Primo — The new sustainable solution for publishing

Rishikesan Nair T, Apu V and Hàn Thế Thành, Jan Vaněk

Abstract

Primo is a cutting-edge, cloud-based authoring, submission, and proofing framework that provides a sustainable solution for academic publishing. It combines the advantages of XML-based workflows that facilitate controlled authoring and/or editing in accordance with specific DTDs and house styles, with the visually appealing and mathematically precise typesetting language of TEX, enabling the creation of high-quality PDFs and mathematical images (offering an alternative to MathML coding).

By speaking the widely accepted communicating lingua of mathematics and science (i.e., TEX), and utilizing the XML/MathML format for archiving, Primo has the potential to revolutionize the publishing industry. This tool caters to both the author and the publisher, bringing their needs together with enhanced participation of authors in the publishing process. The three main modules of Primo include Authoring, Submission/Reviewing, and Proofing, all of which are equipped with usability checks during submission, a collaborative editing feature, a WYSIWYG math editing tool, and publisher/journal-based PDF manuscript rendering. With Primo, authors can be assured that their work will be published with the highest level of precision and quality.

1 Introduction

Primo, the latest addition to the lineup of TEX-based tools, is developed by STM Software Engineering Pvt Ltd. who is a specialized TEX typesetting house renowned for its top-notch typesetting and pre-press services, catering to the needs of STM publishing giants specialized in the complex article typesetting. With its state-of-the-art technologies, STM Software Engineering Pvt Ltd. developed a range of cloud-based typesetting frameworks, including TEXFolio [1] and Ithal [2], primarily designed for in-house typesetting and format conversion purposes within publishing houses. On the other hand, Neptune [3] and Primo target authors directly, providing them with efficient and user-friendly TEX-based tools.

2 Primo

Primo's modular structure and well-designed tools enable authors to navigate the entire publication journey with ease, from initial authoring to final proofing. By integrating these three modules, Primo optimizes the authoring, submission, and proofing processes, making it a comprehensive and efficient platform for scholarly publishing.

2.1 Authoring tool

An intriguing offshoot of the aforementioned processes is a stand alone authoring tool, codenamed Primo Editor. This tool encompasses all the necessary elements to effortlessly compose an article that meets all technical requirements for seamless uploading to a publisher.

2.1.1 Salient features

Please note that while the below list covers the main features, there are undoubtedly additional features yet to be mentioned that further enhance the functionality and user experience of the tool.

1. **Collaborative Editing:** Multiple authors can contribute simultaneously.
2. **Operating System Independence:** The tool shall be operating system independent and will have a WYSIWYG interface.
3. **Cloud-Based with TeX Installation:** A completely cloud-based version with a TEX installation comprising of essential packages, fonts, compiler and utilities. The user need not worry about installation or setting up of a TEX distro in their local system.
4. **Proper template:** With the Primo authoring tool, authors no longer need to spend time searching for the appropriate template for their submissions. The tool offers a comprehensive collection of templates, and by simply providing the name of the journal and publisher, it automatically selects the most suitable template. This eliminates the hassle of manually locating the correct template and ensures that authors can focus on their content without the added burden of formatting.
5. **Math Input:** Users can enter mathematical equations using LATEX syntax or utilize built-in math tools from the menu.
6. **Form-Like Interface:** A user-friendly form-like interface is available to capture front matter data such as author information, affiliations, abstract, keywords, graphical or stereo-chemical abstracts, and more.
7. **WYSIWYG Interfaces for Tables and Figures:** WYSIWYG interfaces for entry of table and figure data.

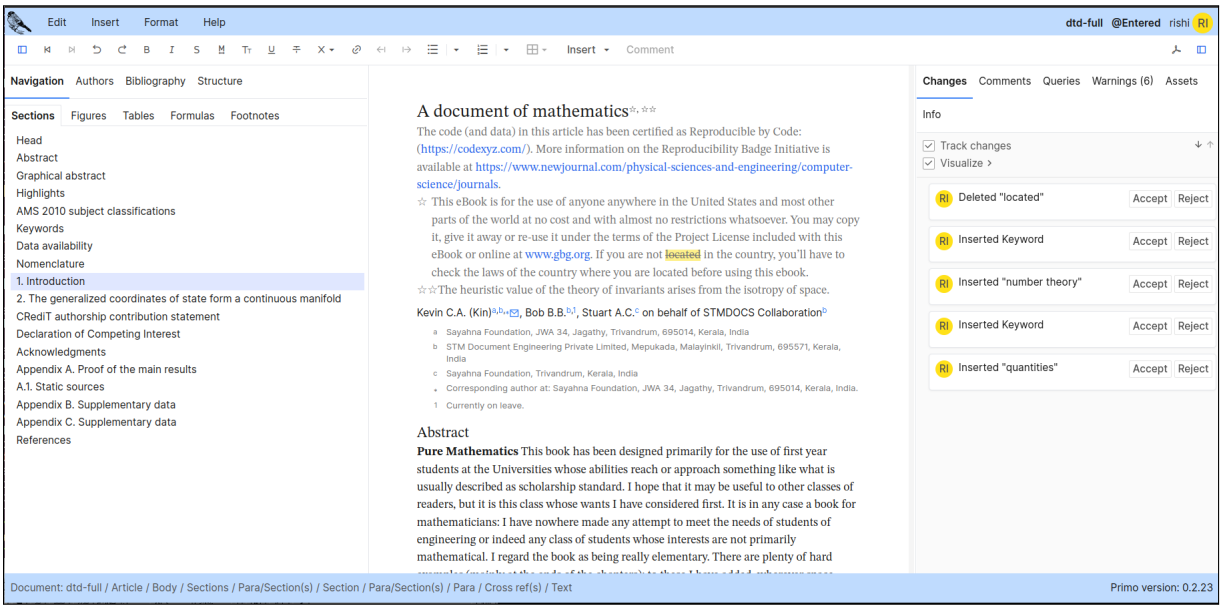


Figure 1: Primo: The main page.

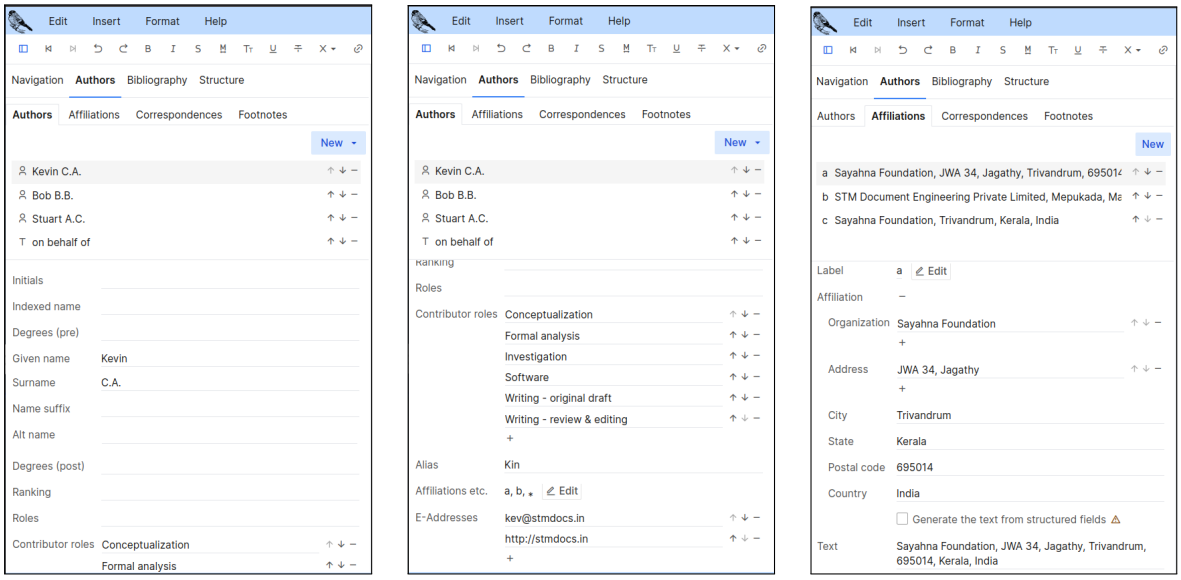
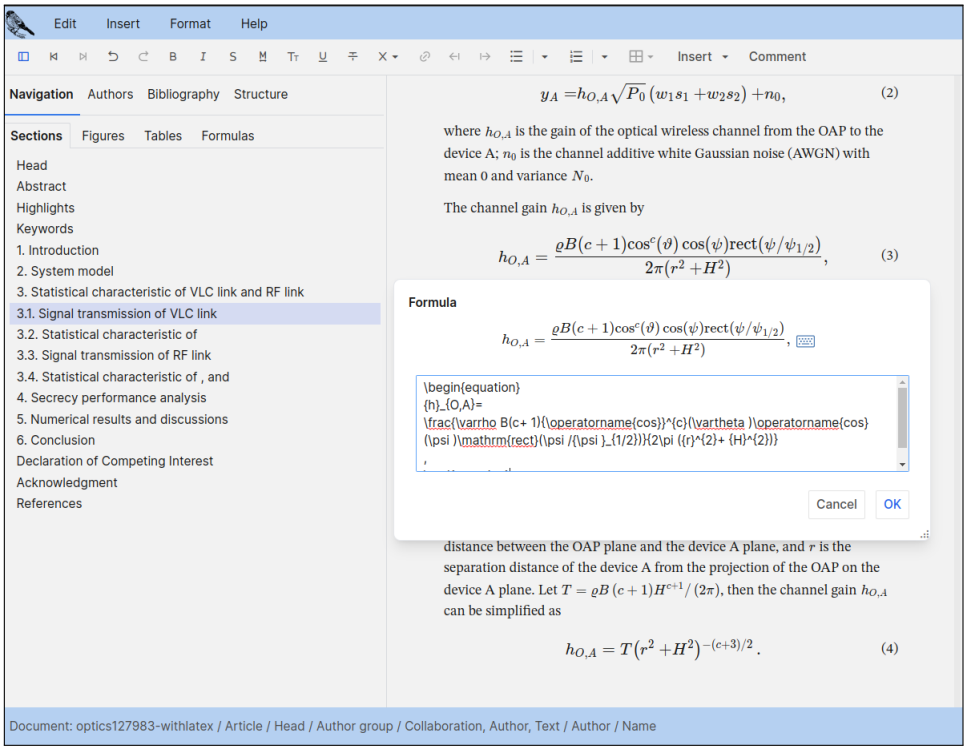


Figure 2: The author and affiliation field.



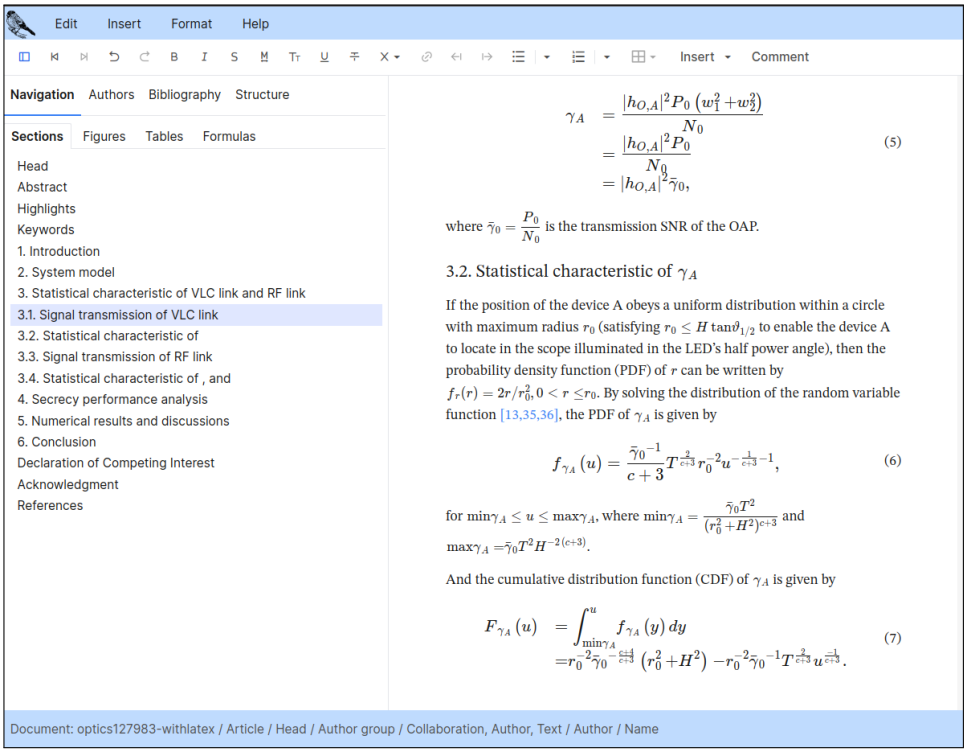


Figure 3: Math content rendering.

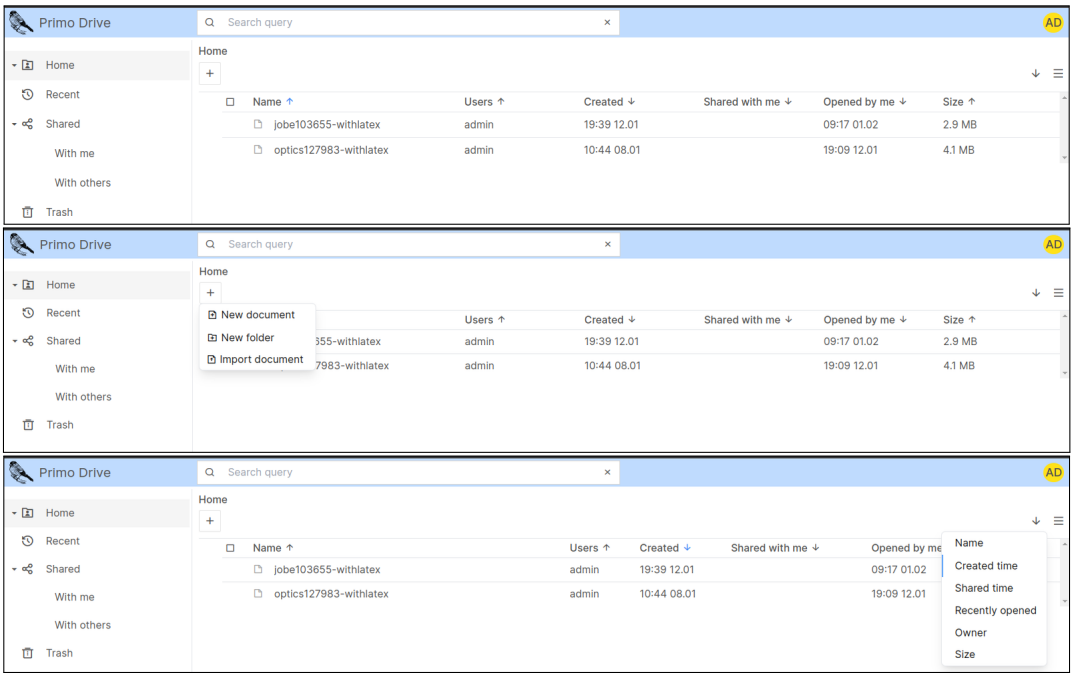


Figure 4: The primo drive.

8. **BibTeX Support:** BibTEX data is always welcome if the user prefers to use the same.
9. **Bibliography Data Checking:** Checking bibliography data with Cross-ref is an added benefit.
10. **Bibliography Import:** With the help of Primo, users can easily import bibliography data using identifiers such as DOIs (Digital Object Identifiers) or PMID (PubMed IDs), streamlining the referencing process.
11. **Enhanced Author Participation:** The tool promotes active author involvement in the publishing process, minimizing errors and semantic inconsistencies introduced by typesetters. This enhances the overall quality and reduces the time gap between submission and publication.
12. **Compliance Checking:** The tool automatically checks the manuscript's adherence to the specific style guidelines of the publisher or journal, ensuring compliance with formatting requirements.

These features collectively provide a comprehensive and user-friendly platform for collaborative manuscript preparation, improving the efficiency and quality of the publishing process.

2.2 Submission tool

The submission process for authors can often be arduous and time-consuming. With strict timelines and numerous procedures to navigate, authors often find it challenging and frustrating. Primo seeks to alleviate these difficulties faced by the author community. In its initial stage, this authoring tool assists authors in crafting their manuscripts in compliance with the specific style requirements of publishers and journals. The subsequent step involves a seamless transfer of the source files to the submission system employed by publishers.

2.2.1 Salient features

1. **Source files:** Since the source files were already prepared as per the specification using the authoring tool, there will not be any surprises in the submission system.
2. **Proper submission:** Transferring files directly from author tool to submission tool and finally to the publishers' submission system helps to eliminate chances for any missing files or materials.
3. **File category:** In the popular submission systems, we have to select the file type of each source file that we upload. For example, "Manuscript", "Revised Manuscript", "Figure", etc. Primo tool helps to sort this out easily and helps authors to identify which is which.
4. **Usability check:** Depending on the compliance of the submission system to which finally the source files are uploaded for the publisher,

the Primo submission tool runs a custom-made usability check on the source files and reports problems any problems with the source files. This will help those authors who prepare manuscript in their system or any other interface and directly upload source files to the Primo submission tool.

2.3 Proofing tool

The proofing tool plays a crucial role in Primo. Once the typesetting process is complete, the typesetter uploads or imports a dataset that includes the article's XML/MathML, figures, supplementary materials such as audio, video, program codes, and alternative images for MathML, among others.

The format of the dataset is simple and it looks like:

```
Archive:  ENDEND_99996.zip
  Name
  ----
  ENDEND_99996.pdf
  main.assets/
  main.assets/fx1006.jpg
  main.assets/gr1.jpg
  main.assets/fx999.jpg
  main.assets/gr2b.jpg
  main.assets/mmc1.pdf
  main.assets/fx1001.jpg
  main.assets/fx1004.jpg
  main.assets/fx1.jpg
  main.assets/fx1005.jpg
  main.assets/fx1002.jpg
  main.assets/gr2a.jpg
  main.assets/fx1003.jpg
  main.assets/fx1007.jpg
  main.pdf
  main.stripin/
  main.stripin/si33.svg
  main.stripin/si121.svg
  main.stripin/si165.svg
 .....................
 .....................
 .....................
```

2.3.1 Salient features

1. **Track changes:** This facility helps the authors to understand the changes made by the typesetters in their document. The accept and reject buttons can be used to accept or reject the changes made by the typesetters. The visualize mode in the track changes further has two features. They are to visualize the changes made by the authors and to visualize the changes made by the copy-editors or co-authors.
2. **Comment:** If authors are unsure about how a change is to be done, then the comment facility can be used to put a comment.
3. **Queries:** Queries raised by the typesetters or copy-editors are available in the query panel. Authors can provide reply to the queries just below the queries.
4. **DTD compliant:** All the features allowed by the DTD can be used. For example, they cannot insert a figure in the author field since the DTD does not support that.
5. **Changing the order:** Order of author names or position of given and surname or just the content can be swapped.
6. **List:** Changing the formats of the list counters or when inserting a new list, selecting the format of the counters is just easy by using the drop-down menu facility.
7. **Intelligent insertions:** Since the tool follows the DTD faithfully, the insert menu will show the items according to the content model only. So if you are in a paragraph, the drop-down list will show sections, math (both inline and display), list, display quote, etc. However, if you are in the reference section, the insert menu will show bib-entry, bibliography section, etc.

3 Technologies behind Primo

Primo is written mostly in Scala, both server-side and client-side. The client-side is compiled using Scala-JS to JavaScript. On the server-side, Scala is compiled to Java byte-code and runs in the JVM. It can seamlessly inter-operate with existing Java libraries. The development environment is Intellij IDEA, the build-tool is SBT. Primo uses its own widget library called VDL, part of the Primo code base.

Primo doesn't have many external dependencies. We use following "major" libraries:

- JDK obviously
- undertow - the web-server, like tomcat, but smaller
- sqlite - for the DB
- lucene - full-text index of the documents

And some "minor" libraries:

- xpp3 - XML parser
- scala-js DOM, java-time, java-logging
- boopickle, scala-css, and some others

References

[1] Rishikesan Nair T., Rajagopal C.V., and Radhakrishnan C.V. TEXFolio — a framework to typeset XML documents using TEX. TUGBoat

40(2), 147–149. 2019.
https://tug.org/TUGboat/tb40-2/tb125rishi-texfolio.pdf

[2] https://ithal.io/main.html

[3] Aravind Rajendran, Rishikesan Nair T., and Rajagopal C.V. NEPTUNE — a proofing framework for LaTeX authors. TUGBoat 40(2), 150–152, 2019.
https://tug.org/TUGboat/tb40-2/tb125rajendran-neptune.pdf

◇ Rishikesan Nair T, Apu V
STM Document Engineering Pvt Ltd.,
River Valley Campus,
Mepukada, Malayinkizh,
Trivandrum 695571,
India
rishi (at) stmdocs dot in, apu.v (at) stmdocs dot in
https://stmdocs.org

◇ Hàn Thế Thành, Jan Vaněk
Trivic s.ro., Družstevni 161, 763 15 Slušovice, Czech Republic
thanh (at) trivic dot io, jan (at) trivic dot io
https://trivic.io