Migrating Language Resources from SGML to XML: the Text Encoding Initiative Recommendations

Syd Bauman

Women Writers Project
Brown University
Providence, RI USA

Alejandro Bia

Dept. of Computer Languages and
Information Systems
University of Alicante
Alicante, Spain

Lou Burnard

Oxford University Computing
Services
Oxford University
Oxford, England

Tomaž Erjavec

Department of Knowledge
Technologies
Jožef Stefan Institute
Ljubljana, Slovenia

Christine Ruotolo

University of Virginia Library
University of Virginia
Charlottesville, VA USA

Susan Schreibman

Maryland Institute for Technology
in the Humanities
University of Maryland
College Park, MD USA

Abstract

The Text Encoding Initiative (TEI), established in 1987, has been the largest effort in the area of standardisation of computer encoding of language resources. TEI chose SGML (Standard Generalized Markup Language) as its underlying standard, and in the years before the inception of XML, a number of projects encoded their data according to some SGML DTD, TEI compliant, or otherwise. These projects could now benefit from migrating their data to XML. Apart from validation, the most compelling reason for migration is the scarcity of SGML-aware software and the abundance of XML-based tools and related recommendations. However, despite the fact that XML is a subset of SGML, migration is not a trivial process, especially in the case of large holdings of legacy language resources. This is why in 2002 the TEI Consortium established a Task Force on SGML to XML migration. The TF has now produced a number of reports that simplify and make explicit the conversion of SGML TEI (version P3) to XML TEI (version P4) documents. The reports are also relevant for a general audience of SGML users that are considering migrating their language resources to XML. This paper presents the recommendations made by the TF, concentrating on strategic considerations, the practical guide, and one case study, the conversion of the British National Corpus.

1. Introduction

The computer encoding of language resources has been, since the late 1980s, subject to an increasing amount of standardisation. The largest effort in this has been the Text Encoding Initiative, established in 1987. TEI became the only systematised attempt to develop a fully general text encoding model and set of encoding conventions based upon it, suitable for processing and analysis of any type of text, in any language, and intended to serve an increasing range of existing (and potential) applications and uses.

While TEI has been used in a number of other areas, it has also been influential in encoding language resources, especially corpora. Here it has been used either directly (for example the British National Corpus), or via its derivative, the EAGLES Corpus Encoding Standard, CES (Ide, 1998).

Both TEI and CES chose as their underlying standard SGML, Standard Generalized Markup Language (ISO 8879:1986). In the years before the inception of XML, eXtensible Markup Language (W3C 2000), a number of projects have encoded their data in these standards, or according to some other SGML DTD, and could benefit from migrating their data to XML.

Migrating SGML resources to XML provides a number of benefits. Many projects have been working with the same SGML DTD for many years and may need to re-examine it. Migration provides an opportunity to revisit DTDs and encoding practices which were developed to facilitate searching or display in a particular SGML-based system but are no longer necessary in an XML-based system. It also creates an opportunity to parse data again and fix errors.

Apart from validation, one of the most compelling reasons for a project or individual to consider migrating data is the scarcity of SGML-aware software and tools and the relative abundance of XML-based tools. Indeed, as XML becomes the industry standard there is a real danger that SGML-aware software will no longer be supported. SGML also lacks a suite of related standards that allow full exploitation of the encoded data. XML, on the other hand, is accompanied by a number of related standards and specifications, e.g. XML Namespaces, XPath, XSLT, XML Schemas, XPointer, XLink, XQuery, etc.

However, despite the fact that XML is a subset of SGML (Clark 1997), migration is not a trivial process, especially in the case of large holdings of legacy language resources. Furthermore, because of the fact that XML is a subset of SGML, XML is more restrictive which makes it easier to use, but this added restrictiveness is one of the reasons why migration from SGML is not a trivial process.

Such a process demands the consideration of both technical and strategic issues.

In the case of the TEI Guidelines, the issue of enabling a smooth transition between SGML and XML is especially pressing, since future releases of the TEI Guidelines will no longer be SGML compliant. The first release of the Guidelines, the SGML-based TEI P3 (Sperberg-McQueen and Burnard 1994) has been now superseded by the XML-based TEI P4 (Sperberg-McQueen and Burnard 2002), which still maintains backward compatibility with P3, and hence SGML. Thus the conversion from P3 to P4 is relatively straightforward, while the ongoing development of P5, the next generation of the Guidelines, will render P3 increasingly obsolete. TEI P5 will be XML-based and will not ensure backward compatibility, so a P3 to P5 migration may be substantially more difficult than P3 to P4. Therefore having TEI P4-conformant XML texts will make life simpler should a P5 migration become necessary.

This is the reason why in 2002 the TEI Consortium established a Task Force on SGML to XML migration. The TF has produced several reports that simplify and make explicit the conversion of SGML TEI (P3) to XML TEI (P4) documents and DTD extensions. These reports are relevant not only for TEI users but for a general audience of SGML users, especially those who hold repositories of SGML encoded language resources.

The TF work is now completed, and the following reports are available in their final form:

- TEI MI W02: *Strategic considerations in migration of TEI documents from SGML to XML;*
- TEI MI W03: *Practical Guide to migration of TEI documents from SGML to XML;*
- TEI MI W04: *Technical Checklist for TEI/SGML documents*, the purpose of which is to compare and classify sample TEI/SGML documents and their properties for the purpose of XML migration;
- TEI MI W06: *Migration case studies* for nine projects: British National Corpus, MULTEXT-East Multilingual Corpus, Corpus of Middle English Prose and Verse, Japanese Text Initiative, Women Writers Project, Thomas MacGreevy Archive, Documenting the American South, Victorian Women Writers Project, and the Thesaurus Musicarum Italicarum.

These documents, together with the samples directory, software tools folder, and meeting minutes, etc., are available on-line at http://www.tei-c.org.uk/Activities/MI/

These guidelines should significantly simplify the migration process for all TEI encoded language resources, and the migration of other SGML-based encodings.

The rest of this paper discusses the most important of these reports, MI W02 and W03, and one migration case study, the British National Corpus.

2. Strategic considerations

The TF report TEI MI W02 “Strategic Considerations in Migration of TEI Documents from SGML to XML” is intended for administrators and project managers; it discusses migration issues from a managerial perspective, with an emphasis on planning and decision-making.

The document contains the following sections:

- *Motivation, Opportunities, and Challenges* discusses some of the many excellent reasons for migrating legacy SGML data to XML, and argues that conversion is well worth the effort despite the challenges involved.
- *Areas of Migration* describes the components of a document production environment — document instances, DTD and extension files, catalog files, and the processing environment — and outlines in general terms how each area must be addressed in a migration to XML.
- *General Recommendations* describes the migration planning and workflow design process. It suggests strategies for analyzing legacy SGML data, allocating resources for migration, automating the conversion, and verifying the results.
- *Special Considerations in Migration* discusses different degrees of migration complexity, from easy conversions that aim for simple XML conformance to more robust conversions that look forward to advanced XML functionality and future versions of the TEI Guidelines.
- An appendix, *Potential Impact of Future Versions of the Guidelines on Migration Issues*, describes some of the changes that are likely to appear in P5, the next iteration of the TEI Guidelines, and how the anticipation of these changes might impact a project's migration strategy.

3. Practical Guide to Migration

The TF report TEI MI W03 “Practical Guide to Migration of TEI Documents from SGML to XML” is a technical report that describes the mechanics of conversion in greater detail, providing solutions to specific conversion problems as well as a recommended conversion workflow, and it is written primarily for the technical staff who will implement the conversion. Its specific recommendations are augmented by a set of Migration Case Studies that outlines individual migration efforts undertaken by members of the TF.

Data migration involves several distinct steps, which include converting document instances from SGML to XML, obtaining an XML DTD, and modifying the processing environment (including catalog files and applications such as parsers and editors) to accommodate XML. Because instance conversion is often the most substantial part of the migration process, the bulk of the MI W03 report discusses this topic.

Specifically, the first section presents a recommended workflow for instance conversion, while the second section discusses conversion tools. The third section discusses conversion of SDATA entities, which can be one of the trickier aspects of instance conversion. The final section of the report provides a tutorial in converting DTD extensions to XML. Each of these topics is covered below in more detail.

3.1. Migration workflow

This section discusses recommended procedures for migrating data from SGML to XML. It focuses mainly on a schematic workflow for individual document instances but also briefly addresses some other considerations related to processing environment and DTDs.

In migrating document instances from SGML to XML, the first stop is to convert the documents to well-formed XML. It will also probably be necessary to normalize tag case, since XML is case-sensitive and the SGML environment may not have been. It may be also useful to format the files to make them easy to read. Finally, the report strongly suggests, procedures for checking the results for any bugs that may have been introduced during the migration process. The report also covers in detail migration issues in relation to DTDs, as well as the processing environment, including editing tools, parsers, transformation engines, stylesheets, and catalogs.

The recommendations to this point are applicable to any SGML to XML migration; however, the discussion on migrating DTDs is TEI specific. For those using an unextended view of the TEI DTD, the procedure of moving from P3 (SGML) to P4 (XML) is straightforward, as care has been taken to make P4 backward compatible. For those that used local extensions, the extension files will need to be migrated manually. For simple extensions like deleting elements, renaming elements, and constraining attributes, the migration is also relatively straightforward; for complex extensions that manipulated the class system or made extensive changes to content models, migration can be difficult. For those who generated a "compiled" or "flattened" one-file DTD with the Pizza Chef (a web based form that, given a particular combination of TEI modules and possibly local extensions, generates a one-file DTD), the migration is very easy once the local extension files, if any, have been migrated, so long as the parameters entered into the Pizza Chef to generate the P3 DTD are remembered.

3.2. Instance Conversion: Tools

The most widely accepted tool in SGML to XML instance conversion has always been James Clark's C++ program sx, part of his (no longer maintained) SP package. It is, in fact, so widely accepted that there are few other widely available general purpose conversion tools. Therefore, this section addresses SGML to XML conversion issues using osx, the improved version of sx, which is maintained as part of the SourceForge OpenJade project.

Some post-processing tools are also discussed, in particular HTML Tidy, xmllint, and, a tool specific to TEI, Sebastian Rahtz's tei2tei.xsl stylesheet, which normalises case to TEI's “camel case” convention, e.g. *<sourceDesc>, <tagUsage>.*

3.3. Handling SDATA entities in the conversion process

SDATA entities are “specific entity references” which were available in SGML, but do not exist in XML. This section of MI W03 gives some simple recommendations for handling them in the migration.

In the SGML world, SDATA entities have been used mainly to provide a handle to characters that were not available in the coded character set used by a document instance. A number of public entity reference sets have been published by the ISO as an informative appendix to the SGML standard. In these sets, each entity declaration usually takes a form similar to

<!ENTITY amacron SDATA "[amacron]">

This provides the parser with a string that is algorithmically derived from the entity name. Many SGML applications take this kind of string and map it to the information that the application needs to handle such a character.

In XML the document character set can always be, and generally is, Unicode (ISO 10646). Most of the characters listed in the ISO public entity sets can be found among the over 100 000 characters available in Unicode. The report discusses both how to find the corresponding Unicode character for an ISO entity name, and (if found) how to map the name to the character.

Conversion of SDATA entities representing characters that exist in Unicode is the simplest case. Usually, it will require replacing the value of the SDATA entity replacement with the appropriate Unicode value. The tool-dependent methods to achieve this are explained in the report.

If no Unicode representation can be found for a character, the remaining possibilities for this conversion are to assign code points from the private use area of Unicode (PUA) or to use markup constructs to represent these characters; both options are discussed further in the report.

3.4. Migrating TEI DTD extensions to XML

If the elements or content models that the TEI provides don't quite meet the requirements of a specific project, the DTD can be modified in a number of well-defined ways and the documents will still remain “TEI conformant.” This modification involves creating two extension files, setting some parameter entities, possibly defining new elements or redefining existing ones, and making these modifications known to the parser in the DTD subset at the beginning of the document.

This section is for projects that have modified the TEI DTD in this manner, and want to migrate these modifications from SGML to XML (i.e., want to use the XML-based P4 DTD with equivalent modifications). It begins with some general remarks, then describes a sample DTD modification that covers the most important issues, then outlines a recommended migration procedure and demonstrates the key steps using the example.

4. Migration case study: British National Corpus

The British National Corpus (BNC) is a 100 million word snapshot of British English taken at the end of the 20th century. It contains 4130 distinct texts, sampled from a very wide range of materials both spoken and written. It is richly annotated in SGML, with markup of a wide range of structural features, and associated metadata, as well morpho-syntactic tagging down to the individual token level. The BNC has its own DTD, using the TEI prose base, the corpus additional tagset, and a number of modifications to the basic TEI model, as described in the Users Reference Guide. The most recent edition of this Guide includes a section on TEI conformance which explains in excruciating detail the TEI Extension files used to define the BNC DTD.

The tagging makes heavy use of SGML minimization features, notably for part of speech (POS) coding. For example, here is a heading at the start of text A1L: *<head type=MAIN><s n="1"><w VVG-AJ0>Ripping <w NN2>yarns <w CJC>and <w AJ0>moral <w NN2>minefields<c PUN>: <w NP0>Allan <w CJC>and <w NP0>Janet <w NP0>Ahlberg <w NN1-VVB>talk …*

We have successfully completed conversion of a 4 million word subset of the BNC, and documented the procedure for use by other BNC licensees; we give below the two steps in the conversion, namely converting the DTD and converting the SGML files.

4.1 Translation of the DTD from SGML to XML

The TEI website currently provides a web-based utility (The Pizza Chef) which can be used to create customized versions of any TEI-conformant DTD in SGML or XML form. We used this to provide an initial XML version of the original SGML BNC DTD, using the procedure described in the Migration report at http://www.natcorp.ox.ac.uk/migration.html.

The only significant problem we encountered arose from the extensive use of specific default attribute values in the original SGML DTD. The original encoders of the BNC often did not realise that a defaulted attribute value was not necessarily the same as a null attribute value. If the DTD includes a default value for some attribute, then that value is actually present in the XML document instance, thus adding considerably prolixity to our final output. We therefore revised the DTD, defaulting all attributes to #IMPLIED rather than supplying explicit values for them.

The BNC DTD originally declared several thousand SDATA character entities to represent non-Latin characters, typographic symbols, and a variety of other characters. We mapped all but 5 of the entity declarations to an equivalent Unicode character; the exceptions were characters representing three mathematical fractions (1/7 1/9 and 4/7), one representing the character "/" when used to separate shillings and pence in old money, and the last had been used to represent any omitted mathematical formula. Appropriate equivalents for these were not hard to define.

It is probably worth recording also that the process of converting the DTD and revalidating the data against it brought to light a number of serious tagging errors in the original, largely due to the use of omitted end-tags. Although SGML minimization features reduce the apparent complexity and verbosity of an encoded text, they may do so at the price of obscuring serious errors in it. This is particularly the case where they are used in conjunction with inclusion exceptions. For example, in the original TEI P3 (SGML) DTD, a number of elements, such as <lb/> and <pb/> were declared as global inclusions. These elements were therefore legal inside the content model of all elements, even those (such as <w>) declared to have only #PCDATA content. In the document instance, a sequence such as *<w>word1 <lb> <w>word2* would therefore be interpreted, when converted to XML, as *<w>word1 <lb/></w><w>word2</w>* rather than as *<w>word1 </w><lb/><w>word2</w>*.

4.2 Translation of BNC documents

We were able to carry this process out entirely automatically, using osx. The conversion was done using a shell script which carried out the following steps:

(a) extract a filename from the BNC user file identifier;
(b) produce a wrapper file which can be submitted to an SGML parser;
(c) run OSX on this file, with parameters which retain both internal and external entity references;
(d) run an XSLT transformation to 'pretty print' the XML file generated in the previous step.

The BNC consists of over four thousand files, and we did not wish to process all of it in one pass, even supposing we could do so with the hardware at our disposal, since this would have produced a single monster XML output file. The shell script therefore operates on one file at a time. Each file has a 3 character identifier which has to be mapped to the directory structure used to store it. To parse it as a valid SGML document, each text file needs to be embedded within a structure that includes invocation of the relevant declaration files and which also includes the corpus header. The other components of this structure are of course common to each case: we therefore represent them as SGML external entity references and instruct osx to include them as references only in the output. This means that, in the final stage, we have a single XML entity which contains only the text being processed. This final stage is carried out by the XSLT processor xsltproc, which does not need any of the declarations necessary for the previous steps: its job is to replace named character entity references by appropriate Unicode values and to reformat the XML text for better readability.

While the XML files are significantly larger than the original minimised SGML files, the increase in size is far less significant (between 1.15 and 1.26) for the compressed files than it is for the uncompressed files (where the factor is a fairly steady 1.8), because of the repetitiveness of the XML encoding.

5. Conclusions

The paper has presented the reports of the TEI TF on SGML to XML migration, which provides detailed instructions for migrating TEI P3 (SGML) documents and DTDs to XML TEI P4 (XML). The reports are meant primarily to serve TEI P3 resource holders, however, they are, in the main, relevant for any SGML to XML conversion project.

The reports are available on the TEI Consortium Web site, at http://www.tei-c.org.uk/Activities/MI/

Acknowledgements

The authors would like to thank the other members of the TF for their contribution: Jessica Hekman, Tobias Rischer, Chris Powell, Natasha Smith, John Walsh, and Franz Wiering. The work reported here was supported by grants from the National Endowment for the Humanities, an independent grant-making agency of the United States.

References

Clark, J. (1997). Comparison of SGML and XML. World Wide Web Consortium Note 15-December-1997. http://www.w3.org/TR/NOTE-sgml-xml-971215

Ide, N. (1998). Corpus Encoding Standard: SGML guidelines for encoding linguistic corpora. In *First International Conference on Language Resources and Evaluation, LREC'98*, (pp. 463-470). ELRA. http://www.cs.vassar.edu/CES/

ISO 8879:1986. Information processing -- Text and office systems -- Standard Generalized Markup Language (SGML).

Sperberg-McQueen, C. M., Burnard, L. (eds.) (1994). TEI P3: *Guidelines for Electronic Text Encoding and Interchange*. Text Encoding Initiative.

Sperberg-McQueen, C. M., Burnard, L. (eds.) (2002). TEI P4: *Guidelines for Electronic Text Encoding and Interchange, XML-compatible edition*. The TEI Consortium. http://www.tei-c.org/P4X/

W3C (2000). Extensible Markup Language (XML) 1.0 (Second Edition). http://www.w3.org/TR/2000/REC