XML-first publishing for full-text, HTML articles

Current development is being tracked here: XML-first Workflow (IMH, SDH)

Workflow for publishing full-text HTML for *Studies in Digital Heritage*

This is a new approach (June 5, 2017) that does not require InDesign .

Goals

1. **Convert Word docx > XML > HTML for full text publishing in OJS. Commonly referred to as x ml-first publishing.**
2. **Use standard format for scholarly publishing: NLM/JATS DTD (defines XML elements).**

Required software

1. Python
2. Java
3. bash
4. meTypeset (docx > XML)
   Purpose-built tool to convert from Microsoft Word .docx format to NLM/JATS-XML for scholarly/scientific article typesetting
   https://github.com/MartinPaulEve/meTypeset
   a. Usage: "meTypeset.py docx <input> <output_folder> [options]"
   b. Installed on OS X machine, required additional installation of lxml and lxslt python packages `https://stackoverflow.com/questions/4598229/installing-lxml-module-in-python`
5. Oxygen (XML > xHTML)
   GitHub tools by NCIB/NLM provide a pathway for standardized XSLT transformations. https://github.com/ncbi
   a. Usage: transform XML with jats-html.xsl transformation (https://github.com/ncbi/JATSPreviewStylesheets/blob/master/xslt/main/jats-html.xsl)
   b. Oxygen "transforms" XML using the format defined by the XSL stylesheet
   c. Requires valid XML using NLM/JATS DTD, e.g., http://dtd.nlm.nih.gov/publishing/3.0/journalpublishing3.dtd
6. CSS options
   a. jats-preview.css
   b. sdh-article.css

Note: Using InDesign to export to XML requires thorough knowledge of XML, with the additional burden of configuring XML DTDs within the InDesign environment.

Comments

It will take approximately a half to a full week to revise the **meTypeset** and **jats-html.xsl** code to fully account for the elements and style used by *Studies in Digital Heritage*. Some of these changes are needed to modify the JATS standard for the SDH context as well as fine-tuning the CSS originally devised for SDH. Redoing SDH's template with all the elements defined as Word "styles" will also help (this latter work could be done by a student).

Both significant pieces of this workflow have solid histories and are emerging as best practice standards digital publishing and preservation, and are included in a PKP/OJS project to incorporate a Word to XML to HTML plugin in OJS 3 (see below). meTypeset, for example, is a fork of the TEI OxGarage (http://www.tei-c.org/index.xml) set of transformations and uses TEI as an interim step in its processing. JATS (Journal Article Tag Suite) is an application of NISO Z39.96-2015, which defines a set of XML

elements and attributes for tagging journal articles (https://jats.nlm.nih.gov). JATS is a continuation of the NLM Archiving and Interchange DTD work begun in 2002 by NCBI.

Eventually, this workflow would work best if the "meTypeset" package were hosted on library servers with a light interface to eliminate command-line requirements.

References:

Eve, Martin Paul. Building a real XML-first (XML-in) workflow for scholarly typesetting; published on July 20, 2015 https://www.martineve.com/2015/07/20/building-a-real-xml-first-workflow-for-scholarly-typesetting/

Garnett A, Alperin JP, Willinsky J. The Public Knowledge Project XML Publishing Service and meTypeset: Don't call it "Yet Another Word-to-JATS Conversion Kit". In: Journal Article Tag Suite Conference (JATS-Con) Proceedings 2015 [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2015. Available from: https://www.ncbi.nlm.nih.gov/books/NBK279666/

O'Connor C, Haenel S, Gnanapiragasam A, et al. Building an Automated XML-Based Journal Production Workflow. In: Journal Article Tag Suite Conference (JATS-Con) Proceedings 2015 [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2015. Available from: https://www.ncbi.nlm.nih.gov/books/NBK279927

Most recent JatsCon proceedings available here https://www.ncbi.nlm.nih.gov/books/.

Other Notes

1. There is a plugin in development for OJS 3 9HtmlArticleGalleyPlugin.inc.php) that may process docx in the future. Some relevant links to track it.

- JIRA task IUSW-1242
- https://github.com/MartinHinz/htmlArticleGalleyJNA
- https://github.com/pkp/ojs/blob/master/plugins/generic/htmlArticleGalley/HtmlArticleGalleyPlugin.inc.php

2. Guide and wiki for OJS 3 https://www.gitbook.com/book/pkp/ojs3/details