DEUTSCHE NATIONAL BIBLIOTHEK

Automatisiertes Abliefern über Harvesting-Verfahren

Wege zur effizienten Ablieferung von Netzpublikationen

Version 1.0
Stand: 12. August 2008

Redaktion: Jürgen Kett, Thomas Seidel

Deutsche Nationalbibliothek (Leipzig, Frankfurt am Main, Berlin)
2008

<urn:nbn:de:101-2008081507>

Inhaltsverzeichnis:

1. Einleitung

Um dem Sammelauftrag der DNB möglichst effizient gerecht werden zu können, sind Automatismen unverzichtbar. Dies gilt für alle Teilaufgaben vom Einsammeln, Archivieren, Erschließen, bis hin zur Bereitstellung von Publikationen und der Verbreitung der zugehörigen Metadaten. Der Schritt des Einsammelns hat hierbei eine besondere Komplexität: Um einen hohen Grad an Automatisierung zu erreichen, ist die DNB auf die enge Zusammenarbeit mit den Verlegern angewiesen.

Als Voraussetzung muss zwischen beiden Parteien, also DNB und Verleger, eine feste Vereinbarung bestehen, auf welche Weise neue Veröffentlichungen automatisch erkannt und abgeholt werden können. Dies erfolgt über ein technisches Protokoll, das den genauen Ablauf bis ins Detail beschreibt und das beide Seiten implementieren müssen – ein *Harvesting-Protokoll*.

Die DNB muss ein Harvesting-Verfahren mit sehr vielen Partnern durchführen. Im Idealfall nutzen alle Partner bis ins kleinste Detail hinein exakt dasselbe Protokoll. Dann müsste die DNB lediglich ein einziges Protokoll implementieren. Das ist unrealistisch: Allein die Vielfalt der Publikationsarten (Monografien, Serials, Web-Seiten, Musik, etc.) erfordert unterschiedliche Metadaten und Objektformate. Auf der anderen Seite sollten die Protokolle möglichst homogen sein, um den Aufwand für Abstimmung, Pflege, Entwicklung und Dokumentation möglichst gering zu halten.

Wiederverwendbarkeit ist hier das Paradigma. Vollständig proprietäre Protokolle, die auf Einzelabsprachen basieren, bilden hierzu keine Grundlage. Es wird beispielsweise nicht möglich sein, jedes beliebige Metadatenformat zu verarbeiten. Deshalb setzt die DNB, wo immer es möglich ist, auf von ihren Partnern akzeptierte Standards wie beispielsweise ONIX oder MARCXML. In solchen Fällen ist es meist leicht, Einigkeit zu erzielen. Leider gibt es viele Bereiche, in denen sich noch kein Standard etabliert hat. Insbesondere dort sind wir auf die Mitarbeit unserer Partner angewiesen, nachhaltige und für alle Parteien strategisch günstige Lösungen zu finden.

2. Welche Voraussetzungen muss ein Harvesting-Protokoll erfüllen?

2.1 Allgemeine Anforderungen

Protokolle sind Regeln, welche das **Format**, den **Inhalt**, die **Bedeutung** und die **Reihenfolge** gesendeter Nachrichten zwischen verschiedenen Instanzen festlegen.[1] In unserem Falle haben wir es mit zwei Instanzen zu tun: dem Client, der Daten abholen möchte, und dem Server, der die Daten zur Verfügung stellt.

Damit die Kommunikation wirklich vollautomatisch laufen kann, muss das Protokoll bis ins kleinste Detail spezifiziert sein. Dies beinhaltet auch die Behandlung von Fehlersituationen.

[1] http://de.wikipedia.org/wiki/Protokoll#Protokolle_in_der_Telekommunikation_und_Informatik

Wo immer möglich, empfiehlt sich der Einsatz von existierenden Standards oder veröffentlichten Quasi-Standards. Das erhöht die Interoperabilität und erleichtert die Implementierung und Pflege. Gerade für Client-Server-Kommunikationen gibt es durch das Internet eine Fülle von Standards und Design-Patterns auf denen ein Harvesting-Protokoll aufbauen sollte wie beispielsweise HTTP, FTP, XML sowie REST- und SOAP-Webservices. Gleiches gilt natürlich auch für Metadatenformate.

2.2 Kernfunktionalität "Synchronisation mit dem Server"

Die Synchronisation zwischen dem Client (DNB) und dem Server (Verleger) hat das Ziel, regelmäßig neue Publikationen samt Metadaten abzuholen. *Neu* meint hierbei: „neu für den Abholenden“. Im Idealfall ist also die Rückgabe des Servers abhängig vom aktuellen Stand des Clients. Das heißt, der Client bekommt alles geliefert, was er bislang noch nicht abgeholt hat. Tritt der Client also beispielsweise zum ersten Mal an den Server heran, müsste er alles bekommen, was dieser Verleger bislang publiziert hat, auch wenn darunter schon zehn Jahre alte Publikationen wären.

Ein solches Synchronisationsprotokoll könnte also wie folgt ablaufen:

```
Client -> Server:
   LIEFERE METADATEN VON 2008-03-01 BIS 2008-03-17
Server -> Client:
   METADATENSATZ 1: xxxxxxx, ERSTELLT 2008-03-01
   METADATENSATZ 2: yyyyyyy, ERSTELLT 2008-03-01
   METADATENSATZ 3: zzzzzzz, ERSTELLT 2008-03-02
   .....
   METADATENSATZ n: qqqqqqq, ERSTELLT 2008-03-17
   ENDE DER LISTE
```

Bei der nächsten Anfrage würde der Client nach Metadaten seit dem 18.03.2008 fragen. Die Zeitintervalle gibt der Client vor, und damit besteht ein einfacher Weg, auch weiter zurückliegende Metadaten abzurufen.

2.2.1 Aufbau der Listenelemente

Grob betrachtet sähe der Ablauf des Protokolls also wie folgt aus:
Der Client schickt eine Anfrage. Der Server antwortet mit einer Liste. Diese Liste enthält Elemente, die die Publikationen repräsentieren. Wie aber müssen die Elemente dieser Liste genau aussehen?
Die DNB benötigt für jede neue Publikation einen bibliografischen Metadatensatz und den automatischen Zugriff auf die elektronische Publikation selbst, um diese abholen und archivieren zu können.

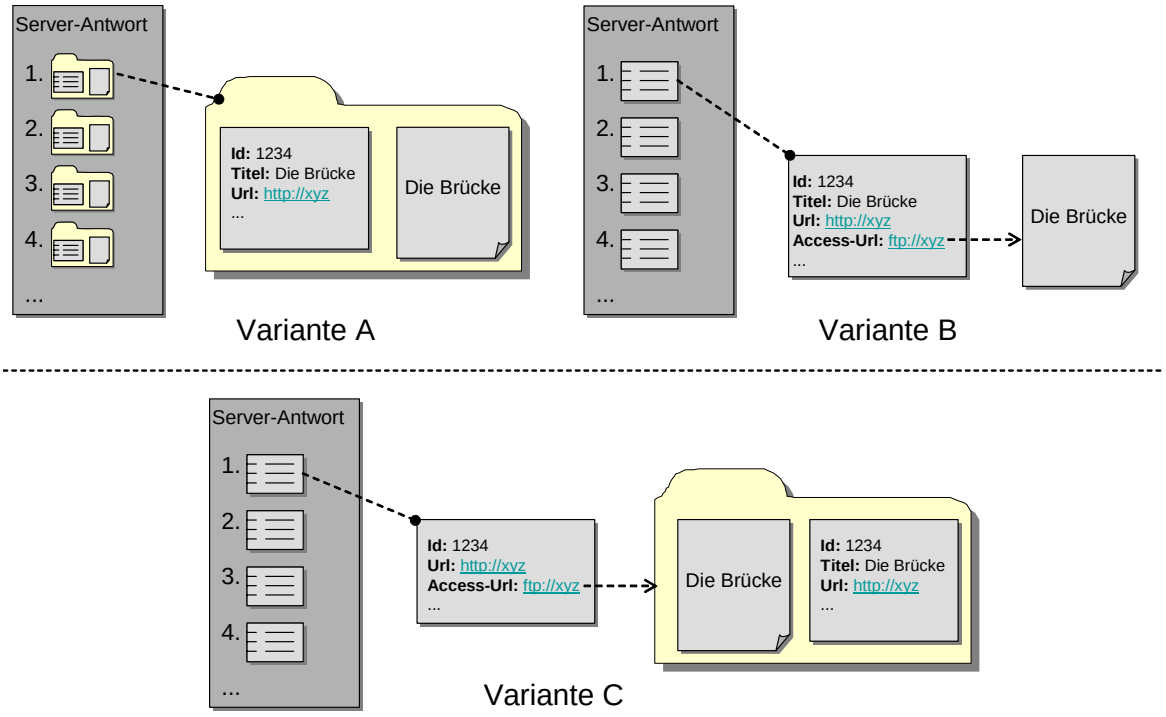

Abbildung 1: Wie Metadaten und Publikation angeboten werden könnten

Es gibt drei Varianten, wie die Elemente der Liste aufgebaut sein können (siehe Abbildung 1).
In der **Variante A** verweist die Liste auf Pakete, die jeweils die Metadaten und die dazugehörige Publikation enthält. Der Nachteil dieser Variante besteht darin, dass die Liste insgesamt sehr groß würde, da sie die Publikationen bereits beinhaltet.
Für das Ablieferungsverfahren der DNB ist die **Variante B** vorgesehen: Die Liste enthält zunächst nur die Metadaten, die die Publikation beschreiben. Innerhalb dieser Metadaten gibt es ein Element (z.B. eine Zugriffs-URL), das auf das Paket mit der Publikation verweist. Dadurch würde die Gesamtgröße der Liste deutlich kleiner. Der Client könnte sich also effizient darüber informieren, was es Neues zur Abholung gibt, bevor er mit der eigentlichen Verarbeitung beginnt. Auch die Aktualisierung von Medadaten wäre damit einfacher. Dem Server könnte eine überarbeitete Version der Metadaten zu einer Publikation, die bereits abgeholt wurde, zur Verfügung stehen. Dem Client wäre es jetzt möglich, diese Korrektur zu übernehmen ohne gleichzeitig auch die Publikation ein zweites Mal abzuholen.
Die **Variante C** sei nur erwähnt, weil sie mitunter vorkommt. In Abweichung zu Variante B, könnte eine Zugriffs-URL nicht nur auf die Publikation, sondern auf ein Paket bestehend aus Publikation **und** Metadaten verweisen. Die im Paket gespeicherten Metadaten sind bei einer solchen Variante meist deutlich umfangreicher als jene in der Server-Antwort.

Im Rahmen des Protokolls zwischen Server und Client muss für die Variante B eine feste Vereinbarung zur Paketstruktur, zum verwendeten Metadatenstandard (z.B. ONIX) und zu Aufbau und Inhalt der Metadaten getroffen werden. Welche bibliografischen

Metadaten gelieferten werden sollten, wurde in einem so genannten Metadaten-Kernset dokumentiert.[2]

2.2.2 Umfang der Liste

Auch was den Umfang der zurückgegebenen Liste anbelangt, gibt es verschiedene Varianten.

- Variante 1: Der Server pflegt eine komplette Liste aller seiner Publikationen, die er permanent aktualisiert, und bietet bei Anfrage stets die aktuellste Version an. Dem Client bleibt überlassen, die für ihn neuen Publikationen herauszusuchen. Vorteil: Sehr einfach. Nachteil: Nicht effizient. Es werden immer alle Veröffentlichungen angeboten, obwohl nur ein Bruchteil davon neu ist.

- Variante 2: Der Server beginnt nach einer festen Regel (z.B. täglich, wöchentlich oder monatlich) eine neue Liste. Alte Listen werden nicht gelöscht. Dem Client ist es bekannt wie er auf eine spezielle Liste (beispielsweise alle Veröffentlichungen der 10. KW 2007) zugreifen kann. Dem Client bleibt es überlassen, aus den Listen die für ihn neuen Publikationen herauszusuchen. Vorteil: Deutlich effizienter als Variante 1, da weniger redundante Information ausgetauscht wird.

- Variante 3: Der Server erstellt die Listen dynamisch in Abhängigkeit von den Wünschen des Clients. Der Server würde also auf Fragen antworten können, wie: "Gib mir alles was vom 01.01.2008 bis zum 10.01.2008 hinzukommen ist". Das Harvesting-Protokoll OAI ist eine Realisierung dieser Variante. Vorteil: Flexibler und effizienter als Variante 2.

Ein Punkt ist allen Lösungen gemein: Dem Client ist es jederzeit möglich, auch ältere Publikationen abzuholen. Alle von der ersten bis zur letzten Publikation sind über dieses Verfahren jederzeit abrufbar. Dies stellt eine starke Vereinfachung dar, da es in vielen Fällen dem Server nicht möglich sein wird, Publikationen beliebig lange vorzuhalten. Dennoch ist es eine wichtige Anforderung, dass es über einen gewissen Zeitraum (ein halbes Jahr ist hier eine gute Marke) möglich sein muss, Publikationen und Metadaten rückwirkend einzusammeln. Das kann insbesondere bei der Analyse und Behebung von Fehlerfällen notwendig werden.

3 Das Harvesting-Protokoll OAI

3.1 Überblick

Die DNB setzt beim Einsammeln zurzeit bevorzugt auf das Protokoll OAI. Das hat folgende Gründe:

- Das Protokoll bildet genau die Funktionalitäten ab, die für das Harvesting benötigt werden.

- Es ist ein offener Standard.

[2] Lieferung von Metadaten für Netzpublikationen an die Deutsche Nationalbibliothek http://www.d-nb.de/netzpub/info/pdf/metadaten_kernset_extern.pdf

- Es wird von vielen Informationssystemen (OPUS, DSpace, Fedora, etc.) und Dienstleistern (z.B.: OCLC und verschiedene Archive) unterstützt

- Es ist im universitären Bereich sehr verbreitet und wird deshalb bereits für das Abholen von Dissertationen genutzt.

- Es ist ein recht einfaches Protokoll und es gibt eine Menge offene Werkzeuge und Implementierungen dazu.

- Die DNB hat Erfahrungen mit beiden Seiten, also der Server- und der Client-Seite des Protokolls, und könnte daher Support und Beratungsleistung erbringen. Die DNB betreibt beispielsweise ein OAI-Repository, an dem sich Partner zeitnah mit unserem Katalog synchronisieren können.

OAI steht für *Open Archives Initiative*. Die Initiative hatte zum Ziel, eine offene Schnittstelle[3] zu definieren, die zum Austausch von XML-Metadaten geeignet ist.
Herausgekommen ist dabei folgendes Modell:

Die Kommunikation läuft zwischen zwei Partnern, dem *Data Provider* und dem *Service Provider*.

- **Data Provider (Server):** Datenanbieter mit einem *Repository* von Metadaten
- **Service Provider (Client):** Sammelt Daten von *Data Providern* ein und nutzt hierzu einen sog. *Harvester*.

Das Protokoll, das für diese Kommunikation verwendet wird, hat den Namen OAI-PMH 2.0 (*OAI-Protocol for Metadata Harvesting 2.0*).

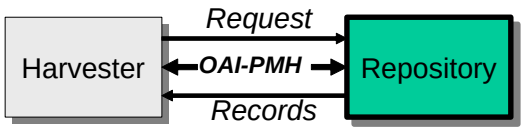

Abb. 1: Kommunikation über OAI

3.2 Hilfreiche Dokumente

Am Ende dieses Papiers wurden einige Referenzen beigefügt. Diese Dokumente beschreiben alles, was man wissen muss, um mit OAI arbeiten zu können. Die OAI-Homepage [1] sei hierbei besonders empfohlen. Hier finden sich auch Beispielimplementierungen für OAI-Harvester.

3.3 Das Protokoll OAI-PMH

Das Protokoll OAI-PMH ist web-basiert: Der Harvester arbeitet mit einfachen Anfragebefehlen per HTTP-GET oder POST und bekommt http-Antworten zurück. Diese Antworten enthalten eingebettet in eine XML-Struktur die angeforderten Daten.

[3] Offen im Sinne einer offenen Dokumentation, nicht im Sinne eines freien Zugriffs

Der Harvester ist daher sehr leicht zu implementieren. Ein weiterer Vorteil ist, dass ein OAI-Repository mit einem einfachen Web-Browser abgefragt werden kann.

3.3.1 Befehlssatz

Das Protokoll OAI-PMH enthält lediglich sechs Befehle:

- Identify
- ListMetadataFormats
- ListSets
- ListRecords
- GetRecord
- ListIdentifiers

Die ersten drei Befehle beantwortet das Repository mit statischen Antworten. Diese enthalten Informationen über die Einstellungen und Fähigkeiten des Repository (z.B. Was enthält das Repository, welche Datenformate werden unterstützt, zwischen welchen Datensets[4] kann man wählen?)

Beispiele:

1. Allgemeine Informationen zum Repository: http://arXiv.org/oai2?verb=Identify
2. Auflisten aller zur Verfügung stehenden Sets: http://arXiv.org/oai2?verb=ListSets.
3. Auflisten aller zur Verfügung stehenden Formate: http://arXiv.org/oai2?verb=ListMetadataFormats

Anmerkungen:

- http://arXiv.org/oai2 ist die URL des Repository.
- Mit „verb=“ wird der jeweilige Befehl eingeleitet

Die übrigen drei Befehle liefern die eigentlichen Metadaten. Auf diese wird im Folgenden näher eingegangen.

3.3.2 ListRecords

Funktion: Harvesting von Datensätzen per Angabe des Zeitraums (from/until) und/oder Sets. Dies ist der Kernbefehl von OAI. Er ermöglicht *selektives Harvesting*, d.h. der Harvester kann seine Anfrage auf Datensätze beschränken, die

1. aus einer bestimmten Menge stammen und
2. in einem bestimmten Zeitraum erzeugt oder geändert wurden. Dies entspricht Variante 3 aus Kapitel 2.2.2.

[4] Ein Set definiert einen beliebigen Ausschnitt aus der Menge der angebotenen Daten.

Parameter:[5]

- *from / until:* Zeitpunkte, die den Zeitraum für das selektive Harvesting definieren. Diese können (je nach Repository) entweder Tagesgenauigkeit (YYYY-MM-DD) oder Sekundengenauigkeit haben (YYYY-MM-DDThh:mm:ssZ).
- *set:* Die Menge, aus der die Datensätze stammen sollen (z.B. ein Set, in dem nur jene Bestandsdatensätze enthalten sind, die einer Bibliothek aus dem BVB zugehörig sind)
- *metadataPrefix:* s.o.

Beispiel:
http://arXiv.org/oai2?verb=ListRecords&from=2003-09-15&set=cs&metadataPrefix=oai_dc

Die Datensätze, die hier zurückgegeben werden, müssten nun natürlich noch wie Abschnitt 2.2.1 (Variante B) beschrieben, ein Element enthalten, das auf die eigentliche Publikation verweist (Zugriff-URL). Der Client könnte dann Datensatz für Datensatz abarbeiten und bei Bedarf die zugehörige Publikation vom Server abholen.

3.3.3 GetRecord

Funktion: Abfrage eines einzelnen Records per ID-Angabe. Dieser Befehl setzt voraus, dass man die ID des gewünschten Records kennt.
Parameter:

- *identifier:* Die Identifikationsnummer des gewünschten Datensatzes
- *metadataPrefix:* Der Code für das Datenformat, in dem der Datensatz geliefert werden soll. Die zur Auswahl stehenden Codes können über den Befehl „ListMetadataFormats“ (s.o.) abgefragt werden.

Beispiel:
http://arXiv.org/oai2?verb=GetRecord&identifier=oai:arXiv.org:cs/0112017&metadataPrefix=oai_dc

- „oai:arXiv.org:cs/0112017“ ist die ID des gewünschten Datensatzes
- oai_dc ist bei diesem Repository der Code für „DC Simple“

3.3.4 ListIdentifers

Funktion: Variante von ListRecords, bei der nur die „Header“ (ID, Set-Zugehörigkeit und Änderungsdatum) der Datensätze angezeigt werden.

Beispiel:
http://arXiv.org/oai2?verb=ListIdentifiers&from=2003-09-15&set=cs&metadataPrefix=oai_dc

[5] Der Parameter resumptionToken wurde weggelassen, weil er für einen ersten Überblick unwichtig ist. Genauere Details finden sich in [2].

3.4 Implementierungen

Implementierungen zu OAI gibt es reichlich. Manche davon sind frei, manche sind kostenpflichtig. Eine (nicht vollständige) Liste von Implementierungen und Tools findet sich unter http://www.openarchives.org/pmh/tools/tools.php. Eine sehr einfache Variante, ein OAI-Repository zu realisieren, bietet die Software OaiCat von OCLC[6]: Sie bietet unter anderem die Möglichkeit, ein vollständiges OAI-Repository auf Basis eines herkömmlichen File-Systems oder einer herkömmlichen SQL-Datenbank (via JDBC) aufzubauen. [3]

4 Sicherheit

Die DNB unterstützt marktübliche Verfahren, um die Kommunikation zwischen der DNB und den abliefernden Stellen sicher zu gestalten und vor Missbrauch zu schützen. Dies betrifft insbesondere die Punkte:

- Ausspähen der übertragenen Informationen
- Identifikation des Harvesting-Clients als DNB-Client
- Sicherheit der Zugriff-URLs
- Speicherung der Metadaten und der Zugriff-URLs

In den folgenden Abschnitten werden diese Fälle näher erläutert.

4.1 Ausspähen der übertragenen Informationen

Das OAI-Protokoll basiert auf HTTP, d.h. sämtliche Sicherheitsmechanismen, die das HTTP unterstützt, können zu Absicherung der übertragenen OAI-Informationen gegen Ausspähen dienen. Die Implementierung des OAI-Clients der DNB unterstützt die ungesicherte Kommunikation per HTTP und die gesicherte Variante per HTTPS.

4.2 Identifikation des Harvesting-Clients als DNB-Client

Der Harvesting-Client der DNB ist sowohl für die Übertragung der Metadaten per OAI als auch für die Übertragung der Publikation per Zugriff-URL jeweils über eine feste IP-Adresse identifizierbar. Eine abliefernde Stelle kann daher sowohl den Zugriff auf das OAI-Repository als auch auf die Publikationen per Zugriff-URL basierend auf der IP-Adresse absichern.

4.3 Sicherheit der Zugriff-URLs

Die Zugriff-URL dient ausschließlich zur Übertragung der Publikation an die DNB. Diese Übertragung wird durch die DNB initiiert. Nach der Übertragung, die zeitnah nach dem Erhalt der Metadaten erfolgt, wird die Zugriff-URL nicht mehr benötigt.

Ein unbefugtes Nutzen der Zugriff-URL durch Dritte kann in Zusammenarbeit der abliefernden Stelle mit der DNB durch die folgenden Verfahren unterbunden werden:

- Der Zugriff auf die Publikation erfolgt per SSL (HTTPS)

[6] Folgender Link führt direkt zur Download-Seite http://pubserv.oclc.org/oaicat/jars/dist/dist.html

- Die abliefernde Stelle beschränkt den Zugriff auf die Publikation auf die feste IP-Adresse des Harvesting-Clients der DNB
- Der Aufbau der Zugriff-URL ist nicht aus den restlichen Informationen der Metadaten der Publikation ableitbar.

4.4 Speicherung der Metadaten und der Zugriff-URLs

Die abgelieferten Metadaten werden nur innerhalb des entsprechenden Kontexts verarbeitet und gespeichert. Insbesondere die Zugriff-URL wird ausschließlich für den Transfer der Publikation verwendet, und niemals an Dritte weitergegeben.

5 Fazit

Die OAI-Spezifikation klärt einen großen Teil der Protokolldetails, die im vorletzten Kapitel vorgestellt wurden. Es definiert bereits genau, wie die Anfragen des Clients und wie die dazu passenden Antworten des Servers aussehen müssen. Ebenso wird die Kommunikation von Fehlerfällen festgelegt.
Natürlich sind damit noch immer nicht alle notwendigen Absprachen zwischen Client und Server getroffen: Man muss sich noch mit dem Partner auf ein Metadatenformat (z.B.: ONIX) und Pflichtangaben einigen, darüber verständigen, wie die Zugriff-URLs aussehen und auf was diese verweisen: auf die Publikation, oder auf ein Paket, das darüber hinaus noch mehr Daten enthält (zusätzliche Metadaten, Abstracts, etc).[4] Im letzteren Fall müssen auch noch Festlegungen zur Paketstruktur getroffen werden. Dies geschieht in individuellen Vereinbarungen.

Es bietet aber einen bereits deutlich konkreteren Rahmen, als einfache Basisnetzwerkprotokolle wie HTTP oder FTP und bietet wegen der guten Verbreitung (außerhalb des Verlagswesens) Perspektiven für weitere Zusammenarbeit. Beispielsweise könnte ein künftiges Modell vorsehen, dass sich Verlage zeitnah die von der DNB angereicherten Datensätze via OAI wieder zurück ins eigene System holen.

Referenzen

[1] OAI Homepage. http://www.openarchives.org
FAQ über OAI. Internet: http://www.openarchives.org/documents/FAQ.html

[2] OAI-PMH-Spezifikation. Internet:
http://www.openarchives.org/OAI/openarchivesprotocol.html

[3] Implementation Guidelines. Internet:
http://www.openarchives.org/OAI/2.0/guidelines.htm

[4] Information zum Thema Ablieferung Netzpublikationen an die Deutsche Nationalbibliothek im Internet:
http://www.d-nb.de/netzpub/index.htm

Ansprechpartner:

Technisch:
Jürgen Kett (Abteilung Informationstechnik)
j.kett@d-nb.de

OAI-Service
E-Mail-Adresse allgemein:
oai-service@d-nb.de

Erwerbung Netzpublikationen:
E-Mail-Adresse allgemein:
np-info@d-nb.de

Telefon-Nummern:
Leipzig +49-341-2271-282
Frankfurt +49-69-1525-1320