$res .= "    <td>\n";
$res .= getformheader($department,$language,"edit",$metadir,$metafile,
    $session);
while (($key,$value) = each %metadata) {
    $res .= "<input name=\"$key\" type=\"hidden\" value=\"$\"";
    $res .= unescape($value)."\n";
}
$res .= "        <input name=\"send\" type=\"submit\" ";
$res .= "value=\"$\".$lang{\"mm_change\"}.\n";
$res .= "    </form>\n";
$res .= "    </td>\n";
$res .= " </tr>\n";
$res .= "</table>\n";
$res .= "<br>\n";

# Ausgabe des resultierenden Metadatenviews
$res .= $lang{mm_result}."\n";
$res .= "<dir>\n";
$res .= "    <table bgcolor=\"ffffcc\" width=\"100%\" cellpadding=\"5\">\n";
$res .= "        <tr><td>\n".$metaoutput."\n</td></tr>\n";
$res .= "    </table>\n";
$res .= "</dir>\n";
$res .= "<br>";

# Fehler und Warnungen ausgeben
if (($metadata{"error"}) || ($metadata{"warning"})) {
    $res .= $lang{mm_error_notes}." : \n";
    $res .= "<dir>\n";
    if ($metadata{"error"}) {
        $value = "<li>".$metadata{"error"}; $value =~ s/:/:<li></li>/g;
        $res .= "<b>".$value."</b>";
    }
    if ($metadata{"warning"}) {
        $value = "<li>".$metadata{"warning"}; $value =~ s/:/:<li></li>/g;
        $res .= $value;
    }
    $res .= "</dir>\n";
}

# Ergebnis zurückliefern
return $res;
}

# -----
# - Routine:
# - save
# - Beschreibung:
# - Speichert die (mit der Variablendeklarationszeile) übergebenen Metadaten.
# - Datz wird zunächst aus den Angaben der resultierende Dateiname ermittelt,
# - die Metadaten gespeichert und dann ein Benachitigungsml an das
# - Administratorteam versendet.
# - Parameter:
# - admin      : Name des Administrators (oder leer, falls normaler Benutzer)
# - department : Fachbereich
# - language   : Sprache
# - basicdir   : Fachbereichspfad
# - metadir    : Verzeichnis in denen der Metamaker die Daten lesen/schreiben
# -            soll
# - metafile   : Datei im Verzeichnis metadir, die der Metamaker verarbeiten
# -            soll
# - formvars   : Variablendeklarationszeile in der Form
# -            "x1 = y1 & x2 = y2 & ..."
# - session    : Session-Variable von KLUEDO
# - Rückgabe:
# - HTML-Text
# -----

sub save {
my($admin,$department,$language,$basicdir,$metadir,$metafile,$formvars,
    $session) = @_;
my($res,$serie,$fserie,$source,$number,$n1,$n2,$metadata,$langid);
my($shadowfilecontent,$noerror,$serie,$filename,$line,$nr,$tag,$mail,$reihe);
my($errmsg,$cmd,$lang,$env,$fileprefix);

# Initialisierung
$res="";
$errmsg="";
%metadata = formvars2hash($formvars);
$langid = getLanguageID($language);
%env = Environment();
$lang = Translations($language,$env{"CONF"}."/language.conf");

# Dateinamen aus Serie und Nummer ermitteln
$serie = $metadata{"source.internal.serie"};
$nr = $metadata{"source.identifizier.nr"};
$tag = $metadata{"source.identifizier.tag"};
if ($nr) {

    # Laufende numerische Nummer, entspricht Dateinummer
    $number = $nr;

} else {

    # Die Dateinummer ist automatisch zu erzeugen
    if ($metadata{"source.internal.nr"}) {

```

```

# Bei Bearbeitung einer Information soll die alte Nummer verwendet werden
$number = $metadata{"source.internal.nr"}

} else {

# Es handelt sich um einen Neueintrag, also nächste frei Nummer ermitteln
$nl = getfilenumber("$basicdir/Metadaten",$serie);
$nl2 =getfilenumber("$basicdir/Metadaten_Eingang",$serie);
$number = $nl;
$number = $nl2 if ($nl2 ge $nl);

}

}
$fileprefix = $serie."_".$number;
$metadata{"metafile"} = $fileprefix.".html";

# Datei schreiben
$shadowfilecontent = generateMetadataStr($department,%metadata);
$filename = "Metadaten_Eingang/".$metadata{"metafile"};
umask 002, -1, 600;
open(FILE,">$basicdir/$filename" ||
($errmsg = "<li>".$lang{"mm_error_file"}."</li>");
print FILE $shadowfilecontent;
close(FILE);

# Mailtext generieren
$reihe = $metadata{"source"};
if ($metadata{"source.identifizier"}) {
$reihe .= " " . $metadata{"source.identifizier"};
}
$reihe =~ s/&auml;/ä/gi;
$reihe =~ s/&ouml;/ö/gi;
$reihe =~ s/&uuml;/ü/gi;
$reihe =~ s/%szlig;/ss/gi;
$mail = "";
$mail .= "From: unescape($metadata{editor})."\n";
$mail .= "To: kluedo@ub.uni-kl.de\n";
$mail .= "Subject: (KLUEDO) Fachbereich $department, ";
$mail .= $metadata{"metafile"}."\n";
$mail .= "X-Real-Host-From: kluedo.ub.uni-kl.de \n";
$mail .= "Content-type: TEXT/PLAIN\n";
$mail .= "Folgendes Dokument wird in KLUEDO eingebunden :\n\n";
$mail .= "Reihe:\n " . $reihe . "\n\n";
$mail .= "email-Adresse der Kontaktperson:\n ";
$mail .= unescape($metadata{"editor"})."\n\n";
$mail .= "Hinweis zur Bearbeitung:\n ";

$line=unescape($metadata{"editor.comment"});
if ($line) {
$mail .= $line."\n";
} else {
$mail .= "(kein Hinweis)\n";
}
if ($errmsg) {
$mail .= "Bei der Bearbeitung traten leider folgende Fehler auf:\n";
$line = $errmsg;
$line =~ s/<li>/ /;
$line =~ s/</li>/\n/;
$mail .= Mail $line;
}
$mail .= "\nHinweis: Diese Nachricht wurde automatisch generiert.\n\n";

# Mail an das KLUEDO-Team versenden
$noerror="true";
open (Mail,"| sendmail -t -oi -n" || ($noerror=""));
if ($noerror) {
print Mail $mail;
close Mail;
} else {
$errmsg .= "<li>".$lang{"mm_error_mail"}."</li>";
}

# Liste der Neuzugänge aktualisieren
$cmd = `env{HOME}"/bin/NeueListen.pl $department_incoming`;
$line = `cmd`;

# Benutzermitteilung
$res .= "<h2>".$lang{"mm_mode_save"}."</h2>\n";
$res .= $lang{"mm_thanks"};
$res .= " " . $lang{"mm_savedname"} . " : ";
$res .= "<dir><i>".$metadata{"metafile"}."</i></dir>";
$res .= $lang{"mm_check"} . "<p><br>\n\n";

# Dateiupload
$res .= $lang{"mm_upload_msg"} . "\n";
$res .= "<dir>";
$res .= "<form enctype=\\"multipart/form-data\" action=\\"upload_\"";
$res .= $language . ".html\" method=\\"post\">\n";
$res .= " <input type=\\"hidden\" name=\\"MAX_FILE_SIZE\" ";
$res .= "value=\\"5000000\">\n";
$res .= " <input type=\\"hidden\" name=\\"department\" value=\\"";
$res .= $department . "\\">\n";
$res .= " <input type=\\"hidden\" name=\\"fileprefix\" value=\\"";
$res .= $fileprefix . "\\">\n";
$res .= startSection("", $lang{"mm_upload"});
$res .= " <br>\n";
$res .= " <b>".$lang{mm_upload_file}."</b><br>\n";

```

```

$res .= " <input name=\"userfile\" type=\"file\" size=\"80\"><br>\n";
$res .= endSection();
$res .= " <input type=\"submit\" value=\"\"";
$res .= $lang{"mm_upload_button"};
$res .= "\>\n";
$res .= "</form>\n";
$res .= "</dir>\n";
$res .= "<p>\n\n";

# Weitere Editierungsmöglichkeiten anbieten.
$res .= $lang{"mm_possibilities"}.":\n";
$res .= "<dir>\n";
$res .= " <li><a href=\"http://kluedo.ub.uni-kl.de/$department\"";
$res .= "/metamaker_$language.html\">";
$res .= $lang{"mm_mode_new"};
$res .= "</a></li>\n";
$res .= " <li><a href=\"http://kluedo.ub.uni-kl.de/$department\"";
$res .= "/Listen/edit_$language.html\">";
$res .= $lang{"mm_mode_edit"};
$res .= "</a></li>\n";
$res .= "</dir><p><br>\n";

# Fehlermeldungen
if ($errmsg) {
    $res .= "<p><hr><b>".$lang{"mm_error_notes"}." : </b>";
    $res .= "<dir>$errmsg</dir>";
}

# Ergebnis zurückliefern
return $res;
}

# -----
# - Routine:
# - newmetadata
# - Beschreibung:
# - Zeigt die Eingabemaske des Metamakers an und verwendet dabei Standard-
# - vorgeben. Diese Routine wird beim Neueingeben von Metainformationen ver-
# - wendet.
# - Parameter:
# - admin : Name des Administrators (oder leer, falls normaler Benutzer)
# - department: Fachbereich
# - language : Sprache
# - basicdir : Fachbereichspfad
# - metadir : Verzeichnis in denen der Metamaker die Daten lesen/schreiben
# - soll
# - metafile : Datei im Verzeichnis metadir, die der Metamaker verarbeiten
# - soll
# - session : Session-Variable von KLUEDO
# - Rückgabe:
# - HTML-Text
# -----

sub newmetadata {
    my($admin,$department,$language,$basicdir,$metadir,$metafile,$session) = @_ ;
    my($metadata,$res,$langid,$env,$lang);

    # Umgebung
    $langid = getLanguageID($language);
    $env = Environment();
    $lang = Translations($language,$env{"CONF"}."/language.conf");

    # Standardvorgaben
    $metadata{"editor"}=$admin if ($admin);
    $metadata{"source.internal.serie"}="no_series";
    $metadata{"creator.affiliation"}="Universit&auml;t Kaiserslautern";

    # Formular zum Editieren anzeigen
    $res = "";
    $res .= "<h2>".$lang{"mm_mode_new"}."</h2>\n";
    $res .= generateForm($admin,$department,$language,$basicdir,"",$session,
        $metadata);

    # Ergebnis zurückliefern;
    return $res;
}

```

```

# -----
# - Routine:
# - metamaker
# - Beschreibung:
# - Je nach übergeben Modus in dem sich der Metamaker befindet, wird die
# - HTML-Seite zur entsprechende Aktion (wie beispielsweise Bestätigung,
# - Eingabe oder Verarbeitung von Metadaten) erzeugt und zurückgeben.
# - Parameter:
# - admin      : Name des Administrators (oder leer, falls normaler Benutzer)
# - mode       : Modus in dem sich der Metamaker befindet.
# - department : Fachbereich
# - language   : Sprache
# - metadir    : Verzeichnis in denen der Metamaker die Daten lesen/schreiben
# -            soll
# - metafile   : Datei im Verzeichnis metadir, die der Metamaker verarbeiten
# -            soll
# - session    : Session-Variable von KLUEDO
# - Rückgabe:
# - HTML-Text
# -----

sub metamaker {
  my ($admin,$mode,$department,$language,$metadir,$metafile,$formvars,
      $session) = @_;
  my ($res,$env,$basicdir);

  $env = Environment();
  $basicdir = $env{DEPTS}."/".$department";
  $res = "";
  if ($mode eq "confirmedit") {
    $res .= confirmedit($admin,$department,$language,$basicdir,$metadir,
        $metafile,$session);
  }
  elsif ($mode eq "edit") {
    $res .= edit($admin,$department,$language,$basicdir,$metadir,$metafile,
        $formvars,$session);
  }
  elsif ($mode eq "checkedit") {
    $res .= checkedit($admin,$department,$language,$basicdir,$metadir,
        $metafile,$formvars,$session);
  }
  elsif ($mode eq "save") {
    $res .= save($admin,$department,$language,$basicdir,$metadir,$metafile,
        $formvars,$session);
  }
  elsif ($mode eq "new") {
    $res .= newmetadata($admin,$department,$language,$basicdir,$metadir,
        $metafile,$session);
  }
  else {

    $res .= "<H2>Unerlaubter Modusaufwurf (".$mode.") des Metamakers</H2>";
    $res .= "Bitte kontaktieren Sie den Administrator.";
  }

  # Ergebnis zurückliefern
  return $res;
}

```

```

# -----
# - Routine:
# - main
# - Beschreibung:
# - Die Hauptroutine des Metamakers wertet die Befehlszeilenargumente aus
# - und ruft den Metamaker entsprechend auf.
# - Parameter (hier implizit die Befehlszeilenargumente):
# - mode       : Modus in dem sich der Metamaker befindet.
# - department : Fachbereich
# - language   : Sprache
# - session    : Session-Variable von KLUEDO
# - formvars   : Variablendeklarationszeile in der Form
# -            "x1 = y1 & x2 = y2 & ..."
# - metadir    : Verzeichnis in denen der Metamaker die Daten lesen/schreiben
# -            soll
# - metafile   : Datei im Verzeichnis metadir, die der Metamaker verarbeiten
# -            soll
# - Rückgabe / Ausgabe:
# - HTML-Text
# -----

sub main {
  my ($res,$mode,$department,$language,$metadir,$metafile,$formvars);
  my ($session,$admin);

  # Programmaufruf prüfen
  $mode = defined($ARGV[0]) ? shift(@ARGV) : "";
  $department = defined($ARGV[0]) ? shift(@ARGV) : "";
  $language = defined($ARGV[0]) ? shift(@ARGV) : "";
  $session = defined($ARGV[0]) ? shift(@ARGV) : "";
  $formvars = defined($ARGV[0]) ? shift(@ARGV) : "";
  $metadir = defined($ARGV[0]) ? shift(@ARGV) : "";
  $metafile = defined($ARGV[0]) ? shift(@ARGV) : "";
}

```



```
if ((!$mode) || (!$department) || (!$language)) {
  $res .= "<H2>Unerlaubter Aufruf des Metamakers</H2>";
  $res .= "Bitte kontaktieren Sie den Administrator.";
} else {
  $admin = "";
  if ($mode =~ /^admin=(.*?):(.*?)$/i) {
    $admin=$1;
    $mode = $';
    $admin = $admin . "@ub.uni-kl.de";
  }
  $res .= metamaker($admin,$mode,$department,$language,$metadir,$metafile,
    $formvars,$session);
}

# Ergebnis ausgeben
print $res;
}

# -----
# - Aufruf der Hauptroutine
# -----

main();
```



```

$res .= "<font color=\"#003399\">".$lang{Year}."</font>&nbsp;</td>";
$res .= "<td valign=top bgcolor=\"#FFFCC\">";
$res .= $year;
$res .= "</td></tr>";
}

# Ausgabe der Referenz
if ($reference) {
$res .= "<tr><td valign=top bgcolor=\"#6699cc\">";
$res .= "<font color=\"#003399\">".$lang{Reference}."&nbsp;</font></td>";
$res .= "<td valign=top bgcolor=\"#FFFCC\">";
$res .= $reference;
$res .= "</td></tr>";
}

# Ausgabe der verfügbaren Dateiformate
$res .= "<tr><td valign=top bgcolor=\"#6699cc\">";
$res .= "<font color=\"#003399\">".$lang{Documents}."&nbsp;</font>";
$res .= "</td>";
$res .= "<td bgcolor=\"#FFFCC\">";
$formats = getFileformats(\%rqpairs);
if ($formats) {
$res .= $formats;
} else {
$res .= $lang{msg_nodocument};
}
$res .= "</td></tr>";
$res .= "<tr><td colspan=2>&nbsp;</td>";
$res .= "</tr></tr>";

### Editmaske erweitern
print EDITLIST "<tr bgcolor=\"#\">";
if (($count % 2) eq 0) {
print EDITLIST"ffffff";
} else {
print EDITLIST"ffffcc";
}
print EDITLIST ">";
print EDITLIST "<td valing=top width=\"1%\">";
print EDITLIST "<a href=../metamaker_.$language_.html";
print EDITLIST "?modus=confirmedit";
print EDITLIST "&metadir=$metadir";
print EDITLIST "&metafile=$metafile";
print EDITLIST ">.$metafile.</a>";
print EDITLIST "</td>";
print EDITLIST "<td>&nbsp;</td>";
print EDITLIST "<td valign=top width=\"99%\">";
print EDITLIST $doctitle;
print EDITLIST "</td></tr>\n\n";

# Werte zurücksetzen
undef %rqpairs;

# Ergebnis zurückliefern
return $res;
}

# -----
# - Routine:
# - RefreshLists
# - Beschreibung:
# - Aktualisiert die übergebene Veroeffentlichungsliste
# - Parameter:
# - language : Sprache, in der die Veroefflichungsliste publiziert
# - werden soll (z.B. "de","en")
# - department_path: Pfad des Fachbereiches
# - serie : interner Name der zu aktualisierenden Liste
# - pagetitle : Seitentitel (Überschrift)
# - documenttitle : Dokumenttitel (Präfix)
# - edittitle : Bearbeitungstitel (Überschrift)
# - refreshhit : ist 1, falls die Liste aktualisiert werden soll
# - (ansonsten wird die Liste nur in dem Verzeichnis der
# - Reihen gelinkt).
# -----

sub RefreshSerie {
my($language,$department_path,$department,$serie,$pagetitle,$documenttitle,
$edittitle,$refreshhit) = @_;
my(@allfiles,@filelist,$file,$count,$newfiles,$metaname,$metadir,$number);
my($tempstr,$reslist,$tmpline,@temp);

# Initialisierung und Benutzermitteilung, Ermitteln der Dateien einer Reihe
$reslist = "";
$metaname = $serie."_.$language_.html";
if ($refreshhit eq 1) {
if ($serie eq "_incoming") {
print " - Erstelle Liste der Neuzugänge (Eingangsverzeichnis)\n";
$newfiles = 1;
$metadir = "Metadaten_Eingang";
opendir(DIR,"$department_path/Metadaten_Eingang/");
@allfiles = grep /html/,readdir DIR;
closedir(DIR);
} else {
print " - Erstelle Liste der Reihe \"$serie\"...\n";
$newfiles = 0;

```

```

$metadir = "Metadaten";
opendir(DIR,"$department_path/Metadaten/");
@allfiles = grep /html/, grep /^$serie_*/,readdir DIR;
closedir(DIR);
}

# Alle Dateinummern einer Reihe in die Abarbeitungsliste aufnehmen und
# sortieren (aus: Perl Kochbuch, p126)
@filelist = map { $_->[1] }
sort { $b->[0] <=> $a->[0] }
map { [ /_(\d+)/, $_ ] } @allfiles;

# Editliste generieren
open (EDITLIST,"> ".$department_path."/Listen/edit_". $metaname);

# Headpart der Editliste
print EDITLIST "<?\n";
print EDITLIST "require(\"$env{HOME}/php/init.pinc\");\n\n";
print EDITLIST "do_header(\"$edittitle\");\n";
print EDITLIST "do_body(\"<h2>$edittitle</h2>\");\n";
print EDITLIST "?>\n";
print EDITLIST "\n";
print EDITLIST " <table cellpadding=1 cellspacing=0 border=0 ";
print EDITLIST "width=\%100%\\" bgcolor=\%6699cc\ ">\n";
print EDITLIST " <tr>\n";
print EDITLIST " <td><I>".$pagetitle."</I></TD>\n";
print EDITLIST " </tr>\n";
print EDITLIST " </table>\n";
print EDITLIST " <table cellpadding=0 cellspacing=0 border=0 ";
print EDITLIST "width=\%100%\ ">\n";

# Veröffentlichungsliste
$reslist .= "<?\n";
$reslist .= "require(\"$env{HOME}/php/init.pinc\");\n";
$reslist .= "require(\"$env{HOME}/php/navigation.pinc\");\n\n";
$reslist .= "\$docs = array(\"\"";
$count=0;
foreach $file(@filelist) {

    # Anzahl der Veröffentlichung pro Liste mitzählen
    $count ++;

    # Nummer der Datei durch Zerlegen herausfinden...
    ($tempstr,) = split('.', $file);
    @temp = split('_', $tempstr);
    $number=$temp[$#temp];

    # Dokumenttitel festlegen
    if ($newfiles eq 0) {
    $tempstr = $documenttitle." ".$number;
    }
    else {
    $tempstr = $file;
    }

    # Metadatenfile-Informationen hinzufügen
    $tmpline = AddMetadataInformation($language,$department,$department_path,
    $metadir,$file,$tempstr,$count);
    $tmpline =~ s/"/\\/"/gi;
    $reslist .= "\n \\".$number." : ".$tmpline."\"";

}
$count++;
$reslist .= ");\n";
$reslist .= "do_header(\"$pagetitle\");\n";
$reslist .= "do_body(\"<h2>$pagetitle</h2>\");\n";
$reslist .= "do_navigation(\".$count.\" ,\$docs, \$POST);\n";
$reslist .= "do_footer(); ?>\n";

# Liste schreiben
open (LISTFILE,"> ".$department_path."/Listen/". $metaname);
print LISTFILE $reslist;
close (LISTFILE);

# Tailpart der Edit-Liste
print EDITLIST " </table>\n";
print EDITLIST "<? do_footer(); ?>\n";
}

# Erweitern der Editmaske
print EDITMASK "<li><a href=\"";
print EDITMASK "edit_". $metaname;
print EDITMASK "\">$pagetitle</a></li><p>\n";

# Werte zurücksetzen
undef @allfiles;
undef @filelist;
}

```

```

# -----
# - Routine:
# - RefreshLists
# - Beschreibung:
# - Aktualisiert die Veroeffentlichungslisten eines Fachbereiches
# - Parameter:
# - language:      Sprache, in der die Veroefflichungsliste publiziert werden
# -                soll (z.B. "de","en")
# - department:   Name des Fachbereiches (wie Sub-Pfad auf Dateiebene)
# -                z.B. "Mathematik"
# - filter_serie : Angabe der zu aktualisierenden Serie (wie in Conf-Datei)
# -                oder leer für alle Serien
# -----

sub RefreshDepartment {
my($language,$department,$filter_serie) = @_ ;
my($url,$count,$line,$serie,@fields,$parameter,$pagetitle,$documenttitle);
my(%series,$languageOK,$editpage,$depname,$title,$CONF_FILE,$fehler);
my($refreshhit);

# Benutzermittteilung
print "\n\nAktualisiere Fachbereich \"$.department.\"";
print "  Sprachcode : \"$.language.\"";

# Initialisierung: Übersetzungstabelle, Pfade, Variablen
$lang=Translations($language,$env{CONF}."/language.conf");
$url = $env{DEPTS}."/".$department;
$depname = "dep_" . $department;
$title=$lang{editShadowfiles}." ".$lang{$depname};
$fehler=0;

# Beginn der Erstellung der Auswahl-Seite "edit_XX"
open (EDITMASK,"> ".$url."/Listen/edit_".$language.".html");
print EDITMASK "<?\n";
print EDITMASK "require(\"$env{HOME}/php/init.pinc\");\n\n";
print EDITMASK "do_header(\"$title\");\n";
print EDITMASK "do_body(\"<h2>$title</h2>\");\n";
print EDITMASK "?>\n";

# Konfigurationsdatei einlesen
%series = Translations($language,$url/conf/Listen.conf");
$count = 0;
$editpage = "";
while (($serie,$parameter)=each %series) {

# Aufsplitten der Konfigurationszeile (der aktuellen Sprache)
@fields = split(":",$parameter);
$pagetitle = $fields[0];
$documenttitle = $fields[1];

# Liste der jeweiligen Reihe aktualisieren
# Bedingung: Korrekte Anzahl der Argumente (gdw. Letztes
# Konfigurationsfeld nicht leer)
if ($fields[1])
{
$refreshhit = 0;
if (($filter_serie eq "") or ($filter_serie eq $serie) or ($serie eq "_incoming")) {
$refreshhit=1;
$count++;
}
RefreshSerie($language,$url,$department,$serie,$pagetitle,
$documenttitle,$title,$refreshhit);
}
}

# Edit-Seite-Erstellung abschließen
print EDITMASK "<? do_footer(); ?>\n";
close(EDITMASK);

# Benutzermittteilung
print "Es wurden ".$count." Reihen aktualisiert.\n\n";
}

# -----
# - Hauptprogramm
# -----

sub main {
my($department, $filter_serie,$lang,@lInfo);

# Programmaufruf prüfen
print "(Skript: $0)\n";
$department = defined($ARGV[0]) ? shift(@ARGV) : "";
$filter_serie = defined($ARGV[0]) ? shift(@ARGV) : "";
if (!$department) {
print "Aufruf: NeueListen.pl Fachbereich [Serie]\n";
} else {

# Einlesen der Umgebungsvariablen
%env=Environment();

# Dateirechte auf 664 setzen
umask 002;

# Aktualisieren der Listen für alle bekannten Sprachcodes
@lInfo = getLanguagesInfo();

```

```
for ($lang = 0; $lang < $lInfo[0]; $lang++) {
    RefreshDepartment(getLanguageCode($lang), $department, $filter_serie);
}

# Alle Rechte auf kluedo.kluedowww setzen
chown 826, 600, glob($env{DEPTS}."/".$department."/Listen/*");
}
}

&main;
```

## 6. Literatur

- [1] Suchmaschine *Altavista*  
<http://www.altavista.de>
- [2] The Apache Software Foundation  
<http://www.apache.org>
- [3] Bolze L., Eichler-Liebenow C.: *Metadaten für die Lehre*. Projektbeschuß. Juni 1998. Universität Halle  
[http://www.mathematik.uni-halle.de/MathNet/teaching\\_metadata.html](http://www.mathematik.uni-halle.de/MathNet/teaching_metadata.html)
- [4] Bowman C.M., Dharap C., Baruah M., Camargo B., Potti S.: *A file system for information management*. Proceedings of the Conference on Intelligent Information Management Systems. Juni 1994  
<ftp://ftp.cse.psu.edu/pub/bowman/doc/iims.ps.Z>
- [5] Bray T., Hollander D., Layman A.: *Namespaces in XML*. W3C Consortium. Januar 1999.  
<http://www.w3.org/TR/1999/REC-xml-names-19990114>
- [6] Bray T., Paoli J., Sperberg-McQueen C.M.: *Extensible Markup Language (XML) 1.0*. W3C Issues Recommendation. Februar 1998  
<http://www.w3.org/TR/1998/REC-xml-19980210>
- [7] Brickley D., Guha R.V.: *Resource Description Framework (RDF) Schema Specification*. W3C Proposed Recommendation. März 1999  
<http://www.w3.org/TR/1999/REC-rdf-schema-19990303>
- [8] Bundesministerium für Bildung und Forschung  
<http://www.bmbf.de>
- [9] *CARMEN: Content Analysis, Retrieval and Metadata: Effective Networking*. SFM Inhaltserschließung im Rahmen des Global-Info-Projektes  
<http://www.mathematik.uni-osnabrueck.de/projects/carmen/carmen.html>
- [10] Dalitz W., Grötschel M., Heyer G., Lügger J., Sperber W.: *Informationsdienste für die Mathematik im Internet (Math-Net)*. DFN-Vorhaben DT11. Technical Report TR 96-13. Dezember 1996. Konrad-Zuse-Zentrum für Informationstechnik Berlin.  
<ftp://ftp.zib.de/pub/zib-publications/reports/TR-96-13.ps>
- [11] Daly J., Mitchell N., Lloyd A.: *W3C Issues Recommendation for Resource Description Framework (RDF)*. Introduces model for defining and organizing information. Press Release.  
<http://www.w3.org/Press/1999/RDF-REC>

- [12] Danzig P.B., Hall R.S., Schwartz M.F.: *A Case for Caching File Objects Inside Networks*. Technical Report CU-CS-642-93. März 1993. University of Colorado at Boulder  
<ftp://ftp.cs.colorado.edu/pub/techreports/schwartz/FTP.Caching-PS>
- [13] Deutsche Forschungsgemeinschaft  
<http://www.dfg.de>
- [14] Deutsches Forschungsnetz  
<http://www.dfn.de>
- [15] Dublin Core WWW Server.  
[http://purl.oclc.org/metadata/dublin\\_core](http://purl.oclc.org/metadata/dublin_core)
- [16] *Dublin Core in der Interpretation des BSZ (Regelwerk)*. Bibliothekservice-Zentrum Baden Württemberg  
<http://www.bsz-bw.de/diglib/medserv/konvent/metadat/dcregelw.html>
- [17] Suchmaschine *Fireball*  
<http://www.fireball.de>
- [18] Global-Info-Projekt: Globale Elektronische und Multimediale Informationssysteme für Naturwissenschaft und Technik  
<http://www.global-info.org>
- [19] Grötschel M., Dalitz W., Winczewski V.: *Math-Net: Ein Informationssystem der Mathematik*. Preprint SC 98-25. Oktober 1998. Konrad-Zuse-Zentrum für Informationstechnik Berlin.  
<http://elib.zib.de/preprints/shadows/SC-98-25.html>
- [20] Hardy D.R., Schwartz M.F., Wessels D.: *Harvest User's Manual*. Technical Report CU-CS-743-94. Januar 1996. University of Colorado at Boulder.  
<http://www.tardis.ed.ac.uk/harvest/docs/>
- [21] Hardy D.R., Schwartz M.F.: *Customized Information Extraction as a Basis for Resource Discovery*. Technical Report. Februar 1995. University of Colorado at Boulder.  
<ftp://ftp.cs.colorado.edu/pub/techreports/schwartz/Essence.Jour.ps.Z>
- [22] Hardy D.R., Schwartz M.F.: *Essence: A Resource Discovery System based on Semantic File Indexing*. Technical Report. Januar 1993. University of Colorado at Boulder.  
<ftp://ftp.cs.colorado.edu/pub/techreports/schwartz/Essence.Conf.ps.Z>
- [23] Suchmaschine *Infoseek*  
<http://www.infoseek.de>



- [24] International Standard. *ISO 8601. Data elements and interchange formats - Information interchange - Representation of dates and times*. Juni 1988. Reference number 8601:1988 (E).  
<http://www.iso.ch/markete/8601.pdf>
- [25] Kahle B., Medlar A.: *An information system for corporate users: Wide Area Information Servers*. November 1991  
<ftp://think.com/wais/wais-corporate-paper.text>
- [26] Dokumentenserver *KLUEDO*: Kaiserslauterer uniweiter elektronischer Dokumentenserver  
<http://kluedo.ub.uni-kl.de>
- [27] Lassila O., Swick R.R.: *Resource Description Framework (RDF) Model and Syntax Specification. W3 Recommendation*. Februar 1999  
<http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>
- [28] Lehmann A.: *Empfehlungen zum Aufbau eines Servernetzwerkes für elektronische Hochschulpublikationen*. September 1998. Arbeitsgruppe Volltexte und Hochschulpublikationen des Ministeriums für Wissenschaft, Forschung und Kunst Baden-Württemberg.  
<http://www.bsz-bw.de/depot/dokersch/6800000/6853000/6853597k.html>
- [29] Lower Saxony State and University Library Göttingen (Germany). *Metadata Server*  
<http://www2.sub.uni-goettingen.de>
- [30] Manber U., Glimpse S. W.: *A tool to search through entire file systems*. Proceedings of the USENIX Winter Conference. Technical Report 93-34. Januar 1994.  
<ftp://cs.arizona.edu/reports/1993/TR93-34.ps.Z>
- [31] Martin N.: *Die elektronische Publikation von Dissertationen - Bibliothekarische Anforderungen*. September 1998. Humboldt-Universität zu Berlin  
<http://dochoost.rz.hu-berlin.de/epdiss/martin/martin/martin2.html>
- [32] Math-Bib-Net-Projekt: Aufbau nutzerorientierter verteilter Informationsstrukturen in Kooperation von Bibliotheken und Fachbereichen am Beispiel des Faches Mathematik.  
<http://harvest.sub.uni-goettingen.de>
- [33] *Mathematics Subject Classification*. Revision 1991.  
<http://www.ams.org/msc/home.html>
- [34] *Mathnet Internet Information Services for Mathematicians*  
<http://www.mathnet.de>
- [35] Math-Net-Projekt: Informationsdienste für die Mathematik im Internet  
<http://www.mathnet.de>

- [36] Münz S.: *SELFHTML 7.0*. April 1998  
<http://www.teamone.de/selfaktuell/>
- [37] Ohst D.: *Dateiformate für das elektronische Publizieren*. März 1998. Humboldt-Universität zu Berlin.  
<http://amor.rz.hu-berlin.de/~h0444saa/didi/formate.html>
- [38] PHP: Hypertext Preprocessor  
<http://www.php.net>
- [39] Rusch-Feja D.: *Metadata-Tags zur Erschließung von Internetquellen*. Dezember 1996. Max-Planck-Institut für Bildungsforschung.  
<http://www.mpib-berlin-mpg.de/DOK/metatagd.html>
- [40] Schön E.: *Das Ressource Description Framework (RDF) - ein neuer Weg zur Verwaltung von Metadaten im Netz*. Oktober 1998. Technische Universität Ilmenau.  
<http://home.t-online.de/home/eckhard.schoen/worksh98.htm>
- [41] Softwarehaus SuSE  
<http://www.suse.de>
- [42] *Syntax des Dublin Core*. Bibliotheksservice-Zentrum Baden Württemberg  
<http://www.bsz-bw.de/diglib/medserv/konvent/metadat/dcsyntax.html>
- [43] *USMARC Code List for Languages*. Library of Congress.  
<http://lcweb.loc.gov/marc/languages/>