Metadata quality control

Members

Dulip Withanage, Heidelberg University
Tim Deliyannides, University of Pittsburgh
Nate Wright, PKP

Background

Why is this topic important?
Editors need to assure the quality of metadata before publically disseminating new works in order to improve efficiency of metadata reuse and to inspire confidence among authors and readers. Failure to check metadata quality can result in data corrections post publication and can be especially problematic once metadata is harvested and reused by others.

Goals

Produce specification details for configurable and extensible metadata validation to answer the following questions: What kinds of validation are needed? What kinds of configuration options are needed? How can we ensure rapid extensibility for technical users?

Produce a working proof-of-concept for a Plugin Category for quickly building metadata checks into the publishing process. No configuration yet, but a base to build on.

Notes

2019-07-29
Dulip Withanage, Tim Deliyannides, Nate Wright

- Define a mechanism for metadata quality control before a document is approved for publication
    - Is this a plugin? Yes
- At time of publication, run metadata validation checks
    - Present modal that lists metadata that needs to be added/corrected
- Possible validation checks:
    - Is the article already assigned to an issue? (this check is already implemented in OJS 3.2)

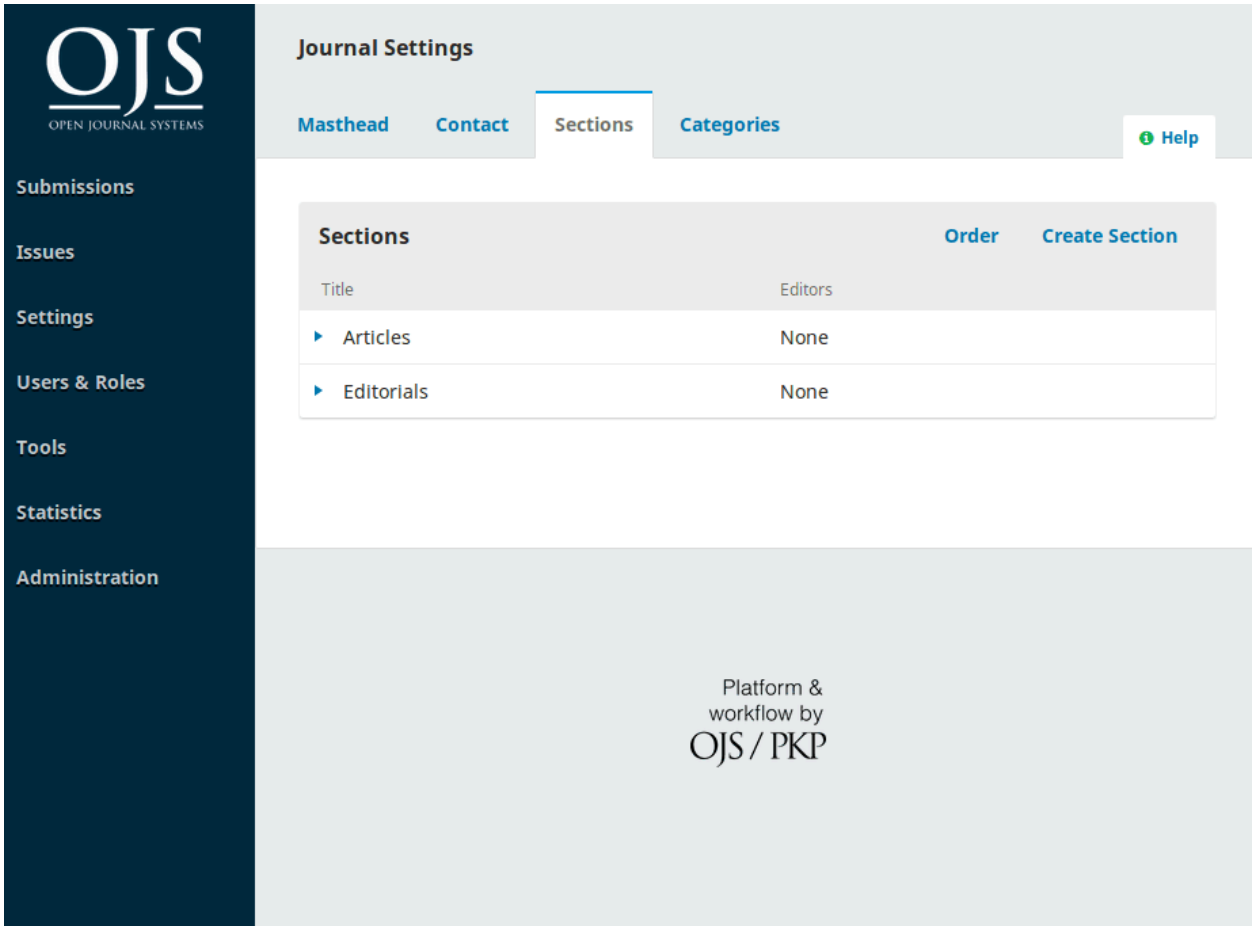
    - Check for presence of required metadata values:
        - Author
        - Title (in primary language)
        - Abstract
            - Option for editor to configure min/max abstract length requirements and validate for it.
        - Multilingual metadata
            - Option to check for required metadata in primary language only / or all languages
    - Check keywords or subjects against standard controlled vocabulary or thesaurus (which?)
    - Are all the required galley formats present? (validation based on file extension)
    - Valid, registered DOI assigned to each galley?
    - Appropriate license present?
    - CrossRef validation
- Non-metadata checks
    - Have a sufficient number of blind reviews been completed?
    - Do specific editorial decisions exist?
- Make it configurable: Give journals control over which checks are required
    - Don't want to force checks that are not relevant
    - Default configuration at journal level - which metadata checks are required?
    - Custom configuration per section (in section set-up) override which metadata checks are required
    - Optionally allow journal manager override of checks for exceptions
- Which checks should be bundled with the core? others not?
    - Define which metadata checks are in the core
    - Some metadata checks not in core, but only apply to certain plugins
- Checks should apply to all publishing functions, INCLUDING when publishing via API
- Future:
    - Should this validation occur at the submission stage?
        - Some journals may want to apply some of these checks during submission, to enforce good quality metadata from the author.
    - We need automated checks to ensure JATS metadata values exactly match submission metadata, i.e., the metadata on the published piece and the metadata in the online interface are one and the same.

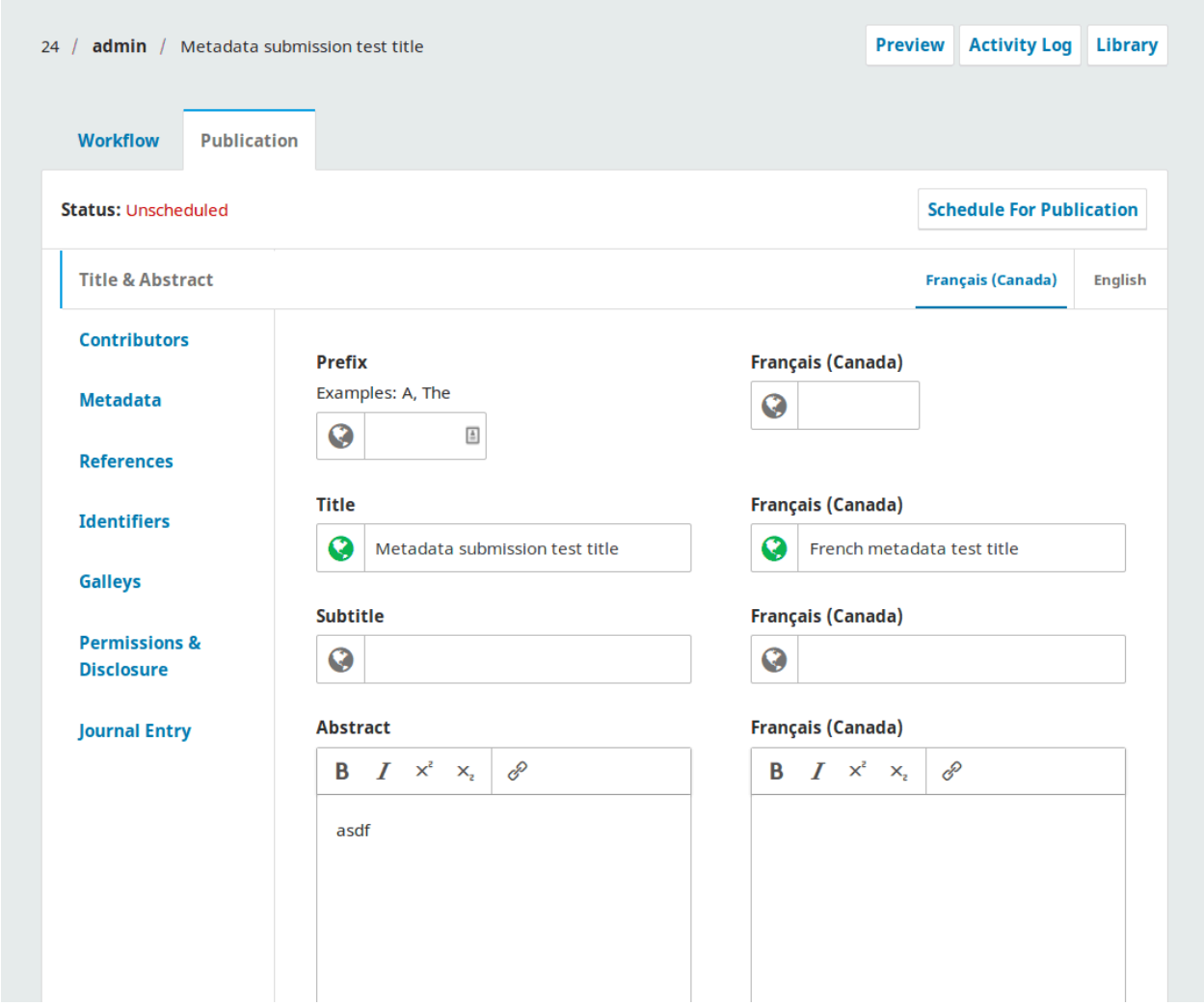
Results

We created detailed notes on specifications that would be good to have now and in the future, in order to guide the development of this feature.

We also created a new plugin category that makes it easy to quickly code custom checks before publication. We wrote two example plugins to check that an abstract exists in all languages and that keywords exist in all languages. And we added options to the section settings to disable pre-publication checks for a setting (see Next Steps below).

Technical users who want to look at this work can see the following branches:
OJS: https://github.com/NateWr/ojs/tree/prepublication_checks
pkp-lib: https://github.com/NateWr/pkp-lib/tree/prepublication_checks
(Warning, these are built off of the bleeding edge of development for the new versioning feature, and so won't be compatible with existing databases.)


OJS
OPEN JOURNAL SYSTEMS
Submissions
Issues
Settings
Users & Roles
Tools
Statistics
Administration
Journal Settings
Masthead
Contact
Sections
Categories
Help
Sections
Order
Create Section
Title
Editors
Articles
None
Editorials
None
Platform &
workflow by
OJS / PKP



Next Steps

We would like to further refine the proof of concept so that it supports some of the specifications, like applying to one or all locales, allowing senior editorial overrides, and bundling lots of checks in one plugin (ie - for standards compliance). This requires making a configuration area in the settings that makes it possible for a journal manager to configure how the checks are applied.

However, what we have now could be a basis for shipping in v3.2 that would allow technical users to write custom checks to suit the needs of their journals. It is not yet ready to unleash on end-users but could get there pretty quickly.

Technical note for future work: the disable-by-section feature is a little bit buggy. Rather than the current approach of checkboxes to enable a check, we should have checkboxes to *disable* a

check. That would make it easier to have checks enabled by default (when the plugin is enabled), so from the UX perspective there's an opt-out approach.

(Note to self: in the $errors array, the error key should match the plugin name to avoid duplicates.)