Speech Synthesis Integration in Open Journal Systems

Peter Vedal Utnes and Obiajulu Odu

Abstract

This case study addresses the issue of poor accessibility and speech synthesis compatibility in research papers published as PDFs. We conducted a pilot study using EPUB format in a selection of our university's open-access journals, employing a template designed for speech synthesis compatibility. Initially, we tested markup designed for speech synthesis software but found it was too labour-intensive. Subsequently, we developed a workflow using a document template with wider compatibility and good universal design. This approach simplifies ongoing management and ensures accessible files with appropriate metadata. Other publishing services, especially those that use Open Journal Systems, can replicate our workflow.

Keywords

Speech synthesis, accessibility, Open Journal Systems, OJS, document conversion

Introduction

In this pilot project, we focused on improving the accessibility of academic journal articles by ensuring compatibility with speech synthesis (text-to-speech) software. For our purposes, accessibility and universal design mean creating documents that are easily adaptable, ensuring that article content can be converted into audible speech and meets the needs of a diverse range of users. The goal is to make content readily available for students, staff, and others who benefit from listening to the content.

Open science removes legal and economic barriers, but it also requires practical accessibility. Paywall removal does not by itself make research usable for readers who depend on assistive technologies. To be genuinely open, articles must be machine readable and interoperable with speech synthesis and related tools. The EPUB-based workflow described below operationalises this requirement in Open Journal Systems (OJS) by reducing the remaining functional barriers after open access is achieved.

This pilot is part of a broader project assessing the potential for speech synthesis compatibility of the papers published in the academic journals of Septentrio Academic

Publishing. Septentrio is managed by the University Library at UiT The Arctic University of Norway, and the journals are hosted on an OJS platform.

In accordance with the Norwegian Universities and University Colleges Act (§ 10.2) and the Equality and Anti-Discrimination Act (§ 17 and § 18), which mandate universal design for digital services, both public and private universities and university colleges are required to ensure that their publications are accessible to all, including individuals with visual impairments. Additionally, some journals incorporate Plan S compliance as part of their mandate, which not only ensures open access but also requires that articles be machine-readable – preferably in JATS XML ("Plan S" and "cOAlition S" – Accelerating the Transition to Full and Immediate Open Access to Scientific Publications, 2024).

The articles in Septentrio have previously been published mostly in PDF format, which for various reasons is not ideal for accessibility tools, especially speech synthesis. Specifically, PDFs do not adjust well to different screen sizes, which can hinder readability on various devices, and they often lack semantic content, making navigational 'tabbing' and interaction with assistive technologies challenging.[1]

We aimed to develop a simple workflow, including document templates, for the conversion of submitted manuscripts from Word format to XML, specifically EPUB (Electronic Publication). EPUB documents can adjust to different screen sizes and resolutions, improving readability and meeting accessibility standards. Additionally, many speech synthesis software programs support EPUB. The main challenge involves handling various input types, namely manuscripts written to align with different journal templates, and converting them into standardized EPUB documents that meet the needs and preferences of all users and software. This option was attractive not only because of EPUB's reflowable design, but also because it aligned with the University Library's intention to introduce XML-based workflows.[2]

Our focus has been on a handful of new articles in three of Septentrio's journals (Målbryting, Nordic Perspectives on Open Science and Ottar). We tested various solutions on an internal test server and ultimately developed a workflow using open-source tools such as Pandoc for document conversion and specific plugins integrated into OJS (Open Journal Systems).

OJS is open software used by many publishers and publishing services in Norway and internationally. Therefore, our solution can be adopted by other publishing services.

Our Approach

The project progressed in two main stages. First, we tried to adapt documents to include specific markup for speech synthesis software. We employed SSML (Speech Synthesis Markup Language), the CSS Speech Module, and pronunciation lexicon specification (PLS) for correct pronunciation (CSS Speech Module Level 1, 2023; Paolo Baggia et al.,

---

[1] Although PDFs can be made accessible through proper tagging and adherence to standards such as PDF/UA, achieving this level of accessibility with free, open-source tools typically requires significant additional effort

[2] We also had a parallel, separate project digitizing the journal Ottar that employed XML in its workflow.

2010; Paul Bagshaw et al., 2008). When this approach proved too labour-intensive, we switched to an automatic conversion from Word to standard EPUB. This conversion relied on standardized document templates designed for accessibility. Our goal was to ensure that documents were created in a way that required minimal ongoing support from Septentrio staff, while meeting Plan S and accessibility requirements, and speech synthesis compatibility.

Using Files Specifically Adapted for Speech Synthesis

Our initial method involved converting Word documents to Markdown and then transforming the Markdown into XML with SSML formatting. SSML provides control over aspects like pronunciation, pitch, speed, and volume. However, this method was not compatible with our journal templates and required significant post-processing.

We then experimented with converting Word documents to HTML and applying a stylesheet that specified parameters for both the CSS Speech Module and PLS dictionaries. The CSS Speech Module controls how a webpage is rendered in speech, while PLS help guide word pronunciation. Developing custom PLS, however, requires extensive manual effort, specialized linguistic expertise, and ongoing maintenance for each language. This approach might be feasible for a journal dedicated to one language but is less practical for a diverse range of journals. In addition, HTML is not easily downloadable as a single file, which limits its practicality for some applications.

One advantage of using stylesheets is that they can be customized once and then applied across multiple documents, reducing the need for individual adjustments. Currently, very few speech synthesis programs support the CSS Speech Module. Most software ignores these settings, as users often override them with their own preferences. This was confirmed in discussions with researchers and students who use speech synthesis daily.

Adapting stylesheets for speech synthesis via the CSS Speech Module does not affect users who do not wish to use such customization. We carried this solution into our work with EPUB in case there is broader support in the future.

Conversion to Accessible EPUB

Building on this experience, we adopted a stylesheet solution in EPUB format, an open e-book standard managed by the W3C. EPUB is essentially an archive containing XHTML files along with stylesheets, images, and other assets; it supports core web technologies such as HTML, CSS, and JavaScript, and includes built-in accessibility features. This format can also embed essential academic metadata. Importantly, graphics, tables, and other elements are included in the single .epub file. By using standardized templates and stylesheets, we can design EPUB files to meet both speech synthesis requirements and universal design principles, such as providing alternative text for images, scalable

layouts that preserve pagination, flexible font options, and a semantic structure that supports tabbing navigation.[3]

Our main challenge was to develop Word document templates that simultaneously met requirements for universal design, journal preferences, and which could be automatically converted to EPUB. We used Pandoc, a command-line tool that can universally convert between most commonly used document types. This tool was integrated into an online app, so that editors could convert the final draft of a manuscript in a user-friendly interface (Obiajulu Odu, n.d.).

Pandoc does not natively support the direct conversion of JATS XML to EPUB format. However, we achieve this conversion by using Pandoc in combination with other tools and scripts like PHP (Hypertext Preprocessor), xmlstarlet, and Extensible Stylesheet Language Transformations (XSLT). This process is automated by our script, which is available on our GitHub repository.

Ensuring Correct Metadata

If authors use the journal's standard template from the outset, or can reformat their manuscript accordingly, the process produces quality EPUB documents that require minimal post-processing. However, the goal was to avoid post-processing, as we cannot assume that editors have the expertise to edit EPUB files. A key challenge is that the commonly used Word format lacks the necessary metadata, resulting in EPUB files that are deficient in this regard. Consequently, we faced a choice between two methods:

The first method involves converting directly from Word to EPUB without embedded metadata, then adding the required Dublin Core metadata either manually or using an external tool. The advantage of this method is that it preserves the visual expression of the original Word template, as EPUB supports similar formatting.
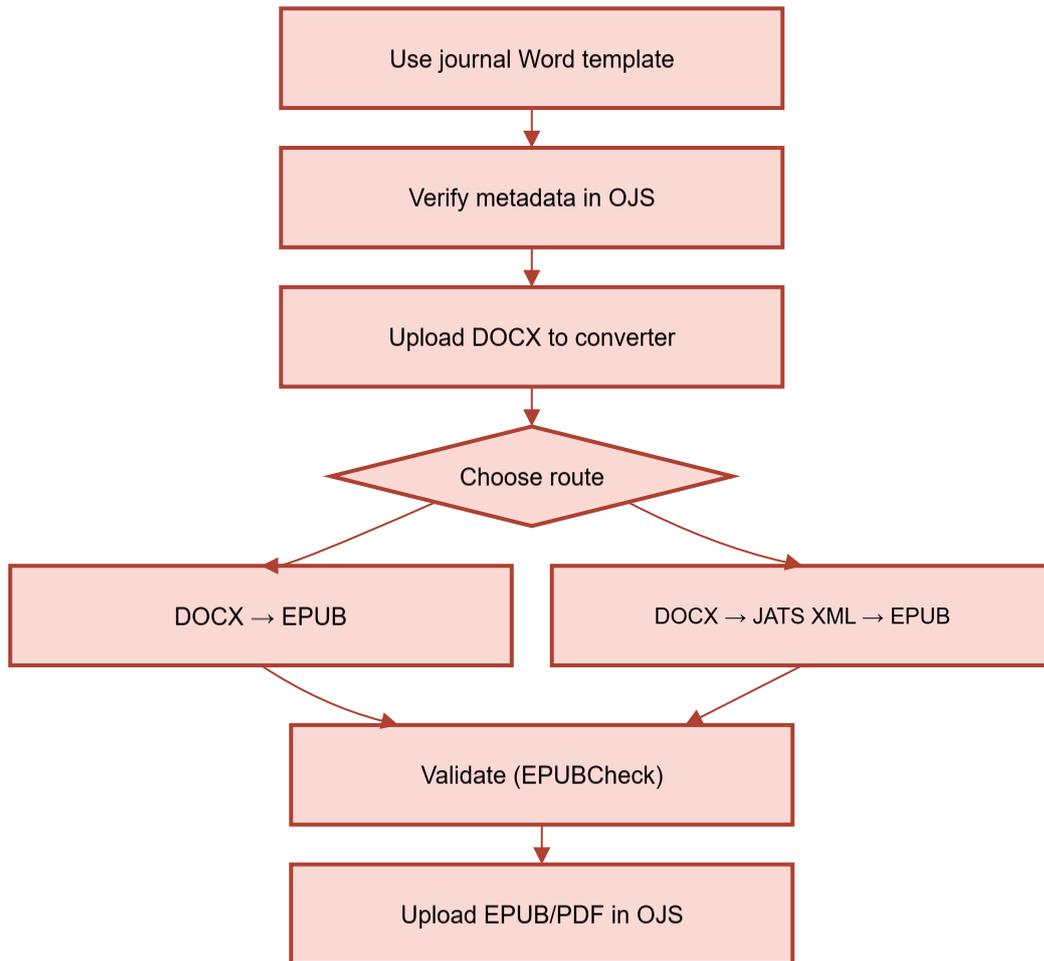
The second method encompasses converting from Word to Journal Article Tag Suite (JATS) XML, where our program automatically retrieves metadata from Open Journal Systems (OJS), incorporates it into the JATS XML file, and then converts the XML to EPUB. The downside is that some customizability is lost because JATS offers fewer formatting capabilities than Word. However, the resulting EPUB file requires essentially no post-processing and meets all necessary criteria.

We found that this latter option was an effective solution for our needs.

---

[3] The XHTML format in EPUB also supports mathematical formulas, using MathML or specialized plugins. However, we did not test this extensively.

Implementation and Workflow

Figure 1. EPUB workflow in OJS

```
┌─────────────────────────────────┐
│     Use journal Word template   │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│       Verify metadata in OJS    │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│      Upload DOCX to converter   │
└─────────────────────────────────┘
                 │
                 ▼
            ◇ Choose route ◇
           ╱              ╲
          ▼                ▼
┌──────────────────┐  ┌──────────────────────────┐
│  DOCX → EPUB     │  │ DOCX → JATS XML → EPUB    │
└──────────────────┘  └──────────────────────────┘
          ╲                ╱
           ▼              ▼
┌─────────────────────────────────┐
│       Validate (EPUBCheck)      │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│     Upload EPUB/PDF in OJS      │
└─────────────────────────────────┘
```

The final workflow we settled on after internal testing, and which other players in university publishing can benefit from, can be summarized as follows:

Preparation

Developing document templates that align with the journal's specifications and universal design principles is the first step. This process involves selecting a high-contrast style and using consistent typographic designs (for example, accessible fonts like Arial or Times New Roman). In addition, authors should be instructed to follow the provided style guidelines rather than applying manual formatting, as manual changes (such as increasing font size instead of using the designated 'heading 2') are not recognized during automatic conversion and can interfere with the navigation features required for speech synthesis.

Next, the document template must be tested using Pandoc, which offers various adjustable parameters. For example, a universal CSS stylesheet can be created for all EPUB files, incorporating accessibility adjustments and voice module settings. During testing, it is recommended to use a test document that includes diverse elements

(tables, figures, etc.) to identify potential issues early. Using an EPUB validator, such as EPUBCheck from the W3C (n.d.), is advisable. Finally, if the EPUB is not rendered as expected in a particular browser, consider trying alternative OJS plugins to optimize how it is displayed.

Next, it is important to ensure that the EPUB file includes the correct metadata. Our preferred method is to use an accessible JATS XML file containing Dublin Core metadata, which our scripting with Pandoc automatically retrieves from OJS. This method provides a replicable solution for other OJS users (cf. section 2.3). Alternatively, one can convert Word directly to EPUB using Pandoc, and then add metadata manually using free online tools or by editing the 'content.opf' file.[4]

Finally, activate a file display plugin in OJS to enable embedded EPUB galleys. This configuration allows users to choose between downloading the EPUB file for use in their document readers or viewing it directly on the website, with the option to have it read aloud via their screen reader.[5] Additionally, the designated 'Galley' in OJS ensures that EPUB appears alongside PDF as a format option.[6]

How to Use our Workflow

Here's a general outline of the steps involved when authors or editors wish to use our online Pandoc converter to perform the conversion:

1. Go to the Online Converter at https://www.ub.uit.no/pandocConverter/

2. Upload Your Word Document: "Choose File" button and click on the button to select your Word document from your computer. We have not implemented the converter to accept documents from cloud storage services like Google Drive or Dropbox.

3. Enter the journal abbreviation from your OJS installation (the abbreviation can be found under Settings -> Journal -> Journal Abbreviation) and the article submission ID. These fields ensure that your EPUB file inherits the metadata registered by the author or editor in OJS.

4. Start Conversion: After uploading the word file, click the "Upload". The online tool will process your Word document and convert it to EPUB format (among other formats).

5. Download the EPUB File: Once the conversion is complete, you will be provided with a download link for the EPUB file in TAR file. Click on this link to save the TAR file to your computer.

---

[4] We experimented with Calibre – E-Book Management (n.d.) for metadata editing prior to finalizing our current workflow.

[5] The default choice in OJS is currently "Bibi EPUB-viewer" by team Lepidus, but at the time of testing this produced an error, so we chose "epubJSviewer" instead, which must be added manually (*Lepidus/epubViewer*, 2021/2023).

[6] For an example of a completed file, see Frantsvåg (2022) at https://septentrio.uit.no/index.php/nopos/article/view/6665

6. Review the EPUB: Open the downloaded EPUB file using an EPUB reader or editor to verify that the formatting, images, content, and metadata are correct. If issues arise, correct them in the source document and repeat the conversion, or make adjustments directly using an EPUB editing tool.

Procedures

Authors should submit their final drafts using the document template available on the journal's website. Both authors and editors are responsible for ensuring that the correct metadata is entered in OJS. Once complete, the editor uploads the Word file to the conversion tool, which generates a TAR file containing the manuscript in PDF, EPUB, JATS XML, and HTML formats. The editor then adds one or more 'Galleys' to the article, uploading the preferred formats (e.g., PDF and EPUB).

Editors should review the EPUB file to verify that images include alternative text and that formatting is correct. Common issues include missing style formatting on headings or errors in references—often due to manual formatting or automatic changes by tools like EndNote or Zotero. If errors are found, the necessary corrections should be made in the Word document, and the conversion process repeated.

Preliminary outcomes and user consultation

The workflow was applied to a small set of new articles across three OJS-hosted journals (*Målbryting*, *Nordic Perspectives on Open Science*, and *Ottar*) with conversions executed on an internal test server. Editors verified rendering and navigation and validated files with EPUBCheck. Informal consultations with students and researchers who use speech synthesis indicated that an EPUB-based approach is presently more practical than CSS Speech or SSML, which are inconsistently supported and require substantial manual effort. Together, this suggests that EPUB delivers reliable reflow, keyboard/tab navigation, and screen reader compatibility for typical use cases. As this is a pilot, we did not conduct a formal user study; the aim was to establish a low-overhead, repeatable workflow that satisfies universal design obligations and Plan S expectations for machine-readable outputs, and that other OJS services can adopt with minimal post-processing.

Final Remarks

A wide range of both open-source and proprietary speech synthesis software is available, and we have found EPUB to be the most practical compromise for our needs. Given the rapid advances in speech synthesis—especially with AI support—it remains uncertain which format will become the future standard.[7] For this reason, rather than committing to more restrictive formats like SSML, our approach preserves PDF for a fixed appearance and uses EPUB for interactivity, readability, and speech synthesis compatibility.

EPUB offers significant advantages, including support for pagination, which is important for researchers and students who rely on consistent page numbering, and the capacity

---

[7] For a review of the field, see (Khanam et al., 2022).

for detailed metadata that is crucial for indexing, reproducibility, and long-term preservation. One weakness of our approach is that it requires substantial initial work and troubleshooting to adapt the workflow for each individual journal. Nevertheless, our solution provides a valuable framework that other OJS users can adopt without starting from scratch.

In sum, institutions should ensure that published files comply with accessibility standards and are machine-readable. We recommend publishing an EPUB galley alongside the PDF to provide more reliable navigation and reflow across devices. Given the rapid evolution of screen readers and AI-driven text-to-speech, institutions should periodically gather feedback from readers who rely on speech synthesis in order to track changing needs.

Acknowledgements

References

calibre—E-book management. (n.d.). Retrieved 26 April 2024, from https://calibre-ebook.com/

EPUBCheck. (n.d.). EPUBCheck. Retrieved 26 April 2024, from https://www.w3.org/publishing/epubcheck/

Frantsvåg, J. E. (2022). Alchemists needed, to transform bronze to gold! *Nordic Perspectives on Open Science*, 7. https://doi.org/10.7557/11.6665

Khanam, F., Munmun, F. A., Ritu, N. A., Saha, A. K., & Mridha, M. F. (2022). Text to Speech Synthesis: A Systematic Review, Deep Learning Based Architecture and Future Research Direction. *Journal of Advances in Information Technology*, *13*(5). https://doi.org/10.12720/jait.13.5.398-412

Léonie Watson & Elika J. Etemad (Eds.). (2023). *CSS Speech Module Level 1*. https://www.w3.org/TR/css-speech-1/

Lepidus/epubViewer. (2023). [PHP]. Lepidus Tecnologia. https://github.com/lepidus/epubViewer (Original work published 2021)

Equality and Anti-Discrimination Act (2017:51). Retrieved 05 March 2025, from https://lovdata.no/dokument/NL/lov/2017-06-16-51/KAPITTEL_3#KAPITTEL_3

MacFarlane, J. (2022). *Pandoc: a universal document converter*. https://pandoc.org/index.html

Obiajulu Odu. (n.d.). *Pandoc converter (UiT)*. Retrieved 30 April 2024, from https://github.com/oodu/docx2Jats_md_epub-for-OJS

Open Journal Systems (OJS)—Software. (n.d.). Public Knowledge Project. Retrieved 25 April 2024, from https://pkp.sfu.ca/software/ojs/

Paolo Baggia, Paul Bagshaw, Michael Bodell, De Zhi Huang, Lou Xiaoyan, Scott McGlashan, Jianhua Tao, Yan Jun, Hu Fang, Yongguo Kang, Helen Meng, Wang Xia, Xia Hairong, & Zhiyong Wu. (2010). *Speech Synthesis Markup Language (SSML) Version 1.1*. https://www.w3.org/TR/speech-synthesis/

Paul Bagshaw, Daniel C. Burnett, Jerry Carter, & Frank Scahill. (2008). *Pronunciation Lexicon Specification (PLS) Version 1.0*. https://www.w3.org/TR/pronunciation-lexicon/

'Plan S' and 'cOAlition S' – Accelerating the transition to full and immediate Open Access to scientific publications. (2024). https://www.coalition-s.org/

Universities and University Colleges Act [Universitets- og høyskoleloven]. (2024). Chapter 10, §10.2. Retrieved 05 March 2025, from https://lovdata.no/dokument/NL/lov/2024-03-08-9/KAPITTEL_10#KAPITTEL_10

Release epubJsViewer-v1_0_0-0 · EKT/epubJsViewer-ojs. (n.d.). GitHub. Retrieved 26 April 2024, from https://github.com/EKT/epubJsViewer-ojs/releases/tag/v1_0_0-0

W3C/WCAG. (2024). [HTML]. World Wide Web Consortium. https://github.com/w3c/wcag (Original work published 2013)