
Vergleich von Repository- Softwarelösungen

Universitätsbibliothek Osnabrück

Motivation

- Einarbeitung in das Thema Open Access und die zugrundeliegenden Technologien
- Beitrag der Universitätsbibliothek zum Open Access, Vorbereitung einer Erklärung der Universität zum Open Access Publishing und Beteiligung der UB an der OA-Community
- Ersatz der von der UB seit 1997 betriebenen *ELib* (Publikation von E-Dissertationen und anderer Arbeiten Osnabrücker Wissenschaftler) durch ein institutionelles Repository (Nutzung einer Open Source Lösung)
- Einwerbung von Publikationen und Umsetzung bestehender Anfragen wie:
 - 66/94: Unbekannte Nutzungsarten
 - Graue Literatur aus der Hebammen-Forschung
 - ...

- Marktübersicht mit Kurzbeschreibung der Systeme → *Long list*
- Vorläufiger Kriterienkatalog:
Muss- und Wunschkriterien → *Short list* nach Musskriterien
- Überarbeter Kriterienkatalog nach folgenden Gesichtspunkten:
 - Kriterien / Funktionen Webanwendung
 - Kriterien / Funktionen Universität
 - Kriterien / Funktionen OA-Repository
 - DINI-Anforderungen für die Zertifizierung
- Testinstallation der Kandidaten der *Short list*
- Bewertung der auf einen Kandidaten angewendeten Kriterien
- Risikobewertung / Entscheidung

Marktübersicht I

- **Recherchekriterien:**
 - Primär: Open Source Lösungen, auch für die Systemumgebung, z.B. Web- und / oder Applikationsserver, Datenbank etc.
 - Vorzugsweise „Out of the box“ Anwendung
- **Attribute der Beschreibungen:**
 - Hersteller (Adresse / Ansprechpartner)
 - URL für Informationen im WWW
 - Letzte Version / Datum
 - Lizenzbedingungen der Nutzung
 - Charakterisierung der Anwendung
 - Personaleinsatz
 - Anzahl Installationen weltweit

Marktübersicht II

- **Beispiel:**

- *Hersteller:* ARNO; Projekt: Academic Research in the Netherlands Online, durch: IWI (NL für: Innovation in Scientific Information Supply)
- *URL:* <http://arno.uvt.nl/~arno/site>
- *Letzte Version:* 1.2 vom 16.03.2005
- *Lizenz:* keine Angabe
- *Charakterisierung:* Designziel: Flexibles Werkzeug zum Erstellen, Verwalten und Präsentieren von OAI kompatiblen Archiven und Repositorien. Kein „off-the-shelf“ Institutional Repository, sondern „Toolbox“. Kein Benutzerinterface, dieses könnte über iPort oder i-Tor realisiert werden.
- *Personal:* Unix Administrator für Installation und Support, Perl Programmierer für Entwicklung, OA Beauftragter für Nutzung und Beratung
- *Installationen:* 7+

Kandidaten

¹ Nicht installiert, keine Informationen zu Shibboleth und Integritätssicherung
² Shibboleth-Anbindung geplant
* Open Source Software
** Enterprise Content Management System

- **Long list:**

- Alfresco [OSS* / ECMS**]
- Archimede [zuletzt: 2006]
- ARNO [OSS* / Toolbox]
- CDS Invenio [OSS*]
- CONTENTdm [OCLC]
- ContentManager [IBM / ECMS**]
- DIAS [Anwendung auf IBM CM]
- Digital Commons [ProQuest]
- DigiTool [ExLibris]
- DiVA [OSS* / XML-zentrisch]
- documentum [EMC**]
- DPubS [OSS* / Toolbox]
- DSpace [OSS*]
- Eprints [OSS*]
- eSciDoc [OSS* / Für Hosting?!]
- Fedora [OSS* / Toolbox]

- **Long list:**

- GAPworks [zuletzt: 2006]
- Greenstone [Digital Library]
- i-Tor [zuletzt: 2004]
- MyCoRe [OSS* / Toolbox (DocPortal)]
- MILESS [Aktuell: MyCoRe]
- OPUS [OSS*]
- Sunsite [keine Information]
- VT-ETD [zuletzt: 2004]
- WildFire [OSS* / nur NL]

- **Short list**

- CDS Invenio¹
- DSpace
- Eprints
- MyCoRe
- OPUS²

Musskriterien I

- **Webanwendung:**
 - Webbasierte Anwendung
 - Persistenz durch DB: Datenmodell, Transaktionen, SQL, etc.
 - Unicode Unterstützung (UTF-8)
 - Anpassbare Benutzeroberfläche: Look and Feel, Sprache, etc.
 - Unterstützung / Anbindung von Webservern, z.B. Apache oder Applikationsservern, z.B. Tomcat
- **Universität:**
 - Open Source Lizenz
 - Einbindbar in Shibboleth-Infrastruktur
 - Betriebssysteme: Unix, oder Unix-artige
 - System wird kontinuierlich weiterentwickelt

Musskriterien II

- **OA Repository / DINI Empfehlungen:**
 - OAI-PMH Support: Zugang zu Metadaten, mindestens DC Simple
 - Persistent Document Identifier, z.B.: Handle, DOI, urn:nbn
 - Metadatenverwaltung: Schema erweiterbar, Validierung, Export (hier auch: Klassifikation/Schlagworte)
 - Kollektionen von Dokumenten: Strukturierung, Gruppierung, Präsentation nach Kriterien, Originalformate
 - Integrierte Kategorien-, Metadaten- und Volltextsuche
 - Funktionen zum Datenimport und -export
 - Statistikfunktionen (Zugriffsstatistik)
 - Indexierung durch externe Suchmaschinen
 - Mechanismen zur Integritätssicherung der Dokumente integrierbar
 - Systemdokumentation

Wunschkriterien I

- **Webanwendung:**
 - Programmierschnittstellen (API) in Java oder einer Skriptsprache: Perl, PHP, Python, Ruby, Tcl/Tk, etc.
 - Serviceschnittstellen (Serviceorientierung), z.B.: SOAP, REST
 - Auf Rollen und Rechte zugeschnittene Benutzersichten und Funktionen
 - Automatisierte Installation (Skript)
 - Updatefähig (ohne Überschreiben benutzerspezifischen Anpassungen)
- **Universalität:**
 - Z39.50 Protokoll Unterstützung
 - Barrierefreie Nutzung der webbasierten Benutzeroberfläche
 - Authentifizierung, Autorisierung: Anbindung LDAP
 - Benachrichtigungssystem: Statusänderung von Dokumenten, etc.
 - Datenimport/-export mit speziellen Werkzeugen: Massendaten, Dokumente und Metadaten, Komprimierung (Fokus: Datenmigration)
 - Anschluss an Print-On-Demand-Service

Wunschkriterien II

- **OA Repository / DINI Empfehlungen:**
 - Workflow-Unterstützung bei Veröffentlichung von Dokumenten: Dokumentstatus etc.
 - Dokumentkollektionen: Quellformate (z.B. LaTeX, XML) zusammen mit Publikationsformat (z.B. PDF) und zugehörige Ressourcen (z.B. XSLT-Skripte) sowie Primärdaten (Programmcode, Messwerte, Multimediadaten, etc.
 - Versionierung von Dokumenten und Varianten: Historie, Label; Pre- oder Postprint, etc.
 - Volltext- und Metadatensuche: Boolesche Anfragen, Trunkierung, Stemming, etc.
 - Exportformate XMETADISS, XMETADISS+, XEPICUR
 - Dokumenten- und Metadatenschutz (Integrität): Verfügbarkeit, Signatur, Hash, Verschlüsselung
 - Zuordnung von Nutzungsrechten / Lizenzen (z.B.: CC) zu Dokumenten

Testinstallation / Testszenario

- **Kriterien:**
 - Verschiedene Nutzer: Systemadministrator, Workflow-Admin, interner Nutzer „Upload“, externer Nutzer „Upload“, externer Nutzer „Recherche“
 - Workflow-Abbildung
 - Datenimport: Datensätze mit Link zu Datei (verschiedener Formate: Text, Bilder, Filme, etc.)
 - Datei-Upload von Wissenschaftlern / Datei-Upload Bibliothek
 - Änderung von Metadaten: durch Wissenschaftler und durch Bibliothek
 - Abbildung eines E-Journals
 - Suche: Stichwort, Autor, Verlag, zeitliche Einschränkungen, Volltextsuche unter Berücksichtigung von Dateitypen / Lizenzen
 - Empirische Datenerhebungen, z.B. Speicherplatznutzung
- **Ergebnisse:**
 - Teil der Bewertung (Beurteilung der Funktionalitäten)

„Barrieretests“

- **Kriterien:**

- Barrieretests nach Empfehlungen des W3C (Web Content Accessibility Guidelines 1.0, W3C Recommendation 5-May-1999, <http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505>)
- Durchführung mit dem Web-Dienst WebXACT (<http://webxact.watchfire.com/>), der zum 1. April 2008 vom Netz genommen wurde (nun Rational Software / IBM)
- Die automatische Beurteilung der vielfältigen Testkriterien ergibt häufig ein kritischeres Bild als tatsächlich vorliegen mag
- Viele Warnungen sind Hinweise auf Verbesserungsmöglichkeiten
- Als ausreichendes Qualitätsmerkmal für Barrierefreiheit des getesteten Repositories gilt die bestandene Überprüfung nach „Priorität 1“
- Je Repository-Software (Short list) wurde eine Stichprobe von drei Installationen gewählt und jeweils Eingangsseite und Suchergebnisliste validiert

- **Ergebnisse:**

- Teil der Bewertung (Universität/Barrierefreiheit)

„Design-Tests“

- **Kriterien:**
 - Anpassbare Benutzeroberfläche (Look and Feel, Kopf- oder Fußzeilen, Auswahl der Sprache der GUI, Hinzufügen neuer GUI Übersetzungen)
 - Layoutkonfigurationen (Positionierung / Darstellung der GUI-Elemente)
 - Repository Titel, Repository Logo, Allgemeines Layout (Schwerpunkt: CSS)
- **Ergebnisse** (nicht Teil der Bewertung da Musskriterium):
 - *DSpace*: Vielfältige Anpassungsmöglichkeiten, die Konfigurationsdateien sind jedoch weit gestreut
 - *Eprints*: Vielfältige Anpassungsmöglichkeiten, aber keine eigene Layoutkonfiguration
 - *MyCoRe*: Vielfältige Anpassungsmöglichkeiten, ohne *DocPortal* als Beispiel-Anwendung jedoch erheblicher Aufwand in der eigenen Realisierung und Gestaltung
 - *OPUS*: Eigene Gestaltung nur mit hohem Aufwand möglich (keine eigene Layoutkonfiguration, nur wenig CSS-Einsatz)

DINI-Anforderungen

- **Kriterien:**
 - Mindeststandards und Empfehlungen, mit technischem Bezug zum Repository
- **Ergebnisse:**
 - Teil der Bewertung (Musskriterien DINI-Empfehlung und Wunschkriterien DINI-Empfehlung)

Bewertungen

- **Methode**

- Musskriterien entscheiden über Kandidaten
- Gewichtung der Wunschkriterien (1...10)
- Bewertung der Erfüllung der Kriterien des jeweiligen Kandidaten (0...10) und Kommentierung der Bewertung
- Summe über die Punktzahl der gewichteten Kriterien (als erster Entscheidungsansatz)
- Differenzierung von eng beieinander liegenden Kandidaten durch Risikobewertung
 - Fehlende Standards / proprietäre Lösungen in der Webanwendung
← Möglichkeit zur Weiterentwicklung / Zukunftssicherheit
 - Beschaffung von Entwicklungs-Ressourcen
← lokale Anpassung und Entwicklung / Integration der Lösung
 - Zu kleiner oder lokal begrenzter Anwenderkreis
← generelle Weiterentwicklung / Performance der Weiterentwicklung / Supportmöglichkeiten

- CDS Invenio¹: 775 Pkt.
- **DSpace**: **966 Pkt.**
- Eprints: 975 Pkt.
- MyCoRe: 924 Pkt.
- OPUS: 611 Pkt.

¹ Nicht installiert, keine Informationen zu Shibboleth und Integritätssicherung

- **Entscheidung für DSpace (April / Mai 2008):**

- Standards bei der Entwicklung der Webanwendung (Java, JSP): gut anpassbar, gut skalierbar, erweiterbar
- Aktualität der Lösung, Dynamik in der Weiterentwicklung, weltweite Akzeptanz, große und aktive Anwendergruppe
- Wachsende nationale Anwendergruppe
- Java-Know-How (und JSP-Know-How) weit verbreitet, Entwicklungs-Ressourcen akquirierbar
- Strukturierung der Dokumente in Bereiche und Sammlungen, gute Integration in die Hochschulstrukturen
- „Fertige“, konfigurierbare Anwendung

- **Pro**

- Python API
- LDAP Anbindung
- Zugriffsrechte nach Benutzer und Gruppen
- Überwachung von und Benachrichtigung bei der Änderung von Dokumentzuständen
- Gute Unterstützung von Publikations-Workflows
- Massen-Import von Dokumenten und Metadaten
- Volltextsuche mit umfangreichen Möglichkeiten der Indexierung
- Versionsverwaltung von Dokument-Metadaten

- **Contra**

- Keine Z39.50 Unterstützung
- Keine Zuordnung von Nutzungsrechten / CC-Lizenzen
- Kein Massen-Export von Dokumenten und Metadaten

Keine Testinstallation, keine Prüfung der angegebenen Funktionalitäten. Bewertung nur nach Dokumentation.

- **Pro**

- Java API
- LDAP-Anbindung
- Zugriffsrechte nach Benutzer und Gruppen
- Überwachung von und Benachrichtigung bei der Änderung von Dokumentzuständen
- Gute Unterstützung von Publikations-Workflows
- Zuordnung von Nutzungsrechten / CC-Lizenzen
- Massen-Import/Export von Dokumenten und Metadaten in einem XML-Format
- Volltextsuche mit umfangreichen Möglichkeiten der Indexierung
- Gute Ergebnisse bei Test auf Barrierefreiheit

- **Contra**

- Keine Z39.50 Unterstützung
- Versionsverwaltung von Dokumenten/Metadaten nur über Referenzen in Dokumenten (z.B. dc.relation), jedoch Änderungsinformationen (Zeit/Datum/Benutzer/Dateiname/Bytes/MD5) stehen zur Verfügung
- Keine Unterstützung von Exportformaten wie: XMETADISS, XMETADISS+, XEPICUR (aber teilweise in Entwicklung)
- Keine Anbindung zu Print on Demand Diensten

- **Pro**

- Perl API
- LDAP-Anbindung
- Zugriffsrechte nach Benutzer und Gruppen
- Überwachung von und Benachrichtigung bei der Änderung von Dokumentzuständen
- Gute Unterstützung von Publikations-Workflows
- Zuordnung von Nutzungsrechten / CC-Lizenzen
- Massen-Import/Export von Dokumenten und Metadaten

- **Contra**

- Keine Z39.50 Unterstützung
- Keine Booleschen Suchanfragen, und Trunkierung, kein Stemming
- Keine Anbindung zu Print on Demand Diensten
- Wenig gute Ergebnisse bei Test auf Barrierefreiheit

- **Pro**

- Java API
- Zugriffsrechte nach Benutzer und Gruppen
- Überwachung von und Benachrichtigung bei der Änderung von Dokumentenzuständen
- Zuordnung von Nutzungsrechten / CC-Lizenzen
- Volltextsuche mit umfangreichen Möglichkeiten der Indexierung
- Versionsverwaltung von Dokumenten/Metadaten über ein internes Datei-Speichersystem
- Gute Unterstützung durch die deutschen Entwickler und Anwender

- **Contra**

- Keine LDAP-Anbindung
- Geringe Unterstützung von Publikations-Workflows, Basis ist Beispiel-Anwendung *DocPortal*, da MyCoRe Werkzeugkasten
- Hoher Aufwand bei Eigenentwicklung einer Anwendung, mittlerer Aufwand bei Ableitung aus Beispiel-Anwendung
- Keine Anbindung zu Print on Demand Diensten
- Wenig gute Ergebnisse bei Test auf Barrierefreiheit

- **Pro**

- Updates ohne Überschreiben benutzerspezifischer Anpassungen
- Zuordnung von Nutzungsrechten / CC-Lizenzen
- Volltextsuche (Google) / Kategoriensuche
- Unterstützung von Exportformaten wie: XMETADISS, XEPICUR
- Anbindung zu Print on Demand Diensten

- **Contra**

- Keine API
- Keine Z39.50 Unterstützung
- Keine LDAP-Anbindung
- Keine Definition von Zugriffsrechten nach Benutzer und Gruppen
- Keine Unterstützung zum Massen-Import/Export von Dokumenten und Metadaten
- Wenig gute Ergebnisse bei Test auf Barrierefreiheit

- Weiterentwicklung von OPUS während der Evaluation (Ende 2007) unklar

Weitere Studien zu OA-Repositorien:

A Guide to Institutional Repositories Software: Open Society Institute. - 3rd Edition. - August 2004.

URL: http://www.soros.org/openaccess/pdf/OSI_Guide_to_IR_Software_v3.pdf

Digitale Repositorien an mittleren Universitätsbibliotheken: Anforderungen und Möglichkeiten am Beispiel des Dokumentenservers GEB des Bibliothekssystems Gießen: Arne Christian Upmeier. – Diplomarbeit. – 2008.

URL: <http://geb.uni-giessen.de/geb/volltexte/2008/6719/pdf/UpmeierArne-2008-09-29.pdf>

Durchführung des Vergleichs:

Robert Roggenbuck [rroggenb@uni-osnabrueck.de]

Worskop-Vorstellung:

Jörg Schmitz [joschmit@uni-osnabrueck.de]

Vielen Dank für Ihre Aufmerksamkeit!