[MIMAS logo]"epub@mimas"

The SuperJournal Project: Data Handling Using SGML

Ann Apps and Ross MacIntyre
Manchester Computing, University of Manchester,
Oxford Road, Manchester, M13 9PL, UK
Email: ann.apps@man.ac.uk, ross.macintyre@man.ac.uk

Publication information.

Abstract

The SuperJournal research project evaluated the usage within UK academia of a set of electronic journals made available through a bespoke electronic journal application via the World Wide Web. This paper describes the data handling processes which comprised the production system required to import publisher supplied journal data into the SuperJournal application. In particular, it focuses on the exploitation of SGML to implement scaleable data handling processes. Data supplied to SuperJournal encompassed multiple publishers' SGML DTDs. Also described is journal article metadata display, and enhanced end-user functionality in electronic journal articles such as "forward" and "backward" citation reference chaining, again implemented via SGML translation.
Keywords: SuperJournal; electronic journals; SGML; article metadata; bibliographic references; citation linking.

Introduction

The SuperJournal research project [1] studied factors which will make electronic journals useful and successful within the academic community. The three year project (1995-1998) developed an electronic journal application making available, via the World Wide Web to particular University test sites, 48 peer-reviewed academic journals, within four subject clusters, provided by major journal publishers (Elsevier, Springer-Verlag, Macmillan, et al). The SuperJournal project was funded by the Joint Information Systems Committee (JISC) of the UK Higher Education Funding Councils, as part of its Electronic Libraries Programme (eLib) [2].

The SuperJournal application, including the production and data handling processes, was developed by Manchester Computing [3] at the University of Manchester, UK. Evaluation of the use of SuperJournal by academic researchers was undertaken by the HUSAT Research Institute [4] at Loughborough University, using the extensive usage statistics logged by the SuperJournal application as a foundation.

The SuperJournal application was controlled by an object database (Fujitsu/ICL ODB-II) which captured the articles' metadata, the actual full articles being held within a directory structure according to a naming convention. Article discovery for end-users was either by browsing through the journal/issue hierarchy or by using one of the three provided search engines.

Journal data delivery was either PDF full articles, with metadata (header) information in SGML mark-up [5,6], or full text SGML, mostly with accompanying PDF. Reflecting each publisher's choice of SGML Document Type Definition (DTD), there were ten different article header DTDs and five different full

article DTDs.

The SuperJournal application required a common article header information format for data capture, the obvious choice being SGML. Thus a common DTD, the SuperJournal Header DTD, was defined. SuperJournal also developed a Generic Header DTD, encompassing all the publishers' DTDs.

A SuperJournal Full Article DTD provided a common format for the full article SGML processing, allowing all supplied full text SGML to be processed by the same data conversion programs following the initial conversion to Full SuperJournal SGML. Full article conversion provided extra functionality to the end-user in addition to HTML article display.

The three SuperJournal DTDs were defined following detailed analysis of the DTDs used for data supply, but also included elements to support the SuperJournal application's functionality. They were developed specifically to provide a common data format within SuperJournal, because of the multiple DTDs used for data supplied to SuperJournal. They were not intended to be exemplary article header or full article DTDs.

Further SuperJournal DTDs defined tables of contents at various levels within the journals hierarchy.

SuperJournal Data Handling Process

During the SuperJournal Data Handling Process supplied header SGML was translated into SGML conformant with the SuperJournal Header DTD by parsing against the SuperJournal Generic DTD, using "sgmls" [7], a freely available validating SGML parser, followed by conversion of the result of this parse to SuperJournal Header SGML, with a C++ program.

Each publisher's supplied full article SGML was translated into SuperJournal Full Article SGML by a separate OmniMark program. OmniMark [8] is an SGML aware, 4th generation programming language which includes powerful pattern analysis and manipulation facilities. SuperJournal full article SGML was then translated into HTML for end-user display and the article metadata was extracted in SuperJournal Header SGML format for SuperJournal application capture, both by OmniMark programs. Further processing, including image manipulation [9], provided extra end-user functionality including: full size figure display via thumbnails in the HTML; article figure thumbnail indexes; display of an article's bibliography in a separate web browser window; hypertext linking from an article's citation references to an appropriate public abstract database; and linking within SuperJournal, both "forward" and "backward", between references and cited articles.

Tables of contents at various levels within the journals hierarchy were created and updated during the SuperJournal Data Handling Process for each new journal issue, resulting in a complete SGML (actually XML [10]) catalogue of the journal data held by SuperJournal. This journal data catalogue was used both for ascertaining references to cited articles within SuperJournal during the data conversion process, and for generating SuperJournal usage statistics from information logged by the SuperJournal application.

A graphical representation of the SuperJournal Data Handling Process is shown in Figure 1.

[process diagram (7772 bytes)]

Figure 1. The SuperJournal Data Handling Process.

DTD Analysis

The initial phase of the SuperJournal DTD definition involved analysing the DTDs used by the publishers involved in SuperJournal for the supplied journals. The detailed result of this DTD analysis was documented in spreadsheet form for input to the definition of the SuperJournal DTDs and for supply to project members.

Most publishers used their own DTD, though some use the Majour [11] or SSSH [12] DTDs for their article headers. In fact, many of the DTDs were customisations of Majour/SSSH or the Elsevier DTD [13], which assisted the process of DTD analysis. But some publishers have developed their own very specific DTD. Some DTDs are very verbose and have been written to include every possible different case which makes the SGML much more difficult to process. Some publishers include print formatting information, which theoretically should not be included in the SGML (which should define document structure). Some DTDs, which appear to have been developed by typesetters who create SGML for several publishers, allow several different means of capturing the same information. At the other extreme, SuperJournal was supplied with very simple DTDs: one specifically for SuperJournal article header supply; and one very minimal full text DTD, though with special enhancements for SuperJournal. Some publishers chose to use the SuperJournal Header DTD, either temporarily or for the duration of the project, for article headers supplied to SuperJournal. For completeness, the SuperJournal DTDs were included in the DTD analysis. Also included in the header data analysis was Dublin Core [14], whose primary use is for Metadata specification, to explore the possibility of the use of Dublin Core for journal article metadata capture.

During the course of the SuperJournal project, new journals with different DTDs were included. Also, the DTDs for existing journals were changed or updated, in some cases changing from article header SGML to full article SGML. The DTD analysis evolved during the project to accommodate these new and updated DTDs.

It was found that, despite differences in tag naming and DTD structures, there was a great deal of commonality amongst the information included in the DTDs. Where the information diverged was generally where publishers had included elements for their own use, particularly during their publishing production processes or for print formatting, information which was of little interest to SuperJournal. There were differences in the levels of structure within the header DTDs, ranging from largely flat structures to much deeper ones.

However, it was discovered that in some DTDs significant information necessary to SuperJournal, such as publisher name, ISSN, or cover date, was missing or optional. Most of this information is deducible or, in the case of cover date, can be included during the data conversion process, but it would seem preferable for all article SGML, both header and full text, to be self-identifying. In some cases, although the information was missing from the DTD, it was included in the actual data supplied, but there were some instances where full article SGML was supplied without page numbers, which are not derivable. Generally, the DTDs where this type of header data was missing or optional were the DTDs where this information is supplied as attributes on the main "article" tag, rather than in separate tags. SuperJournal had a particular problem, which required manual intervention, with publisher supplied article headers where author names were given as a simple text string rather than separately tagged authors with separated name elements.

If SuperJournal were to make recommendations on article header DTD design, following this DTD analysis, they would include:

- Make the data completely self-identifying to allow for any future use of the data in other applications. This should include: Journal title (or journal identifier); Publisher name; ISSN; Copyright text; Volume number; Issue number; Cover date; Article title; Article first page number; Article last page number.
- Include some structuring and multiple elements where necessary, but keep it to a minimum, for example allowing multiple title groups consisting of multiple titles seems rather excessive.
- Within an author group, include each author separately, with separate surname (family name) and first name fields (possibly first name could be optional). Include tagged references between authors and their affiliations, rather than capturing this information by multiple author groups.
- Make SGML tags mandatory where the information or structuring is a requirement, for example an optional article title does not seem sensible.
- Capture dates as separate fields (year; month; day) rather than just text strings, because there is a multitude of ways to format dates.

Generic SuperJournal DTD

The Generic SuperJournal Header DTD encompasses most of the SuperJournal publishers' DTDs, the ones not included being those supplied later in the project. All the supplied header SGML which conforms to the included DTDs should parse against this DTD. Parsing against this generic DTD validated the supplied SGML and produced a data format suitable for entry to the SuperJournal header SGML generation program. For journals whose SGML was not processed by this route, conversion to SuperJournal SGML occurred first, the SuperJournal Header DTD being included in the generic header DTD for completeness.

Defining the generic header DTD was not essentially difficult, but rather extended because of the large number of different possibilities catered for. It was simply a matter of analysing the content of each DTD, and then reflecting its tag names and structure within the generic DTD. The only slight problem occurred where the same name was used by two DTDs for two completely different constructs, for example "<issue>" is used in the Majour DTD to introduce a structure of journal issue information, whereas in another DTD "<issue>" is simple unstructured data containing the issue number.

Because of the size of the resultant Generic SuperJournal Header DTD, it was not easy to maintain or change. This is one reason why DTDs supplied later in the project were not included, the other reason being the change in the data conversion route for later supplied and full text SGML journals after the OmniMark language became available to the project.

SuperJournal did not attempt to develop a generic full article DTD. It was not required for the full article SGML data conversion route, which used a separate OmniMark data conversion program for each publisher's DTD. Although there is much overlap between the content of the various publishers' full article DTDs, which means that a generic full article DTD would be possible, it would have produced a much larger and more unwieldy DTD than the generic header DTD.

SuperJournal Header DTD

The SuperJournal Header DTD was developed to provide a common format for article header data capture by the SuperJournal application. It was not intended to be an exemplary serials header DTD, although some publishers chose to supply their article headers to SuperJournal in this format.

From the header DTD analysis, the significant elements and structures of article headers were identified. In addition, the data content requirement for the SuperJournal application, was considered. Using these elements, the SuperJournal Header DTD was defined. In general, SGML elements which were specific to a particular publisher's production process were not included. The exceptions were some publishers' identifiers and types, which were captured by the SuperJournal application but not displayed on to the end-user.

Particular details to be noted about the SuperJournal Header DTD are:

- Mandatory elements in the DTD for data loaded into the SuperJournal application are: <jtl> Journal title (or journal identifier); <pnm> Publisher name; <issn> ISSN; <crn> Copyright text; <vid> Volume number; <ino> Issue number; <cd> Cover date; <atl> Article title; <ppf> Article first page number; <ppl> Article last page number. Where these elements were not included in the data supplied by the publisher they were either deduced by the SuperJournal Header SGML generation program (eg. Publisher name; ISSN; Copyright) or included as an option to a pre-processing script (eg. Cover date; Volume and Issue number).
- Some elements in the SuperJournal Header DTD were included during data conversion for SuperJournal application data capture. Most of these elements would not be included in any publisher supplied data. They are: <cluster> the SuperJournal cluster to which the journal belongs; <sjaid> a unique SuperJournal identifier for the article; <medline> a Medline [15] identifier for the article if known and applicable; <pdf> and <html> the full article path to the PDF or HTML file of

the article.

- File sizes for full article files were included in the SuperJournal Header SGML file within comments. This information was included within a comment, rather than by the more appropriate method of introducing a new element into the DTD, in order not to make changes to the SuperJournal application data load at a late stage in the project. OmniMark provides the ability to process these comments as well as the SGML elements of the article header for HTML display.
- The SuperJournal application required author names, if present, to have both a surname <snm> and first name <fnms>.
- Author affiliations were captured as a single text string, even if they were split to a finer-grain in the publisher's DTD. It was decided that SuperJournal had no requirement for a finer-grain author address. The only exception to this was that any <email> or <url> tags were captured.
- Keywords were included in SuperJournal Header SGML as a single element containing a semicolon separated list of the keywords. With hindsight, this was possibly an incorrect design decision. The SuperJournal application data load had to split this list into its constituent parts, as did any other operation performed on this field. Keywords were tagged separately in the header part of the Full Article SuperJournal DTD.
- A minimal set of text elements is included in the SuperJournal Header DTD, mainly to allow text formatting within titles and abstracts. HTML tag names are used to simplify subsequent processing.

Article Metadata Display

An article's abstract and header information were displayed to a SuperJournal end-user using an OmniMark program which translated the article's SuperJournal Header SGML file to HTML "on the fly". Displaying article metadata dynamically allowed for the inclusion of hypertext links in the generated HTML which enhanced the functionality provided to the end-user. In addition to a link providing display of the full article, accompanied by an indication of file size, hypertext links were included where appropriate to download the article metadata in bibliographic format, and to view the article's references in a separate web browser window. An example of article metadata display as seen by the end-user is shown in Figure 2.

The same OmniMark script was used to display article metadata when a user followed a citation reference or a "cited by" link, and the display was similar.

Figure 2. Article Metadata Display.

[example (16290 bytes)]

SuperJournal Full Article DTD

The SuperJournal Full Article DTD was developed to provide a common format for subsequent full text SGML processing within SuperJournal. It was not intended to be an exemplary full article DTD.

From the full article DTD analysis, the significant elements and structures of articles were identified. Using these elements, the SuperJournal Full Article DTD was defined. As with the SuperJournal Header DTD, SGML elements which were specific to a particular publisher's production process were not included. The header part of the SuperJournal Full Article DTD is generally identical to the SuperJournal Header DTD, though there are one or two differences introduced to assist the data conversion process. Where appropriate, SGML element names within the Full Article DTD correspond to the equivalent HTML names, to ease subsequent processing, for example table and text elements.

Particular details to be noted about the SuperJournal Full Article DTD are:

- Article sections are nested, thus defining section levels within the article.
- Lists provided are the "ordered", "unordered" and "definition" lists of HTML. Most of the list types used within the publishers' DTDs map cleanly onto these types.
- Figures may include a "caption" and a "legend". Figure filenames followed a predefined naming convention, which could be deduced from an "identifier" attribute during data conversion.
- Tables may include a "caption" and a "legend". They may be supplied by the publishers as either separate graphic files or marked up in SGML. Tables supplied as external graphic files were treated in the same way as figures, with the filename deducible from an "identifier" attribute, and a "type" attribute specifying "external". Tables marked up in SGML have a "type" attribute "SGML". The elements of the table body in the SuperJournal Full Article DTD correspond to HTML table body elements and attributes. Generally tables in the publishers' DTDs mapped easily onto this definition, any information lost during this conversion being print-format related.
- Formulae may be supplied as either separate graphic files or marked up in SGML. Formulae supplied as external graphic files were treated in a similar way to figures, with the filename deducible from an "identifier" attribute. Formulae marked up in SGML presented display problems when converted to HTML. Within SuperJournal Full Article SGML, formulae were marked up utilising the text elements of the DTD and appropriate keyboard characters and SGML entities, by a "best attempt" approach. Some Greek characters were displayed using GIF images [16]. Although this improved the display of formulae it was not an ideal solution because no account could be taken of character size or font.
- Bibliographic references are tagged in a fine-grain way (see below).
- The DTD includes cross reference tags for: sections; citations; footnotes; figures; tables; formulae; lists. These are mapped from similar cross references in the publishers' DTDs.
- Text elements generally correspond to HTML text formatting tags, for ease of use in subsequent processing. Publishers' text elements were mapped onto these where possible. Generally the more esoteric of the publishers' text elements were concerned with print formatting and were ignored or mapped onto another type for HTML display.
- Mathematical text elements also correspond where possible to HTML tags. Other mathematical elements were mapped to appropriate keyboard characters and SGML entities by a "best attempt" approach.

Bibliographic References

Where an article's bibliographic references are marked up using "fine-grain" SGML tagging, processing of these references to increase end-user functionality becomes possible. If references are tagged in a "coarse-grain" way, as just a text string, it is more difficult to parse this text by an automatic program to identify its constituent elements. Though SuperJournal did investigate extracting and parsing references from PDF articles with some success.

Most of the full article DTDs analysed by SuperJournal included some fine-grain reference tagging. In many cases this tagging is optional and was not always reflected in the supplied SGML data, though this situation improved during the time of the project.

From the DTD analysis, the significant elements of a bibliographic reference were identified, and composed into a bibliography section within the SuperJournal Full Article DTD definition. A "text string" option is included to allow for cases where mapping of references in supplied article data was not possible, but this was expected to be a "fall-back" option rather than the norm. The ordering of the bibliographic reference elements is not defined, and they are all optional and repeatable. Generally the element names are different from the names of other similar elements in the DTD to distinguish them as bibliographic reference elements both for readability and to aid processing. Authors are separated and split into separate surname and first name (or initials) elements. Apart from the expected elements of a bibliographic reference two further elements are added during SuperJournal data conversion processing where applicable: <medline> an identifier within the Medline public abstract database [15]; and <sjaid> an internal SuperJournal identifier where the referenced work is held within SuperJournal.

Medline identifiers were added for references to articles in the appropriate subject area, by sending the reference to Medline by email in the required format. Medline's email replies were used to include the Medline identifiers in the SuperJournal Full Article SGML files.

SuperJournal identifiers were determined when a referenced article was in a journal and year range held by SuperJournal, using the SuperJournal journal data catalogues provided by the journal hierarchy tables of contents files. During the determination of this SuperJournal identifier, the referenced article itself was recorded as being cited, thus providing "forward" chaining as well as "backward chaining" citation links.

Bibliographic Reference Display

Where an article's bibliographic references were available, either because they were tagged in SGML or where they were extracted from the PDF article, SuperJournal provided the end-user with the option to view them in a separate web browser window. From these references hypertext links could be followed to: a referenced article's abstract in the Medline public abstract database for journals in the appropriate subject area, implemented by including the captured Medline identifier within the Medline URL; or to the metadata for the referenced article if it was held by SuperJournal and hence the article itself. Activation of the link to the referenced article within SuperJournal was implemented via a CGI-program ("Common Gateway Interface", a technique which allows users to run programs on a WWW server) which utilised the captured SuperJournal identifier. An example of these links within a reference as seen by an end-user is shown in Figure 3. Note that the reference shown in Figure 3 was extracted from a PDF article.

For articles which were recorded as being cited by another article within SuperJournal, the end-user was allowed the option of viewing the list of citing articles, and from this list viewing the article metadata for the citing article and hence the article itself.

Figure 3. Example Reference within an Article.

[example reference (4198 bytes)]

SuperJournal Tables of Contents DTDs

Further DTDs were defined by SuperJournal. These DTDs specify tables of contents files at each level of the "SuperJournal/journal/issue" hierarchy. There is also a DTD which specifies an article's "Cited By" list. It was decided to maintain this information in SGML format, and thus to define DTDs, because it is derived from information already held in SGML. SGML seemed the obvious format for capturing these tables of contents in a rigorous way, and it would simplify subsequent utilisation of these files especially if programs were written in OmniMark or displayed via an XML browser.

Dublin Core Metadata

SuperJournal included Dublin Core metadata [14], using HTML "<meta>" tags at the heads of the HTML articles generated from full article SGML and the journal hierarchy tables of contents files. This metadata was generated automatically during the data conversion process. Strictly, the inclusion of metadata here was unnecessary if the primary use for metadata is web information discovery. The articles within SuperJournal were readable only via the SuperJournal application following user login, and the publisher supplied SGML article header files already provided article metadata. But it was felt that Dublin Core metadata generation was a useful exercise which could provide possible future cataloguing of the data. The generation of Dublin Core metadata using RDF (WWW Resource Description Framework [17]) syntax was not attempted because this was an emerging standard still undergoing definition, and was too immature for use during the timescale of the SuperJournal project.

Conclusion

With hindsight, the original design of the SuperJournal data handling process, using a generic Header DTD and a generic transformation program to the SuperJournal Header DTD, was rather ambitious, though it appeared a good theoretical strategy. In practice, the generic Header DTD became unwieldy and increasingly difficult to maintain and change to incorporate new DTDs. For this reason, not all publishers' DTDs supplied to SuperJournal were included in the generic Header DTD. For a similar reason, a full article generic DTD was not defined.

The SuperJournal DTDs were defined primarily to provide common formats for articles and their metadata. They were born out of necessity because of the multiple DTDs used for the data supplied to SuperJournal. They were not intended to be exemplary header and full article DTDs, nor were they envisaged as competition for emerging standards such as SSSH [12] for serial headers or ISO-12083 [18] for full articles. However both of the SuperJournal DTDs are adequate for serial header and article SGML definition and some interest has been shown in their use. The SuperJournal Header DTD was used for data supply by some SuperJournal publishers either temporarily or for the duration of the project. The bibliographic reference section of the SuperJournal Full Article DTD has been utilised by another SuperJournal publisher. The Full Article DTD was supplied to a European University Library publisher who expressed an interest in using it. From experience gained during SuperJournal, Manchester University are now advising four publishers on their SGML DTD development.

The SuperJournal DTDs continuously evolved during the project. This evolution was necessary when new journals with new DTDs were supplied to SuperJournal or existing DTDs were changed, if this required the capture of new SGML elements, and also when enhancements to the functionality of the SuperJournal application implied the introduction of new elements.

One aim of the SuperJournal project was to produce scaleable data handling processes, because it was necessary to convert and load very large numbers of journal issues and articles with a minimum number of staff. But, at the same time, it was necessary to maintain some level of quality control on the data displayed to the end-user. To a large extent this was successful. Scaleable processes were essential because of the volume of data being handled, some scientific journals being weekly. The data handling process became unscaleable where manual input became necessary, the main reason for which was poor data quality, shown up by data quality checks in the various programs which comprised the data conversion process. Despite significant improvements in the quality of supplied data during the course of the project, it became apparent that the ideal of a single data conversion program, which would take electronic journal data as input and automatically load it into an application with no intermediate control and checking is not feasible.

References

[1] The SuperJournal Project. http://www.superjournal.ac.uk/sj/

[2] eLib (Electronic Libraries Programme of JISC). http://www.ukoln.ac.uk/services/elib (http://www.jisc.ac.uk)

[3] Manchester Computing (at The University of Manchester). http://www.mcc.ac.uk (http://www.man.ac.uk)

[4] HUSAT Research Institute (at Loughborough University). http://info.lut.ac.uk/research/husat (http://www.lboro.ac.uk)

[5] Goldfarb CF. The SGML Handbook. Oxford University Press.

[6] SGML. http://www.oasis-open.org/cover/sgml-xml.html

[7] sgmls (superseded by nsgmls). http://www.jclark.com/sp/nsgmls.htm

[8] OmniMark Technologies. http://www.omnimark.com

[9] MacIntyre R. Digital Publishing - Data Handling. State of the Art Report (STAR), Eurographics'97, Budapest, Hungary, September 1997.

[10] XML. http://www.w3.org/XML

[11] European Working Group for SGML (EWS) MAJOUR DTD. http://www.springer.de/author/sgml/help-sgml.html

[12] SSSH (Simplified SGML for Serial Headers) DTD. http://www.sgml.org.uk/sssh/

[13] Elsevier DTD. http://www.oasis-open.org/cover/elsevier-art-dtd.txt

[14] Dublin Core metadata. http://purl.org/DC/

[15] National Centre for Biotechnology Information, MEDLINE. http://www.ncbi.nlm.nih.gov

[16] Greek graphical symbols. http://www.anachem.umu.se/graphics/symbols/symbols.html

[17] Resource Description Framework. http://www.w3.org/RDF

[18] ISO 12083 Serial Article DTD. http://www.oasis-open.org/cover/gen-apps#iso12083DTDs

---

8 August 2002, epub@manchester.ac.uk

[Go to Electronic Publishing at MIMAS]Electronic Publishing [Go to MIMAS home page]Home Page [Valid XHTML 1.0!]