Identifying XML Issues That Impact Content Interchange

JATS-Con Conference Proceedings 2022

a report from

DCL™
Data Conversion Laboratory Inc.

DCL

61-18 190th Street
Suite 205
Fresh Meadows, NY 11365

+1.800.321.2816

www.dataconversionlaboratory.com

ABOUT DCL

Intelligent data transformations

DCL (www.dataconversionlaboratory.com) provides data and content transformation services and solutions. Using the latest innovations in artificial intelligence, including machine learning and natural language processing, DCL helps businesses organize and structure data and content for modern technologies and platforms. With expertise across many industries including publishing, life sciences, government, manufacturing, technology and professional organizations, DCL uses its advanced technology and U.S.-based project management teams to solve the most complex conversion challenges securely, accurately and on time.

Your data:

transformed, validated, enriched

CONTENTS

Scholarly Publishing Content Collections

Publishers' content collections are complex, often spanning decades, during which time standards have evolved. Version 1.0 of the Journal Publishing Tag Set (aka NLM DTD) was released in February 2003. Thus, content that used the NLM DTD is significantly different from the current specification—NISO JATS Version 1.3, which was released in June 2021. Further, there is more than one way to interpret the specifications, and various parties preparing content over the years may have chosen different interpretations and adopted varying "best practices". The combination of these discrepancies, plus errors that might have crept in, combined with how systems and people discover content from ever larger collections, calls for a systematic review and analysis of a publisher's entire content collection to ensure content structure is optimized for today's platforms, readers, and APIs.

Errors and issues with content structure have serious impact on downstream discoverability and content interchange. Improving and standardizing content structure to take full advantage of JATS 1.3 benefits the entire journal publishing ecosystem. Publishers may be missing out on functionality if content is not consistent and up to date with the latest DTD. Researchers and readers may be missing out on discovering content when it is not optimized to work across the various platforms and systems through which it is disseminated.

In many cases, journal publishers may not review and analyze their entire content collection until it is time to move to a new hosting platform, and that's the point at which you would discover the discrepancies. The last thing you want is surprises when finally loading the content onto a new platform or delivering content to a licensee. This paper reviews how analyzing an entire corpus of a publisher's XML files reveals issues in the JATS XML that do not necessarily invalidate the files but do contribute to interoperability and other issues. Analyzing the content early allows improving the content structure to take full advantage of JATS XML 1.3, benefits the entire journal publishing ecosystem, and facilitates interoperability.

Business Drivers for Change

While system interoperability is a key driver for changes to content structure, other forces are also at play. Publishers change platforms and vendors as well as the tools used to create XML, which results in variations to that XML. At the same time, best practices for content interchange continuously evolve.

Movements to expand online collections to include legacy content have matured. Making content available for historical context is important (e.g., bibliographies, meeting minutes, biographies), and publishers see value in creating uniform well-polished "atoms" from legacy materials for use in rapid development of new product offerings. Items such as equations, tables, funding information, bibliographies, etc. are easily captured when structured in updated JATS format and provide flexibility for new uses.

The complex web of discovery vendors such as EBSCO, ProQuest, Kudos, Web of Science, and many others have varying requirements in content structure that enable libraries, content consumers, and even a publisher's website platform to find and deliver into the content supply chain.

Platform Migration

If lucky, a publisher may change website platforms once or maybe twice every decade. But transitioning from one website platform to another platform introduces errors in tagging, encoding, and spacing that are not apparent until an analysis of the content structure is conducted across the entire journal collection. Nuances in XML structure exist across the various platforms such as Atypon, Highwire, Silverchair, and others, including internally-developed platforms. The analysis of a publisher's entire corpus to identify obstacles in content structure that hinder discoverability and interoperability is critical at the time of platform migration. However, investing in a deep analysis of a publisher's collection outside of platform migration and fixing or updating structural issues contributes to the frictionless flow of content.

When publishers decide to simply move and load existing content onto a new platform, they are not taking advantage of the wealth of their collection and may not realize they are missing out on structural edits and updates that could improve discoverability. Finding problems in advance and fixing them before a migration even begins alleviates many problems and minimizes the "garbage in; garbage out" syndrome.

It's important to also view and consider a publisher's collection across time. Articles published 15 years ago may not have full markup and may not be up to the publisher's current best practices, and therefore are missing out on functionality (e.g., fully-tagged references/affiliations, funding information, ORCIDs, etc.).

Best Practices

The scholarly publishing industry is fortunate to have standards and best practices to guide industry adoption and collectively solve issues around content interchange. Two best practices relevant to this paper include NISO RP-38-2021 and JATS4R.

While JATS XML is the gold standard for content structure across the industry, platforms make use of the DTD differently. For example, some publishers use a subset of the DTD or have various best practices. Standardizing on standards' usage provides increased efficiencies across systems and tools that benefit the end user.

NISO RP-38-2021

NISO published recommended best practice guidelines for platform migrations—NISO RP-38-2021:

> *"The goals of the Content Platform Migrations Recommended Practice ("Recommendations") are to promote a set of guidelines that apply whenever electronic content is migrated from one hosting platform to another, and to encourage the industry to embrace these recommendations as a baseline level of quality."*

Publishers that strategize and plan for organizational change to the development, production, and distribution of content must have a clear picture of content structure across the entire collection.

> *"A publisher with complete knowledge of its content set can provide a detailed inventory report to the party that is normalizing their content to the new standard.*
>
> *…the publisher needs to widely share important variances in its backfile. It is virtually guaranteed that publishers with longer histories will have more challenges to overcome in content normalization."*

JATS4R

JATS for Reuse (JATS4R) is another industry best practice that is devoted to optimizing the reusability of scholarly content. The more standardization there is in JATS usage, the more efficient file exchanges become as well as the tools and services used in the publishing ecosystem:

> *"'Reusability' is the ability of machines to 'reuse' published content for exchange, storage, retrieval, and sharing throughout the scholarly publishing infrastructure. This infrastructure, which is continuously expanding, includes search engines, aggregator and indexer systems, archives, repositories, identifier-assigning authorities, digital catalogs, and databases, making interoperability more important than ever."*

Adopting and following best practices clearly provides advantages that enable the discovery and use of scholarly content.

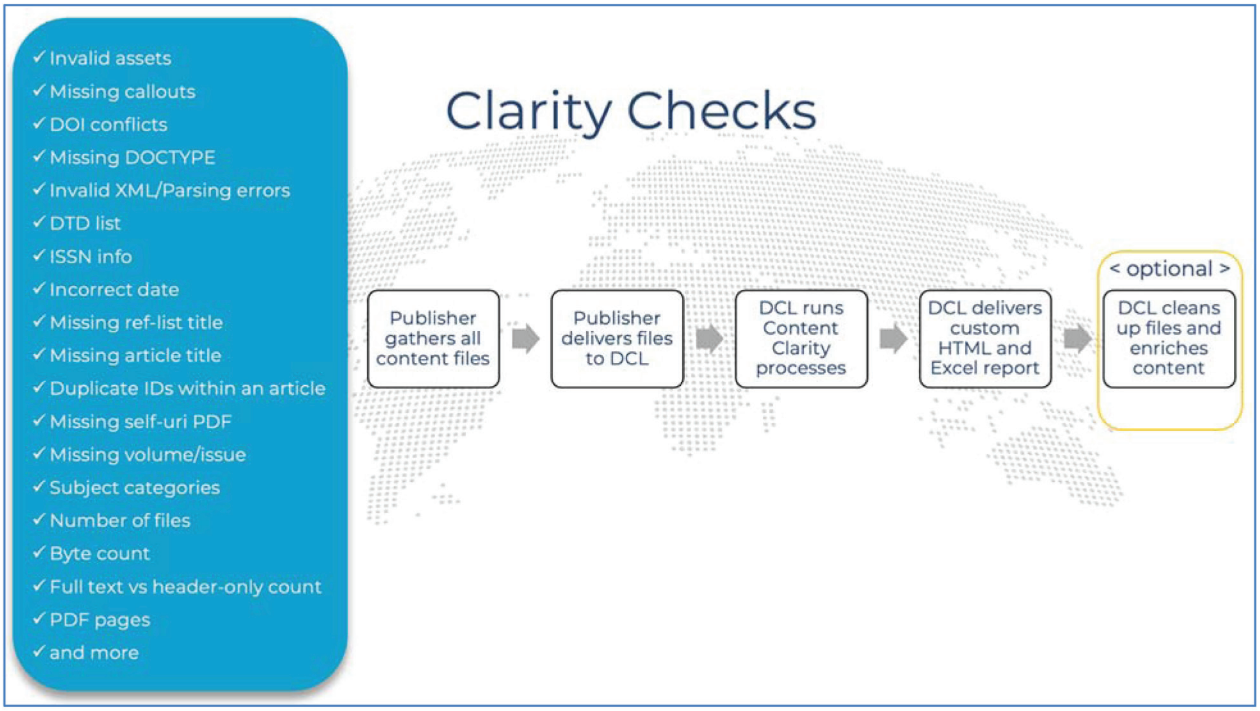
Making the effort to systematically analyze across an entire collection, independent of a platform migration, ensures all content is equally discoverable.

Analyzing a Content Collection

The approach used at DCL to analyze, report, and update content structure across a collection involves a series of what we term "clarity checks." The first step in the process involves a publisher gathering all content files (XML and PDF) in one place. For some publishers, it may be the first time they collect and look critically across a collection. Figure 1 depicts just some of the items we analyze as well as the basic workflow.

Findings from the analysis are grouped into two categories—Summary Analytics and Errors and Warnings. Results are normally presented in a spreadsheet with multiple underlying worksheets that provide additional detail:

- **Summary Analytics**—A series of analytics that report on the state of the collection and how well the collection aligns with the publisher's expectations. Metrics collected might include the following:
  - ✓ Number of XML/SGML files
  - ✓ Total XML/SGML bytes
  - ✓ Files with header/footer content only
  - ✓ Full chapters
  - ✓ PDF-only chapters
  - ✓ Single-PDF book
  - ✓ Wrapper, no PDF
  - ✓ Number of PDFs
  - ✓ Total PDF pages
  - ✓ Total asset bytes
  - ✓ Invalid assets-count
  - ✓ Invalid callouts-count
  - ✓ Data errors-count
  - ✓ Validation errors-count
  - ✓ DTDs
- **Errors and Warnings**—Because JATS is an intentionally robust standard, not



Clarity Checks

✓ Invalid assets
✓ Missing callouts
✓ DOI conflicts
✓ Missing DOCTYPE
✓ Invalid XML/Parsing errors
✓ DTD list
✓ ISSN info
✓ Incorrect date
✓ Missing ref-list title
✓ Missing article title
✓ Duplicate IDs within an article
✓ Missing self-uri PDF
✓ Missing volume/issue
✓ Subject categories
✓ Number of files
✓ Byte count
✓ Full text vs header-only count
✓ PDF pages
✓ and more

Publisher gathers all content files → Publisher delivers files to DCL → DCL runs Content Clarity processes → DCL delivers custom HTML and Excel report → < optional > DCL cleans up files and enriches content

Figure 1.

A few of the clarity checks and basic workflow for DCL's Content Clarity.

all elements are necessary and certainly not all elements exist in legacy content. To that end, these clarity checks detail possible errors in the XML and provide warnings. For example, a true error for <month> would be PCDATA with a value greater than 12. Strictly speaking, <month> represented as "1" would also be wrong because some platforms, but not all, require the month to be a two-digit value. Thus "01" is how the PCDATA should be expressed. Additionally, if there is an <ack> element without a title, we would generate the possible error "Missing ack title." Warning notifications categorize and collect findings so that a subject matter expert can investigate further. Identified warnings and errors might include the following:

- ✓ Duplicate ID
- ✓ Incorrect book structure
- ✓ Incorrect date
- ✓ Incorrect xref
- ✓ Invalid institution id
- ✓ Missing ack title
- ✓ Missing article title
- ✓ Missing asset
- ✓ Missing DOI
- ✓ Missing ref-list title
- ✓ Missing self-uri PDF
- ✓ Missing title
- ✓ Missing vol/issue
- ✓ Multiple citations in one ref
- ✓ Multiple issue-meta
- ✓ No cover image
- ✓ Suspicious abstract type
- ✓ Suspicious contrib type
- ✓ Suspicious footnote label
- ✓ Suspicious related article type
- ✓ Tagging inside subject
- ✓ Volume/issue/ppub different in unit
- ✓ Duplicate article-id
- ✓ Duplicate ISBN
- ✓ Invalid asset type (type) for tag
- ✓ Invalid assets
- ✓ Invalid callout
- ✓ PAP article not replaced
- ✓ Validation errors

Examples of Content Structure Issues

While each publisher's content set is unique with its own errors or issues in the XML, the following are some of the findings we've seen.

Changing Constructs Over Time

There are various constructs that were deprecated over the years as XML usage accelerated among journal publishers and the NLM DTD evolved. Old tags can still be buried deep in legacy files. In the NLM DTD, <appendix> within a <bio> was valid at one point but that is no longer the case. Similarly, the way in which awards and grants are described is updated in JATS 1.3 with the elements

<award-group>

<award-id>

<award-name>

<award-desc>

Funding Information

Funding information in the NLM DTD was often represented as an attribute on <named-content> (e.g., content-type="funder-name" or content-type="funder-identifier"). In JATS 1.3 <funding-group>, <funding-source>, and <funding-statement> provide much more usability to tag and express funding information. PubMed Central will still accept content in the NLM DTD format but there are reasons JATS evolved and thus the structure (even for legacy content!) should also evolve. We now understand that funding agencies, drug companies, and even the general public require or desire access to journal articles and the research that they sponsor. Standardizing on funding information even for legacy content is important.

Corresponding Contributor

A journal article’s corresponding author is the person who takes primary responsibility for communication through the publishing process as well as additional duties such as providing details involving clinical trial documentation, ethics committee approval, and other key tasks.

Often corresponding contributor information was tagged in a footnote. Thus we would see

```
<fn id=“FN150”><p>Correspondence should be
sent to… E-mail:
  <email>authorx@jos.com</email>.</p></fn>
```

The better way to structure corresponding contributor information would be

```
<contrib-group>
        <contrib contrib-type=“author”
corresp=“yes”>
            <name name-style=“western”>
             <surname>Regni</surname>
             <given-names>Marie</given-
names>
</name>
            <xref ref-type=“corresp”
rid=“cor1”>*</xref>
</contrib>
```

Supplementary Material

In today’s era of Open Access publishing, supplementary information in a journal article plays a critical role. Making supplementary material available is critical when thinking about open data initiatives and allowing others access to datasets. Additionally, supplemental material, such as videos or audio files, makes an article more discoverable by providing another route to one’s research.

In the past, supplemental material was often structured as a section in a paragraph tag. But tagging supplementary material within its own element is more useful:

```
<article-meta>
 …
 <contrib-group>
 <contrib contrib-type=“author”>
   <collab collab-
type=“committee”>Accredited Standards
Committee S3,
        Bioacoustics</collab>
 </contrib>
 </contrib-group>
 …
 <fpage seq=“1”>1</fpage>
 <lpage>44</lpage>
 <supplementary-material mime-subtype=“zip”
mimetype=“application”
   xlink:href=“ASASTD.ANSI.ASA.S3.50.
supplementary-material.zip”/>
 …
</article-meta>
```

Missing or Duplicate DOIs

The Digital Object Identifier (DOI) is a unique identifier for a journal article or digital document. DOIs are a critical component in academic citations because a DOI is more permanent than a URL that can change and often does change with platform updates or even company acquisitions. Additionally, journal articles are often found on multiple platforms, websites, and databases. Thus, a DOI is the identifier that ensures identification, discovery, and interoperability across the scholarly publishing landscape.

If two different articles have the same DOI, that is an error that must be corrected. The following two XML files contained *<article-id pub-id-type=“doi”>10.1158/JOS.211.8.1186</article-id>*:

```
JOS\Issues\211_8\1186.xml
JOS\Issues\211_14\2369.xml
```

It’s important to note that not only do we analyze each individual file to reveal structural issues, the cohesive analysis across a content collection is also where findings arise that would not be seen if only analyzing singular XML files. That two articles have the same DOI might not be discovered until a review of the entire collection is done.

Publication Dates

A full issue is expected to have the same publication date throughout its content. When analyzing an archive, we check that all the content for a volume/issue has the same publication date. We also check the consistency of the volume/issue information across the collection.

Following is an example in which the publication date year was different in one file than it was in the rest of the issue.

```
Different volume/issue/ppub value 74/1/2019-01-01 than 74/1/2020-01-01
```

In this example, the month and day were different from the rest.

```
Different volume/issue/ppub value 57/1/2013-05-14 than 57/1/2013-01-10
```

Date Errors

Many errors in tagging publication dates arise across content collections. The publication date of an article is perhaps the most important trigger for downstream discovery, appropriate author credit, Crossref, and other areas.

Following is a typical publication date construct.

```
<pub-date publication-format=“print” date-type=“pub” iso-8601-date=“1999-01-29”>
  <season>Spring</season>
  <day>29</day>
  <month>01</month>
  <year>1999</year>
</pub-date>
```

A publication date error could look like this because JATS expects the iso month and day to be expressed with two digits:

```
<pub-date pub-type=“ppub” iso-8601-date=“2020-06-1”>
    <day>1</day>
    <month>June</month>
    <year>2020</year>
</pub-date>
```

The more accurate representation should be

```
<pub-date pub-type=“ppub” iso-8601-date=“2020-06-01”>
    <day>1</day>
    <month>June</month>
    <year>2020</year>
</pub-date>
```

For most aggregators, every XML article is expected to have either a ppub date or a collection date. An epub date on its own is not enough. The following XML will parse and validate according to the DTD.

```
<pub-date pub-type=“epub”>
    <day>24</day>
    <month>12</month>
    <year>1998</year>
</pub-date>
```

The more accurate representation should also contain a collection date.

```
<pub-date pub-type=“epub”>
  <day>24</day>
  <month>12</month>
  <year>1998</year>
</pub-date>

  <pub-date pub-type=“collection”>
  <day>1</day>
  <month>12</month>
  <year>1998</year>
</pub-date>
```

Missing Abstracts

Today it is generally unacceptable to publish a journal article without an abstract. The first evidence we have of standardized author-written abstracts comes in 1914 when the Royal Society communicated that every paper

> *“‘must be accompanied by a summary not exceeding 300 words in length, showing the general scope of the communication, and indicating points which, in the opinion of the Author, are of special importance.’ [revised standing orders, in RS Council Minutes, 21 May 1914, para 37]”*

Today, abstracts are critical for indexing in Google Scholar as well as the discovery vendor platforms.

In some cases, tagging for <abstract> revealed structurally accurate XML but no true summary of the journal article, which is perhaps the first and often the only part of a published article that prospective readers can readily access from a literature search:

```
<abstract>
<title>Abstract</title>
<p>Abstract not found</p>
</abstract>
```

While there are various AI techniques that might be used to evaluate abstracts, a simple approach might to verify a minimum character count check to ensure that the text within an abstract is indeed a summary of the article and not simply text indicating that there is no abstract; thus the report indicates:

Abstract too short

Invalid Assets and Callouts

In the XML of a journal article, assets are typically images or tables that live as a specific file format and are referenced in the XML. Therefore, for every image there is at least one callout in the XML and for every callout there is an asset.

When publishers submit their corpus for analysis, all XML and related files such as GIFs, JPGs, PNGs, etc. are also included in the package. Validating across files and identifying if corresponding callouts and assets are part of the publisher's package is a critical check that ensures all related content is served cohesively to a reader.

The check for an invalid asset confirms that for every asset callout in the XML, the corresponding file was provided by the publisher. For example, the following callout

<graphic xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="permzspch025.jpg ">

searches the file directory and looks for

Journal-of-Science\171_5\images\permzspch025.jpg

However, if the XML calls out permzspch024.jpg and that image is not found, an error is reported:

Missing Asset: <graphic xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="permzspch024.jpg ">

A reverse check is performed as well to ensure that every asset provided by the publisher has a callout in the XML.

Another asset check is to make sure that the file extension on a given asset is correct and valid. The software looks at the assets and determines what they really are. The filename might include .jpg but in actuality the file is a .gif.

Mismatch between reported image type 'GIF' and file extension 'JPEG' permzspch024.jpeg

Another example combines cross-checking a callout against a series of business rules. For example, the following callout for this image would also verify against the business rule that states the maximum size of a TIF image.

Journal-of-Science\200_7\images\jimmunol_200_7_coverfig.tif

File is too large for TIFF. Size is 210033760 bytes

Information Analysis

Identifying errors in structure is critical but equally important is analyzing a corpus for useful and important information about the collection. While not strictly metadata, some of the analysis performed is information about publishers' data.

DTDs/Schema Used

Global standards are in place for the facilitation of content across the ecosystem benefitting the publisher, technologies used, and of course the end user of that content. Likewise, specifications evolve based on usage, and updates to DTDs are encouraged to keep pace with those changes. When a content collection comprises thousands (millions) of articles, it’s nearly impossible to keep track of the various DTDs used over the years.

To date we have calculated 21 various DTDs/ Schemas used across publishers’ journals collections. Table 1 shows an example from a report that lists the DTDs used across a publisher’s collection.

Article Types

A count of Article Types can be a useful and interesting metric that ensures that what is identified in the publisher’s XML files indeed lines up with other record keeping employed. This report also allows publishers to normalize variations that have occurred over time and update their legacy content.

Table 2 provides an example of an Article Types count. Interesting to note is that in this example, 810 articles were not classified with an article type and 1,646 were classified as “other,” which is something a publisher might want to investigate to understand if the article type information is simply missing or miscategorized.

Subject Categories

Information collection for subject categories reveals issues in consistency across a publisher’s ontologies and taxonomies. A subject-by-date information report provides a way to understand when certain values were used historically, which helps in making decisions when normalizing the values.

Listing and associated counts of subject category metadata provides publishers with a valid starting point to address errors in consistency. In the following example we see five different ways that “immunogenetics and transplantation” are identified:

Immunogenetics and Transplantation

Immunogenetics and Transplantations

Immunogenetics & Transplantations

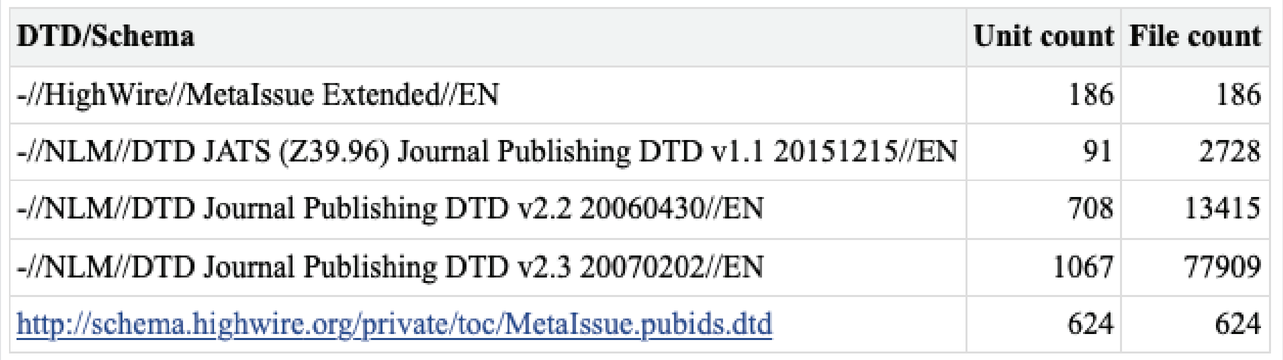
Immunogenetics and transplantation

Immunogenics and Transplantation

Techniques

When performing this analysis, the first step is receiving all XML files from a publisher. We examine the publisher’s corpus, provided in hierarchical folders and files. The first phase of processing separates the content into “units” and extracts relevant metadata from the files. “Units” are typically journal issues but could also be collections of ahead-of-print articles, manuscripts, or conference proceedings; for book content, each book is considered a “unit.” Processing also inventories digital assets (e.g., .jpg, .gif, .XLSX, .r, .JSON, etc.).

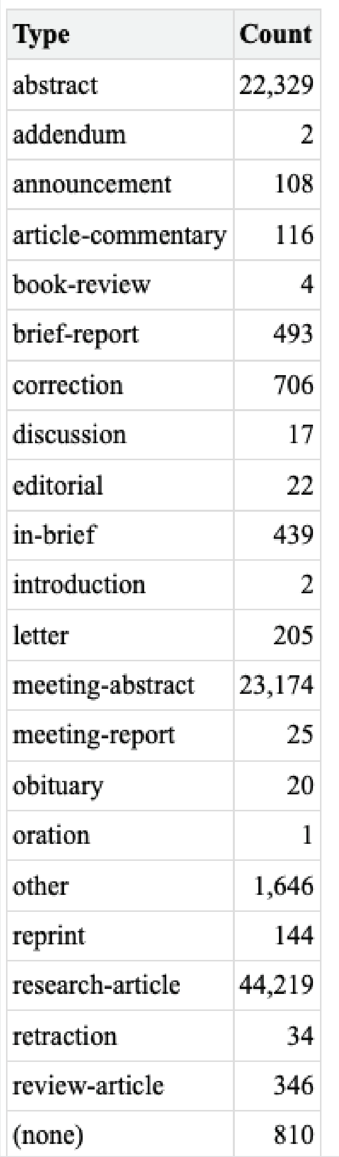
The first phase is multithreaded, restartable, and incremental. Multithreading allows the execution of multiple parts of a



DTD/Schema	Unit count	File count
-//HighWire//MetaIssue Extended//EN	186	186
-//NLM//DTD JATS (Z39.96) Journal Publishing DTD v1.1 20151215//EN	91	2728
-//NLM//DTD Journal Publishing DTD v2.2 20060430//EN	708	13415
-//NLM//DTD Journal Publishing DTD v2.3 20070202//EN	1067	77909
http://schema.highwire.org/private/toc/MetaIssue.pubids.dtd	624	624

Table 1.

Content Clarity report of DTDs and Schemas used across a collection.



Type	Count
abstract	22,329
addendum	2
announcement	108
article-commentary	116
book-review	4
brief-report	493
correction	706
discussion	17
editorial	22
in-brief	439
introduction	2
letter	205
meeting-abstract	23,174
meeting-report	25
obituary	20
oration	1
other	1,646
reprint	144
research-article	44,219
retraction	34
review-article	346
(none)	810

Table 2.

Article types count example from Content Clarity.

program at the same time and improves the responsiveness of a system, which is important when processing large amounts of data. A restartable application can be rerun after system downtime or failure, which allows the system to restart at the point it stopped, ultimately saving processing time and system resources. Incremental processing reduces the total processing required; it does this by processing only a data partition newly added to a dataset when the existing data is already processed, instead of re-processing the complete dataset.

The next phase validates XML files and health checks digital assets. The third phase performs a variety of semantic checks on the individual XML files, at the unit level, at the journal title level, and across the entire corpus.

The relationships between the XML files and digital assets are analyzed and verified.

The final processing phase reports on the findings. Reporting can be selective or all-inclusive.

The technologies used include DOM, XPath, regular expressions, custom algorithms, Exif tools, PDF tools, and Ghostscript.

Deeper Dive on Some of the Checks

ORCIDs

After checking that the ORCID value starts with “https://”, we apply an algorithm to validate the checksum of the value to ensure it follows the numbering format of an ORCID value.

For example, for <contrib-id contrib-id-type=“orcid”>https://orcid.org/1234-5678</contrib-id>, we calculate the checksum for the first seven characters (“1234-567”), following ORCID.org’s instructions, and report if there is an error. For example,

“Calculated checksum for orcid ‘1234-567’ is ‘2’, not ‘8’”

Authority: https://support.orcid.org/hc/en-us/articles/360006897674-Structure-of-the-ORCID-Identifier

ISSNs and ISBNs

A similar approach is used for validating ISSNs/ISBNs, but due to the prevalent use of ISSN and ISBN, we can leverage available modules to validate these values.

ISSN is an eight-digit code, divided by a hyphen into two four-digit numbers. As an integer number, it can be represented by the first seven digits. The last code digit, which may be 0–9 or an X, is a check digit.

For example, given “<issn pub-type=‘ppub’>2049-3631</issn>”, we check the value using the module's algorithm, and report an error:

"Checksum for issn '2049-3631' is invalid"

Authority: https://www.loc.gov/issn/basics/basics-checkdigit.html

Authority: https://www.isbn-international.org/content/isbn-users-manual/29

Conclusion

The purpose of all this is to analyze individual XML files as well as cross-analyze an entire corpus of content to identify issues or errors in content structure that can be improved for discovery and interchange. DCL wraps this entire process and analysis into its service called "Content Clarity." Improving content structure to take full advantage of JATS XML 1.3 benefits the entire journal publishing ecosystem.

Publishers may be missing out on functionality if content is not consistent and up to date with the latest JATS DTD. A periodic assessment of a publisher's corpus is critical to ensure the key drivers for adopting XML are met and continue to support publishers, readers, libraries, funders, and other invested sponsors of the scholarly publishing ecosystem.

Bibliography

1. American National Standards Institute JATS: Journal Article Tag Suite, version 1.2 (ANSI/NISO Z39.96-2019) Baltimore, MD: NISO; approved February 8, 2019.
2. Journal Archiving and Interchange Tag Library NISO JATS Version 1.3 (ANSI/NISO Z39.96-2021) National Center for Biotechnology Information (NCBI) National Library of Medicine (NLM) https://jats.nlm.nih.gov/archiving/tag-library/1.3/element/arc-elem-sec-intro.html.
3. Identifying and Standardizing Funding Information in Scholarly Articles: a Publisher's Solution Schwarzman Alexander B, Dineen M Scott. Journal Article Tag Suite Conference (JATS-Con) Proceedings 2016 https://www.ncbi.nlm.nih.gov/books/NBK350153/
4. Where Did the Practice of "Abstracts" Come From? The History of the Scientific Journal https://arts.st-andrews.ac.uk/philosophicaltransactions/where-did-the-practice-of-abstracts-come-from/

DCL™

Data Conversion Laboratory Inc.

Address	61-18 190th Street Suite 205 Fresh Meadows, NY 11365
Telephone	+1 800.321.2816
Web	www.dataconversionlaboratory.com
Twitter	@dclaboratory
LinkedIn	linkedin.com/company/dclab