Implementing OAI Data and Service Providers

Our institute has implemented (on the basis of sources by the HU Berlin) an OAI data provider for the collections of PhysDoc, a heterogenous repository of physics documents around the world. I will talk about specific problems of heterogenous collections and present our way to cope with these.

We are currently developing a subject specific (physics) OAI Service Provider, which will not only collect data from other data providers but will search through other metadata collections which are not otherwise publically available. There I will present this service provider and explain our way implement this service.
OAI for Archives without Structured Data

Heinrich Stamerjohanns
Institute for Science Networking
at the
University Oldenburg

Overview

• Without databases it will not work...
• Short introduction to PhysDoc
• Harvest Gatherer
• PhysDoc as OAI Data-Provider
PhysDoc

- PhysDoc itself is a distributed document database, which is in service since 1995
- Aims to authors, small institutions and small departments and other small institutions
- Harvest Gatherer collects documents from physics servers europe-wide
- 40000 documents

PhysDoc together with Harvest

- Documents
- Documents
- Documents
- Gatherer
- Summarizer
- Filesystem with SOIF Records
- WWW-Client
- Search Interface
Dublin Core Extension

- Extension of the summarizer to Dublin Core
- HTML-pages describe PDF or PostScript documents
  
  `<META NAME="DC.Title" CONTENT="OAI Talk">`
  `<META NAME="DC.Author"
      CONTENT="H. Stamerjohanns">`

- DC is embedded in the SOIF format
- search interface is extended accordingly

PhysDoc und OAI

- PhysDoc was supposed to deliver data in an OAI-compliant form
- principal possibility to directly use SOIF-Records, and to create by mapping

  ![SOIF to Dublin Core](image)

  OAI-compliant responses
  - not a useful solution
  - bad control of the mapping
Metadata Container

- different approach
- definition of metadata container
- preferably extensive description of the elements (here documents)
- contains Dublin Core elements
- can be easily extended to local requirements
Metadata Converter

- Data is converted offline
- Normalization
- use DC if possible, otherwise SOIF
  - different representations of single metadata elements in one common format
  - DC.language "GER" → "de"
  - DC.date "1.02.1999" → "1999-02-01"
- simple quality function
- better possibility to check consistency, if data is available in a structured form

PhysDoc together with OAI

```plaintext
documents documents

Gatherer

Summarizer

Filesystem with SOIF Records

offline

Metadata container as SQL Database

Mapper

XML on-the-fly

OAI-Gateway

DC, MARC

Mapper

Quality function

Normalizer
```
PhysDoc together with ???

- Use of metadata container yields many advantages
- consistency check of data
- quality assurance
- static HTML export
  - any desired export format besides DC/OAI possible

→ is prepared for any other exchange protocols than OAI

Summary

- Authors deposit their pages on their own WWW-server ("self-archiving")
- may enrich these (hopefully) with DC
- Harvest collects metadata
- Metadata are normalized offline and stored in SQL-database in metadata container
  - local development in PHP
  - mySQL database, check XML-database
  - OAI-Gateway queries database and delivers OAI-compliant output (XML on-the-fly)
  - modified version of PHP-scripts of HU Berlin
PhysDoc Service-Provider

- short clarification of terms
- OAI is not a protocol for the end user
- OAI **data-provider** runs web server, which provides its metadata by OAI-protocol
- OAI **service-provider** queries by OAI-protocol other data-provider and uses the collected metadata to provide extended services (e.g. a query)

PhysDoc as Service-Provider

- PhysDoc wants to provide OAI Services to the physics community
- includes freely accessible documents (e.g. arXiv) as well as metadata (and only metadata) of commercial providers
- local development in PHP
- successful test on own data-provider
- as well as the OAI-interface of arXiv
- other providers are included with proprietary interfaces (IOP)
Technical Details

- local development
- all written in PHP4
- scheduler is based on database
- expat library is used as XML-Parser for OAI and proprietary interfaces
- database is again mySQL
  - with “tricks”
  - full text extensions
- XML database should be checked
Technical Details

- successful implementation by testing on the local data-provider
- Added another data-provider within five minutes
- but yet problems
  - vagueness in protocol definition
  - 503 flow control...
  - bad choice, because it depends on layout
- normalization is again necessary (might raise further technical, textual and legal problems)

Thank You

- OAI at the Institute for Science Networking, Oldenburg:
  http://physnet.uni-oldenburg.de/oai/oai.php
- stamer@uni-oldenburg.de