



Vortrag:

Dokumentenserver 2.0 – Anwendung von neuen Navigationsverfahren, Suchverfahren und Nutzung von Methoden zum Metadatenaustausch.

Vortragender: Trung Kien Chu
Betreuer : David Aumüller
Juni 2008

1



Inhalt

Ausgangssituation

Anforderungen an einen Dokumentenserver

Aufbau eines Dokumentenservers

- Auswahl eines Content-Management-System
- Verwaltungskonzept von Drupal

Navigations- und Suchverfahren

- Faceted Search
- Tag Cloud

Schnittstelle für Metadatenaustausch

- OAI-PMH

Aktueller Stand des Dokumentenservers der Fakultät für Mathematik und Informatik

2



Ausgangssituation

Informationsflut

- Studie der Universität von Berkeley: jede Person produziert im Jahr durchschnittlich 800 Megabyte an Daten \approx zehn Regalmeter mit Büchern
- Ein großen Teil der Informationen aus wissenschaftlichen Publikationen.
- Die Anzahl der Wissenschaftler steigt.

3



→ Bedarf an Archivierung und Verwaltung von Publikationen

→ Der Einsatz von einem Dokumentenserver für die Haltung von elektronischen Publikationen (oder i.A. Dokumente) ist unverzichtbar.

4



Anforderungen an einen Dokumentenserver

- Dokumente für lange Zeit speichern
- Dokumente mit ihren Metadaten verwalten
- Unterstützung der Recherche auf großer Datenmenge
- Verwendung von offenen Schnittstellen für andere Systeme um die Kommunikation und Datenaustausch mit anderen Systemen zu ermöglichen

5



Mögliche Umsetzung

- **Frage 1: Wie wird ein Dokumentenserver aufgebaut?**
- Lösung: Einsatz von einem Content-Management-System

- **Frage 2: Wie kann ein Dokumentenserver die Recherche auf großer Datenmenge unterstützen?**
- Lösung: Einsatz von neuen Navigation- und Suchverfahren wie Faceted Search und Tag Cloud

- **Frage 3: Wie wird die Kommunikation mit anderen Dokumentenserver und mit anderen Systemen möglich?**
- Lösung: Einsatz von Metadaten-Export (offline) oder von einem Protokoll für die Metadaten austausch (online).

6

Wir sind hier

Ausgangssituation

Anforderungen an einen Dokumentenserver

Aufbau eines Dokumentenservers

- Auswahl eines Content-Management-System
- Verwaltungskonzept von Drupal

Navigations- und Suchverfahren

- Faceted Search
- Tag Cloud

Schnittstelle für Metadaten austausch

- OAI-PMH

Aktueller Stand des Dokumentenserver

der Fakultät für Mathematik und Informatik



7

Aufbau eines Dokumentenservers

Auswahl eines Content-Management-Systems

AIGAION oder DRUPAL



8

Aigaion

Web-Content-Management-System für bibliografische Dokumente

- Open-Source-Projekt
- erste Version 1.1.0 im Juli 2006
- aktueller Stand Version 2.0.2 Beta März 2008
- Website: <http://www.aigaion.nl>
- Dokumentenserver mit Aigaion-Demo:
<http://db15.informatik.uni-leipzig.de/aigaion2>

9

Aigaion

Vorteile:

- spezielles System für die Verwaltung von bibliografischen Daten von Publikationen
- gutes Eingabesystem (optimiert für diesen Zweck)
- gutes Benutzer- und Rechteverwaltungssystem
- mächtige Volltextsuche mit gutem Suchergebnis
- Eingebautes Modul für Bibtex-Export und Bibtex-Import

10



Aigaion

Nachteile:

- Format von Publikationen ist unflexibel
- Kein manueller Mapping-Vorgang beim Import
- Keine mehrsprachige Benutzeroberfläche
- Keine Unterstützung von Faceted Search
- Erweiterung vom fixem Entwicklungsteam → individuelle Anpassung zu aufwendig

11



Drupal

Web-Content-Management-System

- Ein verbreitetes Open-Source-Projekt, u.a. Communities, Blogs, University departments (>20 in Stanford*) ...
z.B.: Die Welt, Die Zeit, Eclipse, Yahoo!
- Website: <http://drupal.org/>
- Wird eingesetzt für Dokumentenserver der Fakultät für Mathematik und Informatik.
- Dokumentenserver mit Drupal-Biblio-Demo:
<http://db15.informatik.uni-leipzig.de/doxbiblio/>
- Dokumentenserver mit Drupal-CCK-Demo:
<http://db15.informatik.uni-leipzig.de/dox2/>

* <https://techcommons.stanford.edu/drupal-book-toc/sites-using-drupal-stanford> 12



Drupal

Vorteile:

- Datenbank-Abstraktion-Konzept, das Standard-SQL in SQL-Dialekten übersetzt. Zurzeit werden 2 Datenbanksysteme *MySQL* und *PostgreSQL* unterstützt (*Oracle* und *DB2* in beta).
- Modul-Konzept: Funktionalität erweiterbar durch die Drupal-Module.
- eine große und aktive Entwicklung-Community zur Erweiterung der Funktionalität.

13



Vorteile: (Fortsetzung)

- viele Modul-Pakete zum Aufbau vom Verwaltungssystem mit bibliografischen Dokumenten
 - Biblio** : vorgefertigtes Paket
 - CCK** : manuelle Konfiguration von Metadaten-Felder zur Verwaltung von Dokumenten
- Module für Implementierung neuer Navigation- und Suchverfahren (Faceted Search, Tag Cloud) verfügbar.
- Indexierung der Inhalte für Volltextsuche.
- Eingebautes Taxonomy-Modul für die hierarchische Kategorisierung von Inhalte

14

Drupal

Nachteil:

starke Abhängigkeit der Module untereinander sowie vom Basispaket Drupal-Core.

15

Wir sind hier

Ausgangssituation

Anforderungen an einen Dokumentenserver

Aufbau eines Dokumentenservers

- Auswahl eines Content-Management-System
- Verwaltungskonzept von Drupal

Navigations- und Suchverfahren

- Faceted Search
- Tag Cloud

Schnittstelle für Metadaten austausch

- OAI-PMH

Aktueller Stand des Dokumentenserver

der Fakultät für Mathematik und Informatik

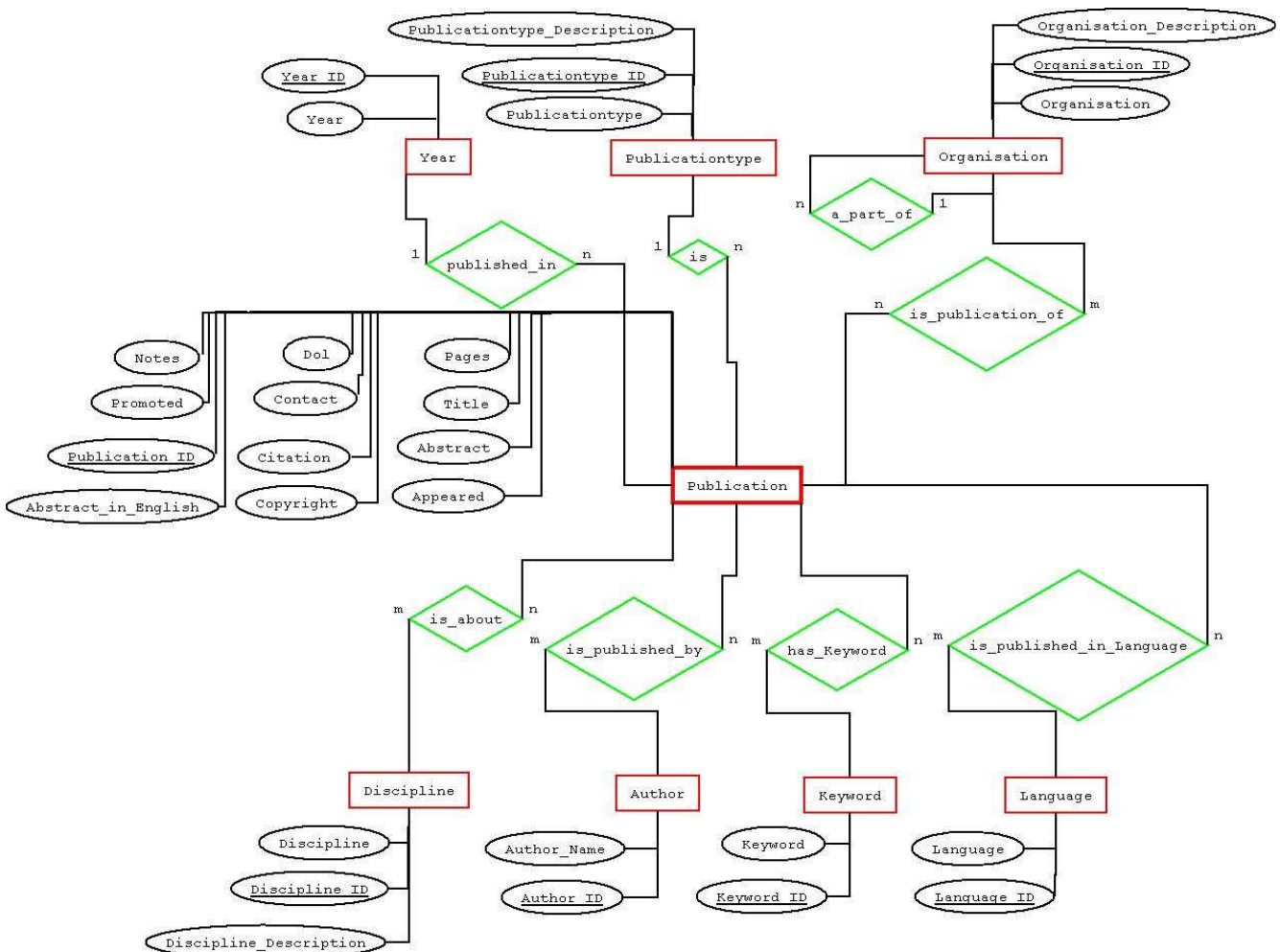


16

Verwaltungskonzept von Drupal

Anforderung für die Verwaltung von Dokumenten

17






Content-Modell

Drupal-Konzept zur Verwaltung von Content

- Content als Knoten (en. node), der als Instanz von einem Knotentyp (en. node type) ist.
- Vordefinierte Knotentypen: Blog entry, Book Page, Comment, Forum, Page, Poll, Story.
- Speicherung allgemeiner Knoteninformationen in der Tabelle node (Typ, Status, Titel, User-ID, Datum Erzeugung/letzte Änderung).
- Speicherung zusätzlicher Knoteninformationen in andere Tabellen

19

- 
-
- Speicherung der Node-Anhänge in Tabelle *files*

- Abbildung in Drupal

Content → Knoten

Content-Metadaten → zusätzliche Knoteninformationen

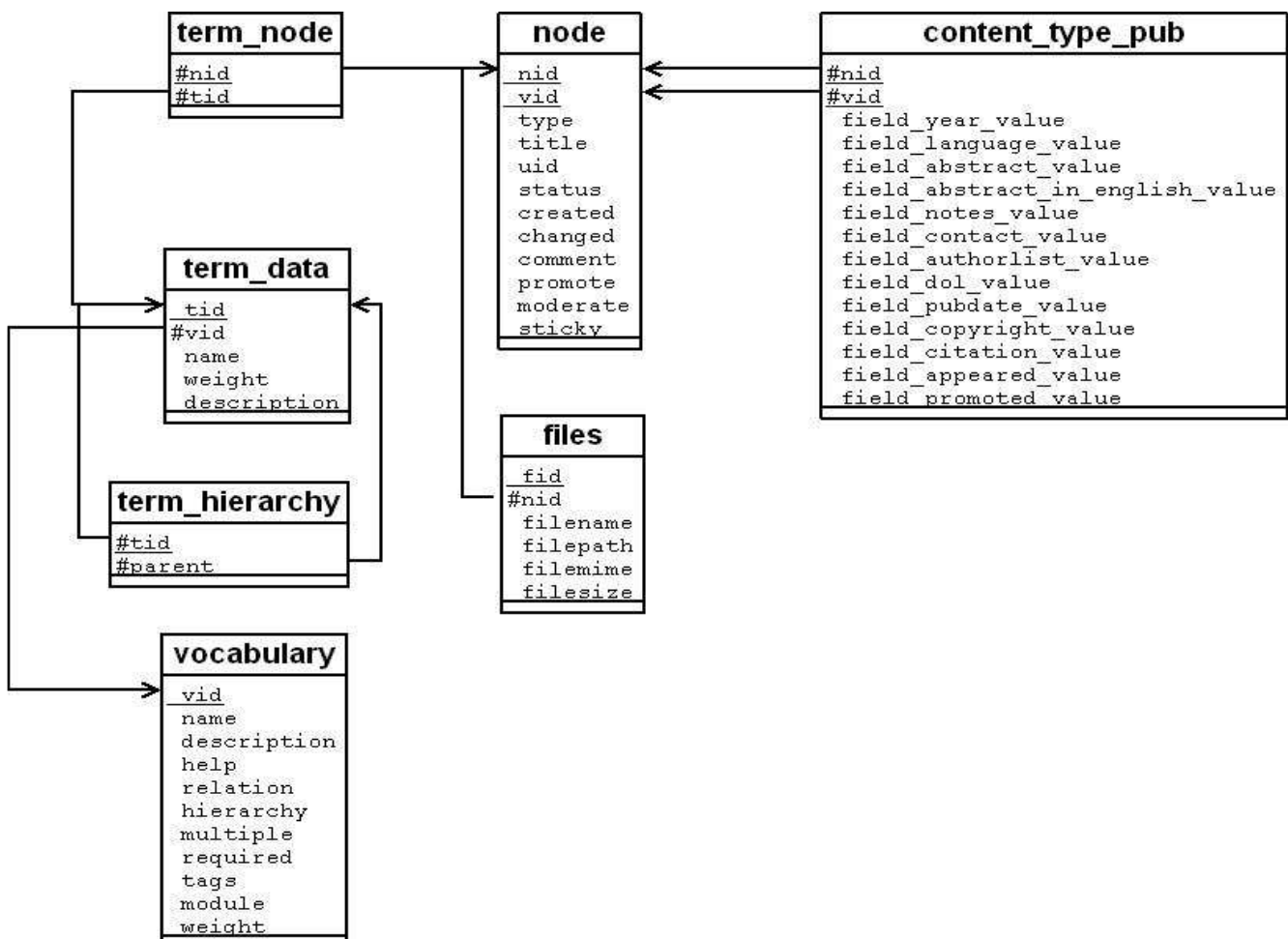
Content-Inhalt → Abhängig von Knotentyp als node-body oder node-Anhänge.

20

Umsetzung der Anforderung mit Drupal und seinem Modulpaket CCK*

- Erstellung eines neuen Knotentyp *Publication*
- CKK fügt eine Tabelle *content_type_pub* ein

21





Wir sind hier

Ausgangssituation

Anforderungen an einen Dokumentenserver

Aufbau eines Dokumentenservers

- Auswahl eines Content-Management-System
- Verwaltungskonzept von Drupal

Navigations- und Suchverfahren

- Faceted Search
- Tag Cloud

Schnittstelle für Metadatenaustausch

- OAI-PMH

Aktueller Stand des Dokumentenserver

der Fakultät für Mathematik und Informatik



23



Navigations- und Suchverfahren

Faceted Search

24



Navigations- und Suchverfahren

Faceted Search in Drupal

Was versteht man unter Faceted Search?

- ein Modul-Paket zur Funktionserweiterung des Content-Management-Systems *Drupal*
(http://drupal.org/project/faceted_search)
- nutzt die Idee vom Ordnungsprinzip *Fassettenklassifikation*
- nutzt die Idee vom Flamenco Project
(<http://flamenco.berkeley.edu/index.html>)
- kombiniert die Navigation und Volltextsuche.
- unterstützt den Benutzer, das Suchergebnis zu verfeinern oder zu erweitern
- hilft dem Benutzer, seine bisherigen Suchaktionen zurückzufolgen. Der Benutzer hat dadurch die Orientierung während der Suche.

25



Exkurs: Ordnungsprinzipien

Fassettenklassifikation

- Objekt unter verschiedene Aspekte betrachten
(z.B.: Autor, Erscheinungsjahr, Herausgeber für Dokumente)
- verschiedenen Aspekte als Fassetten
- Instanz von einer Fassette ist eine Klasse
- Wenn das Dokumentensystem n Fassetten hat, so müssen die Dokumente in genau n Fassetten unterteilt werden.

26



Faceted Search im Dokumentenserver

- Ordnungssystem nach dem Ordnungsprinzip *Fassettenklassifikation*
- Faceted Search präsentiert eine Benutzerschnittstelle um mit diesem Ordnungssystem zu interagieren.
- Ein Dokument wird in folgende Fasette eingeteilt:
 - Type of work (Dokumenttyp)
 - Language (Sprache)
 - Authors (Autor)
 - Year (Erscheinungsjahr)
 - Discipline (Sachgebiet)
 - Organisation (Institut)
 - Keywords (Schlüsselwort)

27

- 
-
- hierarchische Fassettenstruktur durch hierarchische Kategoriestructur unterstützt
 - z.B. bei der Fasette *Organisation*

Univ. Leipzig → Fassetteninstanz

-- FMI → untergeordnete Fassetteninstanz (Tief 1)

----- Inf → untergeordnete Fassetteninstanz (Tief 2)

----- Math → untergeordnete Fassetteninstanz (Tief 2)

-- Med → untergeordnete Fassetteninstanz (Tief 1)

-- Phys → untergeordnete Fassetteninstanz (Tief 1)

28

Wie funktioniert Faceted Search im Dokumentenserver der Fakultät für Mathematik und Informatik?

Beispiel: Gesucht werden Dokumente vom Autor *Apel Joachim* mit Schlüsselwort *algorithm*.

1. Schritt : Von der Fassade *Authors* wird die Klasse *Apel, Joachim* ausgewählt.

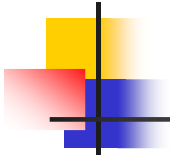
29

- Im Block *Guided Search* bietet *Faceted Search* die Möglichkeit das Ergebnis weiter zu verfeinern.
- Im Block *Current Search* bietet *Faceted Search* die Möglichkeit die bisherigen Suchbegriffe zu finden.
- Mit dem Kreuz (x) neben jedem Suchbegriff im Block *Current Search* kann man den Begriff entfernen, so dass das Ergebnis erweitern wird.

The screenshot shows the 'Dokumentenserver Fakultät für Mathematik und Informatik' interface. The search bar contains 'Search: Apel, Joachim' and shows '13 results'. The results are displayed in a table with columns for Title and Year. The 'Guided search' block on the left shows filters for Type of work, Language, Authors, Year, and Discipline. The 'Current search' block on the right shows the active search criteria: 'Authors: Apel, Joachim'.

Title	Year**
Apel, Joachim	1998
The Computation of Grobner Bases Using an Alternative Algorithm.	
Apel, Joachim	1998
Computational Ideal Theory in Finitely Generated Extension Rings	
Apel, Joachim, Klaus, Uwe	1998
Aspects of Large Scale Symbolic Computation Management	
Apel, Joachim, Klaus, Uwe	1998
A Note on Data Types Supporting Efficient Implementations of Polynomial Arithmetic	
Apel, Joachim, Stokorad, Jürgen; Tworzewski, Piotr; Winiarski, Tadeusz	1997
Term Bases for Multivariate Interpolation of Hermite Type.	
Apel, Joachim	1997
Zu Berechenbarkeitsfragen der Idealtheorie.	
Apel, Joachim	1997
The Theory of Involutive Divisions and an Application to Hilbert Function Computations.	

30



2. Schritt: Das Ergebnis wird mit der Auswahl der Klasse *algorithm* von der Fassetten *Keywords* verfeinert.

31



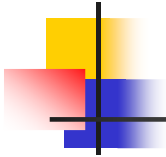
- Das Endergebnis liefert ein Dokument

The screenshot shows a search interface for the 'Dokumentenserver Fakultät für Mathematik und Informatik' at 'UNIVERSITÄT LEIPZIG'. The search query is 'Apel, Joachim, algorithm', resulting in 1 result. The result table shows the following details:

Title	Year
Apel, Joachim Computational Ideal Theory in Finitely Generated Extension Rings	1998

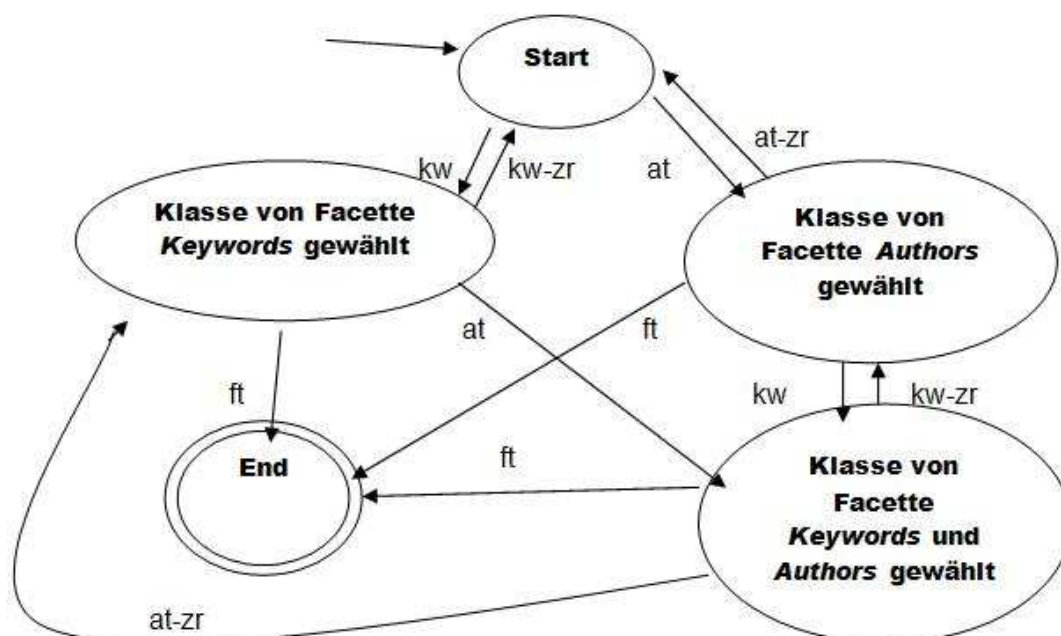
On the left, a 'Guided search' sidebar lists filters: Type of work (Research (1)), Language (English (1)), Authors (all > Apel, Joachim), Year (1998 (1)), Discipline (MATH/INF/NAT (1)), Organisation (Univ. Leipzig (1)), and Keywords (all > algorithm). On the right, a 'Keyword search' section shows the current search filters: [-1] Authors: Apel, Joachim and [-1] Keywords: algorithm.

32



Navigation mit Faceted Search durch einen deterministischen endlichen Automat simulierbar

33



Eingabe

kw.....Wähle eine Klasse aus *Keywords* aus

kw-zr... *Keywords* zurücksetzen

at.....Wähle eine Klasse aus *Authors* aus

at-zr.....*Authors* zurücksetzen

ftFertig

34



Faceted Search kombiniert Navigation und Suche

- Das Ergebnis des Beispiels kann nach dieser Suchstrategie aufgefunden werden.
 1. Schritt: Suche nach *algorithm*
 2. Schritt: Beschränkt das Ergebnis auf Keywords
 3. Schritt: Beschränkt das Ergebnis auf *Apel, Joachim*
→ Das Endergebnis liefert genau ein Dokument.
- *Faceted Search* in Drupal ermöglicht die Kombination der Verfahren: **SUCHE* \Leftrightarrow NAVIGATION***.

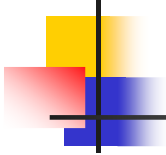
35



Wie ist Faceted Search in Dokumentenserver eingebunden?

- **Verwendete Module:**
 - CCK** (für die Erstellung von spezifischen Feldern)
 - Taxonomie** (für die Bereitstellung hierarchischer Kategorienstruktur)
 - CCK Facets** (für die Erstellung von Fassetten aus CCK-Feldern)
 - Faceted Search**
Nutzt *Taxonomie* und *CCK Facets* für die Bereitstellung von Fassetten und Fassettklassen sowie die Zuordnung von Dokumente zu Fassettklassen.

36



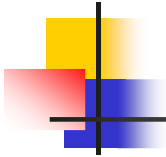
- **Verwendete CCK-Felder:**
 - field_year (Erscheinungsjahr)
 - field_authors (Autoren)
 - field_language (Sprache)
- **Verwendete Taxonomieterme:**
 - Type of work (Publikationstyp)
 - Discipline (Sachgebiet)
 - Organisation (Institut)
 - Keywords (Schlüsselwörter)

37



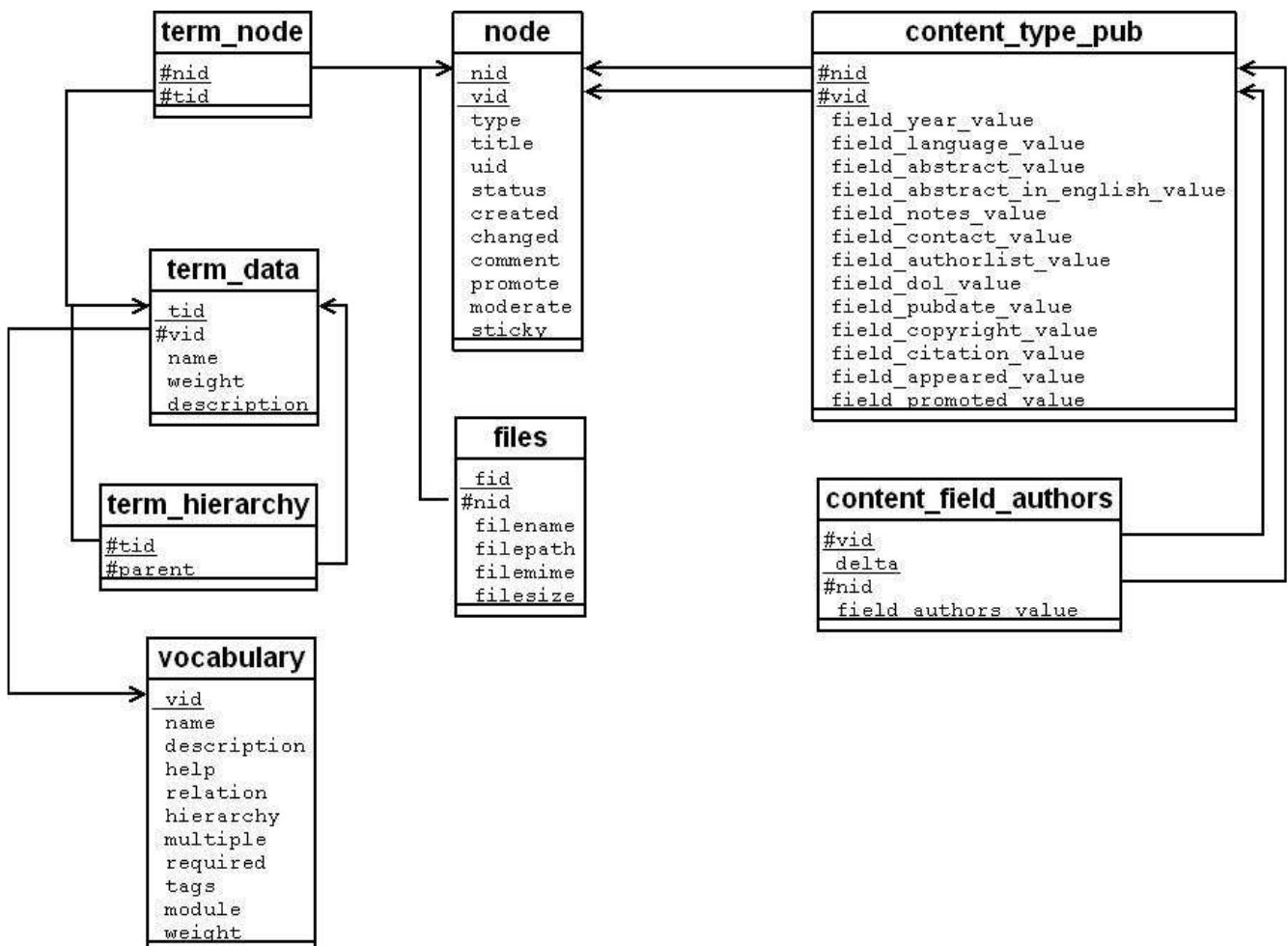
Die Verwendung von *field_authors* (dieses Feld ist nicht in der Tabelle content_type_pub enthalten) ergibt ein Änderung im Datenbankschema.

38



- Ein Author-Extraction-Verfahren wurde verwendet um die Autoren von *field_authorlist* in *field_authors* zu speichern.
- Die Kopplung vom einzelnen Autor mit seinen Publikationen wird in der Tabelle *content_field_authors* mit Hilfe von *vid* und *nid* gewährleistet.

39



Faceted Search in anderen Websites*

- **Nobel Prize Winners**

From nobelprize.org

<http://orange.sims.berkeley.edu/cgi-bin/flamenco.cgi/nobel/Flamenco>

- **UC Berkeley Architecture Image Library**

From the UC Berkeley Architecture Visual Resources Library

<http://orange.sims.berkeley.edu/cgi-bin/flamenco.cgi/spiro/Flamenco>

→Über 40.000 Bilder, 16.000 hierarchische Fassetten

41

Tag Cloud(s)

- Deutsch: „Schlagwortwolke“
- Methode zur Visualisierung von Informationen.
- Schlagwörter werden meistens alphabetisch sortiert. Die jeweilige Schriftgröße entspricht der relativen Häufigkeit eines Schlagworts.

42



Tag Cloud

Wozu?

- Zur Visualisierung strukturierter und unstrukturierter Daten
- Navigation (Schlagwörter mit Links gerüstet)
- Webdesignelement

43



Tag Cloud

Wie wird Tag Cloud benutzt?

- Sortierung:
 - alphabetisch
 - nach Häufigkeit
- Layout:
 - unterschiedliche Schriftgrößen
 - unterschiedliche Schriftfarben
 - Kombination unterschiedlicher Schriftgrößen und Schriftfarben

44



Tag Cloud

Vorteile

- Man hat gleich den Durchblick über den wichtigsten Tags
- Tags sind meistens alphabetisch sortiert, dadurch hat man gleich den Blick was man sucht

45



Tag Cloud

Nachteile:

- Vergleich zwischen Tags mit ähnlicher Wichtigkeit ist schwierig.
- Tags mit ähnlicher Bedeutung liegen weit voneinander (Z.B. *Linux*, *Windows*) → siehe Abbildung
- Aus Tag Cloud kann man keine Taxonomie ableiten (*webdesign* ist Unterbegriff von *design*)
- Mehrdeutigkeit von Wörtern wird nicht erfasst (*Windows* ist ein Betriebssystem oder sind Fenster???)

46



Tag Cloud

Wie kann Tag Cloud als Navigationsverfahren eingesetzt werden?

- Schlagwörter haben Links
- Schlagwörter sind strukturierte Daten (Terme von einer Kategorie oder sind zumindest Daten aus einem Bereich, damit man die Mehrdeutigkeit vermeiden kann)
- Link vom Schlagwort könnte eine Beschreibung haben, das beschreibt, wie viele Sätze von dem Ergebnis ausliefern werden, wenn man den Link folgt.

47



Tag Cloud

- Autor-Tag Cloud vom Dokumentenserver der Fakultät für Mathematik und Informatik
- Verwendetes Modul: **Tagadelic**
- Tag Cloud dient als Einstieg von einer Suchstrategie

Ongoing Research: 193 papers

Apel Aumüller Balzuweit Beckmann
Bernstein Biemann Böhm **Böhme** Der Do-
Do Döhler **Dötsch** Duc Gerber **Gräbe**
Greiner Hamzaoui Hartwig **Haupt** Heller
Hemmecke **Hering** Herre Herrmann
Herrmann Herrmann Hrsg.: Der Rektor
der Universität Leipzig **Institut für**
Informatik Jain Jantke Käthner
Kirsten Klaus Lange Lange Löffler Lummer
Marek **Märtens** Maßmann Melnik Melnik
Moehrke Möhrke Müller Müller **Müller** Nessel
Petri **Rahm** Ramsch Röder Rogler
Saupe Sch **count: 75** ulze Siedschlag **Sosna**
Spruth Spruth Steffen Steinmetz **Stöhr**
Stückrad Thor Tworzewski Villmann
Wagner Welk Winiarski **Wolter**
Zakharyashev Zeugmann

48

Wir sind hier

Ausgangssituation

Anforderungen an einen Dokumentenserver

Aufbau eines Dokumentenservers

- Auswahl eines Content-Management-System
- Verwaltungskonzept von Drupal

Navigations- und Suchverfahren

- Faceted Search
- Tag Cloud

Schnittstelle für Metadatenaustausch

- OAI-PMH

Aktueller Stand des Dokumentenserver

der Fakultät für Mathematik und Informatik



49

OAI-PMH als Schnittstelle für Metadatenaustausch

Was ist OAI-PMH?

- OAI ist die Abkürzung von **Open Archives Initiative.**

- OAI-PMH ist die Abkürzung von **OAI Protocol for Metadata Harvesting.** Protokoll dient zum Metadatenaustausch von Publikationen zwischen Systemen.

50



OAI-PMH

Es gibt 2 Klassen von Teilnehmern
(participants) an OAI-PMH Framework:

- Data Provider
- Service Provider

51



Aufgabe von Teilnehmer jeder Klassen

Data Provider

- Bereitstellung von einem Data-Repository
- Annahme der OAI-PMH-Anfrage von Service Provider in Form von HTTP-Request
- Bereitstellung vom Handlungssystem, das die OAI-PMH-Anfragen bearbeitet.

Die Standard-Anfragen von OAI-PMH sind: *Identify*, *ListMetadataFormats*, *ListSets*, *ListIdentifiers*, *ListRecords*, *GetRecords*

- Übertragen der OAI-PMH-Anwort in Form HTTP-Response an den Service Provider.

52



OAI-PMH

Service Provider

- Bereitstellung der OAI-PMH-Anfrage und Übergabe an Data Provider
- Sammeln von OAI-PMH-Antworten
- Caching von OAI-PMH-Antworten
- Bereitstellung einer Benutzerschnittstelle, Transformierung der Benutzereingabe in OAI-PMH-Anfrage
- Präsentation der OAI-PMH-Antworten auf der Benutzerschnittstelle

53



OAI-PMH

Beispiel für Data Provider:

- Universität Stuttgart, Fakultät 5, Germany, Computer Science Archive
<http://www.informatik.uni-stuttgart.de/cgi-bin/OAI/OAI.pl>
- University of Aachen, GERMANY
<http://darwin.bth.rwth-aachen.de/opus3/oai2/oai2.php>

Beispiel für Service Provider:

- [OAISTER](http://www.oaister.org/) - University of Michigan Digital Library
<http://www.oaister.org/>
- [Scientific Commons](http://en.scientificcommons.org/)
<http://en.scientificcommons.org/>

Quelle: <http://www.openarchives.org/>

54



Dokumentenserver der Fakultät für Mathematik und Informatik Universität Leipzig ist als ein Data Provider an Open Archives Initiative und ScientificCommons registriert.

55

Scientific Commons: Schema and Ontology Matching with COMA++ (2005), 2005-06 [Aumueller, David, Do, Hong-Hai, Massmann, Sabine, Rahm, Erhard] - Mozilla Firefox

http://en.scientificcommons.org/34140924

Scientific Commons beta

deutsch english

Publication View

Export References

- EndNote Download (RIS Format)
- BibTex Download (Tex Format)

Schema and Ontology Matching with COMA++ (2005)

Aumueller, David, Do, Hong-Hai, Massmann, Sabine, Rahm, Erhard

Abstract

We demonstrate the schema and ontology matching tool COMA++. It extends our previous prototype COMA utilizing a composite approach to combine different match algorithms. COMA++ implements significant improvements and offers a comprehensive infrastructure to solve large real-world match problems. It comes with a graphical interface enabling a variety of user interactions. Using a generic data representation, COMA++ uniformly supports schemas and ontologies, e.g. the powerful standard languages W3C XML Schema and OWL. COMA++ includes new approaches for ontology matching, in particular the utilization of shared taxonomies. Furthermore, different match strategies can be applied including various forms of reusing previously determined match results and a so-called fragment-based match approach which decomposes a large match problem into smaller problems. Finally, COMA++ cannot only be used to solve match problems but also to comparatively evaluate the effectiveness of different match algorithms and strategies.

Publication details

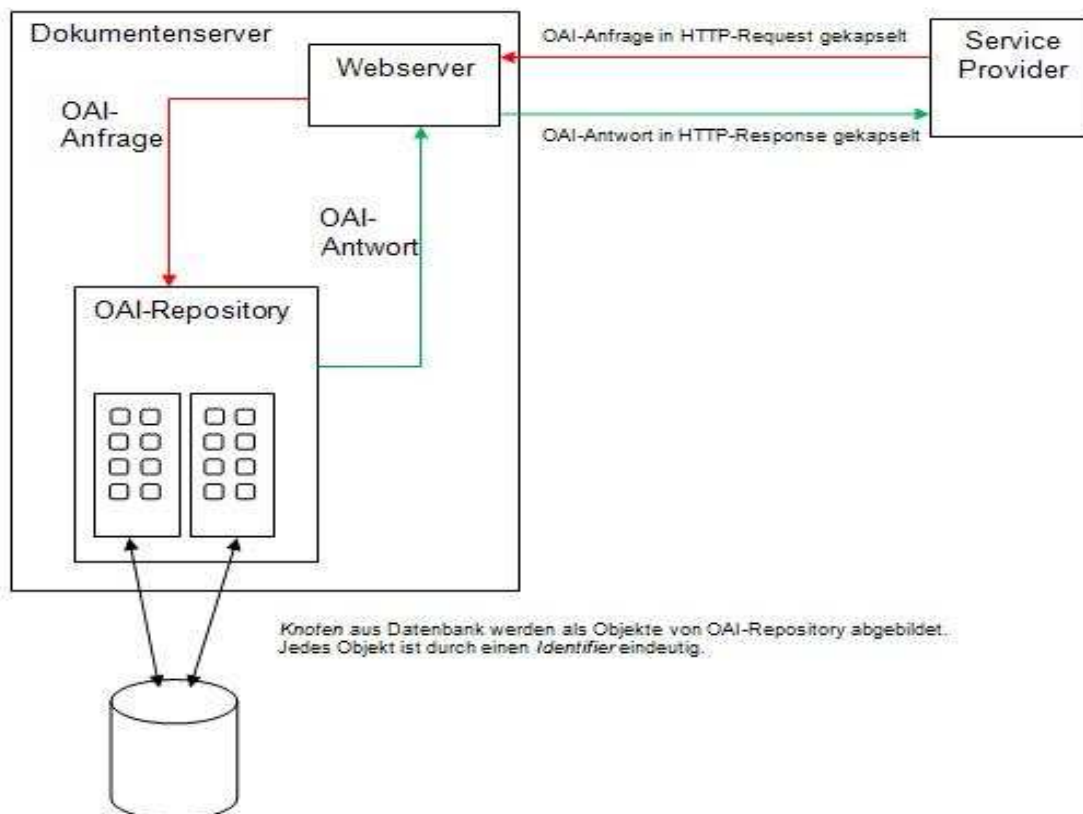
Download	http://lips.informatik.uni-leipzig.de/?q=node/166
Repository	Dokumentenserver Fakultät für Mathematik und Informatik Universität Leipzig (Germany)
Language	English
Relation	SIGMOD 2005

About Scientific Commons Imprint Contact Register URL Help

OAI-PMH

- Wie kommuniziert der Dokumentenserver der Fakultät für Mathematik und Informatik mit dem Service Provider mittels OAI-PMH?

57



58



Format von OAI-Anfrage

- **OAI-Anfrage in HTTP-Request gekapselt**

GET-Methode

`http://lips.informatik.uni-leipzig.de/oai?verb=Identify`

POST-Methode

`POST /oai`

Host: `lips.informatik.uni-leipzig.de`

Content-type: `application/x-www-form-urlencoded`

Content-length: `82`

Connection: `close`

`verb=Identify`

59



Format von OAI-Antwort

- **OAI-Antwort in HTTP-Response gekapselt**

`HTTP/1.1 200 OK`

`Date: Thu, 01 May 2008 22:01:22 GMT`

`Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5 with Suhosin-Patch`

60



Format von OAI-Anfrage

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">

  <responseDate>2008-05-30T22:50:11Z</responseDate>
  <request verb="Identify">http://lips.informatik.uni-leipzig.de/oai</request>
  <Identify>
    <repositoryName>University of Leipzig Faculty of Mathematics and
      Computer Science</repositoryName>
    <baseURL>http://lips.informatik.uni-leipzig.de/oai</baseURL>
    <protocolVersion>2.0</protocolVersion>
    <adminEmail>dbs@informatik.uni-leipzig.de</adminEmail>
    <earliestDatestamp>2008-03-07T15:56:21Z</earliestDatestamp>
    <deletedRecord>no</deletedRecord>
    <granularity>YYYY-MM-DDThh:mm:ssZ</granularity>
    <compression>yes</compression>
  </Identify>
</OAI-PMH>
```

61



Welche Metadatenformate werden unterstützt?

- **Ein Objekt von Repository (record) kann beliebigen Metadatenformate haben. Die enthält zu mindesten**
 - metadataPrefix: Präfix von Metadaten
 - metadataSchema: Schema zur Validierung von XML-Daten
 - XML Namespace: global eindeutiger Kennzeichner von Metadaten
- **Empfohlene Metadata-Format:**
 - Dublin Core
 - Marc
 - RFC 1807
 - OAI-PMH mit
 - metadataPrefix : oai_dc
 - metadataSchema : http://www.openarchives.org/OAI/2.0/oai_dc.xsd
 - XML Namespace : http://www.openarchives.org/OAI/2.0/oai_dc/

62

Wir sind hier

Ausgangssituation

Anforderungen an einen Dokumentenserver

Aufbau eines Dokumentenservers

- Auswahl eines Content-Management-System
- Verwaltungskonzept von Drupal

Navigations- und Suchverfahren

- Faceted Search
- Tag Cloud

Schnittstelle für Metadatenaustausch

- OAI-PMH

Aktueller Stand des Dokumentenserver

der Fakultät für Mathematik und Informatik



63

Aktueller Stand des Dokumentenservers der Fakultät

■ **Anzahl Dokumente: 402 (Stand Juni 2008)**

■ **Funktionalität:**

- Web-basiertes Benutzerverwaltungssystem
- Web-basiertes Dokumentenverwaltungssystem
- Benutzerschnittstelle für die Eingabe von Dokument. Einbettung von Capcha-Module gegen Spam
- Faceted Search für die Recherche und Navigation innerhalb der Dokumentenmenge.
- Autor-Tag Cloud
- OAI-PMH als Schnittstelle für Metadatenaustausch

64



■ **Erweiterungsaspekte:**

- Entwicklung eines Moduls zum Metadaten-Export (Metadaten-Export in Bibtex-Format in Test-Phase)
- Entwicklung eines Moduls zur Verwaltung von Zitierungen der Dokumenten
- Semantic-Web (z.B. für die Darstellung von Informationen über einen Autor)
- Web Service Schnittstelle zum automatischen Einbringen von neuen Dokumenten bzw. zum Abgleich mit anderer Dokumentendatenbank

- **Link:** <http://lips.informatik.uni-leipzig.de/>

65



Wir sind hier

Ausgangssituation

Anforderungen an einen Dokumentenserver

Aufbau eines Dokumentenservers

- Auswahl eines Content-Management-System
- Verwaltungskonzept von Drupal

Navigations- und Suchverfahren

- Faceted Search
- Tag Cloud

Schnittstelle für Metadaten austausch

- OAI-PMH

Aktueller Stand des Dokumentenserver

der Fakultät für Mathematik und Informatik



66