# Bridging Two Biomedical Journal Databases with XML – A Case Study

Glenn Pearson, Ph.D CS, and Chan Moon, MS CS
*Communications Engineering Branch (CEB),*
*Lister Hill Center for Biomedical Communication,*
*US National Library of Medicine (NLM); and MSD, Inc.*
*Glenn_Pearson@nlm.nih.gov* *MoonC@mail.nlm.nih.gov*

***Abstract***

*For transferring data between heterogeneous databases, XML is becoming a method of choice. While deployment is generally successfully, there are barriers to overcome. A recent example at NLM involved transfers between "DCMS" and "Mars". These two document workflow systems, with separate databases, process biomedical journals for incorporation into NLM's "MEDLINE". DCMS acts as an overall manager, while Mars is one of three decoupled methods for harvesting the abstracts and citation information from articles. A pre-existing flow of article information from the "back end" of Mars was first recast into XML for delivery to DCMS by an "Upload" daemon. Next addressed was the lack of data flow from DCMS to the Mars "front end", necessitating duplicative data entry by Mars operators. The desired flow was achieved as XML, mediated by a new "CheckIn" module. Bumps hit on the road to successful deployment of these transfer applications are discussed.*

## 1. Introduction to MEDLINE, DCMS, and Mars

One of the major accomplishments and services of NLM is the creation and maintenance of MEDLINE, a database of journal article citations and abstracts covering much of the world's biomedical literature. Originally for researchers, healthcare providers, and medical librarians, MEDLINE has been publicly available at www.nlm.nih.gov since 1998.

To support this endeavor, NLM subscribes to an extensive collection of biomedical journals. An arriving issue is first given a unique identifier (e.g., a barcode label for physical issues). It is soon processed into "DCMS", the citation Data Creation and Maintenance System of NLM's Office of Computer and Communication Systems (OCCS). This client/server system manages the scatter/gather of journal issues and associated data among three separate data-capture subsystems:

- A traditional off-campus "duplicate keyboarding" approach;
- The portion of the "PubMed" system of NLM's National Center for Biotechnology Information (NCBI) that receives journals electronically from publishers as SGML; and
- The in-house "Mars" (Medical Article Records System [1, 2] by CEB), our focus here.

Output of these systems, once received back in DCMS, are further verified and supplemented by Medical Subject Heading (Mesh) keywords, then placed into MEDLINE's extensive Oracle database, for web retrieval.

Mars workflow begins with a scan of the first page of every journal article. The images are fed to a full-page multi-engine optical character recognition (OCR) system. AI techniques locate textual zones and identify them as title, authors, and so on. Specialized dictionary lookup helps correct OCR errors, and certain fields (authors in particular) are automatically reformatted into a standard style. Duplicate-key "Edit" stations add supplemental information, e.g., GenBank numbers from unscanned pages. Finally, a "Reconcile" operator performs detailed final verification and correction.

## 2. Shipping "back end" Mars data to DCMS with "Upload"

1997 saw the birth of "Mars I", a system with neither database nor AI. Its output was generated, one file per journal issue, in a proprietary text-markup format, and uploaded periodically using ftp to the predecessor of DCMS, the mainframe-based "AIMS" system. From Mars, ftp was done with a batch file, usually invoked automatically at fixed times. But error detection and recovery was minimal, leading its replacement by a C++-based "Upload" module as part of the database-centric "Mars II" in 1998. Upload had a GUI, so that both on-demand and automatic periodic transfers could be accommodated. Ftp functionality was through class wrappers for wininet.dll, an installable component of Internet Explorer.

By 1999, legacy mainframe systems began to be phased out by OCCS in favor of a new "Integrated Library System". A year on, as part of this effort, DCMS became available, and with it the desire from OCCS to phase out the AIMS file format in favor of an XML one. Still retained was the ftp producer/consumer model of transport from Mars. Had the database systems on both ends been identical (e.g., both Oracle) then a direct SQL-to-SQL transfer might work, if versioning, hardware, and security issues didn't intervene. But Mars used MS SQL Server/NT Server, and DCMS used Oracle/Solaris, so file transfer avoids difficulties.

Thus, last winter, Upload was given a new feature (by author CM). Fresh citations retrieved from the Mars database (or, if necessary, from a plain ASCII AIMS file) could now be generated as a file in XML format. There were three requirements to satisfy:

- Compliance with a defined DTD, discussed below;
- Translation of all characters to the standard Unicode/XML "UTF8" [3] encoding;
- Screening of all characters against a "MEDLINE character database".

The detailed specification of these requirements, and their implementations in Upload and DCMS, described next, were refined asymptotically to convergence.

### 2.1 Validating the structure of citation data

XML [4] represents data in a file (or data stream) as a hierarchy of nested tags, whose general syntax is standardized, but whose tag names are customized for the task at hand. For example, a row of data in a relational database table "JournalName" with columns "ISSN" and "Title", might be represented in XML as:

```
<JournalNameRecord>
  <ISSN>01234-5678</ISSN> <Title>Journal of Bio</Title>
</JournalNameRecord>
```

However, the tags can also have "attributes", so one alternative design might be:

```
<JournalNameRecord  ISSN= "01234-5678"  Title="Journal of Bio" />
```

In any event, the data source providers and recipients must reach agreement on the precise nomenclature and parse-tree structure to be generated. This is traditionally formalized with a Data Type Definition (DTD) [4], that is included (typically by reference) as part of the XML file. This was the case with interfacing Upload, which represents one application of the "NLM Common" and "NLM Input Article" DTDs developed over 1999-2000 [5].

The specific XML structure these define (exemplified in Figure 1) consists of a top-level “InputArticleSet”, made up of “InputArticle”s, each of which is an “Article” with optional “MeshHeadingList”. These in turn are further defined in the Common DTD by inclusion.

```
<InputArticleSet>
   <InputArticle>
      <Article>
        <Journal><JournalIssue MRI ="NLM013133587" /></Journal>
        <ArticleTitle>Tumour necrosis factor-alpha in heart failure</ArticleTitle>
        <Pagination><MEDLINEPgn>745-51</MEDLINEPgn></Pagination>
        <Abstract><AbstractText>The experimental and clinical evidence that
           demonstrates the effect of various cytokines…. etc.</AbstractText></Abstract>
        <Affiliation>Baylor College of Medicine, Houston, Texas 77030, USA</Affiliation>
        <AuthorList CompleteYN="Y">
          <Author><LastName>TORRE</LastName><FirstName>G</FirstName>
          </Author>  Additional authors not shown
        </AuthorList>
        <Language>ENG</Language>
        <PublicationTypeList><PublicationType>JOURNAL ARTICLE
        </PublicationType></PublicationTypeList>
      </Article>
      <MeshHeadingList><MeshHeading>
         <Descriptor MajorTopicYN="N">Support, Non-U.S. Gov't</Descriptor>
      </MeshHeading></MeshHeadingList>
   </InputArticle>
</InputArticleSet>
```

**Figure 1. An Example Partial XML File from Upload.** Reformatted here for concision. Just one article is shown, whereas a real file would have all the issue’s articles. Grant information is bundled into the Mesh terms; besides what’s shown, there might be grant numbers for NIH-funded work and Genebank/Protbank numbers.

Generally, both generator and recipient programs have the option to validate against a DTD. DCMS does this with C program calls to an Oracle parser in “SAX mode”. Upload is now adding a DOM object, discussed below vis à vis CheckIn, for generator-side validation.

Alternatives to DTD are emerging: more powerful “schema definition” languages. Three advanced as standards are Relax [6], TREX [7], and the well-known but controversial [8] XML Schema (XSD) [9]. Could these do the extra validation discussed next?

## 2.2 Validating the character set of citation data

MEDLINE covers the biomedical literature for which abstracts are available in English. However, scientific terms, as well as author names and their institutional affiliations, introduce wider symbol sets. MEDLINE strives to permit major European languages with Roman alphabets. (Others have been handled by transliteration to Roman or single-character expansion, e.g., “alpha”.) The traditional set of valid Roman characters closely follows the 8-bit “ANSEL” character set of the Library of Congress MARC [10] bibliographic citation system, with roots in EBCDIC. As a result, DCMS does not accept, for example, the “{“, “}” and “|” characters, nor any accented characters except those in Table 1. Thus, a XML file that contains, for instance, “angstrom e” would be invalid, even though it verifies against the DTD; consequently, Upload pre-screens for such characters. As NLM evolves further towards the Unicode world, a wider set of characters may be possible, while preserving access to the existing MEDLINE corpus. (Note also that current high-end English-capable commercial OCR systems, such as used in Mars, have recognition limited to an 8-bit character set.)

**Table 1. Characters with diacritics currently accepted by DCMS.**

| Name | Symbol | Accepted if composed with (but only with case as shown) |
|---|---|---|
| Grave | ‘ | a, e, i, o, u, w, y |
| Tilde | ~ | a, i, n, o, u |
| Circumflex | ^ | a, c, e, g, h, i, j, s, u, w, y |
| Acute | ´ | a, c, e, i, l, n, o, r, s, u, w, y, z |
| Macron* | ¯ | a, e, i, o, u ** Limitations on generation of these by Mars* |
| Umlaut | ¨ | a, e, i, o, u, w, y |
| Angstrom | ° | a, u |
| Breve | ˘ | a, g, i, o, u |
| Cedilla | ¸ | c, g, k, l, n, r, s, t |
| Slash or bar* | / | o, O (Swedish), l, L (Polish) |

## 3. Front end loading and the birth of "CheckIn"

After the "back end" flow was converted to XML, a new flow route was conceived from DCMS to the Mars "front end", mediated by a fresh Mars application that would work as follows. A journal arriving at Mars has its NLM barcode scanned, giving its unique "MRI" identifier to the application. This is passed as a URL parameter to a DCMS web site. The on-demand query returns data about the given issue (volume number, cover date, etc.) and its series (ISSN, title, publisher, etc.), data that Mars operators were typing or picking by hand.

To quickly demonstrate overall feasibility of this idea, a few dialog-based mini-apps were built in VC++/MFC (by author GP), each to parse a different DTD. The parsing used Microsoft's MSXML implementation of the standard XML DOM object model [4]. DOM is a good choice with small XML files as occur here, since the whole DOM parse tree resides in memory. (Large XML files should use a "SAX2" event-driven parser.) Specifically, the demos (and later CheckIn) were built with MSXML 2.5, conveniently distributed with Windows 2000. (At the time, the next MSXML versions, 2.6 then 3.0, had come out, but perhaps needed a tricky "side by side" installation to avoid affecting the IE browser. At this writing, MSXML 4.0, which offers both DOM and SAX2 parsing and can handle XML Schema, is available as a "Preview", but awaits finalization of the XML Schema standard.)

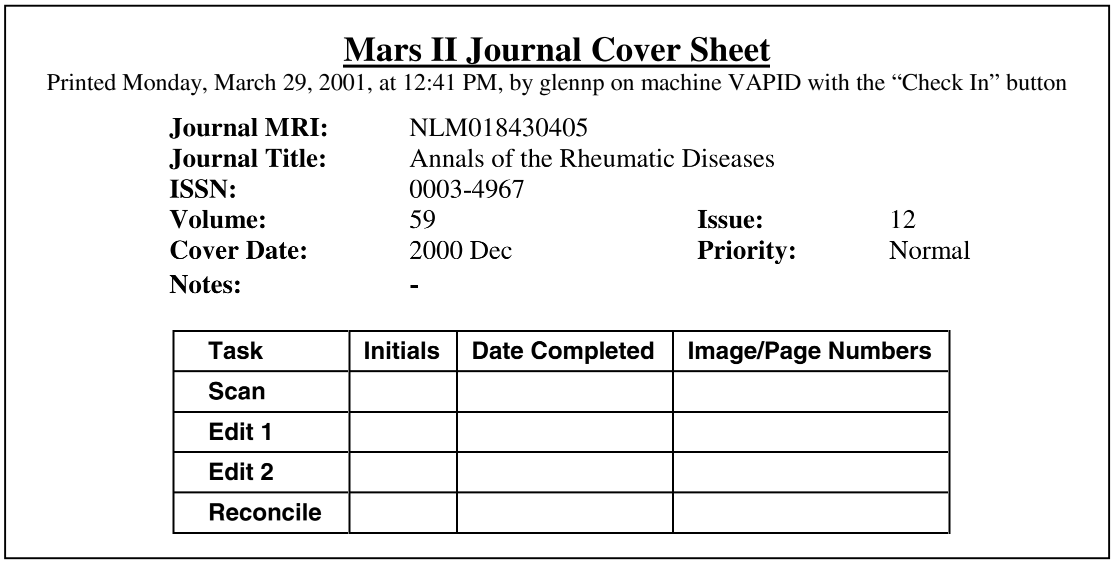
Consider the demo that grew into CheckIn. It fetched live XML data from a pre-existing internal URL that OCCS had recently created for its own purposes, again defined by a DTD derived from "NLM Common". This web service (i.e., XML-over-HTTP provider) has a Cold Fusion script to deliver XML from DCMS with the most vital data needed by Mars: ISSN, journal title (in full and short forms), cover date, volume, and issue. Some of this is a bit problematic. For instance, a cover date, as free text, might be "Spring/Summer 2001", but current Mars uses a more constricted SQL datetime field. And some useful data (e.g., publisher name, and in-house "notes" to NLM operators and indexers) is not available with this DTD, so it is filled in with the somewhat stale values of the Mars database.

Another demo explored what might be ideal for our needs if un-constrained by existing web services. A fresh "reference model" DTD was devised, using vocabulary and structure influenced not just by Mars and DCMS, but by the global Dublin Core Metadata Initiative and its Citations group [11,12]. The latter's XML-binding makes heavy use of recurring attributes like "SCHEME=", but it was found that a DTD can't really validate a subtree whose structure varied depending on the attribute value of its parent. Our reference model chose deeper nesting instead. On another point, the "NLM Common" structure is very flexible, but for Mars, where the journal issue for each article in a set is the same, introduces redundancy. A better design nests per-article data within a given journal.

Experience in these demos with MSXML via the Windows "Platform API" for HTTP and XML revealed that, while it is possible to download a file from the web service, check that the XML is well-formed, and optionally verify against the DTD all in one step, that is a bad idea. It was not unusual for a server-side error to occur, so that what came back to the application was a human-intended web page in HTML. This would fail the XML well-formedness check, and could be further identified by searching for the string "HTML 1.0". A frequent cause of this error was that the database was slow, so the web tier gave up. In that case, creating a code loop to refetch a second time always worked immediately, because the database query and results were now in the server's cache.

## 3.1 From demo to deployed

The mini-apps also fetched data from Mars, besides DCMS. Traditionally, Mars C++ applications used Rogue Wave's DBTools++ class library as client-side middleware to the database. But this suffers on-going per-seat costs. The mini-apps pioneered CEB's use of Microsoft's ActiveX Data Objects (ADO) instead, even if limited to just simple reads. This enthused the Mars developers for both ADO and the front end module idea. Further, there was a pending request from the Mars operations manager to create a front end module to generate cover pages. Each such page (Figure 2), paper-clipped to its journal issue during Mars processing, included the type of information available from the DCMS web site but currently hand researched and written. Merging the operational requirements of the two visions for this module was straightforward. The ensuing CheckIn module (by GP) inherited functionality from the demo code, but got a new main dialog window that looked like a printed cover page, minus the grid. It made sense for this to be a human-operated application, rather than a daemon, because the "Notes" field (initialized as described above) often needed to be edited to pass on supplemental, transient information to the Mars operators.



**Mars II Journal Cover Sheet**

Printed Monday, March 29, 2001, at 12:41 PM, by glennp on machine VAPID with the "Check In" button

| | | | |
|---|---|---|---|
| **Journal MRI:** | NLM018430405 | | |
| **Journal Title:** | Annals of the Rheumatic Diseases | | |
| **ISSN:** | 0003-4967 | | |
| **Volume:** | 59 | **Issue:** | 12 |
| **Cover Date:** | 2000 Dec | **Priority:** | Normal |
| **Notes:** | - | | |

| Task | Initials | Date Completed | Image/Page Numbers |
|---|---|---|---|
| Scan | | | |
| Edit 1 | | | |
| Edit 2 | | | |
| Reconcile | | | |

**Figure 2. Cover Sheet.** Shown reformatted for concision, from its actual full-page size. The grid at the bottom is penciled in by each task's operator as the attached physical journal is relinquished. The citational information, prior to CheckIn, was entered by hand.

When CheckIn was fielded at the Mars production site, a building separate the main NLM facility, some unexpected problems arose. For instance, the very first time CheckIn tried to query the DCMS web site, it was denied access, until the DCMS firewall was tweaked to recognize the off-site Mars IP addresses. A subsequent CheckIn release added a "Lookup Wizard", so that the operator could look up a journal from the Mars database by either title or ISSN, in case of these difficulties with DCMS information retrieval:

1) No or erroneous response from the web service, as discussed earlier;
2) The requested MRI is unknown to DCMS;
3) The returned information is missing an ISSN;
4) The returned ISSN is unknown to Mars.

Case 3 can occur because, in the DCMS database, unlike Mars, ISSN is neither a key nor required field. (If a journal really has no ISSN, the Mars administrator invents an obviously phony one.) For case 3, CheckIn now automatically tries to match the fetched journal title against those in the Mars database. Case 4 will ask the user if such a match attempt by title is desirable. (A planned improvement is to allow a new journal and its ISSN to be added to Mars at CheckIn.) For all cases, the fallback is Lookup Wizard invocation.

## 4. Current prognosis, future plans

Upload has proved to be reliable. Similarly, the first phase of CheckIn's rollout, even though limited by the existing web service, has been successful. Each day, the Mars manager uses it to check in about 40 journals with at least 600 articles. While aspects remain for later phases, overall, our first experiences deploying XML-centric modules have been positive.

## 5. Acknowledgements and References

CEB & the Mars project are headed by Dr. George Thoma. Upload was prototyped by Qianjun Gu & Stanley Huang; Daniel Le and OCCS's John Butler guided XML conversion. For CheckIn, Zhirong Li & Joe Popkin contributed draft Lookup code and cover page design respectively. Merle Chambers & Averi Roberts (Kevric) gave expert user feedback. The Common DTD, begun as SGML by Sergey Krasnov (NCBI), in 1999 became XML due to J. Butler & Mindy Tevlowitz (OCCS), and MEDLINE reps Jane Rosov & Lou Knecht (NLM/BSD).

[1] G.R. Thoma, "Automating Data Entry for an Online Biomedical Database: a Document Image Analysis Application", Proc. 5th Intl. Conf. On Doc. Anal. and Recog. (ICDAR'99), Bangalore, India, Sept. '99, pp. 370-3.

[2] G.R. Thoma, D.X. Le, "Automating Data Entry for Online Biomedical Databases", Proc. 14th Nat. Conf. On Integrated Online Library Systems (IOLS '99), Medford, NJ, Information Today, Inc., May '99, pp. 121-8.

[3] ISO/IEC Standard 10646-1, 1993. (For UTF-8, see Amendment 2 , 1996)

[4] World Wide Web Consortium, XML Working Group, "XML 1.0 Specification 10-February-1998", ed. T. Bray, J. Paoli, C.M. Sperberg-McQueen. Available in various forms: www.w3.org/TR/1998/REC-xml-19980210.

[5] "NLM to use XML and DTD for MEDLINE Data", 10/13/00, www.nlm.nih.gov/news/MEDLINEdata.html; the NLM Common DTD is given in www.nlm.hin.gov/bsd/licnesee.html.

[6] Murata Makoto, "RELAX (REgular LAnguage description for XML)", www.xml.gr.jp/relax. Also ISO/IEC DTR 22250-1, "Regular Language Description for XML (RELAX) -- Part 1: RELAX Core", 2000 October.

[7] Clark, James, "TREX- Tree Regular Expressions for XML", www.thaiopensource.com/trex. OASIS standard

[8] Dumbill, Edd, "W3C XML Schema still has big problems", xmlhack.com, 6 Mar 2001.

[9] "XML-Schema Part 0: Primer, W3C Working Draft", www.w3.org/TR/xmlschema-0/, March, 2001. See also: Allen Brown et al, "MSL: A Model for W3C XML Schema", WWW10, May 1-5, 2001, Hong Kong.

[10] "USMARC Specifications for Record Structure, Character Sets, and Exchange Media", Library of Congress, 1994. For newest MARC 21 specifications, including "crosswalks" with Unicode, see lcweb.loc.gov/marc.

[11] Ann Apps and Ross MacIntyre, "Dublin Core Metadata for Electronic Journals", Proc. 4th European Conf. On Res. And Adv. Tech. For Digital Libraries, Lisbon, Portugal, 18-20 Sept. 2000, Springer-Verlag, pp 93-102.

[12] Bibliographic Citation Working Group, Dublin Core, "DC-Cites", Oct. 1999, http://dublincore.org., www.jiscmail.ac.uk/lists/dc-citation.html.