XML Publishing Workflows

Tools, Practices and the Future
LPF 2020

Who are we?

Vitaliy Bezsheiko:

- Senior Editor for Psychosomatic Medicine and General Practice
- Grobid improvements, JATS Parser Plugin, DOCXTOJATS plugin, more

Dulip Withanage:

- Developer for Heidelberg University Publishing
- meTypeset, heiMPT, OJS typeset & lensGalleyBitsPlugin, Texture

James MacGregor:

Coalition Publica ops/technical teams, PKP|PS

Who are we?

eLife Partnership: eLife, Érudit, PKP, SciELO

Coalition Publica: PKP, Érudit

Semantic Extraction Group

Friends of Manuscripts Press Group

JATS4R (JATS for Reuse) Working Group

Structure of the session

Intro to what we're doing and why

XML workflow demonstration:

- Conversion, using DOCX Converter Plugin and Grobid
- Editing, using Texture
- Typesetting/publishing tools (LensGalley BITS, JATS Parser)

NOT covered in slides, but I can discuss if requested:

- Conversion using meTypeset
- Editing using Prosemirror

Open Discussion:

- Your use cases and questions
- Timelines, next steps

What are we trying to accomplish?

1. Support JATS XML content markup: metadata, fulltext, and references
    a. First, support DOCX/PDF **conversion** to JATS, and WYSIWYG **editing** of results
    b. Long-term, support seamless JATS-first document creation **directly within** OJS
    c. Along the way, support **typesetting** into HTML and PDF
2. Work with partners and community to support wider JATS use cases
    a. First, support Texture's JATS4R subset
    b. Long-term, work to support JATS4R generally, and variances
    c. Do this in an easy to manage, extensible way

Why are we doing this?

1. We've been doing this since ~2005, why stop now?
    a. This is the worst possible answer, but it's one answer.
2. Support reusability
    a. JATS metadata is semantically rich, easily processable/reusable (we do this in Coalition Publica)
    b. JATS full-text is easy to do research on (ditto - Coalition Publica is building a large research corpus on this stuff)
3. Support preservation and long-term access goals
    a. XML is a great preservation schema
    b. HTML and PDF can be generated and re-generated from source texts
4. Support new forms of typesetting, display, integrations

Where are we now?

We are now working on a more distributed/decentralized ecosystem:

- Creation/conversion: DOCXTOJATS, Grobid, meTypeset, from scratch
- Editing/Typesetting: Texture, Prosemirror
- Publishing/Display: Lens Reader, JATS Parser Plugin

What follows is a discussion and demonstration of ***in-progress*** developments.

Not all components are complete yet!

Conversion: Grobid

Grobid is a machine learning software developed for data extraction from scientific articles in PDF format. Originally aimed at metadata extraction, it’s now also capable to extract article’s full text. The main problem of extraction of any data from PDF is that it’s only “visible” as raw and unstructured text. Grobid with machine learning techniques allows to put this data into a structure, like sections, tables, citations, etc. Grobid receives PDF files as an input and converts them to TEI XML, which has similar to JATS XML structure.

Conversion: Grobid

Current development plans:

- Enrich data extraction for fulltext model:
  - Lists. The support for lists is already implemented and pushed on the dev GitHub branch.
  - Tables. Considering the best approach to determine the table structure (rows and cells):
    - Tabula integration. Already implemented, more training data is needed to evaluate accuracy or
    - Determine structure based on PDFAlto results.
  - Blockquotes.
- Provide more annotated training data for machine learning models.
- ~~Develop a converter from TEI XML (Grobid's output) to JATS XML or consider integration of an existing one.~~ (Done!)

Conversion: Grobid

Next steps

- Adapt OJS plugin results to work with Texture/OJS
- More data!

Demo:

- https://grobid.e-medjournal.com/ (Web service **demo only**)
- http://xml.test.publicknowledgeproject.org/index.php/grobid/management/settings/website#plugins (plugin config)
- http://xml.test.publicknowledgeproject.org/index.php/grobid/workflow/index/9/5 (OJS plugin demo)

Conversion: DOCX to JATS XML Converter Plugin

The plugin converts DOCX files which correspond to OOXML format to JATS XML. It's based on a library written in PHP and doesn't have any other dependencies. The output is compatible with the Texture Plugin. As an input can be used DOCX files produced by MS Word, LibreOffice Writer and Google Docs.

Current development plans:

- Extend the support for article elements, like figures, formulas, citations.
- Integration with tools that can help to extract data that aren't regularly present in OOXML.

Conversion: DOCX to JATS XML Converter Plugin

Differences from Grobid (besides input format):

- Doesn’t require anything beyond PHP interpreter.
- The output is restricted to how data is structured in the source DOCX file, although should work well in most of the cases.
- Other limitations for XML to XML conversion.

Readme and demo: https://github.com/Vitaliy-1/docxConverter/blob/master/README.md

Online demo:

- Article: http://xml.test.publicknowledgeproject.org/index.php/docxtojats/article/view/1
- Backend: http://xml.test.publicknowledgeproject.org/index.php/docxtojats/workflow/index/1/5

Editing: OJS Texture Plugin

Functionality

- OJS Plugin with integrated Texture: javascript based XML editor for JATS XML
- Texture Editor is available for integration and as offline application
- DAR as base - Document archive format for research
- Supports images as dependent files
- OJS texture plugin available in plugin gallery
- Import and export DAR files between OJS and Offline Texture Editor
- Creates a galley from a given XML file and it’s dependent files

Editing: OJS Texture Plugin

Hands on Demo

- http://xml.test.publicknowledgeproject.org/index.php/docxtojats/workflow/index/1/5

Editing: OJS Texture Plugin

Current development plans

- Restrict Texture Editing to JATS Body tags
  - Use OJS for frontmatter, references
- Documentation
- Middleware in OJS to validate the supported JATS tags
- Update Texture to version 2.3

Oh, also!

- Texture development has ceased. :-(
- Community moving to Prosemirror.
- This is good news long-term, but annoying short-term.

Publishing: LensGalleyBits Reader

Description

- XML based web-reader (native support for JATS)
- Early implementation by eLife, forked and extended.
- Client-side rendering, no extra libraries
- Additional support for a subset of BITS for monographs and edited volumes
- Limited mobile support : iPad and above
- Standalone application for offline production preview
- Available in OJS plugin-gallery

Publishing: LensGalleyBits Reader

Demo

- Real world example of an edited volume in heiUP

OJS Plugin

- https://github.com/withanage/lensGalleyBits

Lens source code

- https://github.com/withanage/UBHD-Lens#implemented-extensions

Publishing: LensGalleyBits Reader

Current development plans.

- Release for OJS 3.2 planned until 15.June.2020

Publishing: JATS Parser Plugin

JATS Parser Plugin is aimed to convert JATS XML to HTML and PDF and present the article on the front-end. It can be divided mainly into 2 parts:

- JATS Parser library that is written in PHP, thus converts documents on the server side. Current output is HTML (JATS XML -> PHP Objects -> PHP DOM) and PDF produced with TCPDF. Currently, it parses body and references from a given JATS and retrieves article's meta from OJS.
- Front-end part, integrated into the plugin. It relies on Bootstrap 4 and is mobile-friendly.

Requirements: PHP 7.2 or newer, theme that supports Bootstrap 4 (Classic, Immersion, Health Sciences).

Publishing: JATS Parser Plugin

Current development plans:

1. Ensure full compatibility with the output from Texture Plugin.
2. Add compatibility with all OJS themes.
3. Option to display JATS XML on article landing page rather than as a separate galley.
4. Testing and first production release.
5. Extend the support for more JATS XML elements, like formulas and footnotes.
6. Consider the ability to display references from JATS in different citation styles, currently only AMA (similar to Vancouver style).
7. Option to customize PDF output. TCPDF has limited support for styling of produced PDFs, although it's the fastest and most lightweight library.

Demo: HTML, PDF

Next Steps

2020:

1. Release current tools (some maybe still in “beta”)
2. Develop a common evaluation framework for each toolset
    a. Eg., for all conversion tools, develop a “supported elements” matrix
    b. Formalize testing and reporting of results

2021:

1. Implement embedded XML editing functionality within OJS
    a. Gradually retire Texture
    b. Implement Prosemirror JATS support for full-text within OJS

Ongoing:

1. Continue to participate in XML Publishing Community
    a. GROBID, Texture/Libero Editor partnerships
    b. JATS4R Working Groups
    c. Informal community discussion groups

Thank you!

Questions?

James MacGregor: jbm9@sfu.ca

Dulip Withanage: dulip.withanage@gmail.com

Vitaliy Bezsheiko: vitaliybezsh@gmail.com

Resources

GROBID: https://github.com/kermitt2/grobid, https://github.com/Vitaliy-1/grobid/

Texture: https://github.com/substance/texture/, https://github.com/withanage/texture/

Prosemirror: https://prosemirror.net/

meTypeset: https://github.com/MartinPaulEve/meTypeset, https://github.com/withanage/metypeset

Typeset plugin: https://github.com/withanage/typeset

DOCXConverter Plugin: https://github.com/Vitaliy-1/docxConverter

JATSParser Plugin: https://github.com/Vitaliy-1/JATSParserPlugin/

LensGalleyBits Plugin: https://github.com/withanage/lensGalleyBits

Who's who in JATS 2019 (Marc Bria): https://forum.pkp.sfu.ca/t/who-is-who-in-jats-2019/57063

Test Information: email to jbm9@sfu.ca, or try:

DOCX to JATS journal (DOCX to JATS, Texture, JATS Converter)

- http://xml.test.publicknowledgeproject.org/index.php/docxtojats
- User: pkpadmin / pkpadminpkpadmin

Grobid (Grobid conversion, Texture):

- http://xml.test.publicknowledgeproject.org/index.php/grobid
- User: pkpadmin / pkpadminpkpadmin
- Demo web interface: https://grobid.e-medjournal.com/

Questions!

(slide link: https://tinyurl.com/y7mtzlwv)

- Why XML? Why not (stick with PDF; Markdown; HTML; etc.)?
- What's the scope for JATS support right now?
- Is there a timeline for embedded support? AKA this is so janky. When will it be usable?