

Harvesting and application profiles

Report from workshop #1, ELAG 2003 in Bern

Theo van Veen

Participants :

Peter van Boheemen, Netherlands
Ron Davies, ILO
Nanna Hakala, Finland
Harry Janssen
Pierre Clavel, SNL, Switzerland
Hansueli Locher
Petra Otten, NL
Tamar Sadeh, Israel
Susanna Peruginelli, Italy
James Tallon, Switzerland
Graham Tritt, BIT, Switzerland
Bojana Lesnik, Slovenia
Theo van Veen

Below one finds a description of the workshop in chronological order in terms of subjects and questions that were taken into consideration at the workshop on harvesting and metadata profiles.

At the start of the workshop the following terms were explained :

DCMI –Dublin Core, simple, qualified, refinements, application profiles
OAI-PMH, data provider maps own metadata to DC
Distributed versus centralized databases / hybrid combinations
Namespaces and element sets
Application Profiles and Registries

Questions with respect to searching in distributed databases:

How do we deal with a search for fields that are not present in all individual databases? Let the servers decide on the best choice or generate an error message? How should searching in a central index deal with that?
It was decided that the user has control, let him decide. Another principle is: don't fool the user; let him know the differences between the different collections. Let him choose between simple search or advanced search. And finally offer a link to the native interface.

Metadata problems

A source does not have all metadata fields. In case a field is not present the search returns nothing from this source, or ignores it?

The relevant information on this should be made available to the user, e.g.:

- Search found x results
- No "subject" field in 2 or 10 databases searched
- The user should have options on the results screen

Several result sets are possible as a result of a search:

On the main screen in show several parts e.g.:

- Exact search painter=John Smith returned 100 results
- Search for creator / author returned 2000 results
- Searching subject returned 30000 results
- Searchin full text returned 400000 results

By means of these results the user can make a new query or enter an advanced search.

Hybrid solutions

Would a hybrid solution of “distributed (subject or geographical oriented) central indexes” be a useful alternative for pure distributed searching or pure central indexing?

The solution will be a mix depending on:

- the accessibility
- the number of targets
- the similarity of collection metadata

Distributed searching leads to problems in the area of ranking and de-duplication.

Principles

- Index rich data in as high quality as possible
- Don't dumb down the data – make it available in full quality
- Allow dumbing down at the user interface
- Provide data on the collection to the harvester or search engine

Collection Descriptions / “EXPLAIN” functionality

The question is where the information is held for the search and how is it returned to the user. If data collections are unequal we should have this information available. So we need to know what metadata each collection uses –e.g. creator = painter, composer, ... A search for painter=“John Smith” should revolve to a search for creator if the roles have not been defined. The info from the source must be in the index.

How do we find the right targets for harvesting and or searching?

Do we need more sophisticated mechanisms of automated use of databases with collection level descriptions? OAI is not sufficient. We should be able to find collections in possible the following ways:

- Search on google)
- Broadcast a request to the world
- Notify friends and librarian
- Register with a broker

A central register is not necessary.

The search machine

It should be able to use the full quality of the available data and it must be able to handle searches in mixed collections of varied quality e.g. if a field is not present in one database, then the metadata should know this and the search engine should take account of it.

Different Terms

- Users should not be misled by terms, which are not in all collections
- Users do not have the knowledge of the catalogue
- Users who have a detailed query should have direct access to the collection e.g. if a user gets few hits, for those collections which support composer or creator the user should get the opportunity to go to relevant collections directly
- It may well be the users choice to accept being misled!

Distributed portals

In distributed searches we need to be able to integrate descriptions of both traditional catalog objects and searchable collections This needs agreements.

Semantic web approach: “who are you”, then use the answer to define further work. There are multiple approaches.

Definition of base URL: How does the user know what he can search on? The Explain function is implemented with the only information needed = the base URL plusIs it useful? Yes. Is it possible? Technically yes. Is it feasible ? Maybe ...

Questions

What purpose does Dublin Core in the bibliographic world serve and why should we use it?
What do we need to make Dublin Core better usable for exchanging and sharing bibliographic records?
How do we find the balance between introducing new terms and keeping aligned with other applications and application areas.

Increasing usage levels for metadata

The following levels for metadata are recognised:

- DC simple
- DC qualified
- Application profiles

Users may link to the source system if they need more terms. Refine existing elements when possible.

Considerations

- DC was designed as a simple pidgin language and not as a complete set of terms for all applications.
- DC is for sharing a common set. DC is the right starting point! But...the world is developing
- We need a mechanism to add new terms and share evolving metadata definitions.
- Opposing forces: freedom and flexibility versus standardization
- Avoid many different application profiles for almost overlapping profiles

Recommendations

- Use DC or DC qualified as a core.
- Use DCMI element sets when possible.
- Use application profiles when needed but we need a different way of thinking!
- Introduce Dublin Core extended:
 - o Allow for unknown elements above DC qualified
 - o Let applications use the terms they understand and neglect the terms they don't understand
 - o Encourage the sharing of these added terms by submitting them to a central registry