Metadata in journal publishing

Joppe W. Bos, Kevin S. McCurley

Abstract

We discuss how to use LaTeX classes and BibTeX styles to curate metadata throughout the life cycle of a published journal or conference article. Our focus is on streamlining and automating much of the publishing workflow.

1 Introduction

The original goal of TeX was to provide a system for typesetting, namely to control the layout of a document on paper. The later invention of LaTeX was focused on "letting the user concentrate on the structure of the text rather than on formatting commands" [8]. Users were encouraged to write their papers using high-level macros like `\section`, and leave the decisions like how much space to put before or after a section to the style that is used. As a result, an author does not have to worry so much about how the paper looks, but primarily about how the paper is logically structured.

This separation of concerns about appearance versus structure has proved to be very effective and most, if not all, scientific publishers now have their own LaTeX styles. These styles make it easy for an author to conform to a common look and feel in a journal, and can streamline the production steps for a journal if authors comply with the style. Moreover, it is usually easy for authors to convert from one style to another, because most of them adhere to standard macros like `\section`.

There is however at least one area in which the LaTeX community has been slow to adapt to the needs of modern publishing workflows, namely in the *curation of metadata about publications*. This is the main focus of this article.[1] We believe that a LaTeX style serves two roles; namely to provide a mechanism for describing structural information about the document, and a style for describing how to lay it out on the page.

2 Metadata in publishing workflows

When we refer to metadata, we include data objects such as title, subtitle, author names, e-mail addresses, ORCIDs, affiliations, funding agencies, bibliographic citations, journal identifier, page numbers, DOI, etc. Some of this metadata is supplied by the publisher at the time of publication, but much of it is supplied by the authors. Once the author submits their final version, this metadata is typically used to register for DOI, at which time the publisher needs to supply a considerable amount of metadata. Moreover, the web "landing page" for a paper typically has to be created from the metadata. Indexing agencies then step in, either by crawling the data or by receiving metadata feeds from the publisher. This metadata is crucial for ranking, indexing, and organization of scientific publishing.

2.1 Economics of publishing

Part of our motivation arises from our involvement in trying to launch a new open access journal for the professional non-profit society International Association for Cryptologic Research (IACR).[2] The society already runs two diamond open access journals, but experience from running these has shown that on average each published paper requires about an hour of human effort for production and metadata handling. Even then we find that errors sometimes slip through. Another study [4] estimated the amount of human labor for editing and production to be 7.5 person-hours for each published paper. We believe that most of this should and could be automated, and this can help to lower the cost of publishing. This is particularly important for open access publishing, which is heavily dependent on volunteer labor [1] as a way to control costs. It can also be used to improve profitability of commercial publishers.

In some systems, such as Open Journal Systems [12], the submission and curation of metadata is treated as a separate task from submission of the Word, LaTeX or PDF document. This imposes an extra burden on authors, and also renders the workflow vulnerable to inconsistencies with metadata in two places. In our experience, by the time an article has been revised and accepted, there are often changes in titles, abstracts, affiliations, email addresses, references, etc. Checking and correcting these inconsistencies ends up costing time of the human authors and editors.

For this reason, we believe that a LaTeX class should provide a convenient mechanism for authors to enter the metadata only once, in a standard way that encodes relationships between entities. From that point on, it should be possible to generate appropriate machine-parsable formats which can be used at every phase in the publishing pipeline.

3 Our approach at a high level

We automate the capture of metadata during the publishing workflow through the use of a LaTeX class

[1] An earlier and longer version of this article was published at `arxiv.org/abs/2301.08277`.

[2] See `iacr.org`.

doi.org/10.47397/tb/44-1/tb136bos-metadata

iacrcc.cls and a BibTeX style iacrcc.bst.[3] The function of these files is to both display the metadata in the output format, but to also extract the metadata during the compilation process, producing an easily parsable external format as a side product.

When authors supply their final versions, they do so by uploading their LaTeX source to a cloud server, which compiles their sources and extracts all metadata from their sources into a text file with a structured format (together with performing some sanity checks on the provided data). The submission process does not require authors to enter any additional metadata, because it is all encoded into the LaTeX source. The DOI suffix is assigned by the server and the DOI is compiled directly into the PDF at time of submission. A post-compilation step is used to parse the structured metadata and convert it into other formats, including JSON and XML. The DOI is registered with the DOI registration agency once the copyediting phase is complete. The extracted metadata is also used to produce various web pages for the journal site, RSS feeds, OAI-PMH feeds, and register with various indexing services.

The metadata output we require is necessarily *text*, and the lingua franca for encoding of text is UTF-8. With the exception of mathematical structures like inline equations in titles or abstracts, this text is devoid of TeX macros. This causes a few problems in the LaTeX world, which encourages authors to write in 7-bit ASCII text with user-defined macros.

Part of our problem arises from the fact that TeX takes the input format and produces a list of tokens. This sequence of tokens is convenient to produce a list of boxes containing glyphs for layout on pages, but extraction of the author's original text from that token list is problematic. For example, spaces are not space characters but are instead glue between boxes or terminators for macros.

In addition, a core functionality of LaTeX is user-defined macros, so an author might define \pe to represent the text string "Paul Erdős". We only discover this during the LaTeX expansion process when the macros are expanded into glyphs. Macro expansion is one of the most difficult topics in understanding how TeX works.

Our first implementation of metadata capture used the \write macro during the LaTeX compilation process to write an external file containing metadata. The intended function of the \write macro is to expand a list of tokens and write a parsable representation of these tokens into a file. The fact that \write performs expansion is very useful to us, because it expands user-defined macros. Unfortunately \write also causes a few problems when we use it to produce metadata. As an example, \(and \) cannot be used to delimit inline mathematics inside \write, whereas $ works fine.

Another problem arises with pdflatex, because we have found examples like \write{Đ and f\"ur} where the output from \write contains mixed character encodings in a single line. This is apparently due to the fact that while pdflatex handles UTF-8 input, the output tries to use the single-byte Cork encoding for things like ü. For this reason we switched to using \protected@write instead of \write, following a suggestion from the LaTeX team.

One might argue that the author can correct the previous example by avoiding mixed encodings in their input, but this is merely one example of many ways that authors can produce legitimate LaTeX that is difficult to deal with. Our goal is to provide a system that supports whatever legitimate LaTeX the author supplies to us, and to provide them with clear instructions on how to prepare it without causing any interruptions (errors) or other inconvenience to the author's typesetting experience. From an author's point of view, the flexibility of LaTeX can be a blessing, but it's also often a curse for a journal.

3.1 Alternative approaches

We considered several ways to implement the metadata extraction instead of using \write. One alternative approach would be to use a LaTeX parser to extract the metadata directly from the LaTeX. The problem of parsing LaTeX is complicated by the need to expand macros, for which the LaTeX engines themselves are so far the only robust solution. Another approach that we considered involved using Lua within lualatex. Lua is much better suited to text processing than using LaTeX itself, but we had an initial goal to try and make things work with any LaTeX engine.

4 What metadata is required?

Some metadata fields in a journal article are obvious (title, author), but even the obvious fields have nuances in how they are encoded. Examples include:

- Title of the work. In some fields it is commonplace to use mathematics in titles, but TeX formatting in metadata records is often changed to another format like MathML. Titles may also encode face markup (e.g., bold face) or multiple

[3] The authoritative place to download these is publish.iacr.org/iacrcc.

character sets. Extremely long titles are sometimes broken up into a hierarchy, incorporating a subtitle or short versions for running titles.

- Authors of the work. One reason to ask for authors is to give proper attribution in citations, but author names are not unique so we should also use a unique identifier like ORCID.
- Authors may have different levels of contribution. In some cases this is signaled by having author names out of alphabetical order, but in other fields it is common to identify a *role* for author contributions. The CRediT taxonomy is often used to reflect this [11]. Authors may also be categorized as a "corresponding author", with contact information like email.
- Relationships between authors and affiliations and/or authors and funding agencies. It is now very common for authors to have multiple affiliations [6] and for multiple authors to share a subset of affiliations or funding agencies. These many-to-many relationships are best encoded as relations rather than repeating the information for each author. These relationships are shown in Figure 1.
- Bibliographic information (e.g., journal or conference name, volume, year, etc.).
- The list of bibliographic references.
- Submission and acceptance dates.
- Licensing information.
- Funding information.

There are numerous other fields that may be encoded into a LaTeX document or the output format produced from LaTeX. Examples include abstract, number of pages, address information for authors, links to ancillary works like code and data, etc. We come from the world of mathematics and computer science, but other things like chemical structures and clinical trials can also be encoded into metadata. It is beyond the scope of this document to catalog all of them, but rather to focus on the most important elements that are common to all academic disciplines.

4.1 Metadata schemas

Several organizations have defined schemas for the organization of metadata about an article. One of the most important ones is `crossref.org`, which is a non-profit organization whose primary mission is the collection of metadata and the assignment of DOIs. Their schema supports multiple affiliations, author roles, and funding agencies. Other formats include Elsevier's Scopus indexing service and the Clarivate Web of Science.

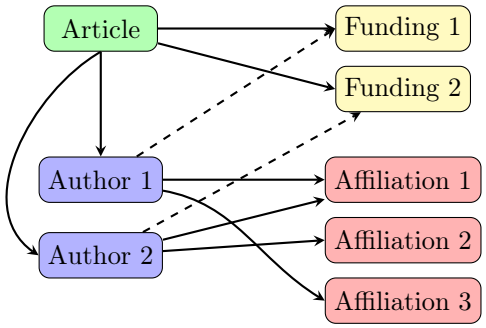

Figure 1: Relationships between major entities. Each entity is listed only once in the LaTeX source. An article may have multiple authors who share relationships to affiliations. Funding agencies are related to the article in the crossref schema, so we chose to link them this way. As an alternative, relations shown with dashed arrows can link authors to their funding sources, in much the same way that we relate authors to their affiliations. We chose to use footnotes to clarify the complex relationships between funding agencies and authors or affiliations. Some funding agencies (e.g., [10]) have strict guidelines for how these annotations should be shown in the paper.

Another important schema is the Journal Article Tag Suite (JATS), which is available in three variations for archiving & metadata, publishing, and authoring [9]. The JATS format may be viewed as a complete structural representation for a publication; in many ways comparable to LaTeX but focused even more on semantic structure rather than typesetting or layout. A JATS document consists of several sections, including front matter, body, and back matter. Most metadata occurs in the front matter and back matter.

There are numerous other formats, but these tend to be less descriptive and incomplete. These include the Dublin Core, the Directory of Open Access Journals (DOAJ), the Extensible Metadata Platform (XMP) that is common in PDF, and PRISM. Among all these alternatives, we found the JATS format to be the most expressive and consistent with others.

5 Using unique identifiers

Unfortunately, things like human names and institution names are not unique identifiers. The DBLP bibliographic website lists 14 authors in computer science who use the exact name "Thomas Müller", and dozens of others that are similar to this, like Thomas F. Müller. There are multiple institutions that go by the names MIT or USC. In order to perform large scale bibliometric analysis for attribution

or duplicate detection, all entities associated with a publication need to be assigned a unique identifier.

Many of the XML schemas such as JATS have embraced the use of unique identifiers. The most notable efforts to assign unique identifiers include:

- DOIs for publications [13],
- ORCIDs for authors [5],
- ROR IDs for research institutions [7],
- Crossref funder registry for funding agencies [2].

Note that in each case where an organization has assigned a unique ID to an entity, there will often be competing organizations with their own ID space. For example, other identifiers for authors have been issued by Clarivate Web of Science, Scopus, SciENcv, Mathematical Reviews, and DBLP.

ROR IDs have coarse granularity, so while there is an identifier for Massachusetts Institute of Technology, they don't distinguish between departments, schools, or programs of the university. By contrast, Mathematical Reviews assigns institution codes at the department level (e.g., 1-SCA-C for the department of computer science at University of Southern California).

A complete list of identifiers associated with scholarly publications is beyond the scope of this document, and we should expect future ID systems to emerge. Because an entity may have multiple IDs from different organizations, we strongly recommend a schema that assigns IDs with a namespace and identifier within that namespace. Thus for example, an organization may have both a Ringgold ID and an ROR ID. Including both can be helpful.

6 Output formats

In our processing, LaTeX is not usually read by humans, but is instead converted into another format like PDF or HTML. To the extent possible, it is desirable to embed the metadata into these output formats in a machine-readable way so that the metadata accompanies the consumable document. Unfortunately the standards for doing so are generally lacking in comprehensiveness.

Probably the most important example of this is the XMP standard, whose standard schema does not even provide a way to identify authors by ORCID. Luckily, as the name implies, this format is extensible, and the XML dictionary may use a schema from a variety of namespaces [14]. Springer does this for ORCIDs by defining their own namespace `sn` and encoding authors as a sequence of `(name, orcid)` pairs. Rather than embracing proprietary extensions such as this, we believe that XMP should use the JATS schema to encode authors, affiliations, funding agencies, and bibliographic references. Unfortunately this is not supported by the `hyperxmp` and `pdfx` packages, but the LaTeX team is engaged in a long-term project to improve the production of XMP in PDF [3].

7 `\author` considered harmful

We now turn to the problem of how to embed metadata into the original LaTeX source. The original LaTeX definition of `\author` provides little help in capturing author metadata, and is also problematic for displaying large numbers of authors. In the standard `article` class, the author defines `\author` to include blocks of formatted text, separated by `\and`. Thus for example, there is no standard way to associate an ORCID with an author's name, or to associate affiliations or funding agencies with an author. Left to their own devices, authors might use various embedded macros or footnotes to link authors to their metadata, and this makes it very difficult to extract metadata from the LaTeX.

Part of the problem here is that the `\author` macro is intricately woven into the *display* of author information on the page. This is an example where the separation of concerns has been neglected, mixing structure with display. Because of this past history with the `\author` macro, we deliberately chose to break `\author` and use `\addauthor` instead. This means authors have to do some work to convert from other standard LaTeX classes to our class, but we judged that to be necessary because of the bad habits that LaTeX has encouraged.

We are not the first to have recognized the deficiency of `\author`. Some LaTeX styles have improved upon the basic use of `\author`, and have adopted metadata capture as part of their authoring process. Examples include `ltugboat`, `elsarticle`, `acmart`, and `amsart`. Each of these uses some variation on `\author` to capture some metadata about an article, but none of them rise to the level of expressiveness contained in something like JATS. Moreover, we are unaware of any that have attempted to provide functionality for a publishing workflow by extracting the metadata from the LaTeX. Publishing workflows tend to be proprietary, but most use significant human labor that is covered by their business model.

8 The `iacrcc` LaTeX and BibTeX styles

Building on what we have learned from previous efforts, we have designed a new document class called `iacrcc`[4] that allows us to capture as much metadata as possible from a document. This may be used with either BibTeX with our own `iacrcc.bst` style, or with the `biblatex` package. These files are designed

[4] May be downloaded from `publish.iacr.org/iacrcc`.

to be used in a publishing workflow to produce metadata in several different formats. Not only do they produce metadata to go back into PDF, but they also produce a plain text version of metadata that can be easily processed for other purposes like DOI registration. We capture a broad range of metadata, including alternate titles, author names, surnames, ORCIDs, affiliations with ROR IDs and addresses, and abstract. An example of author metadata for `iacrcc` is given in Figure 2.

```
\title[running={Emojex documentation},
       onclick={example.com/emo},
       subtitle={Faces in unicode},
      ]{Emojex: use of emojis in \LaTeX}
\addauthor[orcid={0000-0002-0599-0192},
           inst={1,2},
           onclick={www.madmagazine.com/}
           email={fester@example.com},
          ]{Fester \surname{Bestertester}}
\addauthor[orcid={0000-0001-7890-5430},
           inst={2},
           footnote={Thanks mom!},
          ]{Kevin S. \surname{McCurley}}
\affiliation[ror=044t1p926,
             city={New York},
             country={United States}]{MAD}
\affiliation[country={United States}]{Self}
\addfunding[crossref=100011047,
            grantid={A-1234},
            country={Canada}
           ]{AGE-WELL}
```

Figure 2: Sample metadata entry in `iacrcc.cls`.

8.1 How it works

The workflow for an author consists of the usual multiple rounds of running `latex`, `bibtex` or `biber`, followed by two more runs of `latex`. The output from this is not only a PDF file with XMP metadata, but also a file `\jobname.meta` file that contains all metadata in a structured format. The `.meta` file is written with macros using `\write` calls.

The structure of the `\jobname.meta` is similar to YAML. We thought about attempting to write YAML or JSON or XML format, but each output format has its own set of special characters and encoding requirements that are complicated to achieve in LaTeX. It was easier for us to write Python code to parse our custom output format than to write LaTeX code to produce one of the more common formats. This Python code is included in the repository for the `iacrcc` files.[5]

[5] See the github repository at `github.com/IACR/latex`.

The basic metadata from the paper is written to the `.meta` file using macros from the `iacrcc.cls` file. The citation information is written into the `.meta` file in one of two different ways, depending on whether the author chooses to use BibTeX or `biblatex`. Both methods produce a `.bbl` file that contains `\write` macros to append to the `.meta` file during compilation. The `\write` macros are implemented in the `iacrcc.cls` file for `biblatex`, and are implemented in the `iacrcc.bst` file for BibTeX. In both cases, the `.bbl` file ends up containing a structured form of the citations. In theory, this allows us to follow the standard practice of publishers to only require authors to submit their `.bbl` file rather than their entire BibTeX file. In practice we require authors to submit their BibTeX because there is no convenient way to validate the `.bbl` file.

8.2 The submission pipeline

Once a paper has been accepted for publication, the authors need only submit their LaTeX source file(s), including the BibTeX file they used. The submission form is minimal, since all metadata is included in the LaTeX and BibTeX files themselves. We merely capture an authenticated `paperid` and require the submitting author to supply an email address for the contact author. We derive the DOI from the `paperid`, and inject it into the PDF during compilation along with the acceptance and received dates.

Once the authors upload their LaTeX sources, the server runs `latexmk` within a docker container containing an instance of TeX Live. The server validates that the sources were compiled, and provides reports back to the author in case of any errors. We plan to release our server code as open source in the future, but it's premature to do so now, since some basic design decisions are still being made.

Once the document successfully compiles, the server runs a Python script to process the `.meta` file, creating metadata in XMP, JATS, JSON, and crossref formats. The JSON format is convenient for immediately publishing the article on the web. The crossref format may be used to register the paper with a DOI.

If the author is satisfied with the output from compiling their source, then the paper moves to the next step of copyediting. Copyediting is itself a huge topic in publishing that is mostly beyond the scope of this article. In our experience with external publishers, some of the effort is devoted to metadata handling. Our goal is to at least completely automate metadata handling.

Once the paper is given final approval by the copyeditor, the paper may be published without need

for a human to handle any of the metadata. At the time the paper is published, the DOI is registered.

9 Summary

We believe that LaTeX can be used to simplify the processing of metadata in the publishing process, and we have developed a document class that we hope will greatly improve the quality of our metadata. By using this approach, we believe it should be possible to streamline the publishing workflow of an open access journal with a low budget. We are in the early stages of this project, and we welcome suggestions for better ways to capture metadata.

Metadata handling is just one reason why text extraction is important for LaTeX. We are in the midst of a revolution in natural language processing through the development of machine learning for large language models. We are hopeful that this will give rise to better tools for tasks such as copy editing. This includes some fairly mechanical steps like punctuation, spelling, and grammar checking. It may also involve visual aspects of typography (e.g., widows, orphans, under/overfull hboxes). It can also involve more intensive steps like checking consistency in terminology, optimizing word choices, or improving sentence structure.

Unfortunately, one barrier to the use of large language models with LaTeX is the fact that it is relatively difficult to extract the author's text from LaTeX. We encourage the community to think more about this problem — not just within author environments or PDF output, but also within publishing pipelines.

Acknowledgements

The authors would like to thank Gaëtan Leurent and other contributors to the `iacrtrans` document class that was used as the starting point of this project. The authors would also like to thank Enrico Gregorio and David Carlisle for answering questions about the inner workings of LaTeX, and to the reviewers of this paper for making very useful suggestions.

References

[1] J. Bosman, J.E. Frantsvåg, et al. OA diamond journals study. Part 1: Findings, Mar. 2021. This report was supported by Science Europe and cOAlition S. doi.org/10.5281/zenodo.4558704

[2] Crossref funder registry. crossref.org/services/funder-registry/

[3] U. Fischer, F. Mittelbach. Adding XMP metadata in LaTeX. *TUGboat* 135(3):263–267, 2022. doi.org/10.47397/tb/43-3/tb135fischer-xmp

[4] A. Grossmann, B. Brembs. Current market rates for scholarly publishing services. *F1000Research*, 2021. doi.org/10.12688/f1000research.27468.2

[5] L.L. Haak, M. Fenner, et al. ORCID: a system to uniquely identify researchers. *Learned Publishing* 25(4):259–264, 2012.

[6] H. Hottenrott, M.E. Rose, C. Lawson. The rise of multiple institutional affiliations in academia. *Journal of the Association for Information Science and Technology* 72(8):1039–1058, 2021. doi.org/10.1002/asi.24472

[7] R. Lammey. Solutions for identification problems: a look at the research organization registry. *Science Editing* 7(1):65–69, 2020.

[8] L. Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley Publishing Company, first ed., 1986.

[9] National Center for Biotechnology Information (NCBI). Journal publishing tag library NISO JATS version 1.3. jats.nlm.nih.gov/publishing/tag-library/1.3/, June 2021.

[10] National Institutes of Health. Communicating and acknowledging federal funding, 2021. grants.nih.gov/policy/federal-funding.htm

[11] NISO. CRediT: Contributor roles taxonomy. credit.niso.org

[12] Open Journal Systems. pkp.sfu.ca/ojs/

[13] N. Paskin. Digital object identifier (DOI®) system. *Encyclopedia of library and information sciences* 3:1586–1592, 2010.

[14] *Technical Note 0009: XMP Extension Schemas in PDF/A-1*. www.pdfa.org/resource/technical-note-tn-0009-xmp-extension-schemas-in-pdfa-1/

◇ Joppe W. Bos
joppe.bos (at) nxp dot com
ORCID 0000-0003-1010-8157

◇ Kevin S. McCurley
iacrcc (at) digicrime dot com
ORCID 0000-0001-7890-5430