

Prüfung und Erweiterung der technischen Grundlagen des Dokumentenservers OPUS zur Zertifizierung gegenüber der DINI anhand der Installation an der TU Hamburg- Harburg

Projektbericht für das Projektstudium an der Virtuellen Fachhochschule,
Standort Lübeck

Dipl.-Bibl. (FH) Oliver Marahrens, TU Hamburg-Harburg, Bibliothek

März 2005

Betreuung: Dipl.-Ing. Malte Dreyer, VFH / FH Lübeck

Projektzeitraum Wintersemester 2004/05, 18.10.2004 – 04.02.2005

Inhaltsverzeichnis

1 Darstellung des Projektes und seiner Ziele.....	3
1.1 OPUS-Dokumentenserver.....	3
1.2 Projektumgebung	4
1.3 Projektziele.....	4
2 Projektrelevante Zertifizierungsbedingungen und Realisationspläne.....	5
3 Beschreibung der benutzten Technologien.....	7
4 Projektergebnisse.....	7
4.1 Formate zur Langzeitarchivierung.....	7
4.2 Automatische Übernahme der OPUS-Titel in den Bibliothekskatalog.....	7
4.3 Behebung von Sicherheitslücken in OPUS.....	9
4.3.1 Ausführung beliebiger PHP-Dateien im volltexte-Verzeichnis.....	9
4.3.2 Einblick in Konfigurationsdateien im lib-Verzeichnis.....	9
4.4 Eigene Erweiterungen von OPUS.....	10
4.4.1 Autorensupport.....	10
4.4.2 FAQ-Datenbank.....	11
4.4.3 Authentifizierung zur Veröffentlichung eigener Dokumente	13
4.4.5 Automatische URN-Zuweisung für Dissertationen und Habilitationen.....	16
4.4.6 Prüfsummen.....	18
4.4.7 XMetaDiss-Schnittstelle.....	19
5 Fazit.....	24
6 Literaturliste.....	24
7 Anlage1: Diagramme.....	28

Dieser Projektbericht ist online verfügbar unter der URN urn:nbn:de:gbv:830-opus-827

1 Darstellung des Projektes und seiner Ziele

1.1 OPUS-Dokumentenserver

OPUS steht für Online Publikationsverbund der Universität Stuttgart. Der OPUS-Server entstand 1997 aus einem Projekt des Deutschen Forschungsnetzes DFN heraus, bei dem es um den Aufbau eines landes- und bundesweiten Publikationsverbundes ging ([4]).

In der Abbildung 1 ist schematisch in Form eines UML-Diagrammes beschrieben, wofür OPUS eingesetzt wird.

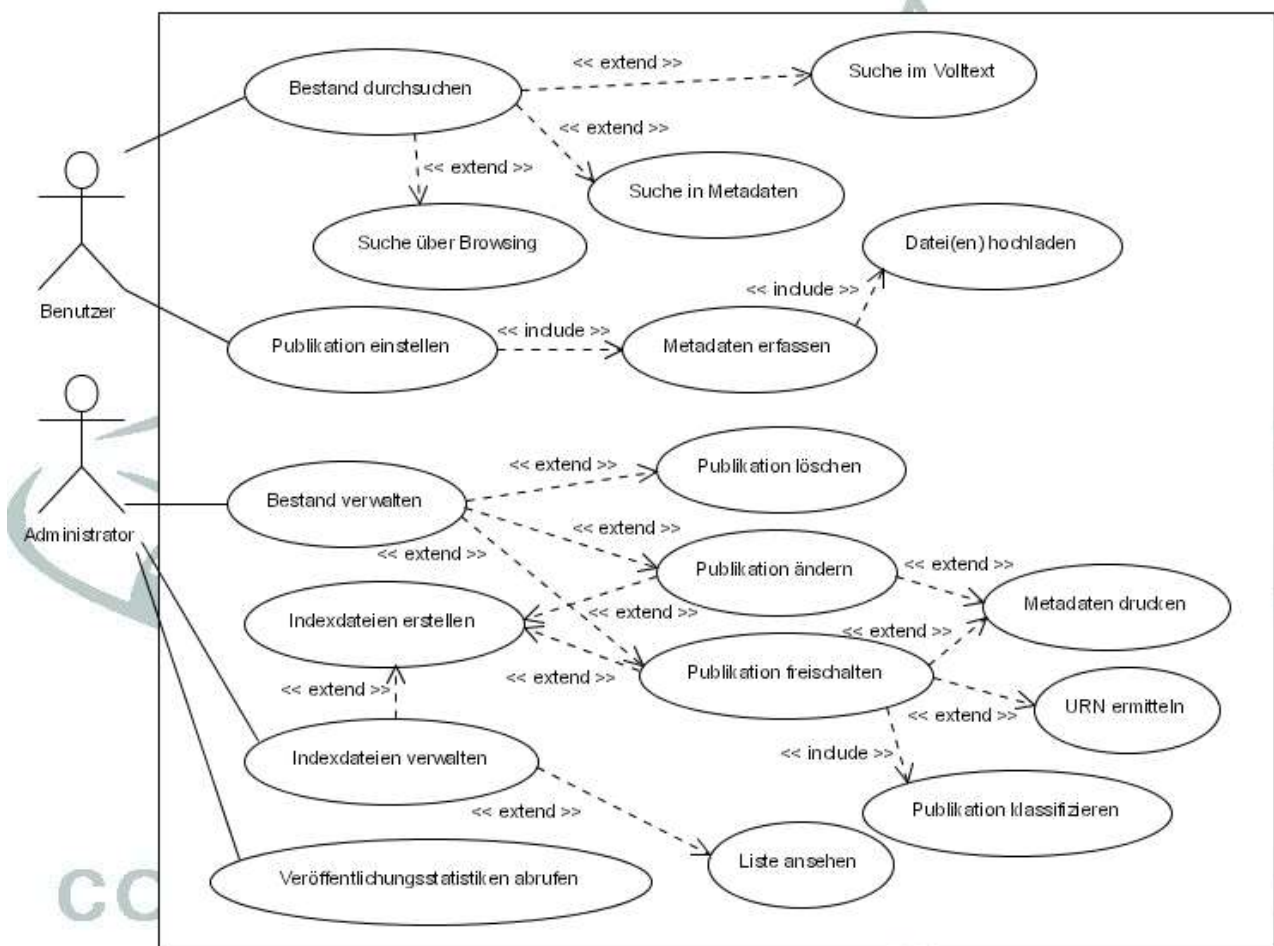


Abbildung 1 UseCase-Diagramm für OPUS (UML)

Für die beiden UseCases **Publikation einstellen** (beinhaltet die Erfassung von Metadaten und das Hochladen von Dateien) und **Publikation freischalten** wurden detailliertere Aktivitätsdiagramme angefertigt, anhand derer der jeweilige Workflow ersichtlich ist (siehe Anlage 1). Die Aktivitätsdiagramme sind detaillierter als gewöhnlich, da sie dazu dienen, in der ersten Phase des Projektes einen genaueren Überblick über die internen Abläufe von OPUS zu gewinnen. Auf diese Weise sollte es vereinfacht werden, später Erweiterungen einsetzen zu können. Die aktuellen Aktivitätsdiagramme, die den Workflow nach dem Projekt zeigen, sind ebenfalls in der Anlage zu finden.

Bei der ersten Sichtung des Quellcodes des Softwaresystems stellte sich heraus, dass die Programmierung teilweise recht unsauber ist. Zudem fehlen an vielen Stellen Kommentare, die einen schnellen Überblick erleichtern würden.

Auch sicherheitstechnisch zeigte sich eine unschöne Lücke: beim Aufsetzen des Testsystems stellte sich heraus, dass die globalen Variablen, die seit PHP 4.2 standardmässig deaktiviert sind, in OPUS nicht konform zu den höheren PHP-Versionen gesetzt werden. Das Verhalten, dass sämtliche Übergabeparameter automatisch in Variablen umgewandelt werden (ob sie nun per GET oder POST an das Script weitergegeben werden), wird in PHP über die Konfigurationsoption `register_globals` gesteuert. Ist sie aktiviert (standardmässig bis PHP 4.2), so kann eine Variable, die per URL übergeben wird, im PHP-Script automatisch angesprochen werden. Im Fall von `register_globals=off` ist dies nicht möglich. Bei ausgeschalteten globalen Variablen werden die Übergabevariablen in Container-Variablen übergeben und müssen anders adressiert werden. Eine Übergabe per URL (GET-Verfahren) heisst in diesem Fall `$_GET[„variablenname“]`; wird über das POST-Verfahren gesendet (zum Beispiel bei Formularen), so kommen die Variablen im Container `$_POST[„variablenname“]` beim Script an. Dieser Mechanismus verhindert Manipulationsversuche per URL und sorgt daher für die Sicherheit des PHP-Scripts. Noch sicherer wäre die Verwendung von Sessionvariablen, auf die OPUS allerdings weitestgehend verzichtet (vermutlich aus historischen Gründen, da die Software bereits recht alt ist und der Session-Mechanismus erst seit etwa Version 4.1 ausgereift ist).

OPUS adressiert die Variablen auf die unsaubere Art einfach über den Variablennamen, was bei Installationen mit `register_globals=off` dafür sorgt, dass die Variablen nicht korrekt angesprochen werden können. Eine Lösung dieses Problems liegt auf der Hand: `register_globals` kann einfach auf `on` gesetzt werden. Doch da sich auf einem Webserver und einer PHP-Installation meist mehrere Websites befinden ist dies eine unschöne Lösung, die bei anderen Webapplikationen evtl. Sicherheitslöcher aufstoßen könnte.

Es gibt eine Möglichkeit, `register_globals` für eine einzelne Site (und nicht für den globalen Server) freizuschalten durch einen Eintrag in einer `.htaccess`-Datei. Diese Lösung wurde im vorliegenden Fall gewählt. Die Sicherheitslücke wurde den Entwicklern von OPUS gemeldet und sie haben zugesagt, das Problem in der nächsten OPUS-Version zu beheben.

1.2 Projektumgebung

Das Projekt wurde im Auftrag der Bibliothek der Technischen Universität Hamburg-Harburg neben der hauptamtlichen Tätigkeit in derselbigen Institution durch den Autor dieser Arbeit durchgeführt. Wie vereinbart wird das Projekt von der Virtuellen Fachhochschule als Projektsemester anerkannt. Es wurde ein Testsystem mit Opus aufgesetzt, wobei das Opus-System nicht neu installiert wurde, sondern eine Kopie der lokalen Produktivversion benutzt wurde. Auch die produktive Datenbank wurde auf das Testsystem kopiert. Erweiterungen und Modifikationen wurden ausschliesslich am Testsystem durchgeführt und sorgfältig protokolliert. Nach Projektende soll durch intensive Begutachtung des Testsystems entschieden werden, welche der Änderungen auf das Produktivsystem übernommen werden. Diese Begutachtung hat noch nicht stattgefunden.

1.3 Projektziele

Durch das Projekt sollte zunächst überprüft werden, welche Voraussetzungen dem Dokumentenserver OPUS in der vorliegenden Installation an der Bibliothek der TU Hamburg-Harburg noch fehlen, um gegenüber der DINI (Deutsche Initiative für Netzwerkinformation) zertifiziert werden zu können. Die Zertifizierung an sich war nicht Teil des Projektes und es war von vornherein so geplant, dass der Zertifizierungsantrag nach Projektende (im Laufe des Jahres 2005) gestellt werden soll.

Durch das Projekt sollten die Grundlagen geschaffen werden, damit der Dokumentenserver von den Studierenden und Doktoranden als Mittel der Veröffentlichung ihrer Abschlussarbeiten akzeptiert und genutzt wird. Dafür ist nicht nur die Bereitstellung des Systems notwendig, sondern auch Hilfestellungen und Ansprechpartner innerhalb der Bibliothek als anbietende Institution. An Stellen,

wo das System hier noch nicht ausreichend Hilfestellung gibt, sollten zusätzliche Mechanismen entworfen und implementiert werden, mit denen ein solches Ziel erreicht werden könnte.

2 Projektrelevante Zertifizierungsbedingungen und Realisationspläne

Es wurde relativ schnell klar, dass OPUS an sich schon sehr viele der von der DINI vorgegebenen Zertifizierungsbedingungen erfüllt. Einige Bibliotheken betreiben schon heute OPUS-basierte Dokumentenserver, die das DINI-Zertifikat erhalten haben (z.B. die Universitätsbibliothek der Universität Stuttgart, wo OPUS auch entwickelt wird).

In der folgenden Tabelle ist ersichtlich, welche Anforderungen der DINI Bestandteil des vorliegenden Projektes waren. Ebenso sind Bedingungen ausgeblendet, die von vornherein erfüllt waren.

<i>Anforderung</i>	<i>Umsetzung</i>	<i>Projektbestandteil</i>	<i>Pflicht</i>
Autorenbetreuung			
Nach außen sichtbares Beratungsangebot (z.B. WWW-Seiten und E-Mail-Kontaktadressen, telefonische und persönliche Autorenbetreuung)	Einbindung von Kontaktadressen auf der Dokumentenserver-Seite sowie in einer (neu zu schaffenden) FAQ-Datenbank	ja	ja
Unterstützung des gesamten Publikationsprozesses (einschließlich rechtlicher und technischer Problemstellungen)	Einbindung von FAQs auf der Dokumentenserver-Seite (in Form einer Datenbank geplant, damit die FAQs dynamisch und auch anderweitig verwendet werden können)	ja	ja
Rechtliche Aspekte			
Der Betreiber des Servers muss berechtigt sein, die eingereichten Dokumente auf dem Server zu veröffentlichen (Recht zur elektronischen Speicherung, insbesondere in Datenbanken, und zum Verfügbarmachen für die Öffentlichkeit zum individuellen Abruf, zur Wiedergabe auf dem Bildschirm und zum Ausdruck beim Nutzer [Online-Nutzung], auch auszugsweise)	Knüpfung der korrekten Rechte durch zu bestätigende Klausel vor dem Upload von Publikationen	ja	ja
Authentizität und Integrität			
Dokumentation des technischen Systems	OPUS-Änderungen und -Erweiterungen sorgfältig dokumentieren	ja	ja

<i>Anforderung</i>	<i>Umsetzung</i>	<i>Projektbestandteil</i>	<i>Pflicht</i>
Existenz eines Systems zur Sicherung der Server, Metadaten und Dokumente	Entwicklung einer Backup-Strategie (z.B. Mirroring oder DB-Dump; teilweise schon realisiert in lokaler Umgebung)	ja	ja
Kontrollierte und nachweisbare Aufnahme von Dokumenten aus technischer Sicht	Prüfung der MD5-Summe vor und nach dem Upload	ja	ja
Persistent Identifier	Automatisierte URN-Vergabe und Meldung bei DDB über OAI (teilweise schon erfüllt; XEpicur)	ja	ja
Einsatz eines Verfahrens zum Nachweis der Unversehrtheit der Dokumente (z.B. Hash-Wert)	Hinweise auf MD5-Prüfsummenbildung, Kontrolle der MD5-Prüfsumme vor und nach Upload bzw. Download	ja	nein
Darstellung der Dokumente in einem gängigen Präsentationsformat.	Bereitstellung aller Dokumente als HTML oder PDF, ggf. Online-Umwandlung	ja	nein
Metadatenexport			
Export in Langzeitarchivierungssysteme	Evtl. Teilnahme an DDB-Archivierungsangebot (Bereitstellung der XMetaDiss-Schnittstelle)	ja	nein
Export in bibliographische Datenbanken (z. B. Literaturverwaltungsprogramme, Verbünde, OPACs)	Untersuchung: Bereitstellung einer Schnittstelle von OPUS zum LBS/CBS mit weitestgehend automatischer Eintragung möglich?	ja	nein
Zugriffsstatistik			
Die Statistik sollte als dynamisches Metadatum dem jeweiligen Dokument zugeordnet sein	Prüfung, ob dem schon so ist, ansonsten einbauen	ja	nein
Langzeitverfügbarkeit			
Sicherstellung der Langzeitverfügbarkeit durch ggf. Kooperation mit einer Archivierungsinstitution	Evtl. Kooperation mit der DDB und redundante Archivierung auf deren Dok-Server (zumindest Dissertationen und Habilitationen) -> Automatismus zur Spiegelung, Schnittstelle XMetaDiss	ja	nein

Tabelle 1 projektrelevante DINI-Zertifizierungsbedingungen (angepasst nach Quelle [5])

Wie oben erwähnt gehören weitere Bedingungen zur endgültigen Zertifizierung, diese sind aber weniger technischer Natur, so dass sie im Projekt weitgehend ausgeblendet wurden.

Zur Zertifizierung müssen beispielsweise noch Richtlinien für den Dokumentenserver-Betrieb verfasst werden und rechtsgültige Autorenverträge formuliert werden.

3 Beschreibung der benutzten Technologien

Die zugrundeliegenden Basistechnologien waren vorgegeben, da die Eigenentwicklungen auf OPUS basieren und demzufolge die Rahmengrundlagen bereits abgesteckt waren. OPUS ist in PHP programmiert und bietet eine Weboberfläche, die über den Browser bedient werden kann. Somit ist Bedingung für den Betrieb eines OPUS-Dokumentenservers ein Webserver mit PHP-Unterstützung. An der TU Harburg ist dies mit einem Apache-Webserver umgesetzt, der auf einer Linux-Plattform läuft. PHP ist als Modul in den Apache-Webserver eingebunden.

OPUS kann wahlweise eine mSQL- oder eine MySQL-Datenbank als Datengrundlage benutzen, für andere Datenbanksysteme ist keine Schnittstelle vorhanden. Eine Erweiterung wäre unproblematisch, da die Datenbankschnittstelle objektorientiert realisiert ist und somit einfach ausgetauscht werden könnte. Dies war jedoch in keiner Weise Aufgabe in dem vorliegenden Projekt. An der TU Harburg wird eine MySQL-Datenbank als Backend benutzt, womit die LAMP-Umgebung¹ komplettiert wird.

4 Projektergebnisse

4.1 Formate zur Langzeitarchivierung

Es muss festgelegt werden, welche Dokumentformate von den Studierenden einzureichen sind, wenn sie über den Dokumentenserver publizieren wollen. Für Hochschulschriften (und ausschliesslich solche sollen auf OPUS angeboten werden) hat die Deutsche Bibliothek (DDB) recht genaue Vorgaben². Das präferierte Abgabeformat ist demzufolge XML/SGML, es folgen HTML, PDF, PS und sonstige (als Beispiel sind hierbei RTF, DOC, TEX, DVI, TXT genannt, es werden aber auch weitere Formate akzeptiert). Daher sollten die Studierenden auch für den Dokumentenserver der TUB dazu angehalten werden, ihre Dokumente möglichst in einem XML-Format einzureichen. Da moderne Office-Programme normalerweise diese Möglichkeit bieten, muss dies den Studierenden nur noch mitgeteilt werden. Denkbar wäre es, ausschliesslich XML-Formate zu akzeptieren (zum Beispiel OpenOffice/StarOffice-Dateien oder in MSOffice verfasste, aber als XML gespeicherte Dateien). In dem Fall muss aber eine ausführliche Hilfestellung bereitgestellt werden, da die Speicherung in XML nicht unbedingt als Allgemeinwissen von den Usern vorausgesetzt werden kann.

Eine Frage, die bislang noch nicht geklärt wurde, ist die nach der Archivierung von Multimedia-Daten. Dies ist problematischer als bei Textdateien, da Multimedia-Daten nur schwierig in einem Klartextformat gespeichert werden können und es verschiedene Ansätze dazu gibt.

In diesem Punkt sind also noch offene Fragen vorhanden, die für die Richtlinien beim Betrieb des Dokumentenservers geklärt werden müssen.

4.2 Automatische Übernahme der OPUS-Titel in den Bibliothekskatalog

Die automatische Übernahme von Titeln, die in OPUS veröffentlicht werden, in den lokalen oder gar den zentralen (Gemeinsamer Bibliotheksverbund - GBV) Bibliothekskatalog der TU Harburg

1 LAMP = Linux, Apache, MySQL, PHP, gängige Systemumgebung für Webanwendungen (Quasi-Standard)

2 Siehe [7]

wurde nicht erreicht. Es wurden jedoch einige Überlegungen angestellt, wie man eine solche Übernahme realisieren könnte.

Überlegung 1: Meldung an die Katalogisierung

Diese Überlegung umfasst Optimierungen am Status quo. Der Status quo kann so zusammengefasst werden, dass Neueinträge in OPUS ausgedruckt und an die Katalogisierungsabteilung weitergeleitet werden. Eine mögliche Optimierung dieses Workflows könnte erreicht werden, indem die Lieferung der Daten nicht über ein ausgedrucktes Blatt Papier geschieht, sondern per eMail. Eine doppelte Datenerfassung würde damit weitgehend vermieden, da die Daten nur per Copy&Paste aus dem Mailbody übernommen werden müssten.

Überlegung 2: Halbautomatische Übernahme

Übernahme über ein Exportformat (z.B. MARC-Daten), die aus OPUS exportiert und in das LBS/CBS importiert werden könnten. Die MARC-Daten könnten dann in der Katalogisierungsabteilung eingelesen, kontrolliert und in das LBS/CBS übertragen werden. Problem dabei ist, dass es noch keine Schnittstelle gibt, die aus OPUS MARC-Daten erstellt und so der Export derzeit noch nicht gewährleistet werden kann. Außerdem ist nicht sicher, ob die Daten per MARC in das LBS/CBS eingespielt werden können. Zudem müsste eine manuelle Dublettenkontrolle vorgenommen werden.

Überlegung 3: Vollautomatische Übernahme

Übernahme über eine direkte Schnittstelle von OPUS in die lokale oder zentrale Bibliotheksdatenbank. Hierbei stellt sich nicht nur das Problem, dass es eine solche Schnittstelle noch nicht gibt, sondern auch, dass eine automatische Dublettenkontrolle problematisch ist (schon eine manuelle Dublettenkontrolle ist nicht trivial, umso problematischer ist es, Daten aus verschiedenen Datenquellen automatisch auf Doppelungen zu überprüfen). Das CBS (Centrale Bibliothekssystem) bietet zwar eine rudimentäre Dublettenkontrolle, jedoch ist es trotzdem angeraten, auch manuell darauf zu prüfen.

Eine automatische Datenübernahme für den GBV ist nach aktuellem Stand offenbar bei noch keinem Dokumentenserver realisiert. Auf dem Dokumentenserver Monarch gibt es eine vergleichbare Meldeschnittstelle an den SWB (Südwestdeutscher Bibliotheksverbund), auf dem Dokumentenserver Miles ist eine Schnittstelle bislang lediglich in Planung. Von der offiziellen OPUS-Distribution ist zunächst keine GBV-Schnittstelle zu erwarten, da OPUS aus dem SWB stammt und daher in erster Linie eine dementsprechende Schnittstelle zu erwarten wäre.³ Hier müsste also vermutlich weitere Eigenentwicklungsarbeit investiert werden. Interesse an einer derartigen Schnittstelle wäre allerdings auch in anderen OPUS einsetzenden GBV-Bibliotheken vorhanden (z.B. die Helmut-Schmidt-Universität der Bundeswehr in Hamburg oder die Staats- und Universitätsbibliothek Carl von Ossietzky Hamburg).

³Annette Maile, Opus-Entwicklerin an der UB Stuttgart, merkt dazu an: „zum SWB existiert bereits eine Schnittstelle. Das BSZ holt sich ueber eine spezielle Oai 1.1 Schnittstelle mit DC qualified die Daten ab und importiert sie in den SWB.“ [10]

4.3 Behebung von Sicherheitslücken in OPUS

4.3.1 Ausführung beliebiger PHP-Dateien im volltexte-Verzeichnis

[...] sofern Sie OPUS (trifft aehnlich evtl. auch auf andere Volltextserver-Software zu) betreiben, sollten Sie unbedingt Ihren Web-Server so konfigurieren, dass in dem Verzeichnisbaum, in dem die Volltexte liegen, keine PHP-Skripte ausgefuehrt werden koennen. Andernfalls ist es moeglich, per Dokumentanmeldung php-Skripte (z.B. als angebliches Originalformat) hochzuladen und spaetestens nach Freischaltung (je nach Berechtigung fuer den Zugriff auf die incoming-Verz. auch schon direkt nach dem Upload) beliebige Operationen unter dem Benutzer des WWW-Servers durchzufuehren (z.B. Loeschen von anderen OPUS-Dokumenten, Ausnutzen eines lokalen Exploits zur Erlangung von root-Rechten, etc.).

Unter Apache sollten eigentlich folgende Zeilen ausreichen:

```
<LocationMatch "/<tragen Sie hier Ihr Verz. ein>/volltexte">  
  php_flag engine off  
</LocationMatch>  
[6]
```

Nachdem dieser Hinweis über die Opus-Mailingliste weitergereicht worden war, wurde die Relevanz dieser Sicherheitslücke für die lokale Installation überprüft. OPUS ist auf bestimmte Dateiformate eingegrenzt und es lassen sich nur Dateien bestimmten Typs hochladen. Dies wird anhand der Dateiendung überprüft, was als hinreichend sicher akzeptiert werden kann. So liessen sich zwar theoretisch auch gefährliche Dateien diesen Typs hochladen (oder sich durch Umbenennung tarnen), jedoch können sie nicht übers Web ausgeführt bzw. wieder zu ihrer Originalendung umbenannt werden.

Ein gefährliches Szenario kann jedoch entstehen, wenn ein Angreifer PHP-Dateien (mit der Endung PHP) hochladen kann. Diese Dateien würde der Webserver als Script interpretieren und bei Aufruf automatisch über PHP ausführen. Kennt ein Angreifer den Pfadnamen, so müsste er nur diesen Pfad mit seinem Browser aufrufen und würde auf diese Weise sein Script ausführen können und damit unter Umständen das System schwer beschädigen oder an Daten gelangen, die nicht frei zugänglich sind.

Tatsächlich gibt es in OPUS neben den begrenzten Dateiformaten auch eine Möglichkeit, beliebige Dateien hochzuladen und damit die Sicherheitslücke ausnutzen. Bei Dissertationen und Habilitationen können bleibige Dateiformate als „Originalformat“ hochgeladen werden; sie werden ebenso wie die anderen, ungefährlichen Dateien, bis zu ihrer Freischaltung in einem temporären Ordner gespeichert, der allerdings für den Webserver zugreifbar ist.

Für das lokale Testsystem ebenso wie für das Produktivsystem wurde ein Mechanismus antsprechend dem oben genannten Zitat über eine .htaccess-Datei eingepflegt.

4.3.2 Einblick in Konfigurationsdateien im lib-Verzeichnis

In Version 1 von OPUS lag das lib-Verzeichnis innerhalb des normalen Baumes, in dem auch die anderen Webdocumente liegen. Dies bedeutet jeodch ein Sicherheitsrisiko, da Dritte sich problemlos die zentrale config-Datei opus.conf anschauen können, indem sie einfach im Browser die Serveradresse mit angehängtem /lib/opus.conf eintippen. Da .conf normalerweise nicht beim Webserver als gültiger Dateityp eingetragen ist, wird einfach der Klartext des Dokuments ausgegeben. In der opus.conf stehen zwar keine Passwörter, aber einige Variablen, die zum Beispiel

auf das verwendete Datenbanksystem schliessen lassen und weitere Rückschlüsse auf das verwendete System zulassen. Dies sind zwar keine hochsensitiven Daten, aber sie sollten gleichsam nicht unbedingt an dieser offenen Stelle einsehbar sein. Die Passwörter stehen auch im lib-Verzeichnis, allerdings in einer PHP-Datei, so dass hier bei korrekter Serverkonfiguration keinesfalls der Klar- bzw. Quelltext einsehbar ist.

In neueren OPUS-Versionen ab Version 2 ist das lib-Verzeichnis nicht mehr im sichtbaren Web-Baum untergebracht, sondern liegt oberhalb des sogenannten DocumentRoot. Die Opus-Scripte greifen über relative Pfade darauf zu. Auf diese Weise können sie nicht im Browser sichtbar gemacht werden. Da eine solche Änderung verlangt, dass Pfade angepasst werden, wurde in der lokalen Installation eine .htaccess-Datei für das lib-Verzeichnis hinterlegt, die jeglichen Browserzugriff blockiert. Beim Inkludieren der dort befindlichen Scripte wird die .htaccess ignoriert, so bringt diese Massnahme keine Funktionseinschränkungen mit sich.

4.4 Eigene Erweiterungen von OPUS

4.4.1 Autorensupport

Neu geschaffen wurde ein Bereich, in dem sich ein Autor (mit seinem Bibliotheksausweis und den dazugehörigen Zugangsdaten, siehe Abschnitt 4.4.3 Authentifizierung zur Veröffentlichung eigener Dokumente) einloggen kann und auf einen Blick sehen kann, welchen Status seine eigenen Veröffentlichungen derzeit haben. Eine Screenshot des neuen Bereiches ist in Abbildung 2 gezeigt.

Statistiken zu Ihren Dokumenten

Ihre Dokumente in der Warteschlange (noch nicht freigeschaltet):

- Testhabilitation
- Test
- Blablafaselfasel
- Ollis Preprint
- Testdiss
- Blabla-Buch
- Mein kleiner grüner Kaktus

Ihre freigeschalteten Dokumente:

EIS-Board

- Indexseite Aufrufe: 3
- lexikon.pdf Prüfsumme: b395ef6b703cfb1bb84a478f442c0715 Aufrufe: 1

Testaufsatz

- Indexseite Aufrufe: 0
- ratgeber_urheberrecht.pdf Prüfsumme: 5e1dc261693e95ad1f4a9142e9fed0eb Aufrufe: 0
- emotionen_elearning.pdf Prüfsumme: 10daa9fcb5ef2bad40cab55d9d6a6567 Aufrufe: 0

Home | Suchen | Veröffentlichen | Hilfe | Viewer | Autorenlogin | Logout

Fragen und Anregungen an die Bibliothek

Letzte Änderung: 09.12.04

Abbildung 2 Neu geschaffener Autorenbereich in OPUS

Da der Autorenbereich im originären Opus gar nicht vorhanden ist, wurde er „from scratch“ neu programmiert. Es wurde auch PHP als Programmiersprache genommen. Bei der Programmierung wurde objektorientiert vorgegangen; programmiertechnisch erfolgt die Authentifizierung über Session-Bildung und Session-Variablen.

Auch im Autorenbereich enthalten ist eine Statistik darüber, wie häufig die eigenen Dokumente aufgerufen wurden. Diese Statistik wird anhand eines Programmes namens accesslog2mysql erstellt. Mit Hilfe eines Bash-Scriptes wird turnusmässig die Apache-Access-Logdatei von OPUS in eine

eigens dafür angefertigte Statistikdatenbank übertragen. Diese Datenbank enthält prinzipiell die gleichen Angaben wie die Access-Logdatei des Apache-Webservers, ist aber entsprechend einfacher durchsuchbar und auswertbar. Auf diese Weise können auch rückwirkend Dokumentaufrufe sehr einfach festgestellt werden (sofern eine Logdatei existiert). Im Grunde reicht es, wenn die Logdatei die aufgerufenen Dateien enthält, andere Daten werden hier nicht überprüft. Zu Problemen kann es jedoch kommen, wenn eine Datei zu groß ist und daher fragmentiert ausgeliefert wird. In der Datenbank wird nicht der Statuscode festgehalten, der Aufschluss darauf geben könnte, ob ein Logeintrag die Fortsetzung eines anderen ist. Dieses könnte noch entsprechend geändert werden und die Datenbank bzw. accesslog2mysql entsprechend abgewandelt werden; im Laufe des Projektes reichte die Zeit hierfür nicht mehr aus. Auch eine automatische Konvertierung der Server-Logdatei in die Datenbank findet derzeit noch nicht statt; dies kann jedoch (theoretisch) über den Cron-Dämon leicht realisiert werden.

4.4.2 FAQ-Datenbank

Die FAQ-Datenbank soll dazu dienen, Fragen und die dazugehörigen Antworten zu allen Belangen aus OPUS (technische Erklärungen, Kontaktdaten, Fragen zur Publikation eigener Arbeiten, Autorensupport, etc.) zentral zu speichern und dynamisch zur Verfügung stellen zu können. Auf diese Weise ist es zum Beispiel möglich, nähere Erläuterungen zu bestimmten Formularfeldern aus dem Veröffentlichungsformular hier abzuspeichern und dann im Formular einen Link einzubauen, der genau diesen Eintrag anzeigt. Bisher gibt es HTML-Dateien mit derartigen Erläuterungen, die mit Hilfe von Ankerpunkten aufgerufen werden. In anderen statischen Dateien finden sich weitere Hinweise. Eine Datenbank macht diese HTML-Dateien obsolet, kann alle Fragen und Antworten sammeln und ist übersichtlicher bzw. besser auffindbar und abfragbar. Zudem ist die Datenbank leichter erweiterbar als die statischen Dateien.

Als Datenstruktur für die Einträge wird zunächst HTML genutzt; sollte sich im Laufe der Eintragungsarbeiten herausstellen, dass XML hier hilfreicher wäre, so kann dies noch geändert werden.

Eine Verschlagwortung der Einträge sollte bei dem voraussichtlich nicht allzu großen Umfang der Datenbasis nicht notwendig sein, daher wurde auf Verschlagwortungstabellen und ein entsprechendes System zur Verschlagwortung verzichtet.

Eine Kategorisierung der einzelnen Fragen ist vorgesehen, wobei eine Frage nur genau einer Kategorie zugeordnet werden kann. Eine Mehrfachkategorisierung ist voraussichtlich nicht notwendig und daher nicht vorgesehen.

Sowohl die Kategorien als auch die einzelnen Fragen können in eine Reihenfolge gebracht werden. Dadurch wird es möglich, aus der gesamten Datenbank ein logisch aufgebautes FAQ-Dokument zu erzeugen, mit dem eine gute Übersicht über das gesamte Hilfesystem gewonnen werden kann.



Abbildung 3 ER-Diagramm der FAQ-Datenbank

Die FAQ-Datenbank ist bislang nur sehr rudimentär verwirklicht. Es wurde eine Struktur für die Datenbank angelegt (siehe Abbildung 3), jedoch sind noch keine Einträge vorhanden und die

Datenbank ist noch nicht in die lokale OPUS-Seitenstruktur eingebaut worden. Daher ist auch die Formatfrage für die Einträge noch nicht abschliessend geklärt; weiterhin wird aber zunächst eine HTML-Formatierung angestrebt.

4.4.3 Authentifizierung zur Veröffentlichung eigener Dokumente

In der Originalversion von Opus ist der Zugriffsschutz an bestimmte IP-Bereiche geknüpft. Dies ist zwar eine recht zuverlässige Methode, um Missbrauch zu verhindern, aber wenig serviceorientiert, da sie voraussetzt, dass der Autor einer Arbeit an einem Rechner sitzt, für dessen IP-Adresse diese Funktion zugänglich ist.

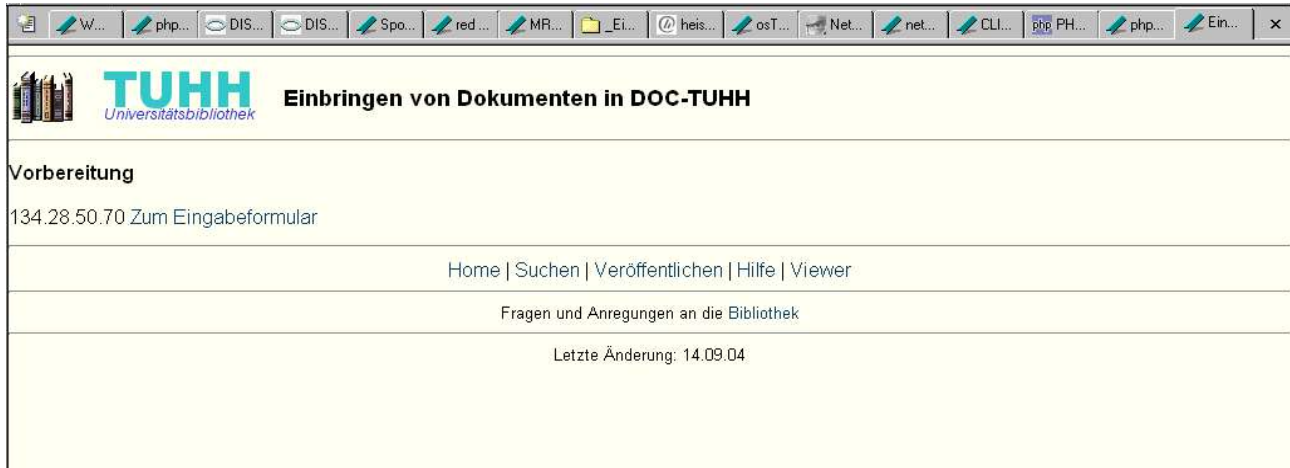


Abbildung 4 Früherer Eingangsscreen für das Veröffentlichungsformular

Diese Methode wurde gründlich überarbeitet und gegen einen Passwortschutz ersetzt. Die ursprüngliche erste Seite zum Veröffentlichen eines neuen Dokuments ist in Abbildung 4 zu sehen; die Ansicht nach der Umarbeitung ist in den Abbildungen 6 und 5 zu sehen.

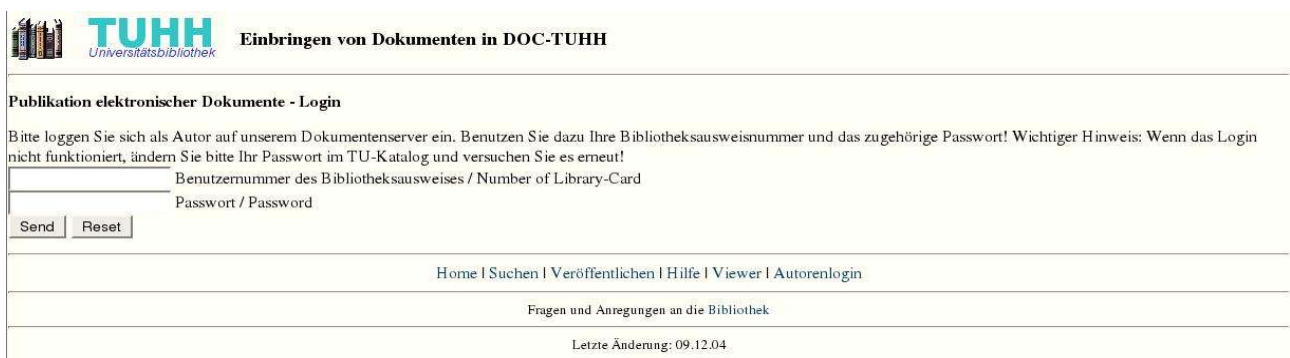


Abbildung 5 Login-Bildschirm nach der Umarbeitung



Publikation elektronischer Dokumente - Hinweise

In diesem Bereich können Sie Dokumente auf unseren Dokumentenserver hochladen und damit einer breiten Öffentlichkeit zur Verfügung stellen. **Zum Upload benötigen Sie einen gültigen Benutzerausweis bei der Universitätsbibliothek der TU Hamburg-Harburg!** Bitte beachten Sie, dass Sie das Veröffentlichungsrecht für die Arbeit benötigen (dies haben Sie, wenn Sie die Arbeit selbst geschrieben haben und die Rechte nicht an einen Verlag oder eine andere dritte Person übergeben haben). Sollte Ihre Publikation bereits durch einen Verlag veröffentlicht worden sein, so benötigen Sie in der Regel die Genehmigung, sie zusätzlich als elektronisches Dokument auf einem Dokumentenserver bereitzustellen. Bitte holen Sie vor dem Upload diese Genehmigung ein!

Durch den Upload willigen Sie ein, dass Ihr Dokument für eine von Ihnen bestimmbare Gültigkeitsdauer von uns in einer Datenbank gespeichert und elektronisch verfügbar gemacht wird. Das Dokument wird dabei weder von der TUB noch von Dritten inhaltlich verändert, technische Änderungen (zum Beispiel das Format) zum Zwecke der Langzeitarchivierung sind jedoch nicht ausgeschlossen. Eine Weitergabe von Daten an die Deutsche Bibliothek ist ggf. erlaubt.

Hiermit übertrage ich der Technische Universität Harburg das Recht, das übermittelte Dokument elektronisch zu speichern und in Datennetzen öffentlich zugänglich zu machen. Ich versichere, daß mit einer derartigen Veröffentlichung keine Rechte Dritter verletzt werden. Meine Rechte zur Verwertung in einer anderen körperlichen Form bleiben davon unberührt.

☐ Ich habe diese rechtlichen Hinweise gelesen und bin damit einverstanden

[Home](#) | [Suchen](#) | [Veröffentlichen](#) | [Hilfe](#) | [Viewer](#) | [Autorenlogin](#)

[Fragen und Anregungen an die Bibliothek](#)

Letzte Änderung: 09.12.04

Abbildung 6 Neuer Publikationsbildschirm, rechtliche Hinweise (Eingangsbildschirm)

Der neue Mechanismus basiert auf einer Authentifizierung mit dem Bibliotheksausweis und dem dazugehörigen Passwort. Über diese Bibliotheks-Zugangsdaten können auch Bücher entliehen werden. So wird keine weitere Nutzerkennung angelegt, der User braucht sich kein neues Passwort zu merken und muss sich nicht für den neuen Dienst extra registrieren. Auch in der Umsetzung hat dieser Aufsatz Vorteile: es muss kein neues Benutzer-Verwaltungssystem implementiert werden.

Diese Funktionalität kann nur von der Bibliothek der TU Harburg genutzt werden und ist nicht ohne weiteres verallgemeinerbar, da sie sehr stark auf dem lokalen Bibliothekssystem in der TU (PICA) basiert. Es wäre allerdings denkbar, ein solches Zugangssystem auch in anderen Bibliotheken, die PICA anwenden, umzusetzen. Dies würde allerdings weitere Erfordernisse an den Webserver mit sich bringen (eine Sybase-Schnittstelle müsste in PHP einkompiliert werden).

Die Modifikation macht es auch möglich, dass für die Publikation im Datenerfassungsformular der Name des eingeloggten Autors voreingestellt wird. In der Originalversion war das Feld in der Voreinstellung frei.

4.4.4 Zugangsbegrenzung für Dokumente

Die Möglichkeit, Dokumente nur einem begrenzten Nutzerkreis zur Verfügung zu stellen, ist ganz neu.

The screenshot shows a web form titled 'Anmelden von Dokumenten für DOC-TUHH' from the 'TUHH Universitätsbibliothek'. The form is for restricting document access to specific users. It includes fields for 'IP-Adresse', 'Rechnername (FQDN)', 'Username', and 'Passwort'. A section titled 'Verfügbar für' allows selecting specific work areas from a list: 'keine Auswahl', 'FSP 1: Stadt, Umwelt, Technik', 'Abwasserwirtschaft 1-02 (Kommunale und industrielle Abwasserwirtschaft)', 'Technikbewertung und Technikgestaltung 1-11', and 'Prozeßtechnik und Berufliche Bildung 1-12 (Berufswissenschaft Elektrotechnik/Informatik)'. Navigation buttons 'Weiter' and 'Formularinhalt löschen' are at the bottom. A footer contains links like 'Home', 'Suchen', and 'Veröffentlichen', along with the date 'Letzte Änderung: 21.01.05'.

Abbildung 7 Dokumente nur für einen begrenzten Nutzerkreis freigeben

Der in Abbildung 7 gezeigte Bildschirm präsentiert sich dem Nutzer, nachdem er den Dokumenttyp für seine neue Publikation ausgewählt hat. Es kann eine frei festlegbare IP- oder Netz-Adresse oder ein Rechnername bzw. eine Domäne angegeben werden, für die das betreffende Dokument dann ausschließlich zugänglich ist. Außerdem ist es möglich, aus einer Liste bestimmte TU-Arbeitsbereiche und Institute zu wählen, für die dann automatisch die IP-Adresse eingetragen wird. Die zugehörigen Daten sind in der OPUS-Datenbank gespeichert.

In der Realisation sieht diese Begrenzung so aus, dass im Hintergrund eine .htaccess-Datei erstellt wird, in der die angegebenen Parameter eingetragen werden. Über eine .htaccess-Datei wird beim Apache-Webserver eingestellt, welche Bereiche Zugriff auf eine Seite oder einen bestimmten Seitenteil haben dürfen. Die IP-Bereiche bzw. Domänen der Arbeitsbereiche wurden in Form einer Ergänzung der OPUS-Datenbank implementiert. In einer Tabelle sind die entsprechenden Daten gespeichert und werden zum Aufbau der .htaccess-Datei ausgelesen und genutzt. So ist auch sichergestellt, dass niemand über die Veröffentlichungsseiten das lokale Netz und seine IP-Adressen ausspionieren kann, da die Informationen nicht auf den Seiten preisgegeben werden. Parallel zum Aufbau der .htaccess wird auch die robots.txt modifiziert. In der robots.txt kann Suchmaschinen verboten werden, bestimmte Ordner zu indexieren. Die robots.txt versteht sich aber nur als Richtlinie; nicht alle Suchmaschinen halten sich daran. Da Suchmaschinenrobots aber sowieso keinen Zugriff auf die Dokumente haben, wenn sie zugriffsbeschränkt sind, können sie sie auf keinen Fall indexieren. Insofern dient die robots.txt in diesem Fall der Arbeitserleichterung für die „braven“ Robots, die dann von vornherein bestimmte Ordner übergehen.

Der Autor darf selbst bestimmen, für wen er seine Arbeit freigeben möchte; der bibliothekarische Bearbeiter kontrolliert diese Eingabe bei der Freischaltung und kann sie bei Bedarf auch ändern.

4.4.5 Automatische URN-Zuweisung für Dissertationen und Habilitationen

Ein URN (Uniform Resource Name) ist ein sogenannter persistenter Identifikator (persistent identifier, PI), der ein einzelnes Dokument eindeutig kennzeichnet. URN ist damit eine Erweiterung der flüchtigen URLs (Uniform Resource Locators) und stellt einen Mechanismus zur Verfügung, dass Dokumente auch über den URL hinaus identifiziert und aufgefunden werden können. Neben dem URN gibt es weitere (auch internationale) Ansätze zu Persistent Identifiern (z.B. die Persistent URL PURL des amerikanischen Bibliothekskonsortiums OCLC⁴ oder der DOI⁵), jedoch hat sich die Deutsche Bibliothek auf das URN-Modell festgelegt und dementsprechend verbreitet ist der URN mittlerweile in der deutschen Bibliotheks- und Online-Publikationslandschaft.

Die Deutsche Bibliothek betreibt eine Resolving-Tabelle, in der dem URN ein URL zugeordnet wird. Ändert sich der URL eines mit einem URN ausgezeichneten Dokumentes, so hat die Bibliothek, aus deren Namensraum dieser URN stammt, die Aufgabe, die Tabelle zu updaten und die neue Verknüpfung einzutragen.

Für die Browser Mozilla und Internet Explorer steht ein Plugin zur Verfügung, durch das es möglich ist, einen URN direkt in die Adressleiste des Browsers einzugeben und daraufhin mit dem entsprechenden Dokument verbunden zu werden. Diese Plugins wurden im Projekt Epicur an der Deutschen Bibliothek entwickelt und werden auf deren Seiten zum Download angeboten ([3]). Alternativ kann ein URN auch über einen Web-Resolver über http aufgelöst werden. Dazu muss der URL <http://nbn-resolving.de/urn/resolver.pl?urn=<urn>> aufgerufen werden. Statt dem <urn> ist dann ein realer URN einzusetzen.

Im Laufe des Projektes wurde zunächst ein eigener URN-Namensraum für die TU Harburg bei der DDB beantragt. Es wurde der Namensraum mit dem Präfix `urn:nbn:de:gbv:830` zugewiesen, mit dem nun eigene Dissertationen versehen werden können. Im Falle der lokalen Installation wird dieser Präfix folgendermassen fortgesetzt: Die Zeichenkette „-opus-“ wird angehängt, gefolgt von der internen Opus-ID des Datensatzes. Danach folgt eine angehängte, nach einem Algorithmus der Deutschen Bibliothek berechnete, Prüfziffer. Ein Beispiel für einen von der TU Harburg vergebenen, gültigen URN lautet `urn:nbn:de:gbv:830-opus-514`. Mit diesem beispielhaften URN wird eine an der TU Harburg angefertigte Dissertation von Markus Venzke gekennzeichnet.

Der URN ist Teil der zum Dokument gehörenden Metadaten und wird künftig für elektronische Publikationen in Ergänzung zum URL im Bibliothekskatalog nachgewiesen. Alle bisher eingetragenen Dissertationen wurden retrospektiv mit einem URN versehen, der manuell an die DDB gemeldet wurde. Im Bibliothekskatalog ist der URN für diese Publikationen bislang nur bei einer der Dissertationen vermerkt⁶; für die anderen Dissertationen soll der URN alsbald nachgetragen werden.

Das OPUS-System wurde im Rahmen des Projektes dergestalt angepasst, dass für Dissertationen und Habilitationen (also Hochschulschriften im engsten Sinne) automatisch ein URN aus dem Namensraum der TU Harburg vergeben und per XEpicur über das OAI-Protokoll an die Deutsche Bibliothek gemeldet wird. Über XEpicur wird lediglich die URN/URL-Verknüpfung übertragen, keine weiteren Metadaten zur Publikation. Die Übermittlung der Daten über OAI/XEpicur ist bereits seit längerem Bestandteil der offiziellen OPUS-Distribution, wurde aber in der lokalen Installation noch nicht genutzt. Dies soll sich nach Projektablauf und der Umstellung des Produktivservers ändern.

Im originären OPUS ist die Zuweisung eines URN nicht voll automatisiert. Ein URN wird dann zugewiesen, wenn ein Dokument freigeschaltet wird, und auch nur dann, wenn der Bearbeiter diese Funktion aufruft (über einen Button in der Freischaltmaske möglich).

4 OCLC = Online Computer Library Center, näheres zur PURL ist unter <http://purl.oclc.org> zu finden

5 DOI = Digital Object Identifier, <http://www.doi.org>

6 Dieser Datensatz ist unter <http://s250.b.tu-harburg.de:8080/DB=1/SET=8/TTL=2/SHW?FRST=1> zu finden

Eine automatische Vergabe von URNs wäre jedoch wünschenswert, da in den Richtlinien für die URN-Vergabe an der Deutschen Bibliothek erwähnt wird, dass der URN auch in der Publikation vermerkt werden soll:

Persistent Identifier für digitale Inhalte dienen einerseits der Langzeitverfügbarkeit der Online-Ressourcen, andererseits der dauerhaften sicheren Referenzierbarkeit. Im Hinblick auf die persistente Zitierfähigkeit einer URN durch einen Nutzer empfiehlt Die Deutsche Bibliothek, die im Projekt vergebenen URNs für digitale Dissertationen zusätzlich in die Ressource selbst einzubringen.

In Abstimmung mit der Koordinierungsstelle "DissOnline" an Der Deutschen Bibliothek, können URNs formal unterhalb des Titels, aber vor dem Beginn des Textes positioniert werden.

Damit Print- und Online-Dissertation identisch sind, muss diese Angabe in beiden Versionen erfolgen d.h. das Einbinden der URN in die Ressource ist vom Doktoranden selbst auszuführen. [1]

Dies ist nur dann möglich, wenn der Autor den URN erfährt, bevor er seine Arbeit hochlädt. Dafür ist eine vollautomatische Zuweisung notwendig, da spätestens nach der Erfassung der Metadaten der URN angezeigt werden muss, der dann für genau diese Publikation reserviert wird. Für die lokale Installation wurde diese Automatik bei Dissertationen und Habilitationen eingeführt (siehe Abbildung 8). In der offiziellen OPUS-Distribution wird diese Automatik erst später (nämlich bei der Bearbeitung der Dokumente) ausgelöst. Auf diese Weise kann dem Autor der URN erst nach dem Erstupload mitgeteilt werden; das Dokument muss später ausgetauscht werden.

Die lokale Änderung wurde realisiert, indem die interne OPUS-ID bereits vor dem Upload kalkuliert wird (normalerweise wird diese ID erst nach dem Upload der Dateien vergeben, wenn die Metadaten in die Datenbank übertragen werden). Diese Methode funktioniert vermutlich nur auf einem Server mit wenig Betrieb zuverlässig. Wenn viel Betrieb ist und die Möglichkeit einer Paralleleintragung besteht müssen Vorsichtsmaßnahmen getroffen werden, dass ein URN nicht doppelt vergeben wird. Eine Alternative wäre, den URN nicht mit Hilfe der internen ID zu berechnen, sondern stattdessen den ersten Titel, ein Autoren kürzel o.ä. zu benutzen.

The screenshot shows the 'File-Upload-Formular' from TUHH (Technische Universität Hamburg) Universitätsbibliothek. It is titled 'Schritt (1) - (2) - (3)' with step 3 highlighted in red. The text explains that documents are assigned a URN (Uniform Resource Name) and provides an example: 'urn:nbn:de:gbv:830-opus-1253'. A red circle highlights the text about URN assignment and the example URN. Below this, it mentions that graphic files can be in JPG, JPEG, and GIF formats. At the bottom, there is a note 'Anmerkungen zum HTML-Format'.

TUHH File-Upload-Formular
Universitätsbibliothek

Schritt (1) - (2) - (3)

Mit Hilfe des **Browse-Buttons** können Sie auf Ihrem lokalen Rechner nach den Dateien suchen, die Sie übermitteln wollen.
Zur Zeit werden die Formate **PDF (Portable Document Format)**, **PostScript**, **HTML** unterstützt. Bitte bringen Sie die Arbeit zusätzlich im Originalformat (zu Archivierungszwecken) ein.

Ihrem Dokument wird ein sogenannter URN (Uniform Resource Name) zugewiesen. Dieser URN kennzeichnet Ihr Dokument eindeutig und sollte auch in Ihrer Arbeit als Referenzquelle genannt werden. Bitte ergänzen Sie Ihre Dokumente um diese Angabe, wenn dies möglich ist!
Der URN Ihrer Arbeit lautet voraussichtlich: **urn:nbn:de:gbv:830-opus-1253**

Graphikdateien in Verbindung mit HTML-Dokumenten können in den Formaten **JPG**, **JPEG** und **GIF** vorliegen.

Anmerkungen zum HTML-Format

Abbildung 8 Anzeige des URNs vor dem Upload

Eine weitere Problemstellung ist dabei noch ungelöst: es kann passieren, dass eine Publikation aus mehreren Dokumenten besteht, die alle verschiedene URNs erhalten. Die Regel ist, dass ein URN pro Publikation vergeben werden soll, aber es kann auch Ausnahmen geben, in denen eine Vergabe mehrerer URNs zur Identifikation bestimmter Publikationsteile sinnvoller erscheint. Wie kann man

nun automatisch feststellen, wie viele URNs an eine Publikation gebunden werden sollen?

Es erscheint sinnvoll, standardmässig einen einzigen URN zu vergeben, den der Autor in allen Einzeldokumenten (die zu einem Gesamtdokument gehören) angibt. Glaubt der Autor, dass eine Unterscheidung sinnvoll ist, dann muss er selbst tätig werden und seine Publikation in mehrere autonome Teile einteilen, die er dann allerdings auch separat in das System einspeisen muss – für jedes der Objekte muss ein eigener Datensatz angelegt werden. Dieses sollte dem Autor im Hilfesystem mitgeteilt werden.

Auf der anderen Seite muss ebenfalls sichergestellt sein, dass eine Publikation nur einen einzigen URN erhält. Diese Regel soll dadurch befolgt werden, dass auf dem lokalen Dokumentenserver ausschließlich Publikationen aus dem eigenen Haus verbreitet werden (also zum Beispiel Dissertationen eigener Doktoranden).

Die automatische Zuweisung eines URN wurde im Laufe des Projektes für die lokale Installation eingeführt, allerdings nur für Dissertationen und Habilitationen.⁷ Im Prinzip kann jede Publikation (nicht nur Dissertationen oder Habilitationen) einen URN bekommen, Sinn macht dies aber nur für langlebige Dokumente. Für den Mensaspiseplan o.ä. ist die Zuweisung eines immergültigen Identifiers aus naheliegenden Gründen nicht sinnvoll.

4.4.6 Prüfsummen



The screenshot shows the 'File-Upload-Formular' from TUHH. It includes instructions for file formats (PDF, PostScript, HTML, etc.) and a section for uploading files. The 'Format' dropdown is set to 'PDF (Portable Document Format)'. The 'Datei 1' field contains the path '/home/olli/studium/Diplomarbeit_Marahrens.pdf'. The 'Prüfsumme dieser Datei' field displays the calculated MD5 checksum: 'd0f15deac36a5fb6cc05bdd24462c30'. A button 'Datei(en) auf den Volltextserver übertragen' is visible at the bottom of the form.

Abbildung 9 Angabe der Prüfsumme beim Dateiupload

Eine Ergänzung des ursprünglichen Opus-Systems, die möglicherweise auch in die offizielle Distribution übernommen wird, ist die Speicherung und Berechnung von Prüfsummen für Dokumente. Dabei wird die md5-Prüfsumme benutzt.

Bereits vor der Übertragung des Dokuments muss der Autor die Prüfsumme selbst berechnen und im Upload-Formular angeben. Die Prüfsumme wird dann auf dem Server erneut berechnet und mit der angegebenen Prüfsumme verglichen. Stimmt sie nicht überein, so wird der Upload abgebrochen,

⁷ Annette Maile vermerkt dazu: „in der jetzigen Opus-Version wird fuer Diss. und Habil. automtisch beim Freischalten eine Urn vergeben. In Opus 3 kann man wahlweise fuer alle Dokumnte oder nur fuer Diss. und Habil. eine Urn vergeben.“ [10]

da in dem Fall Verdacht besteht, dass jemand die Übertragung manipuliert hat. Andernfalls wird die Prüfsumme in einer neu geschaffenen Datenbanktabelle gespeichert und so in die Metadaten des Dokuments aufgenommen.

Da der Autor die Prüfsumme selbst berechnen soll muss ihm mitgeteilt werden, wie er dies machen kann. Die Berechnung einer Prüfsumme gehört mitnichten zum Allgemeinwissen, was man beim Endnutzer voraussetzen kann. Hierzu soll ein Eintrag in der FAQ-Datenbank untergebracht werden, der beschreibt, wozu die Prüfsumme benötigt wird und wie man sie ermittelt. Ein zu empfehlendes Tool, mit dem Autoren auf einfache Art und Weise die MD5-Prüfsumme ihrer Dokumente bestimmen können, findet sich unter <http://www.carstens-informationsmanagement.de/prod/md5f/index.shtml>. Für Linux-Nutzer kann ein Hinweis auf das Kommando md5sum gegeben werden, das auf einem normal ausgestatteten Linuxsystem vorhanden sein sollte.

Im Autorenbereich kann der Autor jederzeit sehen, ob sich eine Prüfsumme eines seiner Dokumente geändert hat und kann so erkennen, ob es nachträglich verändert wurde. Eine periodische automatische Prüfung der Prüfsummen wäre denkbar, wurde aber bislang nicht realisiert. Sie kann aber ggf. auch von der DDB übernommen werden, da diese Dienstleistung mit im Angebot der DDB steht.

4.4.7 XMetaDiss-Schnittstelle

In Rücksprache mit den offiziellen OPUS-Entwicklern und Distributoren wurde die Entwicklung der XMetaDiss-Schnittstelle in das Projekt mit aufgenommen. Eine solche Schnittstelle existierte bislang noch nicht, wird jedoch benötigt, um mit den Entwicklungen bei der Deutschen Bibliothek Schritt zu halten und künftig auch in OPUS die Lieferung von XMetaDiss-Daten über OAI-PMH⁸ an die DDB zu ermöglichen.

Zum besseren Verständnis zunächst ein paar Worte zum OAI-Protokoll. OAI-PMH ist ein Protokoll, das dazu dient, Metadaten plattformunabhängig bereitstellen und austauschen zu können. OPUS unterstützt OAI-PMH in Version 2.0 bereits seit OPUS 2.0 standardmäßig. Benutzt wird dabei eine PHP-Implementierung des OAI-Protokolls, die von Heinrich Stamerjohanns als GPL-Software bereitgestellt und von Anette Seiler (HBZ NRW⁹) für OPUS adaptiert wurde. Dabei wird ein Harvesting per URL durchgeführt: OAI-Parameter werden per GET an den Server übergeben und von der Schnittstelle entsprechend ausgewertet. OAI-PMH ist ein XML-Format, das als Container für verschiedene Metadatenformate dienen kann.

XMetaDiss ist ein solches, ebenfalls XML-basiertes Metadatenformat, das benutzt wird, um Metadaten über Dissertationen und Habilitationen auszutauschen. Andere Dokumenttypen werden nicht in XMetaDiss dargestellt. XMetaDiss wurde von der Deutschen Bibliothek entwickelt und soll das alte Format MetaDiss ablösen. Seit Januar 2005 akzeptiert die Deutsche Bibliothek Datensätze, die in XMetaDiss abgeliefert werden. Dabei wird ein Harvesting bei den teilnehmenden Bibliotheken in einem vorher abgesprochenen Rhythmus über OAI-PMH 2.0 durchgeführt. Die XMetaDiss-Daten werden in einen OAI-Container gekapselt und vom Harvester der Deutschen Bibliothek (oder anderen Harvestern, die XMetaDiss interpretieren) ausgewertet. Die Spezifikation von XMetaDiss findet sich unter [9].

Diese oben beschriebene OAI-PMH-Implementierung für OPUS wurde modifiziert und um XMetaDiss erweitert. Für den Endnutzer ist die Schnittstelle weitestgehend unsichtbar. Die Daten werden vom Harvester der Deutschen Bibliothek über HTTP angefordert (indem der Harvester einen festgelegten URL auf dem jeweiligen Dokumentenserver aufruft) und über die Schnittstelle im XMetaDiss-konformen XML zurückgegeben. Der Harvester sorgt dann DDB-intern für die

⁸ OAI-PMH = Open Archives Initiative Protocol for Metadata Harvesting, <http://www.openarchives.org>

⁹ HBZ-NRW = Hochschulbibliothekszentrum Nordrhein-Westfalen, Serviceeinrichtung für Bibliotheken

Weiterverarbeitung der Daten (Eintrag in den DDB-Katalog, Anforderung dazugehöriger Volltext-Dateien etc.). Die Metadaten stehen frei zur Verfügung, das heisst auch andere Clients als der Harvester der DDB können den entsprechenden URL aufrufen und ihrerseits die Daten auswerten. XMetaDiss ist jedoch (ebenso wie XEpicur, aber anders als zum Beispiel OAI_DC) primär für den Harvester der DDB gedacht. Das Metadatenformat OAI_DC ist für OAI-basierte Suchmaschinen interessant; technisch ist der Ablauf identisch mit dem der Auslieferung der XMetaDiss-Daten.

Alle OAI-Optionen stehen auch bei der XMetaDiss-Schnittstelle zur Verfügung. Die Optionen sind in Tabelle in übersichtlicher Form dargestellt und werden im Folgenden detailliert beschrieben.

from	Startdatum (Format YYYY-MM-DDTHH:MM:SSZ oder YYYY-MM-DD)
until	Enddatum (Format YYYY-MM-DDTHH:MM:SSZ oder YYYY-MM-DD)
metadataPrefix	xMetaDiss
verb	ListRecords GetRecord ListIdentifier
identifier	OAI-Identifier-String (bei verb=GetRecord)

Tabelle 2 OAI-Optionen für die XMetaDiss-Schnittstelle

Der Harvester der Deutschen Bibliothek ruft die Daten nach bestimmten Zeitabständen, die mit den teilnehmenden Bibliotheken vereinbart werden, ab. Da noch kein Produktivbetrieb an der TU Harburg stattfindet ist dieser Zeitraum noch nicht vereinbart; die Tendenz geht zu einer Abholung alle vier Wochen, da der Datenbestand an Online-Dokumenten in der TUB nicht hoch ist. Start- und Endzeitpunkt des Harvesters werden in Form von Parametern an OAI weitergegeben, das dementsprechend die Daten raussucht und nur die zutreffenden zurückliefert. Die OAI-Abfrage sieht also in etwa folgendermassen aus:

from=Startdatum

until=Enddatum

Die Granularität der Daten ist dabei entweder nur das Datum im Format YYYY-MM-DD oder zusätzlich eine genaue Zeitgrenze. OPUS speichert den sekundengenauen Zeitstempel für jedes Dokument. Die Zeit wird (wenn sie abgefragt werden soll) durch ein T vom Datum abgetrennt und mit einem Z abgeschlossen, so dass der komplette String dann lautet YYYY-MM-DDTHH:MM:SSZ. Auch das gewünschte Metadatenformat muss in der Abfrage mitgeliefert werden:

metadataPrefix=xMetaDiss

Der Harvester lässt sich als Option alle zutreffenden Datensätze komplett zurückgeben, dies entspricht dem Parameter

verb=ListRecords

Die Abfrage funktioniert auch für spezielle Datensätze, dann muss verb=GetRecord lauten und ein weiterer Parameter (identifier) mit der genauen Datensatz-Nummer folgen.

Mehr zur Grundlage des OAI-Protokolls findet sich unter [8].

Bei der PHP-Implementierung von OAI werden diese Parameter per GET übertragen, also direkt an den URL angehängt.

Die Rückgabe der Schnittstelle enthält dann XMetaDiss (oder, falls angefordert, andere unterstützte Metadatenformate), gekapselt in einen OAI-Header. Der OAI-Header für die Anfrage

<http://linws.b.tu-harburg.de:81/oai/oai2.php?verb=ListRecords&metadataPrefix=xMetaDiss>

sieht auf dem Testserver an der TU Harburg folgendermassen aus:

```
<OAI-PMH xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2005-03-06T17:08:59Z</responseDate>
<request verb="ListRecords"
metadataPrefix="xMetaDiss">http://linws.b.tu-
harburg.de/oai/oai2.php</request>
<ListRecords>
...
<resumptionToken expirationDate="2005-03-07T16:08:59Z"
completeListSize="9" cursor="0">2023777304</resumptionToken>
</ListRecords>
</OAI-PMH>
```

Dort, wo die drei Punkte als Platzhalter stehen, werden die einzelnen zurückerhaltenen Einträge gekapselt. Jeder Eintrag bekommt wiederum einen kurzen Header zur Herstellung der OAI-Konformität, der OAI-relevante Metadaten enthält. Ein beispielhafter Datensatz-Header sieht folgendermassen aus:

```
<record>
<header>
<identifizier>oai:oaiharvester.opus.tu-harburg.de:76</identifizier>
<datestamp>2004-08-24T10:00:18Z</datestamp>
<setSpec>ddc:004</setSpec>
<setSpec>pub-type:8</setSpec>
</header>
<metadata>
...
</metadata>
</record>
```

Auch hier sind wieder drei Punkte als Platzhalter gesetzt. Dort erscheint in der Ausgabe das angeforderte Metadatenformat. Ein beispielhafter Metadatensatz in XMetaDiss sieht folgendermassen aus:

```
<xMetaDiss:xMetaDiss
xsi:schemaLocation="http://www.ddb.de/standards/xMetaDiss/
http://www.ddb.de/standards/xmetadiss/xmetadiss.xsd">
<dc:title xsi:type="ddb:titleIS0639-2"
lang="ger">Testhabilitation</dc:title>
<dc:title xsi:type="ddb:titleIS0639-2" lang="eng"
ddb:type="translated">Testing</dc:title>
<dc:creator xsi:type="pc:MetaPers">
<pc:person>
<pc:name type="nameUsedByThePerson">
<pc:foreName> Oliver</pc:foreName>
<pc:surName>Marahrens</pc:surName>
</pc:name>
<pc:email>o.marahrens@tu-harburg.de</pc:email>
</pc:person>
</dc:creator>
<dc:subject xsi:type="xMetaDiss:DDC-SG">020</dc:subject>
<dc:subject xsi:type="xMetaDiss:SWD">Informatik</dc:subject>
```

```

<dc:subject xsi:type="xMetaDiss:noScheme">
Informatics</dc:subject>
<dcterms:abstract xsi:type="ddb:contentISO639-2" lang="ger"
ddb:type="ddb:noScheme">YeahYeahYeah</dcterms:abstract>
<dcterms:abstract xsi:type="ddb:contentISO639-2" lang="eng"
ddb:type="ddb:noScheme">YeahYeahYeah</dcterms:abstract>
<dc:publisher xsi:type="cc:Publisher">
<cc:universityOrInstitution>
<cc:name>Universitätsbibliothek der Technischen
Universität</cc:name>
<cc:place>Hamburg-Harburg</cc:place>
</cc:universityOrInstitution>
<cc:address cc:Scheme="DIN5008">Denickestr. 22, 21071
Hamburg</cc:address>
</dc:publisher>
<dc:contributor xsi:type="pc:Contributor" thesis:role="advisor">
<pc:person>
<pc:name type="nameUsedByThePerson">
<pc:foreName/>
<pc:surName>Dipl.-Bibl. Hans Dampf</pc:surName>
</pc:name>
<pc:academicTitle/>
</pc:person>
</dc:contributor>
<dcterms:dateAccepted xsi:type="dcterms:W3CDTF">2004-01-
01</dcterms:dateAccepted>
<dc:type
xsi:type="dcterms:DCMIType">ElectronicThesisandDissertation</dc:ty
pe>
<dc:identifier xsi:type="urn:nbn">urn:nbn:de:gbv:830-opus-
1142</dc:identifier>
<dcterms:medium
xsi:type="dcterms:IMT">application/pdf</dcterms:medium>
<dcterms:medium
xsi:type="dcterms:IMT">application/msword</dcterms:medium>
<dcterms:medium
xsi:type="dcterms:IMT">application/vnd.sun.xml.writer</dcterms:med
ium>
<dc:language xsi:type="dcterms:ISO639-2" lang="ger"/>
<dc:rights xsi:type="ddb:noScheme">
Copyright der Metadaten: Universitaetsbibliothek TU Hamburg-
Harburg
</dc:rights>
<thesis:degree>
<thesis:level>thesis.habilitation</thesis:level>
<thesis:grantor xsi:type="cc:Corporate">
<cc:universityOrInstitution>
<cc:name>Technische Universität</cc:name>
<cc:place>Hamburg-Harburg</cc:place>
<cc:department>
<cc:name>FSP 5: Werkstoffe, Konstruktion, Fertigung</cc:name>
</cc:department>
</cc:universityOrInstitution>
</thesis:grantor>

```

```

</thesis:degree>
<ddb:contact ddb:contactID="F6000-0198"/>
<ddb:fileNumber>2</ddb:fileNumber>
<ddb:fileProperties ddb:fileName="Protokoll_13.pdf"
ddb:fileID="file114-0" ddb:fileSize="69597"
ddb:fileDirectory="/">aus:
Praesentationsformat</ddb:fileProperties>
<ddb:fileProperties
ddb:fileName="Evaluationsbericht_Online_Kurs_pp-guide.pdf"
ddb:fileID="file114-1" ddb:fileSize="234767"
ddb:fileDirectory="/">aus:
Praesentationsformat</ddb:fileProperties>
<ddb:transfer
ddb:type="dcterms:URI">http://linws:81/volltexte/2005/114/containe
r.tgz</ddb:transfer>
<ddb:identifizier ddb:type="URL">
http://linws:81/volltexte/2005/114/original/Protokoll_12.doc
</ddb:identifizier>
<ddb:identifizier ddb:type="URL">
http://linws:81/volltexte/2005/114/original/clix_protokoll11.sxw
</ddb:identifizier>
<ddb:identifizier
ddb:type="URL">http://linws:81/volltexte/2005/114/</ddb:identifizier
>
<ddb:rights ddb:kind="free"/>
</xMetaDiss:xMetaDiss>

```

Die Daten, die hier geliefert werden, zieht sich das Schnittstellenscript direkt aus der OPUS-Datenbank. XMetaDiss holt sich dabei nur Datensätze vom Typ Dissertation oder Habilitation, da das Format XMetaDiss ursprünglich nur für derartige Dokumenttypen vorgesehen ist. Weitere Bedingung ist, dass dem Dokument ein URN zugewiesen ist. Dies geschieht in der lokalen Installation sowie in der offiziellen Distribution für Dissertationen und Habilitationen automatisch. Der Grund dafür, dass nur Datensatz mit URN einbezogen werden, liegt darin, dass das Element `<dc:identifizier xsi:type="urn:nbn">` ein Pflichtelement in XMetaDiss ist. Ohne dieses Element ist das XML-Dokument also nicht valide und würde vom Harvester nicht anerkannt werden.

Eventuelle Archivdateien, die nur temporär benötigt werden, werden vom Script aufgebaut und können nach erfolgreichem Harvester-Durchlauf entfernt werden. Diese Archivdateien werden dann erstellt, wenn mehr als eine Datei zu übertragen ist, da der Harvester der DDB nur eine Datei pro Datensatz abholen kann. Gehören mehrere Dateien zum Datensatz wird also ein entsprechendes Archiv aufgebaut.

Die XMetaDiss-Schnittstelle liegt inzwischen in funktionierender Form vor und die letzten Testläufe mit der Deutschen Bibliothek sind positiv verlaufen. Ein Problem ergab sich noch bei der Codierung, da XMetaDiss standardmässig nach UTF-8 codierte Daten erwartet. Dies war anfangs nicht berücksichtigt worden und in der lokal installierten Version wurde mit ISO8859-1 codiert. Daher kam es zu Unverträglichkeiten. Dieser Fehler wurde inzwischen behoben und alle Daten werden nun in UTF-8 ausgeliefert.

Abschliessende Tests von Seiten der OPUS-Entwickler (Prüfung auf das korrekte Zusammenspiel mit der offiziellen Distribution und letztlich die Einpassung in die Distribution) sind noch nicht abgeschlossen. Nach Abschluss dieser Tests wird die Schnittstelle zu einem AddOn-Paket zusammengefasst und über die offizielle Seite der OPUS-Distribution zur Verfügung gestellt. Die Schnittstelle wird dann in die Version 3.0 von OPUS eingepasst und dort enthalten sein.

5 Fazit

Durch das vorliegend beschriebene Projekt wurde ein wichtiger Schritt in Richtung Zertifizierung des Dokumentenservers an der TUB Hamburg-Harburg geleistet. Als Nebenprodukt entstand die wichtige Schnittstelle zu XMetaDiss, die von der Deutschen Bibliothek und den offiziellen OPUS-Entwicklern dankbar aufgenommen wurde. Auch andere Anregungen für mögliche OPUS-Erweiterungen konnten gelegt und sogar realisiert werden. Insofern kann das Projekt als Erfolg gewertet werden.

Eine Zertifizierung setzt bestimmte Dinge voraus, die nicht Teil dieses Projektes waren. Dies war von vornherein klar und es war auch vorgesehen, dass diese Dinge erst nach Ablauf des VFH-Projektes geklärt werden. Es sind keine technischen Aspekte. So überrascht es nicht, dass die lokale OPUS-Installation im jetzigen Stand noch nicht alle Zertifizierungsbedingungen erfüllt.

Bislang zu kurz kam der Bereich der rechtlichen Rahmenbedingungen und die Formulierung von Leitlinien den lokalen Server betreffend. Durch die Bereitstellung der FAQ-Datenbank sollte für derartige Fragen jedoch in einem gewissen Maße vorgesorgt sein und zumindest technisch die Möglichkeit gegeben sein, die rechtlichen Rahmenbedingungen nachzupflegen. Auch darüber hinaus wurden im Rahmen des Projektes (im technischen Bereich) einige Schritte unternommen, die weiterführen in Richtung der Zertifizierung.

Nach noch durchzuführenden letzten Funktionstests aller gemachten Änderungen und Ergänzungen sowie nach Prüfung auf Übernahme in den lokalen Produktivserver kann die Umstellung des Produktionsbetriebes erfolgen. Anschließend kann auch der Antrag auf Zertifizierung bei der DINI gestellt und eventuell weitere Schritte zur Bekanntmachung des Servers in der eigenen Hochschule unternommen werden. Das Dienstleistungsangebot der TUB hätte damit ein weiteres festes Glied und könnte eine stabile und abgesicherte Möglichkeit zur Online-Publikation anbieten und damit einen kleinen Beitrag zur weiteren Verbreitung des Online-Publizierens leisten.

6 Literaturliste

[1] Informationen für die Uniform Resource Name (URN)-Verwaltung und -Geschäftsverteilung für Online-Hochschulschriften

<http://www.persistent-identifier.de/?link=312>

[2] OPUS – Beschreibung und Grundsätze (abgerufen Oktober 2004)

<http://elib.uni-stuttgart.de/opus/doku/about.php>

[3] URN (NBN) Resolving

<http://www.persistent-identifier.de/?link=600>

[4] Annette Maile und Frank Scholze: Online Publikationsverbund der Universität Stuttgart (OPUS)

In: BI - Informationen für Nutzer des Rechenzentrums Nr.11/12, 1997, S. 21-24

<http://elib.uni-stuttgart.de/opus/volltexte/1999/226/pdf/226.pdf>

[5] DINI: Zertifikat Dokumenten- und Publikationsserver (Stand: November 2003, abgerufen Oktober 2004)

<http://www.dini.de/documents/Zertifikat.pdf>

[6] Maylein, Leonhard <Maylein@ub.uni-heidelberg.de>: [opus-I] Sicherheitsproblem in OPUS

in: persönliche eMail an opus-liste <opus-1@listserv.uni-stuttgart.de> vom 15.11.2004

[7] Präferenzregelung der Deutschen Bibliothek in Punkto Dokumentformate
<http://www.ddb.de/wir/pdf/praeferenzregelung.pdf>

[8] The Open Archives Initiative Protocol for Metadata Harvesting
<http://www.openarchives.org/OAI/openarchivesprotocol.html>

[9] XMetaDiss 1.2 Referenz
URN: urn:nbn:de:1111-2004072137
<http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:1111-2004072137>
http://www.ddb.de/standards/xmetadiss/ref_xmetadiss_v1-2.pdf

[10] Maile, Annette: Re: Projektbericht
Persönliche eMail vom 22.03.2005 von Annette Maile <maile@ub.uni-stuttgart.de>

Vollständige Mail:
Hallo Herr Marahrens,

vielen Dank fuer den Projektbericht. Hier ein paar
Anmerkungen/Hintergrundinformationen:

- S. 9 4.2 letzter Absatz
zum SWB existiert bereits eine Schnittstelle. Das BSZ holt sich ueber
eine spezielle Oai 1.1 Schnittstelle mit DC qualified die Daten ab und
importiert sie in den SWB. Vielleicht waere es auch sinnvoll, dass der
GBV sich die Daten der diversen Publikationsserver ueber Oai abholt. Mit
XMetaDiss hat man ja jetzt auch DC qualified. Man muesste XMetaDiss
wohll nur geringfuegig aendern, dass eben alle Datensaeetze ausgegeben
werden und nicht nur Diss. und Habil.

- S. 17 in der jetzigen Opus-Version wird fuer Diss. und Habil.
automtisch beim Freischalten eine Urn vergeben. In Opus 3 kann man
wahlweise fuer alle Dokumnte oder nur fuer Diss. und Habil. eine Urn
vergeben.
Meinen Sie im zweiten Absatz mit "automatische Vergabe von Urn", dass
diese schon vor dem Upload bereitgestellt wird? Das war mir in diesem
Absatz nicht so ganz klar.
Zu ihrem letzten Absatz: Selbst bei einem Server mit wenig Betrieb ist
Ihre Methode nicht so zuverlaessig. Ich denke, die Metadaten muessen
zuerst abgespeichert werden; dem Autor kann dann die verbindliche Urn
mitgeteilt werden. Eine Moeglichkeit waere vielleicht, dass der Autor
selbst die Dateien nachladen kann oder dass er sie eben dem Opus-Admin
irgendwie zukommen laesst und der laedt dann die Dateien nachtraeglich
hoch (so ist ja unser Vorgehen).

- S. 19 Der Vorname von Frau Seiler schreibt sich mit einem "n", also
"Anette".

Die md5-Sache ist ganz interessant. Koennen Sie mir die entsprechenden
Skripte schicken, die die Aenderungen enthalten? Was ist, wenn im
Admin-Bereich mit dem Dateimanager Dateien ausgetauscht werden muessen?
Werden dann die Pruefsummen neu berechnet?
Wir hatten urspruenglich daran gedacht, die md5 Summen erst beim
Freischalten einzutragen. Gruende sind das Austauschen der Dateien (ist
bei uns sehr oft der Fall) und dass die meisten User nicht in der Lage

sind bzw. es den meisten zu viel Arbeit bereitet, die Pruefsummen selbst zu berechnen. Wir muessen das noch ueberlegen.
Vielleicht koennte man Ihre Aenderungen auch ueber opus.conf steuern, so kann jeder Opus-Nutzer selbst entscheiden, ob er die Pruefsummen beim Upload zulaesst oder nicht.

Die Tests mit XMetaDiss werde ich wohl auf Anfang April verschieben. Ich habe naechste Woche Urlaub und danach werde ich mich darum kuemmern.

Frohe Ostern
Annette Maile

Oliver Marahrens wrote:

> Hallo Frau Maile,
>
> anbei wie versprochen der Projektbericht von meinen Opus-Aktionen.
> Über Feedback würde ich mich freuen, später werde ich diesen Bericht
> auch auf unserem Opus-System publizieren. Bislang haben wir die
> beschriebenen Änderungen noch nicht in den Produktivbetrieb
> umgesetzt, das wird in den folgenden Wochen hoffentlich passieren
> können. Wenn Sie also auch noch inhaltliche Anmerkungen haben so
> nehme ich die natürlich auch gerne entgegen!
>
> Viele Grüße
> Oliver Marahrens
>
> --
> Oliver Marahrens
> TU Hamburg-Harburg / Universitätsbibliothek / EDV / DISCUS
> Denickestr. 22
> 21071 Hamburg - Harburg
> Tel. +49 (0)40 / 428 78 - 32 91
> eMail o.marahrens@tu-harburg.de
>
>
>
> -----
>
> Der folgende Teil dieser Nachricht enthält einen Anhang im
> sogenannten Internet MIME Nachrichtenformat.
> Wenn Sie Pegasus Mail oder ein beliebiges anderes MIME-kompatibles
> Email-System verwenden, sollte Sie den Anhang mit Ihrem Email-System
> speichern oder anzeigen können. Anderenfalls fragen Sie Ihren Administrator.
>
> The following section of this message contains a file attachment
> prepared for transmission using the Internet MIME message format.
> If you are using Pegasus Mail, or any another MIME-compliant system,
> you should be able to save it or view it from within your mailer.
> If you cannot, please ask your system administrator for assistance.
>
> ---- Datei Information/File information -----
> Datei/File: projektbericht.pdf
> Datum/Date: 18 Mar 2005, 9:07
> Größe/Size: 1470388 bytes.
> Typ/Type: Unbekannt

--
Annette Maile Tel: ++49 (0)711 121-4189
Universitaetsbibliothek Stuttgart Fax: ++49 (0)711 121-3502

Holzgartenstr. 16, 70174 Stuttgart maile@ub.uni-stuttgart.de
<http://elib.uni-stuttgart.de/opus/>

7 Anlage1: Diagramme

