ECP 2006 DILI 510049

ENRICH

Report on Development and Validation of Migration Tools

Deliverable number	*D-3.3*
Dissemination level	*Public*
Delivery date	*28 February 2009*
Status	*Draft*
Author(s)	*James Cummings*

e*Content*plus

This project is funded under the *e*Content*plus* programme[1],
a multiannual Community programme to make digital content in Europe more accessible, usable and exploitable.

1 OJ L 79, 24.3.2005, p. 1.

Document Version Control

Version	Date	Change Made (and if appropriate reason for change)	Initials of Commentator(s) or Author(s)
0.0	28 Feb 09	Draft Deliverable	JC,OUCS

Document Review

Reviewer	Institution	Date and result of the review

Approved By (signature)	Date

Accepted by at European Commission (signature)	Date

1 Executive Summary

This report (D3.3) investigates the development and validation of migration tools for the ENRICH project. In doing so it conducts two case studies into the migration of popular formats for manuscript description to the ENRICH Specification, and develops tools to successfully complete these migrations. It analyses problems that are inherent to any form of migration based upon retrospective conversion and the methodologies and technologies used for legacy metadata migration. It reports briefly on the two case studies and provides pointers to the tools and input and output data used for them. The report then considers the validation of migration tools and the output data they create, before concluding with recommendations for the development of migration tools and those migrating to the ENRICH Specification. More information is available at http://tei.oucs.ox.ac.uk/ENRICH/Migration/index.xml.

The general recommendations (from section 5.4) of this report are:

1. If possible use technologies that are mature, open source, cross-platform, human-readable, text-based scripting languages with well-developed support options.
2. Methodology for migration should be modular and take multiple forms, at least building both against the specified data format and a testbed representative sample of the data to be migrated.
3. Additional testing of the output should be done by targeted searches of the output data and proofreading a statistically significant randomly-selected sample. Any errors should be corrected in the migration tool and the conversion re-run from the start.
4. With migration to the ENRICH specification there are three approaches:
   - Archive-specific migration route: this is best done with human interaction customising the available scripts to the specifics of the data format. ENRICH partners can contact enrich@oucs.ox.ac.uk to discuss the migration needs. Non-ENRICH partners can also contact us as above, and we will attempt to assist on a best-effort (or optionally consultation) basis.
   - Self-guided migration route: those with sufficient XSLT experience available to them can use or modify for use the migration tools provided. They are available under a Creative Commons Attribution license and so freely able to be used and modified.
   - ENRICH Garage Engine migration route: the project will be producing a web application to enable migration through multiple formats. If you are interested in this option, check the website once it has been released.
5. The process of migration chosen should be publicly documented and this documentation stored alongside the migration tools and input and output formats.

TABLE OF CONTENTS

2 Development of migration tools

This section looks at the development of migration tools for ENRICH through examining the problems inherent in such conversions, the methods of tool development and the options of technologies for migration. The development of migration tools for any retrospective conversion of existing data formats to a new standard format poses a number of obvious problems. The methodology chosen for migration tool development can significantly affect the quality of migration or amount of work required to produce a satisfactory conversion. The technologies available, and those which might be appropriate to a given migration task, are numerous and various, and as such the motivations behind particular choices of technologies should be investigated.

2.1 Problems inherent in retrospective conversion

In retrospective conversion to migrate existing data formats to a new standard, in this case the ENRICH specification, the goal is to have a lossless conversion. However, in some instances this may be an impossible goal. While every care should be taken to attempt a lossless conversion where possible, this may not indeed be possible owing to incompatibilities between the original and target format. With the ENRICH specification, for example, numerous attribute values have been tightly constrained to allow only a small number of values. This has been done to encourage standardisation and interoperability across resources. And while these values have been chosen to accommodate the broadest range of expected needs, there will always be cases where the original material has an attribute value that was not predicted by the designers of the new schema, or does not sit comfortably in only a single category. This is not to be seen as a failing in the new target standard, as some earlier standards may contain data models which are inherently incompatible with a newer standard. In cases such as these either the migration tools may need to be manually adapted to suit one particular source's input data, or a loss of a certain amount of data tolerated. An example of this might be where the input file contained a non-standard single text element whose content was required in two separate places in the output specification. The options would be:

- to modify the migration tool to attempt to split this text element into two portions (and the related diagnostics that guessing as to its content implies),
- to place the information in both locations,
- to put the full content in one place and a place-holding note in the other location or,
- to substitute the information with some default 'other' value acceptable to the target specification.

For lossless conversion only really the first of these is satisfactory as the others misrepresent the data. For general retrospective conversion the final one of these may also need to be used if there is no reliable method for guessing as to how to fragment the data for the first solution.

Another problem inherent to retrospective conversion from older formats is that it is dependent upon the rigour of the validation or constraints originally applied to the input format. This is in addition to any general worries concerning the reliability of the source data. In many cases the validation of the original may solely be dependent upon the vaguely-written prose of a set of local encoding guidelines. More recent widely-accepted standards tend to use some form of schema to attempt to ensure that document instances match the prose of the specification. If the target specification requires that an attribute value is one of a short list, and the schema for the source's version allowed any text content at this point, then lossless conversion can be problematic. In cases such as this, the migration tools must

inductively attempt to guess at which of the allowed list of possibilities should be used for the possibly infinite values of free-text source.

An example of this in the ENRICH project is seen in converting MASTER files with MASTER's <handDesc> element. In the ENRICH schema this is equivalent to the <handNote> element (grouped together inside a <handDesc> element). The @script attribute on MASTER's handDesc element is declared in its DTD as 'CDATA'. This means that any and all character data is freely allowed here, including whitespace. In the ENRICH specification, the very similar @script attribute on the <handNote> element is declared as only allowing a small set of permissible values. For example, our testbed of MASTER files includes values such as: “caroline minuscule”, “Carolingian minuscule”, and “carolingian-insular minuscule”. It is pretty obvious that these values should be assigned the permitted ENRICH Specification value of 'carolmin'. However, while this does lose the finer-grained distinction between Carolingian Minuscule and Carolingian-Insular Minuscule, this would still be preserved in the text content of the element. The real difficulty comes in creating the tool to guess that this value should be applied because of the free-text nature of the incoming source data. For all of these examples it is straightforward to guess that any incoming @script attribute (once transformed to lower-case) with the content 'carol' should be given the 'carolmin' value. This accounts for the terms 'Caroline' and 'Carolingian' as synonyms, and indeed appears to work for all values in our samples, but is necessarily not exhaustive because of the limitless possibilities of the incoming source. If a value had said 'Car. Minusc.' in the source data, it would be converted to 'other' in the output even though a human proofreader could understand it should have been assigned to 'carolmin'. The only course in such instances is to continually reassess the migration tool's method based on proofreading of the result, especially in the cases where a default 'other' value is applied.

The discussion of this concern, of course, is tremendously understating to what degree this may be a problem in a European context. With a wide range of different input languages the migration tool cannot hope to make reliable guesses on free-text values unless the tool is highly modified. In such cases a variation of the migration tool needs to have all text-based guessing replaced with similar phrases in the input data format's language.

Loose validation in the original data format is certainly a problem, but in some ways this is less problematic than undocumented deviation from the original format. In these cases a resource creator has started with an acceptable standard, but then when faced with challenging data that the standard doesn't (or doesn't appear to) cater for, they have extended the specification to suit their local encoding guidelines. In the best cases they have rigorously documented the differences between the original and their local modification of this standard, and even better submitted their proposed changes back to those who created the specification in the first place. However, in cases of very small variations from the published specification this is quite unlikely to happen. In addition, the person responsible for introducing the deviation from the standard may have left the content-providing institution long ago and those responsible for the legacy data may not even be aware that their encoding guidelines have modified the standard. If cases such as these are detected then they should be treated as separate (but related) data formats and the migration tools modified to cope with the variance in the two specifications.

A related problem to this is simple abuse of the original standard, where data creators have used a known standard but when faced with problematic data requirements have misrepresented that data through abuse of the semantics of the existing standard. This kind of semantic tag abuse is quite common and sometimes derives from a misunderstanding of the meanings of the tags, or unfamiliarity with the standard, but also from specifications that do not cope well with the input data leaving a data creator to find a place to put the information which validates syntactically according to the specification they are using, regardless of the intended semantics of that location. In such cases if these abuses are not documented, and they tend not to be because they are a misuse of the standard, the input data will need to be rigorously checked and documented prior to conversion.

Regardless of the problems inherent in the migration of legacy data to a more modern format, it is usually a worthwhile process in order to gain the benefits, especially those of interoperability, provided by access to tools available for that new format. In the case of ENRICH this is evident in the cross-searching and other benefits available in the manuscriptorium platform.

2.2 Methods of migration tool development

Given the inherent difficulties mentioned above in building migration tools for retrospective conversion, the methodology for building these tools must necessarily depend on the nature of the legacy data available. There are two basic approaches in the development of migration tools. One can build the tool based on:

- the input specification (e.g. its DTD and Guidelines) or,
- a testbed of real-world samples which match this specification.

Either of these approaches has benefits and drawbacks. Building a migration tool based upon the input specification follows the rules and allowed distinctions as specified in the standard the original data creators were meant to be following. However, such a tool isn't able to cope reliably with the infinite variety possible in under-specified aspects of the original standard (e.g. free-text attributes as considered above). While this approach is good for creating the basic structure of a migration tool, in that one can understand the possibilities the data model allows, it is limited in its understanding of the ways in which the specification has been used by data creators.

Building a migration tool based upon a testbed of real-world examples has been benefit of examining real-world document instances whose formats supposedly reflect the input specification that they have all followed. Moreover it allows one to sample data values from free-text fields to attempt to derive some commonality between them and the target format. However, the success of this method is inherently and statistically limited by the size of the sample data set. If too few samples are chosen then this method may miss infrequently-used but equally-valid data models. Conversely if too many examples are used then every aspect of the testbed may not be able to examined for each example in detail. In both of these instances it may be the case that valid but highly-unusual applications of the original specification may be lost because they don't appear in the sample or being so peculiar that it doesn't stand out against the noise of a larger data set.

In reality, any sufficiently robust methodology for migration tool development will use a combination of these approaches. For example, it may build the basic structure for conversion based upon the published standard, and then expand upon this based on a reasonable-sized corpus of real-world examples. This has the benefit of coping with variants as allowed by the original specification, but provides real-world data to enable handling some of the common English free-text values. This is an iterative methodology, colloquially referred to as “lather, rinse, repeat”, in which the work is carried out in small steps, and then the overall goals and methodology reassessed after each iteration of work has been completed. Moreover, such a methodology helps to modularise the tasks undertaken by the migration, which should be done wherever it is possible that the migration might introduce unintentional errors into the conversion process. For example, the preparation of input data into a form suitable for conversion should be separate from the actual conversion itself. The ENRICH case study migrations discussed below have been developed with this modular combined methodology. Nonetheless, it is still the general recommendation of this report that in the majority of cases, except where the input data rigorously follows the standard in the expected manner, migration will still need to be mitigated by human interaction or customisation of the tool to the data.

2.3 Legacy metadata conversion technologies

In the development of any migration tool the technologies chosen to implement the tool will depend on numerous factors, most notably the relevant technological experience of those creating the tools. Nonetheless the choice of technologies deserves some attention. The tools for migration of legacy metadata collections should, wherever possible, use technologies which are open source to help with both long-term preservation of results and portability of migration tools. In addition technologies should use human-readable text-based scripting languages for conversion where possible. While these recommendations are best practice, it is not always the case to follow them. In some cases the legacy data may be stored in a binary and proprietary system with only one closed route to export the data in a single (possibly proprietary and binary) format. In cases such as these it is best to modularise the migration to allow for separate processing of the the exported data to a known textual format and the migration from that format to the target specification. This separation of concerns will help with debugging when errors have been introduced by indicating whether these errors are in the initial conversion or the migration itself.

In creating migration tools there are, of course, numerous technologies available. Care should be taken to use technologies that are not only open source and human-readable as mentioned above, but also sufficiently mature in their development and support. This should help to ensure not only that expertise will exist to modify the migration tools to deal with local encoding differences, but also that the migration can be updated and extended many years later if conversion errors have been discovered or local encoding practices have changed. Another reason for this is that it makes it more likely that sufficiently-complete implementations of the technology exist on multiple operating platforms, or that a cross-platform implementation has been developed.

If the input records are text-based, then a tool such as PERL is an example of an appropriate technology. It has a long history of development and is thus mature and well-supported internationally. However, for the case studies below, XSLT has been used as a more appropriate technology because the input record format is already in an XML format. This is not to imply that PERL couldn't also cope with XML, or that XSLT can't cope with non-XML data, but that XSLT was deemed to be more appropriate for these migrations. This provides a human-readable text-based language for migration which is template or rule-based and thus easy to modify when new variations are encountered. It is a mature technology with cross-platform implementations and a well-developed support options.

While it is obvious that migration technologies need to be fit for the purpose for which they are being chosen, it is the recommendation of this report that where possible these be mature, human-readable, open source, cross-platform, text-based scripting languages applied in a modular manner if necessary. Such a technological choice will help to alleviate some migration problems, or at least enable the tracking and correction of scripting errors where possible

3 Two case studies: MASTER and EAD

As part of this workpackage the ENRICH project undertook two case studies in the development of migration tools. These were to provide migration tools from two legacy data formats to the ENRICH specification developed by the project.

3.1 MASTER:

The Manuscript Access through Standards for Electronic Records (MASTER) was an EU project funded to create a single XML-based standard for computer-readable descriptions of

manuscripts. In many ways this project was the pre-cursor to ENRICH. The MASTER data format was updated and modified and eventually incorporated as a module into the Text Encoding Initiative (TEI) P5 Guidelines. MASTER itself was based on an earlier (P4) version of the Guidelines. The TEI P5 module has since evolved further and in turn has been used as the basis of the ENRICH specification. The ENRICH project has contributed its resolutions on the creation of manuscript descriptions back to the TEI and they have been ratified by the TEI Technical Council and adopted back into their Guidelines. Owing to the history of relationship between MASTER, TEI, and ENRICH, it was an obvious candidate for a case study on the development of migration tools.

3.1.1 Description of format

The MASTER data format is an XML vocabulary that is an extension to the TEI P4 Guidelines. Its DTD is a customisation of the TEI P4 DTD, and as such it starts with a <TEI.2> element and contains a <teiHeader> and other aspects one would associate with a TEI P4 document. However, it allows an <msDescription> element (and children) inside the <sourceDesc> and at other textual locations which is not present in standard TEI P4. Some useful reference material is available and so the details will not be repeated here :

- MASTER Reference Manual (available at http://www.tei-c.org.uk/Master/Reference/oldindex.html)
- MASTER DTD page (available at http://www.tei-c.org.uk/Master/Reference/DTD/)
- MASTER Examples page (available at http://www.tei-c.org.uk/Master/Examples/)

In addition, Matthew Driscoll has written an article containing a description of the history of MASTER and its relationship to the TEI in the development of the TEI P5 module. See: http://www.digitalmedievalist.org/journal/2.1/driscoll/

3.1.2 Differences from ENRICH specification

The differences to the ENRICH specification are, of course, too numerous to detail here. However, in looking at some of the differences which affect the migration from MASTER to the ENRICH Specification, the ENRICH project produced a document outlining most of the changes from MASTER to TEI P5 (upon which the ENRICH specification has been based). This is available from http://tei.oucs.ox.ac.uk/ENRICH/Deliverables/WP3-00.1.0.xml and details content model changes, elements that have been renamed, elements that have been added, elements that have been removed, and changes to attributes. Many of these changes are concerned with the integration as the manuscript description module in TEI P5. They include simple things such as basic renamings, the addition of the TEI namespace, and additional constraints on textual attributes. These can be as simple as the <msDescription> element changing its name to <msDesc>. However, in some cases more significant changes have resulted in a very different data model such as when a datatype is now applied to an attribute which had no such constraints upon it before, or an element which allowed a particular content model no longer does.

3.1.3 Sample datasets

The dataset for the MASTER migration tool development case study was based on the sample MASTER records that were deposited with the MASTER project as examples. This provided a corpus of well over a thousand records from the following institutions:

1. **AMI**: About 500 full records relating to Icelandic manuscripts created by the Stofnun Árna Magnússonar (Arní Magnússon Institute) in Reykjavik
2. **BMR**: About 30 full records from the Manuscritos de América en las Colecciones Reales(American manuscripts in the Royal Collections) portal created at the University of Alicante

3. **IRHT**: About 350 short records relating to French manuscripts, extracted from the Médium database maintained at the Institut de Recherche et de l'Histoire des Textes, Paris
4. **KB**: About 90 records relating to Dutch and Flemish mediaeval manuscripts, from the collections of the Koninklijke Bibliotheek (Royal Library), The Hague
5. **NLP**:About 50 records relating to mediaeval manuscripts in the collections of the Národní knihovna České republiky (Czech National Library), Prague
6. **Well**: a small collection from The Wellcome Trust

These may not be the most up-to-date versions of these records. Instead these are the ones which these institutes had deposited as samples with the MASTER project. As illustrative examples it seemed reasonable to use them to create our testbed corpus.

The sample datasets are available in two forms: as they were deposited with the MASTER project, and in a processed format to prepare them for conversion. This second format expands any entities, and includes any default attributes from the DTD, and creates a single document (duplicating any additional material) for each and every manuscript description provided. In development of the migration methodology for MASTER it was decided that not only should the files be pre-processed to be in a standalone form (i.e. not needing or referencing the MASTER DTD), but also that there should be a one-to-one relationship between the input and the output. Having one manuscript description per file allowed significantly greater ease of debugging during the evolution of the migration tool.

The original MASTER files used to create the testbed sample are available as tar archive files linked to above, or on the MASTER Examples page (available at http://www.tei-c.org.uk/Master/Examples/). These are also available at: http://tei.oucs.ox.ac.uk/ENRICH/Samples/MASTER/. The input files used for the conversion after pre-processing are also available at http://tei.oucs.ox.ac.uk/ENRICH/XSLT/testbed/MASTER/.

3.1.4 Individual problems

The migration from MASTER to the ENRICH specification posed a number of problems. In pre-processing the files for conversion it was noticed that ID/IDREF references in the original meant that a lot of duplication of content had to be provided in the output. This was so that any references to IDs could be preserved. Another option could have been recreate only the needed structures in the preprocessed input but it was decided that would provide unnecessary complexity since if a corpus of files were so presented, the resulting output files would probably be merged together again after the migration.

The stylesheet developed was built on top of a default stylesheet for TEI P4 to TEI P5 conversion provided by the TEI, since this handled many of the issues not directly related to manuscript description. Some of the problems relate to the tightening of constraints decided upon by ENRICH where free-text values were allowed by MASTER. In some cases this simply means that what was allowed in MASTER as 'CDATA' which are now a TEI datatype which is more stringent in its requirements. For examples, the <handNote> element has a @scribe attribute which is a TEI datatype that resolves eventually to xsd:name. This means that it is unable to contain whitespace and various other punctuation characters, and so these must be removed from the input value before it is migrated to the output. This is much simpler than those cases where an input MASTER free-text value must be found to correspond to a very short list of options. For example the <supportDesc> element in the ENRICH schema requires the presence of a @material attribute indicating the nature of the material described. The only likely source for this information is in the text content of the MASTER <support> element. In this case the textual value of an element (which can contain various child elements) is interrogated to try to guess the correct value for this attribute. The allowed values are 'perg' (parchment), 'chart' (paper), 'mixed', and 'unknown'. If any of these (or 'paper' or 'parch') are found in the text content, then the attribute is given its corresponding value. Otherwise, the attribute must be given 'unknown' as a value to produce a valid output.

While this is partly unsatisfying, since human proofreading of the original against the converted output would identify a number of instances where this necessarily very liberal guessing has gone wrong, it is a necessary compromise in the development of mass migration tools.

Some other individual problems involved mass reorganisation of the location of elements. For example, before adoption as a TEI P5 module the data model of <physDesc> was significantly changed by the addition of grouping elements that had not existed in MASTER, or in the case of <accMat> were located elsewhere. This entails careful handling and creation of these new grouping elements only if the necessary data for the required child elements is indeed available.

One of the most awkward aspects of the migration was attempting to preserve the intellectual content of date-related attributes which were 'CDATA' in MASTER but use standard W3C-style dating in the ENRICH specification. This involved migrating dates in all sorts of free-text formats to a W3C standard 'YYYY-MM-DD' style of format (Y=a year numeral, M= a month numeral, D= a day numeral, for example 1415-10-25 for the 25 October 1415). This input data includes formats such as 'c. YYYY', 'YYY' (3 digit year), 'MM.DD.YYY', 'DD.MM.YYYY' and various other formats involving other combinations or character separators. The type of guessing involved at such points is always going to be approximate and prone to error in unusual circumstances. For example, if a date attribute ends (with a dot or dash separator) with a three digit numeral it is always assumed that it represents a year prior to 1000. While this may not be strictly accurate it is a result of building a transformation against a testbed of records. During the debugging phase any unrecognised date was output and rules developed to deal with each case in turn until no more errors were given. The results were then randomly sampled for proofreading.

The XSLT stylesheet to convert MASTER records to the ENRICH specification is available as: master2enrich.xsl (available at http://tei.oucs.ox.ac.uk/ENRICH/XSLT/xsl/master2enrich.xsl).

3.1.5 Transformation results

The results of applying the master2enrich.xsl stylesheet to an input file is a valid ENRICH file containing a single <teiHeader> root element which, inside the <sourceDesc> element, contains a single <msDesc> element. As ENRICH is a more tightly-controlled subset of TEI P5, any valid ENRICH output file could be incorporated inside a TEI template to replace a <teiHeader> and form a valid TEI file.

A webpage listing all of the files converted and giving an HTML proofreading view of them is available from: http://tei.oucs.ox.ac.uk/ENRICH/XSLT/master.xml. The XML version of the converted files are also available from: http://tei.oucs.ox.ac.uk/ENRICH/XSLT/testbed/MASTER-ENRICH/.

3.2 EAD:

Encoded Archival Description (EAD) is a set of guidelines describing the intellectual and physical aspects of archival finding aids so that the information they contain may be easily searched, retrieved, displayed, and exchanged in a predictable platform-independent manner. It was produced by the Society of American Archivists and the Library of Congress. Many of the EAD concepts are based upon early work with the TEI and there is a large crossover of users between EAD and TEI. EAD is sometimes preferred for archival collection metadata and TEI is generally preferred for textual content, though increasingly TEI is becoming a de facto standard.

3.2.1 Description of format

The EAD format used by the Bodleian Library, University of Oxford, is based on version 1.0 of the standard. This was then superseded by a second version of the EAD standard in 2002. No successive versions of the EAD standards have since been released, though it is still used in many libraries. A variety of useful background information concerning EAD is available:

- EAD Website (available at http://www.loc.gov/ead/)
- EAD Tag Library, version 1.0 (1998) (The now superseded version of EAD used by the Bodleian, available at http://www.loc.gov/ead/tglib1998/index.html)
- EAD Application Guidelines, version 1.0 (1998) (available at http://www.loc.gov/ead/ag/aghome.html)
- Various EAD Help Pages (available at http://www.archivists.org/saagroups/ead/)

However, it should be stressed that the format used here is a particular instance of the version 1.0 format as used by the Bodleian Library. In this case the methodology was partly based on structural observations of the EAD samples provided, and partly on building up the output slowly for each aspect present in the EAD and required in the ENRICH specification.

3.2.2 Differences from ENRICH specification

The format of EAD is entirely different from that of TEI and the ENRICH Specification. In the case of the Bodleian catalogue records these are also a very individualistic use of the EAD standard. Each manuscript description is contained inside a <c01> level element with a @langmaterial attribute giving multiple possible languages of the text. Structurally under the <c01> element the Bodleian records contain children of <did> (Descriptive Identification), <scopecontent> (Scope and Content description), <odd> (Other Descriptive Data), and <add> (Adjunct Descriptive Data), usually in that order. In Bodleian records the <did> contains some of the source material for conversion to an <msIdentifier> including id numbers such as classmarks, titles and dates. Moreover it also contains a <physdesc> element with physical description suitable for ENRICH's <physDesc> element. The <scopecontent> in Bodleian is used to record information about the decoration found in the manuscript which is equivalent to ENRICH's <decoDesc> element. <daogrp> elements inside the <odd> element are used to provide links to illustrative manuscript images, and also images of the paper catalogues which were the source for manuscript description. The <add> element and its <bibliography> child element is used to store bibliographic information which should end up in the <additional> element of the ENRICH specification.

3.2.3 Sample datasets

The files provided by the Bodleian for this included:

- additional-a.xml (Additional A Manuscripts, available at http://tei.oucs.ox.ac.uk/ENRICH/XSLT/testbed/EAD/additional-a.xml)
- additional-b.xml (Additional B Manuscripts, available at http://tei.oucs.ox.ac.uk/ENRICH/XSLT/testbed/EAD/additional-b.xml)
- barlow.xml (The Barlow Manuscripts, available at http://tei.oucs.ox.ac.uk/ENRICH/XSLT/testbed/EAD/barlow.xml)
- don.xml (MSS Don., Donations, http://tei.oucs.ox.ac.uk/ENRICH/XSLT/testbed/EAD/don.xml)
- The input files to the conversion as a whole are available at: http://tei.oucs.ox.ac.uk/ENRICH/XSLT/testbed/EAD/

In this instance it was decided to keep the files one-per-collection as they were delivered from the Bodleian rather than burst them out into individual files one-per-manuscript-description as was done with the MASTER files. This meant that individual output files needed to be created by the conversion, and XSLT2's <xsl:result-document> was used to do this. This

allowed a number of benefits in the creation of the output file and control over its name which were implemented as a separate step in the MASTER migration.

3.2.4 Individual problems

The problems in converting EAD records were not numerous but can be classified into two categories. Either they were absence of data in instances where ENRICH requires data, or they were problems with parsing free-text elements to gain some sort of information. An example of the first case is the with the @form attribute on the ENRICH specification's <objectDesc> element. This is a required element, and if it exists it is allowed to have 'codex', 'leaf', 'scroll' or 'other'. However, this information is not recorded (in any reliably accessible means) in the Bodleian EAD records. It would seem in that case that one should use the value of 'other', but since that implies that this manuscript is not a codex, leaf, or scroll, it was decided in consultation with the Bodleian to use 'codex' as a default since the vast majority (if not all) of these manuscripts happened to be codices.

The parsing of free-text elements to extract necessary information has the same kinds of problems as in the MASTER conversion. However, there was an interesting variant upon this problem in that the Bodleian records do not structurally separate the distinct parts of a composite manuscript. While the ENRICH schema allows an individual <msPart> element for each part of a manuscript that was once separate but is now bound together for some reason, the Bodleian indicates this with two vertical bars '||' dividing the element. An example of this might be:

```
<unittitle>
  <title>Nicolaus Praepositus of Salerno, <emph render="italic">Antidotarium
   parvum</emph>. || <emph render="italic">Antidotarium</emph> ('Aurea Alexandrina
   faciens ad reuma'). || Roger de Baron, <emph render="italic">Rogerina major</emph> and
   medical texts.</title>
  <geogname>Italian</geogname>
  <unitdate>14th century, first half || 12th century, first half || 13th or 14th century</unitdate>
</unittitle>
```

In this case there are three separate manuscript parts bound together as a single manuscript. They have three separate titles, three separate dates, but are all Italian in provenance. Care must be taken when splitting each of these attributes to create a separate <msPart> element for each of them not to throw away the italicisation that is marked (in this case for titles). Although there are a number of ways to accomplish this in the chosen technology, this used XSLT2's <xsl:for-each-group> in combination with modes and <xsl:analyze-string> to separate out each section on the basis of these two vertical bars '||' whilst preserving any internal markup.

The XSLT stylesheet to convert Bodleian EAD records to the ENRICH specification is available as: ead2enrich.xsl (available at http://tei.oucs.ox.ac.uk/ENRICH/XSLT/xsl/ead2enrich.xsl).

3.2.5 Transformation results

The results of applying the ead2enrich.xsl stylesheet to an input file is a valid ENRICH file containing a single <teiHeader> root element which, inside the <sourceDesc> element, contains a single <msDesc> element. As ENRICH is a more tightly-controlled subset of TEI P5 any valid ENRICH output file could be incorporated inside a TEI template to replace a <teiHeader> and form a valid TEI file.

A webpage listing all of the files converted and giving an HTML proofreading view of them is available from: http://tei.oucs.ox.ac.uk/ENRICH/XSLT/ead.xml. The XML version of the converted files are also available from: http://tei.oucs.ox.ac.uk/ENRICH/XSLT/testbed/EAD-ENRICH/.

4 Validation of migration tools

The validation of migration tools is inherently bound up with the validation and authentication of their resulting output. And while it does not matter, per se, how that output is created, the authentication of the process, the debugging of errors, and the long-term preservation of the tool chain are significantly aided by using mature, open source, cross-platform, human-readable scripting technologies. The validation of these tools must examine their intention (migration versus crosswalks), the limits inherent to structural validation of results, methods of testing through additional constraints or transformations, and proofreading the results.

4.1 Migration versus crosswalks

In converting between formats for the purposes of migration it is necessary to distinguish between migration and the production of crosswalks for conversion to and from formats. Crosswalks are quite frequently used in library communities, for example where data needs to be transformed from a storage system that uses one format into another format (for example, for purposes of display) but also that modified versions of this converted format need to be able to be ingested back into the original storage system. In this case two conversion tools (From format A to format B, and from format B to format A) are needed, or a tool that is sufficiently sophisticated to handle both of these. While it is more than possible to move from ENRICH to either MASTER or EAD, because the project is interested in migrating resources to more up-to-date formats, it has not produced any tools to reverse this process. The point has been migration rather than conversion to and from these formats. However, this is quite straightforward and may be enabled as part of constructing an ENRICH Garage Engine which is a format migration API (discussed below in the conclusions) as an additional part of the ENRICH project. However, since the ENRICH project is recommending against storing records in EAD or MASTER, it would seem detrimental to its stated aims to be giving data creators the means to migrate to these now-dated formats.

4.2 Limits of schema validation of migration results

As part of the ENRICH project a specification has been produced using the TEI Documentation Elements module. This so-called TEI ODD file can be used to generate documentation, such as the ENRICH Specification Reference Manual, and various schemas to validate one's individual document instances. These schemas are available in Relax NG (Compact Syntax), Relax NG (XML Syntax), W3C Schema, and DTDs. However, the Relax NG formats are the recommended versions of the schema to use because of the known limitations of W3C Schema and DTDs. See http://tei.oucs.ox.ac.uk/ENRICH/ODD/RomaResults/ for access to the schemas.

These schemas necessarily have the same inherent limitations as to the validation they can provide that any other schemas would have: they can tell if a document instance has all required structural elements and attributes; they can warn if an attribute value doesn't match the required datatype; or if an element has been placed somewhere it is not allowed. However, this leaves significant room for error. For example, other than closed attribute value lists, one of the most rigorous attribute datatypes in the ENRICH specification is that of dates. The Relax NG schemas can complain if you don't properly put the date in 'YYYY-MM-DD' format (e.g. if you put in 14151025 without the hyphens) and indeed if the 'MM' section of this is greater than 12, or if the date chosen wouldn't exist (the 31st of April for example). However, if our input date had been in 'MM-DD-YYYY' format, and the 'DD' as in in this case was less than 12 (e.g. 03-09-1415), it might have accidentally been converted to 1415-09-03. This would be a valid date according to the schema, but obviously confuses the 9th of March with the 3rd of September. Likewise, if the input data has put something that is not a title in a <title> element, the migration tools have no means to be able to detect this. A

lot of the success of migration to a new format depends on the quality and consistency of the original data.

Accordingly, it is important to remember the often subtle difference between validity and truth. Something can be structurally valid according to its schema, but contain data that is entirely erroneous. While the migration tools provided as part of the case studies discussed here attempt to do a reasonable amount of testing, they assume that the person migrating the manuscript descriptions is familiar enough with both those descriptions and the migration tool to be able to spot where inconsistencies in the input data may produce erroneous, but valid, output. It is partly for this reasons such as this that one of the recommendations of this report is that successful migration will be archive-specific with human interaction in an iterative process, rather than automated format conversion.

4.3 Additional constraints or transformations

In addition to validation against the ENRICH Specification's schema, there are additional automated forms of validation that could be implemented. Some of these were used during the debugging process in the initial creation of the migration tools for the case studies above, but were ephemeral as once the particular inaccuracy had been found and incorporated in the tool, the additional checks were not needed. Multiple possibilities exist for methods of doing such checking. Hand in hand with schema validation, one could use schematron to check further structural rules (of the type if element A exists, then is attribute B value Y). And yet, other than agreeing to the ENRICH Specification, the project has not agreed additional checks that could be rigorously enforced.

Another form of checking can be done over a whole collection of resulting output by generating proofreading lists of distinct values for certain suspect elements and attributes. For example, while any free text is allowed inside the <institution> element, inside ENRICH's <msIdentifier>, the likelihood is that for any particular migration the content of this element should be the same. Producing a quick output where only distinct values are presented for the <institution> element is quite simple (it is a basic XPath) and would allow a researcher to see whether there is any unintended variance in fields such as this. This technique was also used against in the input during the debugging process in order to establishing possible values in the incoming data. Currently the <msIdentifier> content is hard-coded for the Bodleian, and this (along with various other easy-to-change aspects) would have to be modified for use in migrating other institution's EAD records.

4.4 Proofreading conversion output

Another method of validating the output of migration tools is to assiduously proofread the input against the output looking for inconsistencies in migration. This can be costly and time consuming, and is prone to human error. In an extremely large collection of records it might be functionally impossible to proofread every record carefully. In these cases the generation of proofreading lists as suggested above can help to pinpoint areas that need improvement, but more often a sampling technique to highlight problems can be employed. Such a method would randomly select a statistically significant proportion of records and proofread them. When generalised mistakes were found these would be searched for or corrected in the migration tool itself not the output files as the migration would then be re-run to generate a new set of output files. If too many errors are found that are able to corrected reliably in the migration tool, then the proofreading sample size will have to be increased or the migration methodology reconsidered.

In cases where the input format and output format differ significantly, more proofreading will be necessary. With the MASTER case study, the formats are largely similar since MASTER lead directly to the development of the TEI P5 module for manuscript description, and thus

while proofreading was deemed necessary it was carried out on a fairly small random sample of output files. Conversely, with the EAD case study, although still an XML format, it differs significantly from the ENRICH specification and thus more proofreading was necessary. In this case a small number of records were provided to begin with, and then after the initial migration results had been proofread another batch of records were added, mined for differences, the migration tool improved, and then the new set of records proofread again. The proofreading was done both by the creator of the migration tool and a manuscript cataloguer from the Bodleian familiar with these records. This was repeated a number of times and will continue as the Bodleian wishes to migrate more records. Eventually a larger collection of records converted from the Bodleian's EAD will be released to the public.

4.5 Evaluation of migration case studies

External evaluation of the migration case studies produced by the ENRICH project is being undertaken as a part of the project. However this will commence at the same time as this report is due, so the results cannot be included in the report. The online version of this report will include these results when they are available, and this section will be updated.

5 Conclusions

One of the reasons for undertaking the migration tools case studies was to suggest recommended methods for ENRICH partners to migrate their legacy data to the ENRICH specification. The migration route chosen depends entirely upon the nature of the legacy data and the technical expertise and other resources available at the partner's institution.

5.1 Benefits and limitations of self-guided migration

The migration tools that have been developed are suitable for the conversion of the testbed to the ENRICH Specification. It is impossible to develop migration tools that will function for all input data, and thus the right route is the creation of tools such as XSLT stylesheets which can be easily modified to cope with local needs. These are flexible enough that they can be modified to cope with most divergences from published standards or specific encoding decisions undertaken by a particular archive. However, this requires either sufficient technical expertise to be available as a resource in the institution wishing to migrate to the ENRICH Specification or resources to employ someone to customise the migration tools to the archive's needs. While it is possible that properly prepared records will convert seamlessly to the ENRICH Specification, it is thought unlikely that this will happen without some degree of customisation on an archive by archive basis.

5.2 Benefits and limitations archive-specific mitigated conversion

The ENRICH project is able to undertake a reasonable amount of archive-specific migrations on behalf of ENRICH Content Partners during the length of the project. This has the benefit of being able customise the migration route for the specific archive's needs so that no intellectual content of the original records is lost in migration to the ENRICH Specification. It will be expected that a sample set of data and corresponding documentation will be provided to begin with, and then the results proofread by someone at the institution who is familiar with the input data. After that a larger amount of data will be migrated and it is expected that the same person will proofread a statistically significant randomly-selected sample of the data. The archive will be provided with a copy of the customised migration tool in case it wishes to migrate more records later. While such support can be provided for ENRICH Content Partners during the term of the project, it cannot be relied upon to continue after the project. Similarly, we are happy to provide best-effort advice to those who are not part of the project to a certain level, but the development of customised migration tools would

only be done on a proper consultation basis. Please contact enrich@oucs.ox.ac.uk to discuss your migration needs.

5.3 The case for the ENRICH Garage Engine

The ENRICH Garage Engine (EGE) is a migration tool technology which was not conceived for the original Description of Work for the ENRICH project but is now being developed by the PSNC ENRICH Partner as part of WP3. This tool will be a Java-based migration engine accessible in a number of different manners, such as through a web-form or directly via a REST API, which will facilitate the migration of legacy data through a number of different formats. Partly it builds on work that the TEI has done for the International Standards Organization (ISO), in seamlessly round-tripping ISO standards documents from Microsoft Word to TEI P5 XML and back again. The EGE will allow users to submit a document for conversion to another format, and it will be analyzed, recognized as a particular type, converted, validated, and returned in a user-friendly manner. This builds upon a number of existing stylesheets and conversions to allow conversion through multiple formats using TEI P5 XML as an intermediate format. Pending decisions made in the creation of the EGE it may also be possible for users to provide customised converters before or after any step in the migration process to accommodate specialised migration needs or standardisation of their data to the expected input format.

5.4 Recommendations for migration to the ENRICH Specification

The migration case studies undertaken lead us to make a number of recommendations for successful migration, some of which may be applicable to other forms of migration:

1. If possible use technologies that are mature, open source, cross-platform, human-readable, text-based scripting languages with well-developed support options.
2. Methodology for migration should be modular and take multiple forms, at least building both against the specified data format and a testbed representative sample of the data to be migrated.
3. Additional testing of the output should be done by targeted searches of the output data and proofreading a statistically significant randomly-selected sample. Any errors should be corrected in the migration tool and the conversion re-run from the start.
4. With migration to the ENRICH specification there are three approaches:
   - Archive-specific migration route: this is best done with human interaction customising the available scripts to the specifics of the data format. ENRICH partners can contact enrich@oucs.ox.ac.uk to discuss the migration needs. Non-ENRICH partners can also contact us as above, and we will attempt to assist on a best-effort (or optionally consultation) basis.
   - Self-guided migration route: those with sufficient XSLT experience available to them can use or modify for use the migration tools provided. They are available under a Creative Commons Attribution license and so freely able to be used and modified.
   - ENRICH Garage Engine migration route: the project will be producing a web application to enable migration through multiple formats. If you are interested in that, check the website once it has been released.
5. The process of migration chosen should be publicly documented and this documentation stored alongside the migration tools and input and output formats.

It is hoped that these case studies on the development and validation of migration tools will be of benefit to those undertaking migration to the ENRICH Specification, as well as those migrating to and from other formats.