S:t Laurentius Digital Manuscript Library: An excursion along the border between resource discovery and resource description

Sigfrid Lundberg (sigfrid.lundberg@lub.lu.se)

NetLab, Lund university Libraries
PO Box 134
SE-221 00 Lund
Sweden

ABSTRACT

This note has three aims. First it discusses the experiences gained by the development of a specific service, which, using a collection of detailed XML descriptions, provides its users access to a collection of digitized medieval manuscripts. Secondly, it discusses the database used from the point of view of both retrieval of the intellectual content (the texts) of these manuscripts and of the retrieval of the manuscripts as unique entities. Finally, it explores the possibilities fo searching a collection of complex XML documents using a full text retrieval engine used together with the Z39.50 information retrieval protocol.

1. Introduction

Medieval manuscripts are complicated. Each manuscript is a unique entity with its own history. Often, not always, manuscripts contain several pieces of intellectual content, each of which may be well known in the sense that this content appear in many contemporary manuscripts, and this content may appear in printed editions today.

To catalogue medieval manuscripts requires a complex descriptive metadata schema, which is capable of capturing not only the individual manuscript as a work of art but also as a unique blend of well known intellectual content.

That is, you should be able to search for works by (say) Boethius or Virgil. But you should also be able to search for manuscripts that have been owned by for example the monastic society of the Lund cathedral, or manuscripts that were illuminated in Italy.

The S:t Laurentius Digital Manuscript Library[1] is an attempt to combine these two aspects (Figure 1). A collection of medieval manuscripts at Lund university library is being digitized, and cataloged using the Master XML DTD[2] developed by the Master project. This note describes only internals of the search system of the service, and does so from an entirely conceptual point of view.

The complexity of a Master document is close to or surpasses what one expects in

[1] Nylander, Eva; Borell, Mattias and Lundberg, Sigfrid: *S:t Laurentius Digital Manuscript Library* , Lund University Libraries2002. <http://laurentius.lub.lu.se>.

[2] MASTER Work Group: *Reference Manual for the MASTER Document Type Definition* , Lou Burnard (ed.) Text Encoding Initiative2001. <http://www.tei-c.org/Master/Reference/>.

[3] *DocBook Technical Committee,* <http://www.oasis-open.org/committees/docbook/>.

XML documents in general, for example DocBook[3] or TEI[4] documents of similar length. Conceptually, the problem of retrieving information on a page range in a medieval manuscript described using Master DTD is basically the same as retrieving a section or paragraph in e-texts in general.

Although the scope of Laurentius is information retrieval in a very specialized context, I take a more general approach. The solution to our specific problem applies to a general class of information and text retrieval problems.

This note has the following structure. First, I discuss generalities, namely the principles for transformation of XML documents into a format that simplifies information and text retrieval. The approach used builds upon an extension of the thinking and procedures we earlier applied to the indexing of the web. I thereafter I explore how Z39.50[5] can be employed to retrieve simplified documents. I then go on to discuss methods for fielded search in this environment. Finally, I return to a more general question, if there is a need for a standardized text retrieval syntax, as opposed to text encoding syntaxes.

2. Automatic indexing[6] : The fitting of a document to a database

2.1. Documents and data

The general problem with searching tagged texts is that they are documents, not database records. That is, they are not tagged versions of already structured data. Bourret[7] makes a distinction between document-centric and data-centric XML documents. The latter are usually constructed with a specific data-model in mind and may be created from or digested into a relational database with the aid of an XML parser and only a few lines of code in your favourite scripting language. Documents, however, differ from simple data. Tags may nest into an arbitrary depth, and may also, in spite of the mark-up language's content model, appear in haphazard combinations. Furthermore, the manner in which tags nest and the way they are combined may have an important bearing on the semantics of the tags. In the following I will refer to these two kinds of tagged text as Document Like Objects (DLOs) and Record Like Objects (RLOs)

There are a number of different solutions to the problem of searching XML objects (DLOs as well as RLOs). One such solution might be to use a native XML database solution.[8] Current implementations of such databases are, in general, very good for XML data (where relational databases already excel). Some of the existing solutions are building on a persistent form of the Document Object Model (DOM), and at least some of them have problems with scalability, since they keep the entire database in RAM memory. Undoubtedly, these are problems with native XML databases will

[4] *The TEI Guidelines*, <http://www.tei-c.org/Guidelines2/index.html>.

[5] *Z39.50 Maintenance Agency Page*, Library of Congress Network Development and MARC Standards Office <http://lcweb.loc.gov/z3950/agency/>.

[6] There is a problem of vocabulary connected to the term ”indexing”. In a bibliographic context indexing means the assigning of keywords form a controlled list of terms to a document, whereas in the context of full text searching, the term refers to building a computerized ”table of contents”, an index, to a collection of documents. Typically, the words loaded into the index are also assigned to different fields in relation to the tagging in the document.

[7] Ronald Bourret: *XML and Databases* , 2002. <http://www.rpbourret.com/xml/XMLAndDatabases.htm>.

[8] Kimbro Staken: *Introduction to Native XML Databases*, 2001. <http://www.xml.com/lpt/a/2001/10/31/nativexmldb.html>.

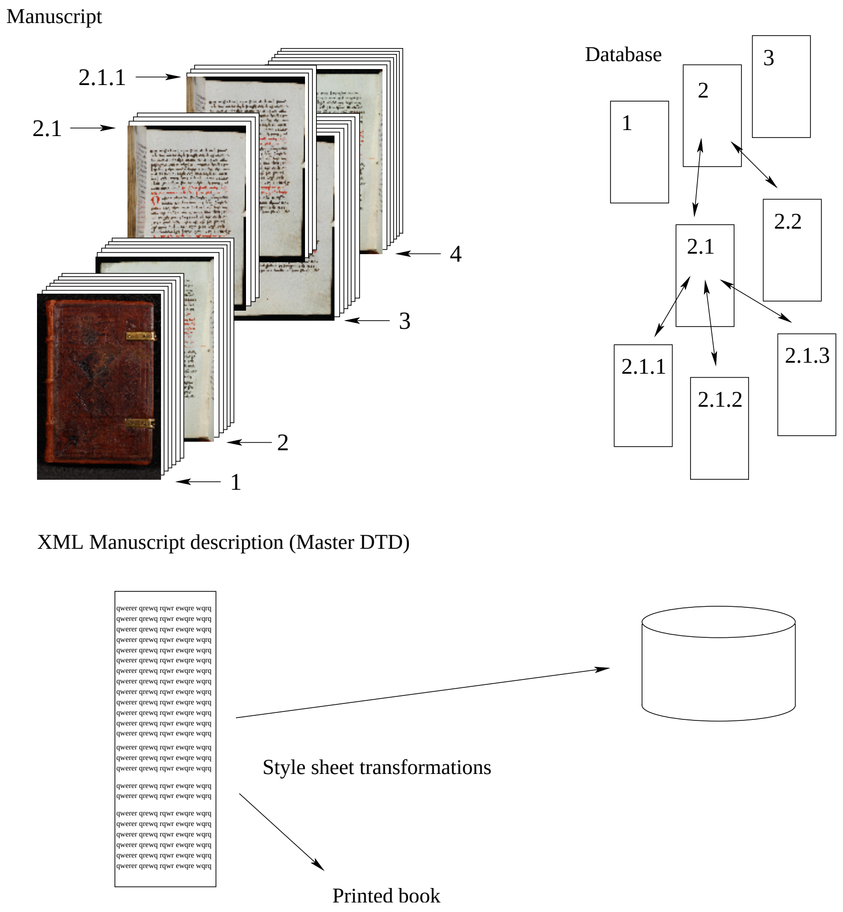
Manuscript
2.1.1
2.1
4
3
2
1
Database
3
2
1
2.2
2.1
2.1.1
2.1.2
2.1.3
XML Manuscript description (Master DTD)
Style sheet transformations
Printed book

Figure 1. The architecture of the S:t Laurentius Digital Manuscript library. The manuscripts are scanned and the resulting images are stored. The texts are cataloged using the Master DTD. The descriptions are parsed and stored in a database, the data model of which is structured hierarchically to reflect the Master content model. The cataloger assigns page ranges in the text describing the intellectual contents of manuscripts. In the web interface the ranges are translated to hypertext links into the image repository, making it possible for users to retrieve images based on content.

eventually be solved.

Apart from the fact that there are few software solutions available at the time of writing, there are other more fundamental problems in connection with using native XML databases for text retrieval. The most important one being that such databases are too closely connected to the XML structure. It is inevitable, since these databases are (currently) queried using the Xpath language[9] or (more recently) the Xquery language.[10] This precludes simultaneous searching of collections of heterogeneous document types, for instance a mixture of TEI, DocBook and Open eBook[11] documents.

The Z39.50 information retrieval protocol is based on a different philosophy. The main design goals behind it being:

□ To make a distinction between ”fields” for searching (so called *search attributes*) and those in the database (SQL tables or XML files or whatever).

[9] World Wide Web Consortium: *XML Path language (XPath)*, James Clarkand Steve DeRose (eds.) Worldwide Web Consortium1999. `<http://www.w3.org/TR/xpath>`.

[10] World Wide Web Consortium: *W3C XQuery Language*, `<http://www.w3.org/XML/Query>`.

[11] *Open eBook*, `<http://www.openebook.org/oebps/history.htm>`.

□ To permit the definition of different *record syntaxes* which are standardized abstract representations of search hits independent of the actual data model of the database.

There are several record syntaxes available. The most common one is MARC followed by XML. In addition there is a native Z39.50 syntax called Generic Record Syntax (GRS). The hits are record-like, they are RLOs, but importantly they are independent of how they are represented in the database. I will not discuss MARC further in this note.

As regards to XML and GRS in a Z39.50 context there is not much to say, other than that you could think of them as being more or less the same thing, namely a kind of simplified XML not supporting attributes, just elements. But the elements, or tags, come from vocabularies, so-called tag sets. The tag sets used in Z39.50 are partly standardized, but it is also possible to define customized tags for a specific application. Laurentius takes advantage of this feature.

The distinction between search attributes and record syntax has proved very useful in this project, as well as in bibliographic searching in general. It is a notion similar to the thinking behind SGML and XML as it aims at making information search and retrieval independent on the underlying data structures.

Everything we did in Laurentius could be done using other protocols and methods, but I firmly believe that Z39.50 and full text indexing is simpler and cheaper than most of the alternatives.[12]

2.2. XML documents and Z39.50 tag paths

The records delivered from a search engine in Z39.50 are more record like than document like, still they are more flexible than most database records. As far as the protocol itself is concerned, the fields may in principle be repeated arbitrarily, and furthermore there are no constraints as regards the lengths of the values. Now, in order to search XML DLOs we have to transform them into RLOs. The procedure for achieving this, involves the solution of two problems. First, to split the document itself into suitable portions, and to do so in a way which is meaningful given the tag semantics and which appear logical from a user's perspective. Secondly, to recast the tagging of each portion into something which becomes record-like, with Z39.50 friendly tag paths. I will start with the latter, and then continue with the former.

To transform a piece of a DLO into a RLO means that the nesting structure will have to be changed so that we obtain a sequence of predictable tags. See Example 1.

```
<p>That means that someone (actually it was
<name type="person">Sigfrid Lundberg</name>)
cheated when he late autumn <date>2001</date>
pushed a lot of XML data into a full-text Z39.50 server</p>
```

Example 1. DLO fragment with valid TEI (or Master) mark-up. Even this simple code would not make sense to my Z39.50 server.

Recasting a document into the form shown in Example 2, is a straight forward exercise in XSLT scripting. I use a style sheet which reads the Master description from the storage, and remap DLO (i.e., Master) syntax into RLO tagging. Each small RLO is then stored in my database. The main difference is that the tags appear sequentially at the same depth. Below I will discuss further how I connect these constructs to search

[12] We use the *The Zebra Server* (cf., `<http://www.indexdata.dk/zebra/>`) , It is capable of handling millions of records loaded as tagged text. The package is distributed with a *GPL* (cf., `<http://www.fsf.org/licenses/gpl.txt>`) license.

```
<record>
   <text>That means that someone (actually it was</text>
   <name-person>Sigfrid Lundberg</name-person>
   <text>) cheated when he late autumn</text>
   <date>2001</date>
   <text>pushed a lot of XML data into a
        full-text Z39.50 server</text>
</record>
```

Example 2. Hypothetical fragment RLO preserving most important aspects of the mark-up in Example 1.

attributes. Naturally, problems will arise that I have not been able to resolve in a general manner (see Example 3).

```
<p>The search engine in
<title><name type="person">Sigfrid Lundberg</name>’s
database</title> is using Z39.50 for access</p>
```

Example 3. DLO fragment that cannot be indexed without either losing some semantics (name or title), or entering redundant information in the database (both name and title).

In the automatic indexing process described above, the main issue is about getting an optimal balance between granularity and aggregation. This in turn will be reflected in the balance between search recall and precision. The original semantics of the DLO mark-up which is aggregated into the `<text> ... </text>` is lost. A number of factors affect the decisions, which are also related to the size of the database, the frequency by which mark-up is actually used and how ”heavy” the most advanced search forms are expected to be.

To be more concrete, using Master it is possible to tag names of scribes, illuminators etc. This is important information for research. However, if a database is small, a field like ”scribe” would be scarcely populated and hard to search unless you know in advance exactly what to search for.[13] In the Digital Scriptorium[14] it is circumvented by exporting lists of names from the database. This helps users and decreases the need for authority files (Consuelo Dutschke, pers. comm.). Interestingly the scan service defined in Z39.50 does exactly this. The opposite problem, an annoyingly high recall, may arise even for small databases, and the problem may be aggravated by aggregation because users may then not have facilities to improve their searches.

In a subsequent section, I will present another, much more powerful mechanism that may be used to find the optimisal balance between granularity and aggregation. Therefore, it suffices to say that whatever user interface one plan to build, it is usually advantageous to preserve as much DLO semantics as is possible.

The other aspect of fitting a DLO into a database is to split it into chunks suitable for searching. Again, this is a matter of granularity and aggregation. For prose, such as a novel in TEI, one obvious unit would be chapter. However, inside chapters there will be paragraphs. In the development of the Z39.50 details of the Laurentius search engine, my main source of inspiration was the ”Application-Support” Z39.50 Profile for

[13] Database size is in the eyes of the beholder. In my view any database containing less than 100 000 objects is indeed small (Laurentius is a very small database). A union catalogue covering all preserved medieval manuscripts in the world would be big but still only of modest size.

[14] *Digital Scriptorium*, `<http://sunsite.berkeley.edu/Scriptorium/>`.

Table 1. The seven top elements of the Master DTD, permitted directly below the top element msDescription

Element	Description
msIdentifier	Identifies the manuscript by naming its physical location and mentioning its call number.
msHeading	Basic metadata, such as author, title and date and place of origin.
msContent	Basically a table of contents. A list of manuscript items, msItems. Each msItem may contain other items, so this is a hierarchical structure.
physDesc	The physical description, such as extent, form, binding etc, but also writing system, number of hands and numerous other aspects of a manuscript.
history	Information about the origin, provenance and acquisition of the volume
msAdditional	Placeholder for various other information, including metadata for the catalog entry itself and a bibliography.
msPart	For composite manuscripts, that is, manuscripts arising at some later time by binding together a number of items that in a previous life lead an independent life. The DTD permits the msPart element to contain other msParts.

Access to Digital Collections.[15] I do not think there are many applications or profiles using the facilities provided by the Collections profile, which is a pity. Laurentius does not use it either. We did, however, borrow the notion of *hierarchical digital collections* from it.

In the Master description schema there are seven top level elements (Table 1). Of these, msIdentifier, msHeading, physDesc and history are descriptors of the volume, whereas msContents describes its intellectual content. msPart adds complexity, by making it possible to describe a manuscript as a composite structure consisting of multiple instances of its own kind. Obviously, in a database there will be one record for each manuscript or manuscript part (msPart). That is a record describing the actual physical object, its origin and appearance.

In addition, each manuscript contains intellectual content described as manuscript items in the Master DTD (Table 1). The items may be nested, such that a given manuscript item may consist of other items, and it may be a part of some other item.

Each piece of intellectual content is described in the msItem element at a Dublin

[15] Library of Congress:
Z39.50 Profile for Access to to Digital Collections, Draft Seven (Final Draft for Review)1996. `<http://lcweb.loc.gov/z3950/agency/profiles/collections.html>`.

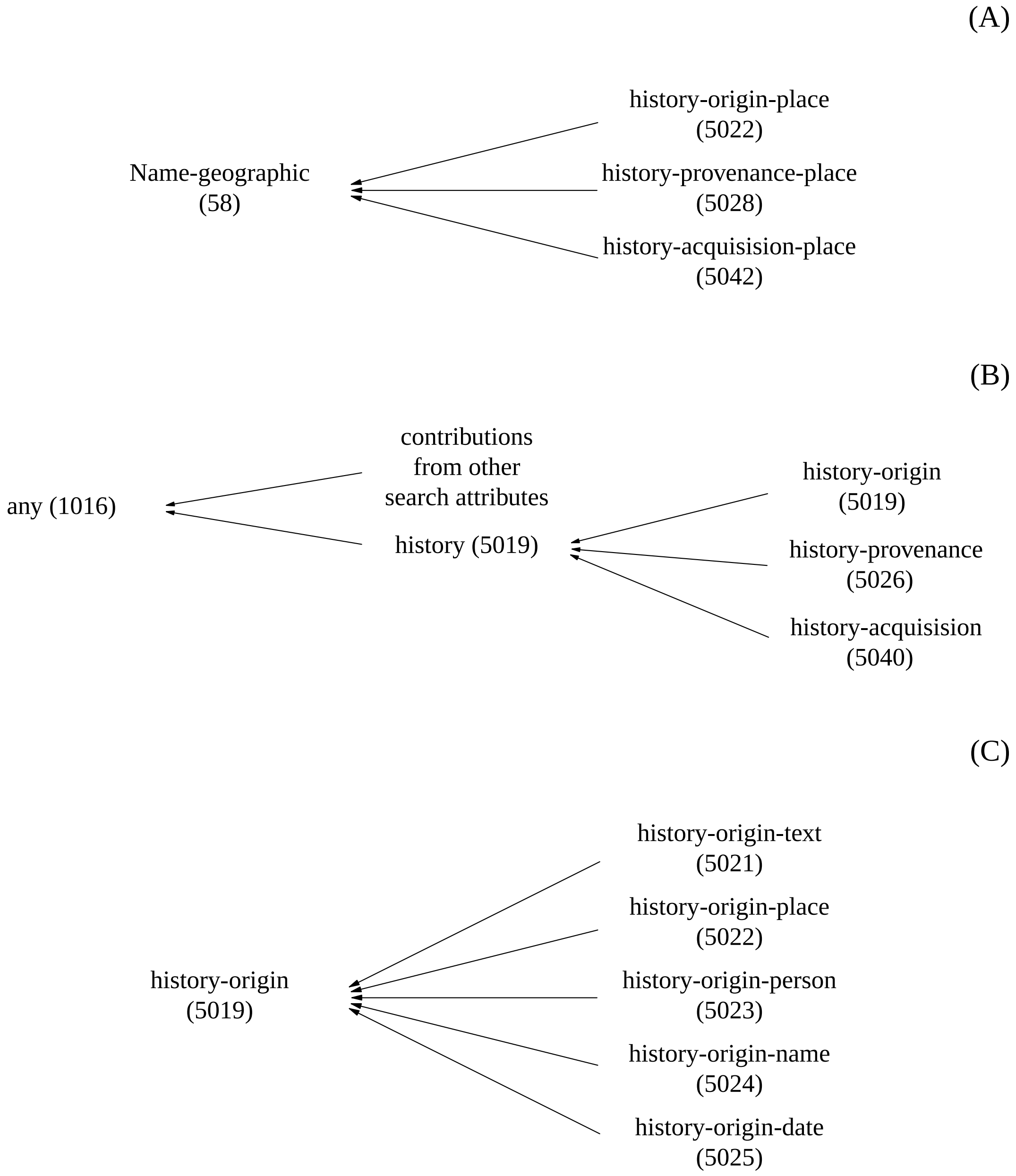
(A)
Name-geographic (58)
history-origin-place (5022)
history-provenance-place (5028)
history-acquisision-place (5042)
(B)
any (1016)
contributions from other search attributes
history (5019)
history-origin (5019)
history-provenance (5026)
history-acquisision (5040)
(C)
history-origin (5019)
history-origin-text (5021)
history-origin-place (5022)
history-origin-person (5023)
history-origin-name (5024)
history-origin-date (5025)

Figure 2. Examples of the hierarchical search attribute architecture of the Laurentius database. Numbers in brackets refer to Z39.50 attribute number. The ones above 5000 are non-standard and used in Laurentius only. (A) The searching for geographical names can be made globally, using the bib-1 standard search attribute Geographical-name. However, geographical names may also be searched in a context specific way, namely historical ones, related to origin, provenance and acquisition of a manuscript. (B) and (C) give examples on how the standard bib-1 search attribute is fed with other, more detailed, search attributes. Please refer to the main text for further discussion on how the search attributes are connected to XML mark-up.

Core like level of detail.[16] It is identified using two methods. First, internally to the manuscript by a locus, i.e., the page (or folio) range it occupies in the volume. Secondly, more specifically, it also identifies its intellectual content by incipit, explicit and rubric. More basic metadata, such as title and author are given.

[16] *Dublin Core Metadata Initiative (DCMI) Home Page*, <http://dublincore.org/>.

Thus we decided to model a manuscript as consisting of one manuscript item, the root item, possibly containing pointers to other manuscript root items (if the manuscript is a composite one). Each root item describes its corresponding physical object. The items become individual records connected to other records through *is part of* or *has part* relations as indicated in Figure 1. A worked example of how this is done is given in Appendix A.

2.3. A search attribute ontology

Like most other Z39.50 based services, Laurentius is using the bib-1 attribute set for providing a standardized interface to fielded search.[17] However, the Laurentius database has a much more intricate search attribute architecture than most other such services. For instance, the bib-1 defines a search attribute *name*. So does Laurentius. All the different kinds of names defined by the Master DTD are searchable through that search attribute. And in the the same way we define *name-geographic, title, author* and other popular search attributes implemented in bibliographic search.

Unlike other services, Laurentius is using a hierarchical search attribute structure. An ontology, if you like. The attribute 'name' is an aggregate of fields that are connected to locally defined names of persons or corporate bodies, like historical names, names of historical persons involved in the origin of the manuscripts, or persons involved in the acquisition of the object etc (Figure 2).

The same reasoning is applied to place names, dates and so forth. Even plain text (i.e., descriptive prose not bound to any particular field in the database) is entered in this hierarchical structure. This means that we can define a search attribute *history* combining all historical data, and *place* through which all place names can be searched, but there is also an attribute *history-acquisition-place* for searching only this aspect of our collection. All fields contribute to the field ”any”.

Above I pointed to the problem arising in the design of user interfaces because of the trade-off between granularity and aggregation in searching and indexing. The hierarchical search attributes architecture is a powerful tool in addressing that problem.

In Laurentius we work towards a detailed indexing. The implication of this is that we try to preserve as much of the semantics provided by the tagging in the Master description as possible, while aggregating by use of search attributes. We do so since, for example, there are not enough names in the database for making it useful to make detailed distinctions between various kinds of names in the user interface. We still have the option to implement more advanced search options at a later stage without changing the way the documents are indexed.

3. Discussion

3.1. On searching

In building the Laurentius search system we have demonstrated that it is possible to search a corpus of complex XML ”document like objects” by

- □ simplifying each document to a number of much simpler records that are connected by *is part of–has part* relations
- □ replacing nested tagging by sequences of tags with equal depth, serialized tagging

[17] The Z39.50 Implementors Group: *Attribute Set bib-1 (Z39.50–1995): Semantics* , Library of Congress Network Development and MARC Standards Office `<ftp://ftp.loc.gov/pub/z3950/defs/bib1.txt>`.

□ defining a hierarchical set of search attributes

The system is using Z39.50 for searching, mainly for pragmatic reasons; Z39.50 provided the machinery for achieving our goals. The incentive for choosing Z39.50 was thus not to provide an interoperable search environment, and thus enable cross search compatibility with library OPACs, or more generally with services using the Bath[18] or CIMI[19] Z39.50 profiles. Indeed at the current state of development, the compatibility between Laurentius and those profiles is very poor. S:t Laurentius Digital Manuscript Library Z server is a experimental from an orthodox Z39.50 point of view, but we managed to build the search service we needed. Cross searching and interoperability are fringe benefits that could and should be developed further. There are two possible directions for such development.

The first interoperability goal would be to develop virtual ”union catalogues” for detailed searching of multiple collections of manuscripts. Although interesting technically, I think, to be honest, that the benefit of such a project would be limited if it is restricted to a search for the content of Master manuscript descriptions. Few manuscript catalogues in the world are larger than that they could be sent as a single mail attachment (admittedly, Laurentius is unusually small). A monolithic central database would do the job. This was also the road taken by the Master project itself. If, however, we add complexities, like links to (XML) full text, scanned images etc, the situation may change, and this is an issue that needs investigation.

A second interoperability goal would be to provide simple resource discovery like access points to manuscript collections, that could be used from more general digital library services and portals. Again, such access points are more or less trivial if the material shared is Master records only, but beyond that level there are complications. How would a manuscript collection in the UK be integrated into JISC’s DNER[20] integrated framework of digital information services?

Lund university library has in its manuscript collection a copy of Boethius *De consolatione philosophiae*

(Incidentally, it is bound together with incunables by the same author in a composite volume.) We obviously possess modern printed editions in different languages, available via the OPAC. We may imagine that someone writes an article about the manuscript in question. That article may become available through an eprints server. Then there is an abundance of related material — printed and digital — about the author and his life and work. Ideally all these resources should be presented and somehow linked together in a way which is meaningful for research workers and students in the humanities. There are prospects for some further work.

3.2. A general text retrieval encoding?

Currently there is a vast interest in developing XML/SGML mark-up languages for various types of texts. This is an area where TEI and DocBook have held the hegemony for a decade (disregarding HTML, which has developed into a presentation language). With XML capable word processors appearing we are now facing a situation where we can expect:

□ Electronic editions of text in TEI

[18] *The Bath Profile: What is it and why should I care*, <http://www.nlc-bnc.ca/bath/prof.pdf>.

[19] *CIMI Profile, Release 1.0H (Section 1), November1998*, <http://www.cimi.org/public_docs/HarmonizedProfile/HarmonProfile1.htm>.

[20] *Distributed National Electronic Resource*, <http://www.jisc.ac.uk/dner/>.

- □ Dissertations on those texts, possibly published in a specialized ETD[21] format.
- □ Papers written on related issues using DTDs proscribed by publishers.[22]
- □ Eventually, documents written in, for example, DocBook will become sources to researchers in the humanities (like historians of science and technology).

While initiatives like DCMI[23] and OAI[24] provide a least common denominator metadata semantics and methods for metadata dissemination, similar facilities for searching and navigation of texts are nonexistent. Clearly, we need a way to search heterogeneous text databases and return result sets, where hits should be in a predictable format, regardless of the DTD of the original document.

Acknowledgments: — Comments from Sara Kjellberg, Jessica Lindholm, Consuelo W. Dutschke, Mattias Borell and Colm Doyle have much improved this note.

[21] *Electronic theses and dissertations in the humanities,* `<http://etext.lib.virginia.edu/ETD/ETD.html>`.

[22] *BioMed Central - XML/XSL Technology page,* `<http://www.biomedcentral.com/xml/dtd.asp>`.

[23] *Dublin Core Metadata Initiative (DCMI),* `<http://www.dublincore.org/>`.

[24] *Open Archives Initiative,* `<http://www.openarchives.org/>`.

Appendix A. Transformation example

Two manuscript items, in original form

```
<msItem n="2.1.16">
   <locus from="41r:4" to="41v:8">41r:4-41v:8</locus>
   <title type="supplied">Prayer to Jesus Christ in his
   pain (<ref>MDB84</ref>)</title>
   <rubric>S<expan>an</expan>c<expan>t</expan>us
   gregori<expan>us</expan> paffuæ gaff til the<expan>n</expan>næ
   effth<expan>e</expan>r skreffnæ bøn saa myghæt afflath
   som ... ee huo th<expan>e</expan>m læs m<expan>eth</expan>
   gudælígh<expan>e</expan>t Amen</rubric>
   <incipit>O kiære h<expan>er</expan>ræ
   ih<expan>es</expan>u
   <expan>christ</expan>e thu som æst alzom
   nadhæ fullæste</incipit>
   <explicit>och giiff mik ryffwilsæ i mith hiærtæ
   for allæ mynæ
   syndær Amen p<expan>ate</expan>r
    n<expan>oste</expan>r</explicit>
</msItem>
```

and transformed into the Laurentius Z39.50 friendly tagset

```
<manuscriptitem>
   <msid>Mh_35</msid>
   <manuscript>Medeltidshandskrift 35</manuscript>
   <title>Prayer to Jesus Christ in his
   pain (MDB 84)</title>
   <identifier>Mh_35-item2.1.16</identifier>
   <from>41r:4</from>
   <to>41v:8</to>
   <incipit>O kiære herræ ihesu christe thu som
   æst alzom nadhæ fullæste</incipit>
   <explicit>och giiff mik ryffwilsæ i mith hiærtæ
   for allæ mynæ syndær Amen pater noster</explicit>
   <rubric>Sanctus gregorius paffuæ gaff til thennæ
   effther skreffnæ bøn saa myghæt afflath som ... ee huo
   them læs meth gudælighet Amen</rubric>
   <item>
      <ispartof><identifier>Mh_35-item2.1</identifier></ispartof>
      <title>Prayers to Our Lord Jesus Christ
      and to the Trinity</title>
   </item>
</manuscriptitem>
```