

“Books are social spaces”: Challenges in Collaborating Across File Formats in Independent Publishing

YANN TRIVIDIC, Université Paris-Saclay, CNRS, Inria,

Laboratoire Interdisciplinaire des Sciences du Numérique, France

CAMILLE GOBERT, Université Paris-Saclay, CNRS, Inria,

Laboratoire Interdisciplinaire des Sciences du Numérique, France

MICHEL BEAUDOUIN-LAFON, Université Paris-Saclay, CNRS, Inria,

Laboratoire Interdisciplinaire des Sciences du Numérique, France

WENDY E. MACKAY, Université Paris-Saclay, CNRS, Inria,

Laboratoire Interdisciplinaire des Sciences du Numérique, France

This paper investigates the challenges faced by independent publishing professionals as they collaborate on multiformat books. We ran four structured participatory design sessions with 16 professionals to collect detailed examples of how they appropriate their specialised but isolated authoring tools, and negotiate the tensions between their personal needs, objectives, expertise and even values. We found that collaboration breakdowns arise primarily from interoperability problems across file formats rather than a lack of coordination across applications. We introduce the novel concept of *ecologies of formats to characterise these findings and show that the interplay of formats and people* is essential for reducing friction in publishing. We then introduce *information synchronisation graphs*, a notation tool that reveals recurring patterns in ecologies of artifacts and use it to analyse four publishing workflows. We conclude with recommendations for the design of collaborative publishing systems and suggestions for improving the organisation of their work practices.

CCS Concepts: • **Applied computing** → **Format and notation**; **Annotation**; **Multi / mixed media creation**; *Text editing*; **Media arts**; • **Information systems** → **Document structure**; • **Software and its engineering** → *Software design tradeoffs*; **Interoperability**; • **Human-centered computing** → **Collaborative interaction**; **Graphical user interfaces**; • **Social and professional topics** → **Project and people management**; **Systems analysis and design**; **Systems development**.

Additional Key Words and Phrases: book publishing, designers, editors, knowledge work, document management, data provenance, file conversion, file formats, collaboration, ecologies of artifacts

ACM Reference Format:

Yann Trividy, Camille Gobert, Michel Beaudouin-Lafon, and Wendy E. Mackay. 2026. “Books are social spaces”: Challenges in Collaborating Across File Formats in Independent Publishing. In *Proceedings of Conference on Computer Supported Cooperative Work (CSCW’26)*. ACM, New York, NY, USA, Article CSCW056, 31 pages. <https://doi.org/10.1145/3816904>

Authors’ Contact Information: Yann Trividy, ytrividy@lisn.fr, Université Paris-Saclay, CNRS, Inria, Laboratoire Interdisciplinaire des Sciences du Numérique, 91400 Orsay, France; Camille Gobert, gobert@lisn.fr, Université Paris-Saclay, CNRS, Inria, Laboratoire Interdisciplinaire des Sciences du Numérique, 91400 Orsay, France; Michel Beaudouin-Lafon, mbl@lisn.fr, Université Paris-Saclay, CNRS, Inria, Laboratoire Interdisciplinaire des Sciences du Numérique, 91400 Orsay, France; Wendy E. Mackay, mackay@lisn.fr, Université Paris-Saclay, CNRS, Inria, Laboratoire Interdisciplinaire des Sciences du Numérique, 91400 Orsay, France.

CSCW’26, USA

© 2026 Copyright held by the owner/author(s).

This is the author’s version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Proceedings of Conference on Computer Supported Cooperative Work (CSCW’26)*, <https://doi.org/10.1145/3816904>.

1 Introduction

Publishing refers to the organisation and structuring of content into a document that can be distributed, whether digitally, such as via a website, or in physical form, such as a book. The publishing process is often depicted as a straightforward sequence of steps: an author writes, then a designer lays out the pages, and the book is finally printed [5]. In reality, publishing involves complex and diverse roles, technical tasks and software—what we refer to as a *workflow* [24]. Publishing professionals must also perform several rounds of additional actions, including copy editing, illustrating, proofreading and typesetting. Since each professional usually masters only one or two aspects of the process, they must coordinate with other professionals with different tools and work practices, which often leads to extra work and unexpected problems.

The advent of the internet has expanded publishing beyond print to include diverse digital formats, including web-based and electronic publications. Publishers are also beginning to produce multiformat publications for accessibility, design or commercial reasons. However, they rely on a single-user, print-centric workflow that was developed for personal computing in the 1980s, making it poorly suited for today's multi-format, collaborative publications. To the best of our knowledge, although many Computer-Supported Cooperative Work (CSCW) researchers have examined small-scale collaborative writing and academic publishing, none have studied trade publishing, which poses distinct challenges due to the complexity and diversity of collaborators' roles, skills and work practices.

This paper focuses on independent trade publishers, a group of “extreme” users [19] whose limited resources often lead them to improvise and innovate. Unlike large, multinational corporations, they act as early adopters of new workflows, e.g., by implementing novel web-to-print techniques,¹ and provide a microcosm of work organisations that blend traditional and emerging practices.

This paper explores two key research questions:

RQ1 How do independent publishing professionals collaborate across heterogeneous software environments to produce multiformat publications?

RQ2 How can we enhance their workflows to help mitigate problems that arise from conflicting constraints and work practices?

We first review related work on publishing workflows, collaborative authoring tools, and the methods used to frame and study them. We next present insights from a study of four structured participatory design sessions with 16 publishing professionals, including specific examples of the tensions that arise when professionals with different roles, training, values and software skills collaborate with each other. We then describe the concept of *ecologies of formats* to frame collaborative processes through the interplay of the file formats they rely on. We apply this concept by introducing *information synchronisation graphs*, a custom notation that helps emphasise how files are exchanged and converted in four publishing workflows. We analyse the results of the participatory design sessions and the ecologies of formats of the workflows and highlight the shortcomings of existing tools and practices and the need for better collaboration support. We conclude with social, technical and design implications for collaborative publishing tools and work organisation, along with recommendations for improving existing publishing workflows.

2 Brief history of publishing workflows

Book publishing has evolved considerably over the past century, but its core tasks remain unchanged. A book begins as a manuscript, then is edited, proofread, designed, produced, and distributed. A hundred years ago, typesetters laid out letterpress characters one by one until a proofreader

¹The term “web-to-print” refers to technologies that produce PDF files out of browser-rendered webpages.

approved the final version. Starting in the 1950s, digital technology slowly replaced physical typesetting, but the overall process did not fundamentally change.

The advent of personal computing radically changed publishing in the mid-1980s. Suddenly, a single person could perform the entire chain of tasks from authoring to composition to typesetting. At the same time, users began sharing files electronically—initially via local networks and, increasingly, by email, facilitating collaboration.

Two releases marked the birth of modern desktop publishing: PostScript, the page-description language later refined into the PDF format, and Aldus PageMaker, the first GUI-based tool for print publishing, built on PostScript. Publishing professionals finally had graphical software capable of producing high-quality print-ready files in far fewer steps than ever before. In parallel, inspired by Bravo [59], Microsoft Word was launched in 1983 and became the dominant word processor for writing and editing manuscripts, but it lacked the level of layout control offered by desktop publishing software [36].

Forty years later, trade publishing still depends on the same dominant, traditional split workflow: on one side, “text workers” (authors, editors, proofreaders) rely on a word processor, such as Microsoft Word or LibreOffice Writer, while on the other side, “design workers” (graphic designers, layout artists, typesetters) use Adobe InDesign [8], the descendant of PageMaker. As a consequence, files continue to be exchanged, primarily by email [44, §4], requiring both sides to repeatedly export, convert and import content across a variety of formats² (listed in Appendix A).

Meanwhile, the development of collaborative web authoring tools such as Google Docs (2006), Overleaf (2011), Canva (2013), Figma (2016) reshaped editing and design practices. Many designers now work across media, ranging from print to web. Over the past decade, web technologies have enabled multiformat book production with built-in accessibility features that remain underdeveloped in print-oriented desktop publishing tools. These web-based techniques are gaining traction: they build directly on collaborative web infrastructures and drive some small presses—which are often more flexible and willing to experiment—to explore alternatives [9].

Because of this complex history, current book publishing tools span a wide range of practices, from single-user WYSIWYG (What You See Is What You Get) projects to complex, collaborative multi-user workflows. We are interested in studying how professionals work together within and across these heterogeneous environments to identify breakdowns and workarounds in order to create smoother publishing workflows.

3 Related work

We review research related to alternative publishing workflows, collaborative authoring tools and theoretical frameworks for studying sociotechnical systems.

3.1 Alternative publishing workflows

As noted in the previous section, digital publishing workflows emerged in the 1980s with the advent of desktop publishing software [51]. These single-user tools were designed to support a linear, asynchronous workflow [44], inherited from processes dating back to the early days of personal computing. In practice, however, real-world publishing is rarely sequential, and authors, editors and designers often perform their tasks in parallel [23]. This goes against the assumptions underlying the software design and creates a variety of coordination problems.

In response, over the past decade, trade book publishers have adopted alternative structured workflows. These include web-to-print, with experimental tools such as PubLink [4] and RASH [17],

²We use the term *file format* to refer to the encoding conventions by which a file’s content is stored and interpreted.

as well as commercial products such as Kotahi³ and Quire⁴ for academic and institutional publishing. These tools were inspired by technical documentation systems from the 2000s, ranging from modest but efficient tools such as Sphinx⁵ to full-featured, multiformat book publishing platforms such as O'Reilly's Atlas⁶. Simultaneously, in newspaper and magazine publishing, platforms such as the Méthode framework⁷ were developed as newsroom management systems, enabling the industry to adapt to the web and support cross-media publication [21].

All these projects follow a “single-source” approach, in which a single source of data—one file or a coordinated set of files—generates multiple output formats targeted for both print and digital distribution. This approach not only improves accessibility but also supports content replication and role separation [14, 23]. It was first popularised by FrameMaker⁸ in the 1980s and was used to digitise office documentation in the following decades [67, 30]. More recently, several studies examined the rise of *modular* publishing workflows [15, 3, 16], in which many document fragments are assembled into a single output using custom-built “publishing factories” [22]. This modularisation of the publishing process reinforces the need to track the provenance of text and images to support diverse data formats [40].

We are interested in extending research from design and media studies [50, 62, 1] by focusing on the collaborative aspect of publishing, especially regarding small independent publishers interested in creating novel forms of multiformat books.

3.2 Collaborative authoring tools

Collaborative authoring has long been a key topic in CSCW [53]. Collaboration can take place asynchronously, using email attachments [64], cloud storage [65], and versioning systems such as Git [72], or synchronously, using tools such as Google Docs⁹ and Etherpad¹⁰.

Synchronous editors were originally developed using operational transformation [34]. More recently, the introduction of conflict-free replicated data types (CRDTs) [66] enabled the development of *local-first* applications [38] that support real-time collaboration without requiring persistent network access. Unlike traditional versioning systems, these applications let users work offline without committing their changes, which are synchronised and merged automatically when network access returns. While CRDTs were initially restricted to simple data types, they now support rich text with formatting [47] and DOM fragments [39], which opens up new collaborative publishing opportunities.

Annotations represent another important vector for asynchronous collaboration. Prior studies have shown that people annotate documents to communicate [18] and have proposed models for annotation systems [49, 2]. More recent work has explored the idea of universal, tool-agnostic commenting systems¹¹. Yet, studies show that authors often continue to use traditional word processors designed for solo work [37, 60] and that they usually do not share what McDonald et al. call the same *representations* of the authoring process [57].

Conversion tools also play a central role in publishing workflows. Over the past decade, HTML-to-PDF converters [32] have become popular in some independent publishing houses—especially

³<https://kotahi.community>.

⁴<https://quire.getty.edu>.

⁵<https://www.sphinx-doc.org>.

⁶<https://atlas.oreilly.com>.

⁷<https://www.eidosmedia.com/government/standard-technologies>.

⁸https://en.wikipedia.org/wiki/Adobe_FrameMaker.

⁹<https://workspace.google.com/products/docs/>.

¹⁰<https://etherpad.org>.

¹¹<https://www.inkandswitch.com/patchwork/notebook/11>.

in those aligned with the free software movement [9]—with the help of tools such as Paged.js¹² [8]. Generic converters such as Pandoc¹³ have been used increasingly for handling many-to-many format conversions through an interchange format, acting as a bridge between otherwise incompatible tools [6]. However, many conversion challenges persist. For example, a format as common as PDF is poorly accessible to readers with disabilities [43] and remains difficult to work with: state-of-the-art technologies for extracting content from PDF files rely on complex techniques [68, 48] and working with legacy encodings requires idiosyncratic workarounds [13].

Overall, our understanding of publishing workflows is limited, as previous studies have focused on individual tools rather than on how professionals interconnect them—especially when producing multiformat publications and accessible books. We are interested in understanding how these professionals adopt and adapt collaborative authoring tools and the barriers they face when doing so.

3.3 Theory and concepts for studying collaboration

Independent publishing typically involves non-standard forms of knowledge work, which is generally more fragmented than typical office work. These settings often include freelance work, volunteer work and loosely organised collectives, which are often fuelled by passion and by community engagement rather than by adherence to rigid processes and profit motives [9]. These hybrid, informal work settings remain largely understudied, as emphasised by Nouwens and Klokmose’s call for additional research into this type of practice [61, p. 10]. Two theoretical concepts are particularly relevant to study such contexts: boundary objects and artifact ecologies.

Boundary objects were introduced by Star in the late 1980s to “maintain and develop coherence across intersecting social worlds” [69], though they were later criticised for their potential to impose a hegemonic view over work practices [55]. A classic example of boundary object for editorial work is the *file*, although, according to Harper et al. [31, p. 1127], its effectiveness as a boundary object is now waning, as its shared definition and the practices around it have shifted with the rise of cloud computing. Artifact ecologies [52] help analyse how tools and practices evolve together through the study of “a set of artifacts (software, hardware, and devices) that a person (or a group of persons) uses while supporting their cooperation” [25]. They were later refined into sibling concepts, such as aligned ecologies [63] and collective ecologies [41], which offer ways to describe how toolsets reflect and shape collective identities, values and working relationships.

Following research on design-based experimental publishing, such as Groten’s study of collective “platform bricolage” [29], we engage with these theoretical frameworks to study how individuals appropriate shared ecosystems of tools and data formats in publishing.

4 Participatory design workshops with publishing professionals

We conducted four structured participatory design workshops [56, 54] to better understand the work practices and challenges faced by independent publishing professionals (RQ1). We were particularly interested in how they collaborate within a heterogeneous software environment to support multiformat publishing.

4.1 Methodology

4.1.1 Participants. We recruited 16 publishing professionals (7 women, 9 men, age 28 to 54, all French, 9 years of experience on average), most of whom are currently working for small, independent publishing companies or associations. The majority of participants engage in multiple activities, including nine graphic designers, eight editors, four developers, three researchers, two

¹²<https://pagedjs.org>.

¹³<https://pandoc.org>.

artists and one translator. Eight run their own independent publishing houses, two are academics, three work in academic publishing and three are freelancers. Eleven use publishing software daily, three use it several times a week, and two use it several times a month. All have mentioned either in the questionnaire or during the workshop that they used Adobe InDesign for at least one publication and all also use text processors including LibreOffice Writer, Microsoft Word or Google Docs. They use an average of eight software applications (ranging from 1 to 19). Appendix B summarises relevant characteristics about participants (Table 2) and their software use (Tables 3 and 4).

4.1.2 Setup. Participants met in person in a meeting room with a central table and a flatscreen display at one end so the authors and the participants could share their screen when needed.

4.1.3 Procedure. We organised four workshops over two days, each with one session in the morning and one in the afternoon. The authors and the participants from the morning and afternoon sessions had lunch together, which allowed everyone to meet each other and engage in open-ended discussions in a convivial setting. Each participant received 80 euros as compensation for their participation.

Prior to each session, participants were asked to provide basic demographic data and to describe a recent, memorable story about the last time they had trouble publishing a book in several formats, using a story interview question format [54]. We assigned four participants to each of four participatory design sessions, with two or three authors in attendance. We used the data provided in the preliminary questionnaire to maximise the diversity of profiles within each group of four.

Each session begins with an introduction followed by four structured activities. The introduction explains the structure and expectations from the workshop, and obtains informed consent from participants. The next two activities are designed to gather detailed examples and insights about how the participants produce multiformat publications. The first activity lasts approximately 30 minutes. Participants first introduce themselves and then share one or more stories about memorable or critical publishing events. Participants are encouraged to comment on each others' stories and add additional, related examples.

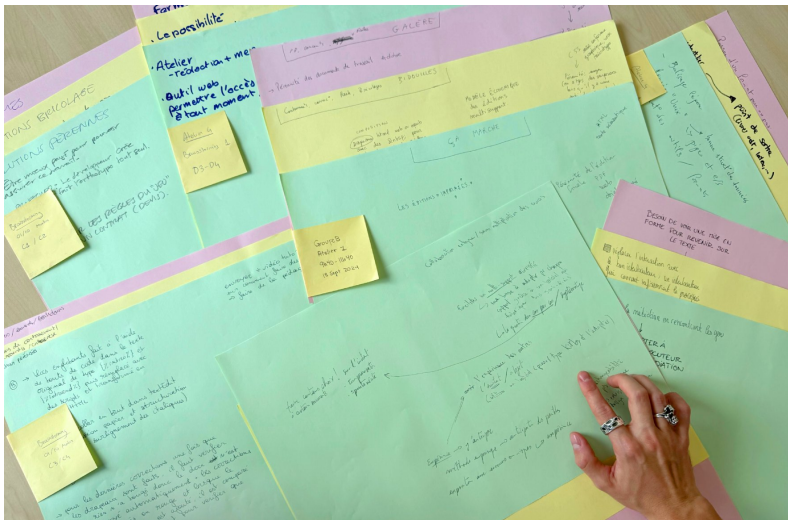


Fig. 1. Workshop materials produced by the participants during the second activity. Green, yellow and pink refer to positive, neutral and negative examples, respectively.

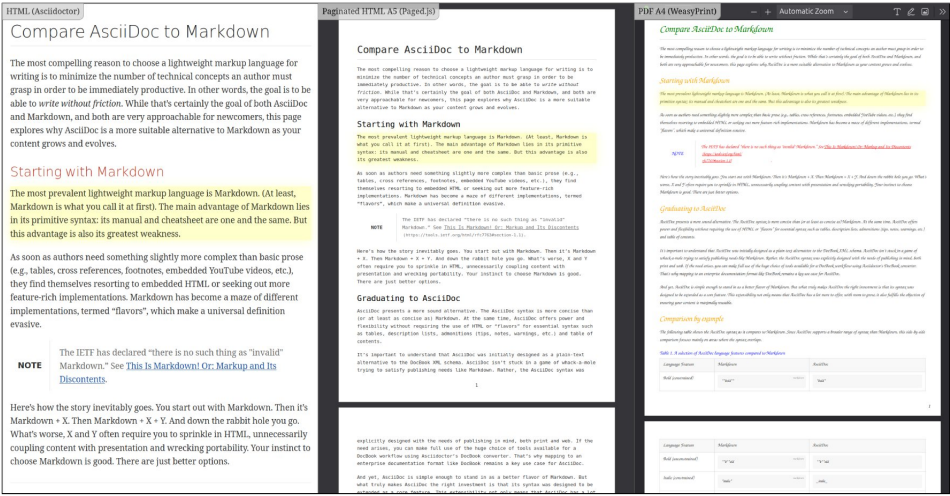


Fig. 2. The prototype shown to the study participants at the beginning of the third activity. The same source document is displayed in plain HTML (left), paginated HTML (center) and PDF (right). Hovering over the text in any of the views highlights the encompassing text block in all three views. Scrolling is synchronised among the three views.

The second activity also lasts 30 minutes. Participants work in pairs, together with at least one author, to brainstorm ideas related to the first activity. Participants colour code their ideas, solutions, and problems with coloured sheets of paper: green, yellow and pink refer to positive, neutral and negative examples, respectively (Figure 1).

The third activity lasts approximately 20 minutes and introduces a live demonstration designed to provoke discussion about the possibility of bridging heterogeneous systems together. We show a simple prototype¹⁴ (Figure 2) that takes an AsciiDoc¹⁵ file as input and enables synchronised scrolling and content highlighting of an arbitrary number of windows that display diverse output formats, including raw HTML, paginated HTML and PDF. Participants are encouraged to reflect on how cross-format tracking at the word level could benefit their current work practices.

The final activity lasts approximately 40 minutes. All participants work together to brainstorm new ideas, based on the ideas and concepts that arose during the previous activities. The session ends with a quick final debriefing.

4.1.4 Data collection. All group sessions were audio recorded. The authors took handwritten notes, and collected the sketches and notes produced by the participants. We also recorded the screens that participants shared with the group.

4.1.5 Data analysis. We transcribed and analysed the session recordings using reflexive thematic analysis [11]. All authors identified specific breakdowns, workarounds, user innovations and general surprises from their own notes. One author then generated codes using an inductive (bottom-up) approach. Two authors then grouped the codes into a small set of categories, with multiple iterations and discussions amongst the three authors who attended the participatory design session. We tested the codes by extracting 50 random quotes and comparing how the authors categorised them. After discussing disagreements, the authors rechecked the full set of quotations. After two additional

¹⁴<https://propage.editionsburnaout.fr>.

¹⁵<https://asciidoc.org>.

iterations, which included separating themes T1 and T2, all three authors reached agreement on the final set of themes.

4.2 Results

We collected and transcribed 10 hours of audio from the four participatory design sessions. This resulted in 350 quotes which were categorised into 58 initial codes. We then identified four themes:

- T1 Participants cannot always collaborate due to lack of content synchronisation across formats.
- T2 Participants struggle to achieve multiformat output from a single source.
- T3 Participants find compelling workarounds to reduce loops in publishing workflows.
- T4 Participants trade values and autonomy for productivity.

4.2.1 T1 — Participants cannot always collaborate due to lack of content synchronisation across formats. Many participants (9/16) described situations in desktop publishing where the plurality of authoring formats makes it nearly impossible to maintain consistent content across files. They described how they work asynchronously with text processors and desktop publishing software, and fail at tracking changes. P12 stated: *“There are plugins that allow us to export to Word from InDesign, but if you already started doing your layout, it is a disaster”*¹⁶. While some export functions exist, they rarely match the participants’ expectations, leading to frustration and higher workload. P1 recalled that he could not understand the structure of an enormous EPUB file until he realised that it had been exported from InDesign. P2 used a metaphor to illustrate her take on the problem: *“I need a tool that makes it so there is a single format that merges all formats instead of multiplying them. Like a universal plug. InDesign and Word superimposed.”*

Likewise, participants (11/16) often have to completely restructure files received from authors. They use workflows to “reset” the document’s structure that is often, at this point, a mix of poor semantics and direct formatting¹⁷. Rather than collaborating directly within a shared format, they assemble conversion workflows that strip and rebuild the semantic structure from scratch: *“To clean up the HTML I get from InDesign, I use Pandoc to convert it to DOCX. It loses a lot of stuff... Then I get the DOCX back and convert it to Markdown, before going back to HTML... it’s the best method I’ve found”* (P13). This approach often sets aside others’ formatting decisions entirely: *“We pass it through TextEdit as plain text, we erase everything, reflow it entirely, and reapply all the styles, because otherwise there are conflicts”* (P10).

This lack of structure in the authors’ document often leads to frustration: *“As authors, they should be mastering the tool they write with. Not knowing how to make a footnote in Word... it is just not possible”* (P11). P9, an editorial secretary, said her role is explicitly to make this step of the process smoother: *“You receive poor-quality—nonsensical—files, and you say ‘thanks a lot! That’s perfect!’ and you do your thing, adapt it and clean it up.”* This effort—normally done by editorial secretaries but often transferred to graphic designers, who do not consider it as their role (P11)—allows authors to keep using their preferred tools, even when those tools make collaboration more difficult: *“We leave it up to each of them to use and choose the tools they prefer... I always say: if you want, you edit the DOCX; if you want, you make annotations on the PDF”* (P9).

Finally, most participants (12/16) expressed their need to retain control over source files, even when the people they collaborate with need an extra interface layer to contribute to the project. P13 explained how he coordinates his command-line operations on the source files while the author used a CMS and a rich text editor because it was *“easier for them.”* P4 added: *“If I want to modify stuff, I know that via FTP I have access to the whole thing”*. On the other hand, several participants

¹⁶All quotes were translated from French by the first author.

¹⁷“Semantics” describe the role of document elements to both systems and humans. “Direct formatting” is formatting applied without semantics, e.g., identifying titles only with a bold typeface and a large font size.

(5/16) noted that certain formats, especially closed-source or highly structured ones, are difficult to parse and modify. *“We spend so much time with super complex file formats that we just can’t strip open, such as TEI”* (P14). P3 noted that enriching source content often backfires as *“the more we try to enrich plain text [with specific semantics], the more we lose in perennality”*.

4.2.2 T2 — Participants struggle to achieve multiformat output from a single source. Participants offered a variety of definitions of multiformat publishing, with different output formats. The most vanilla examples are printed books, eBooks, and websites. For some, multiformat publishing is a deeply experimental or even playful act: *“We made a party into a book. This is multiformat publishing, but weird”* (P3). They also mentioned books as performances (P4), radio shows (P3), or even a publication for which every print is completely personalised and unique (P8), often requiring idiosyncratic code for their workflow to function.

All participants (16/16) view their publications as socially situated artifacts that must be adapted to different media and contexts: *“Books, whatever the media they rely on, are social spaces, and serve as vectors for other practices to unfold”* (P2). For P13, this adaptability even extends to not considering his publishing practice as tied to printed output: *“In the DNA of our publishing house, there has always been the idea of publishing whatever the cost, i.e., whether it is books or something else.”* However, some participants (5/16) agreed that this quality is conditioned by the requirement of perennality—publications are expected to last, and some media, such as eBooks, are less long-lasting than others.

Some participants (6/16) highlighted the role of PDF as a key file format in collaboration. This “digital facsimile” (P1) is an interoperable output medium to work around software that could otherwise be incompatible across people. *“When I start doing Markdown, the author and the copy editor don’t intervene anymore. When I export a PDF proof, I get comments and we implement them”* (P13). The role of PDF as a compromise is a “work tool”: *“it allows me to copy-paste, to share, to do searches”* (P12).

Half the participants (8/16) said they needed to copy-paste and tweak content manually to generate different outputs, especially when synchronisation between formats is not possible. P3 explains: *“In the web version, I have to remove the ragged paragraphs... it’s not exactly the same content. So up until now, I have always duplicated my content... and that’s really a problem”*. To avoid full duplication, some rely on workarounds such as *“literally add[ing] HTML spans to adjust the letter spacing”* or inserting non-breaking spaces in authors’ documents.

In order to try to truly enable single-source publishing, some participants develop and adapt ad hoc infrastructures—often outside mainstream desktop publishing software (5/16). For instance, P3 designs a new and unique workflow for each project, piecing different software together to fit the specific needs of the situation with *“flexibility and modularity”*. Others create highly technical workarounds to accommodate unsupported features, such as right-to-left languages: *“There is no documentation, but I was able to get well-structured XML files... I was then able to understand what parameters I had to use”* (P15). Participants frequently turn to free and open-source tools because commercial publishing software limits their capacity to generate adaptable outputs: *“Free software lets us generate a world of possibilities [...] which wouldn’t be possible with closed-source software”* (P8).

4.2.3 T3 — Participants find compelling workarounds to reduce loops in publishing workflows. A majority of participants (9/16) highlighted the problems that can arise as collaborators go back-and-forth over a document: these iterative loops are necessary but create friction. For example, P10 described a particularly painful loop: *“They sent us back the [text file] with the whole paragraph. We didn’t know where the mistake was... We had to reintegrate the whole thing, redo all the orthotypography¹⁸ and do it in both languages”*. However, despite the amount of work it may incur,

¹⁸Orthotypography is the set of rules that defines the rightful usage of punctuation marks and layout principles.

participants do not wish to automate those loops entirely: *“We still need to see in InDesign how the modification was made in the PDF. It has to be visible, we have to keep a hint”* (P6), and P12 suggested a compromise between control and traceability: *“Corrections, instead of being directly implemented, have to be reviewed and validated”*. Some participants (6/16) ease the process by adopting early layouts: *“The ultimate solution is to send very early laid out PDF files to the authors... I do a rough first draft, and this is what I tell them to copy edit”* (P12). This trick improves communication by casually preserving page numbers and representations between the authoring file and its PDF export.

Besides acknowledging the need for loops, some participants emphasised that some expert tasks are inherently sequential. For example, P1 stated that authors must be *“unplugged”*. P4 said that she needs *“people to be able to access only this chapter or that part”* and *“to distribute different access rights to different places in the project, at different times”*. For P10, synchronous collaboration is even a threat to layout design: *“It’s something we avoid... it’s a bit like opening the door to the destruction of our design”*. The importance of typographic quality, especially ragged paragraphs, was a recurring point of tension. All graphic designers (9/16) felt responsible for perfect typography in the final publications: *“It is very usual for me to go on the client’s website back office long after the delivery to secretly do some orthotypography and adjust the ragged paragraphs”* (P10).

Underlying these frictions are power conflicts across roles: *“When we give power to an author, they ask for more...”* (P8). This tension arising from the pressure of deadlines is inherent in print publishing. *“At some point, the axe falls and we say that the book is done, and we print it as it is”* (P4). Several participants said that the irreversibility of print imposes a retro-planning logic that is less salient in other models, such as those used in web development (P8, P13, P14).

4.2.4 T4 — Participants trade values and autonomy for productivity. Participants are constantly balancing between their ideals of autonomy and coherence on the one hand, and the pressure of economic realities, production timelines, and collaboration constraints on the other hand. All participants (8/16) who work for their own publishing house rely on external financial resources to support their publishing activities. This trade-off between their personal values and practical workflows comes at a cost. For example, participants trying to use free and open-source tools in alignment with their political values—such as autonomy, transparency, or anti-capitalistic stances—often accept imperfect results (6/16). *“It might make sense to imagine this collection using free tools... that’s the artisanal approach we wanted to take, accepting mistakes and hazards”* (P6). Yet this acceptance does not mean full satisfaction. Others expressed frustration at the limits of these tools: *“Reproducibility issues are why I don’t use tools like Paged.js... L^AT_EX isn’t perfect either. I’m on Linux, with Chrome, but the typesetting engine isn’t as good as on Mac.”* (P13).

This compromise extends to concerns about automation and deskilling, particularly for graphic designers. P6 and P10 particularly fear that productivity gains might devalue their profession. *“Would this time saving be a loss of earnings for us graphic designers?”* (P10). Beyond professional roles, collaboration was sometimes described as being made difficult by unaligned ecologies of artifacts. Some participants (6/16) have witnessed people be disregarded for not mastering the technical stack, or for using the “wrong” tools. P16 told the story of a colleague being misjudged because she was not able to structure a document correctly: *“She switched to ChatGPT to put fake markup back in... Now, she’s considered incompetent”*.

To limit the number of loops and thus be more efficient, participants constrained their collaboration by adding clauses to contracts as coordination mechanisms (4/16), in which they formalise their workflow and clarify the final deliverables. P7 described: *“We want to include in the contract that the final version will be the [digital] version and not the print version... this leads to more mediation”*, and P10—to work around the problems outlined in T3—said: *“At the quotation stage, we try to be as detailed as possible... to avoid going back and forth”*.

Despite these strategies, many participants felt that certain defaults—such as the dominance of InDesign—are immutable. “*Compatibility will be a problem anyway... InDesign will always be the main tool*” (P8). P5 blamed most of his issues on the technical debt embedded into his tools through outdated design choices. Incompatibilities within current artifact ecologies—across devices, operating systems, and file types—further constrained collaboration: “*They don’t want to agree on the same tools. I have a Mac, she has a PC... I don’t have the software because it’s not free*” (P2).

5 Ecologies of formats

A key finding of the thematic analysis is that communication breakdowns are due to the diversity of *file formats* rather than the different *applications* involved. In order to further analyse these breakdowns, we considered the concept of artifact ecologies, as it was used to study work organisation in collaborative writing [44] or the establishment of shared practices through tool choices within collectives [41]. However, while artifact ecologies are useful to understand how people coordinate through ensembles of devices and applications, they do not provide much insight for analysing how publishing professionals traverse through formats to collaborate.

We therefore develop a complementary analytical framework to address this gap by foregrounding file formats as a central element of collaborative practice. We introduce the notion of *ecologies of formats* to further investigate the interplay of people and file formats in collaboration. We then situate it in relation to other concepts, including information ecologies, artifact ecologies and boundary objects.

5.1 Definition

We define an *ecology of formats* as the ensemble of file formats a person, a group or a community mobilises, together with the tools, scripts, and conventions that make these formats interoperable in practice. Some file formats work well together, enabling smooth *transitions* [44] and iterative loops, while others hinder collaborative workflows because they cannot encode the same semantic elements, lack conversion tools, or hold information in ways that are difficult to extract or reuse.

Format ecologies therefore emerge from the *de facto* dependencies between formats, which are determined both by their inherent characteristics and by their compatibility with other formats (or lack thereof). This approach focuses on informational structures rather than applications. It puts an emphasis on format interoperability as a critical dimension of collaboration and draws attention to the underlying formats that are often rendered invisible in software.

5.2 Related concepts

In their investigation of the theoretical landscape of “ecology” metaphors in the fields of HCI and CSCW, Lyle et al. [52] identify four broad categories: artifact ecologies, communicative ecologies, device ecologies and information ecologies. In addition, they also identify three analytical levels: a *macro* level (organisations and activity systems), a *meso* level (people and practices) and a *micro* level (individuals, artifacts and tasks). We compare ecologies of formats with the two most relevant categories: artifact ecologies and information ecologies.

Relation to information ecologies. Nardi et al. [58] define an information ecology as “*a system of people, practices, values, and technologies in a particular environment.*” Examples include libraries and hospital units. A publishing house can likewise be considered as an information ecology. Ecologies of formats intersect with this perspective, but they operate at a different resolution. Format ecologies target the meso and micro levels while information ecologies primarily target the macro level [52, §5.1]. In our study, the meso level becomes visible in the organisation of roles and practices, while the micro level emerges in the sequences of concrete actions on files made

by individuals. Information ecologies are most often mobilised to analyse contexts at the macro level—such as library infrastructures managing thousands of titles or systems built on the semantic web—and tend to pay less attention to the specific constraints and affordances of specific actors. Information ecologies have also been concerned with formats, for example when discussing the societal implications of the standardising practices around SGML and XML [26, p. 115], but again at the macro level rather than, e.g., the micro level of format conversions performed by one individual.

Relation to artifact ecologies and boundary objects. While artifact ecologies focus on devices and applications [42] that people own, access and use [35, p. 201], format ecologies focus instead on the structure of the information that circulates through these artifacts. Format ecologies are therefore not reducible to artifact ecologies, nor do they seek to replace them. In fact, file formats are not artifacts, files are. Formats are part of the glue that holds artifacts together: they shape the rules through which files become boundary objects [69] and support collaboration. Therefore, format ecologies provide a complementary analytical perspective that makes visible this specific dimension of artifact ecologies that often remains implicit.

Relation to other sub-concepts of artifact ecologies. Several sub-concepts derived from artifact ecologies can be transposed to format ecologies. For example, collective artifact ecologies [41] ask questions about how a collective's identity becomes embedded in its tools and practices; these questions can likewise be applied to examine how shared identities are expressed through the formats a collective adopts and maintains. Similarly, analyses of aligned ecologies and the exploration of potential ecologies [63], as well as research on transitions within and across ecologies [44], can be transposed to the format ecologies to inform the design of collaborative workflows: aligning a format ecology comes down to figuring out the best conversion route between formats, as well as considering the values and capabilities of the people involved.

Despite operating at the same level as artifact ecologies (mostly the meso level), we are not aware of previous work that focuses on formats as a structuring concept. Format ecologies address this gap by foregrounding interdependencies between formats as a new kind of ecologies capable of reusing the analytical tools of the artifact ecologies framework.

6 Information synchronisation graphs of publishing workflows

Just like artifact ecologies can be described by graphs connecting people and artifacts, ecologies of formats can be described by *information synchronisation graphs* (or *infosync graphs*), which connect the files as collaboration is enacted. For a given publishing workflow, each node of the infosync graph corresponds to a particular file and each arc corresponds to a synchronisation step that involves creating, updating and/or exchanging a file. We first describe the infosync graphs of the “traditional” workflow and of three additional workflows from actual projects. We then use these examples to identify recurring or unique patterns and understand where to expect breakdowns.

6.1 Traditional workflow

During the participatory design sessions, participants repeatedly referred to the traditional publishing workflow described in Section 2: a sequence of steps performed on word-processing files, InDesign documents and PDF proofs. Even though not all participants actively used this workflow (12/16), they all (16/16) referred to it with only minor variations. We thus used this workflow as an entry point to re-examine the corpus from another angle.

This workflow includes four individuals—an author, an editor, a graphic designer and a printer—who use email to communicate and exchange files. First, an author writes a manuscript using Microsoft Word (DOCX file). Once the manuscript is ready, the author sends it to an editor, who

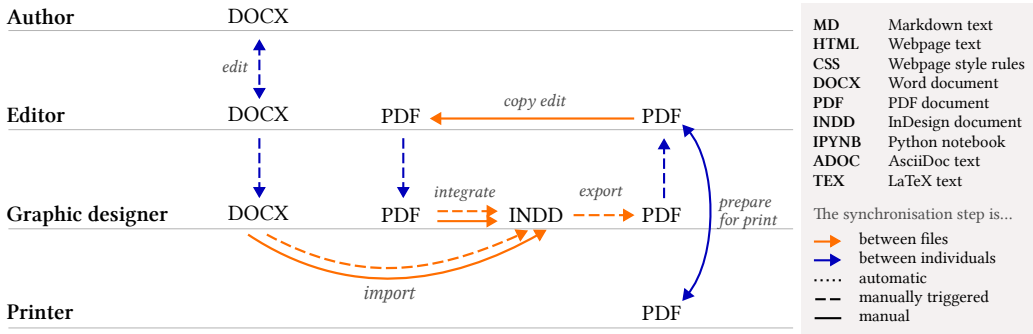


Fig. 3. Infosync graph of the traditional book publishing workflow (W1) showing how the author, editor, graphic designer and printer exchange files in different formats to edit a traditional paper book. Refer to Appendix A for details on the formats listed in the legend.

reads it and, assuming that the manuscript is accepted, sends it to a graphic designer. The graphic designer imports the text from the Word file into Adobe InDesign (INDD file), either by manually copy-pasting the text or with the help of InDesign’s import feature if the DOCX file has been structured correctly. The designer then uses InDesign to create the book’s layout and exports it as a PDF file to send back to the editor. The next steps involve the editor and the graphic designer, in which the editor(s) correct the PDF and the graphic designer makes corresponding changes in the layout, until the editor validates the final version.¹⁹ At that point, the editor sends the PDF to the printer, who engages in a round of back-and-forth communication with each other, before the latter finally approves the PDF, which is then sent to the printing presses.

When we sketched the graph of the workflow described above, we noticed that its synchronisation steps involved either synchronising copies of the same file across two users or synchronising two local files, often with different formats. We also noticed that different synchronisation steps incurred different costs to the individuals involved: some were effortless and transparent whereas others were manual and demanding. We therefore decided to organise the graph along two dimensions: type and difficulty of the synchronisation step. Figure 3 shows the resulting graph and its legend.

In this notation, each node represents a file on an individual’s computer, identified by a code that corresponds to its data format, such as “DOCX” for a Word document. Nodes are aligned along a two-dimensional grid so that each user is represented by a single horizontal axis and each file is represented by a single vertical axis. This means that a single file shared by n users will result in n vertically aligned nodes. Arrows between nodes represent information synchronisation steps. Vertical arrows (in blue) capture synchronisation *between individuals* who share copies of the same file, corresponding to coordination practice at the meso level. Horizontal arrows (in orange) capture synchronisation *between files* managed by the same individual—possibly across different formats—corresponding to actions at the micro level. The arrow’s stroke style indicates the step’s difficulty. Plain arrows represent *manual data transposition*, where users are fully in charge of transferring information from one file to another, such as when manually editing a Word document or $\text{\LaTeX}$ file to incorporate text annotations written on a PDF file. Dashed arrows represent *manually triggered*

¹⁹If the editor uses Adobe Acrobat Pro systematically to annotate the PDF and the graphic designer uses the appropriate Adobe InDesign import mode, without modifying the InDesign document in the meantime, any text changes can be applied automatically to the INDD file. Similarly, Adobe InCopy can be used for proofreading content without preventing the graphic designer from validating and reviewing each change. However, these processes rarely happen in practice, and graphic designers often update the text manually—as reported by all the participants of the first study who use InDesign.

synchronisation, which is partly automatic but requires some user actions, such as emailing a new version of a Word document to a collaborator or running a script to convert a file into another format. Finally, dotted arrows represent *automatic synchronisation* that is performed transparently for the user, such as when the typeset version of a document written in a document description languages such as AsciiDoc or $\text{\LaTeX}$ automatically updates when the code is modified.

6.2 Additional workflows

We constructed the infosync graphs of three additional workflows that represent real situations with a similar scale in terms of human, economic and hardware resources. We chose these workflows to maximise the diversity of publishing practices, software tools and file formats, both to complement the situations described in the first study and to validate the robustness of the graph notation. This selection process resulted in three types of publications: a scholarly article for an ACM conference (W2), a web-to-print catalogue for a museum (W3), and a computer science textbook (W4). Their infosync graphs are depicted in Figure 4.

6.2.1 Scholarly article for an ACM conference (W2). The first workflow (Figure 4a) is a classic academic publishing workflow. It was recently experienced by two of the authors when they jointly published an ACM conference article that requires uploading the source of the accepted manuscript to an online platform (TAPS²⁰), which compiles it into both print (PDF) and web (HTML) versions that are published electronically. This workflow requires the authors to use a predefined layout in the form of a $\text{\LaTeX}$ class. They use Overleaf, an online $\text{\LaTeX}$ editor featuring real-time collaborative editing, and continuously recompile the $\text{\LaTeX}$ source as they write so they can preview the typeset version as a PDF document. Once the manuscript is ready, one of the authors—designated as the *contact author* by ACM—becomes responsible for uploading the source of the document to the online submission system.

The TAPS system attempts to compile the source into a print version using a toolchain similar to that of Overleaf, and into a web version using a toolchain that is unknown to the authors. The contact author receives an email if the system fails to compile the document, and is asked to edit the source to fix the errors and re-upload the revised source file. The author can also contact the customer service of the online system to ask for help. If the system succeeds, the contact author must then download and review both the print and web versions of the article and either accept them as is or engage in another cycle to fix the source and re-upload it to the submission system.

6.2.2 Web-to-print catalogue for a museum (W3). The second workflow (Figure 4b) is a web-to-print workflow implemented in 2020 by Julie Blanc and Antoine Fauché—two researchers with experience in graphic design and publishing—in order to create the web and print versions of a catalogue for an archaeological artifact collection.²¹ This workflow is documented in a blog post²² and was validated by its author.

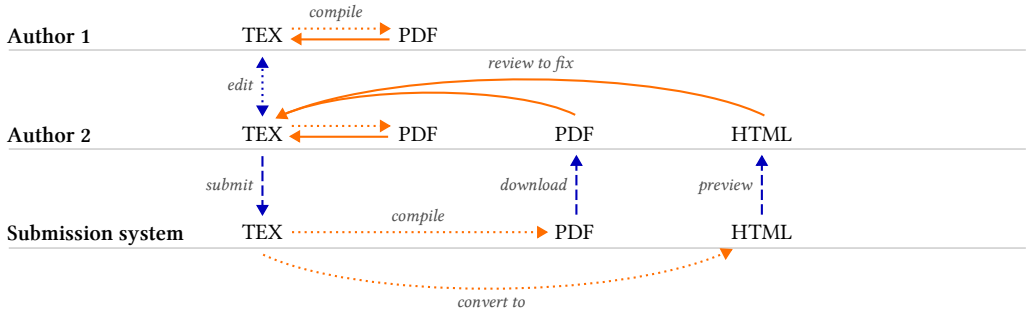
In this workflow, the author and the editor write the content of the catalogue by editing Markdown files with the help of a headless content management system (CMS) that was set up for this project and that is synchronised with a Git repository. They could also upload images and interact with the bibliography using Zotero,²³ a reference management software. References were then exported as Bib $\text{\TeX}$ files, but we do not include this part in the graph for the sake of simplicity. In parallel, the graphic designer writes two CSS files containing rules to style the web and print versions of the catalogue, which they commit to the same Git repository.

²⁰<https://authors.acm.org/proceedings/production-information/taps-production-workflow>.

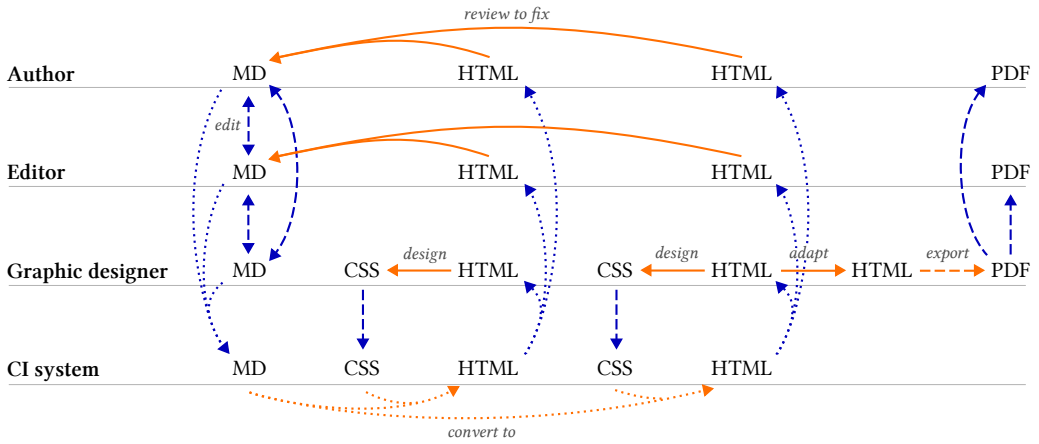
²¹<https://villachiragan.saintraymond.toulouse.fr/en>.

²²https://julie-blanc.fr/blog/2020-11-05_chiragan (in French).

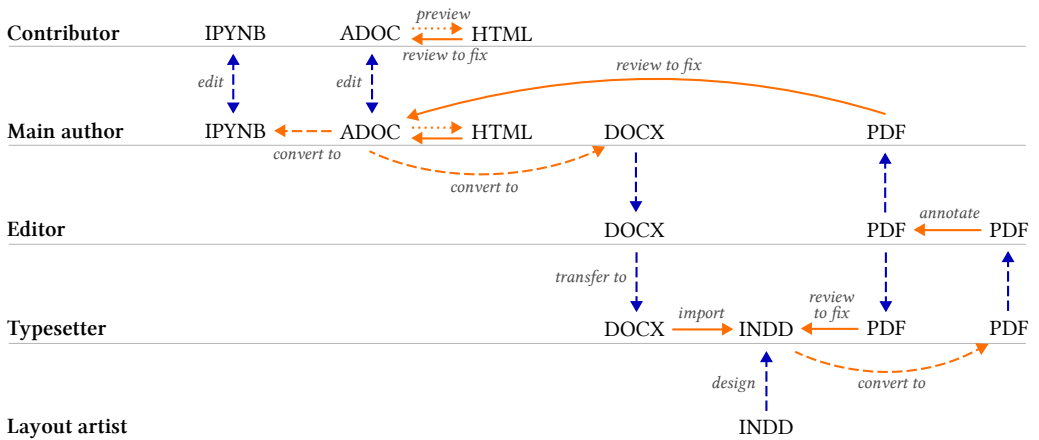
²³<https://www.zotero.org>.



(a) Online scholarly publishing (W2).



(b) Museum catalogue publishing (W3).



(c) CS textbook publishing (W4).

Fig. 4. Infosync graphs of the three additional publishing workflows described in subsection 6.2. In each case, the participants exchange files in different formats in a complex series of inter-connected loops using various communication protocols.

Every time the author, the editor or the graphic designer commits to the repository, a continuous integration service (CI) automatically executed on the Gitlab repository runs a script that compiles the Markdown, media and BibTeX files into the two versions of the catalogue. Both versions are encoded as HTML files, but the print version is paginated using Paged.js and only includes a subset of the content published in the web version. Both versions are made accessible by the CI via a shared URL so that the authors can preview either of them. If needed, the author and the editor can then update the Markdown files while the graphic designer can update the CSS files.

Once the editor validates that the catalogue is ready to be printed, the graphic designer duplicates the paginated version to add content specific to the print version, such as additional images, and performs typographic tweaks by editing the HTML directly. They finally export it as a PDF by using the in-browser print function of Google Chrome and send the final print version to the rest of the contributors.

6.2.3 Computer science textbook (W4). The third workflow (Figure 4c) is a computer science (CS) textbook publishing workflow. It describes the experience of two of the authors writing computer science textbooks for high school students, respectively as a contributor and as the main author.

In this workflow, the textbook's content is authored using the AsciiDoc document description language. The code snippets in the course part of the chapters and the exercises are converted from AsciiDoc into interactive Python notebooks. Once published, both teachers and students can use the notebooks to interactively edit and run code to solve the exercises. The main author decides on the textbook's structure and content and writes parts of it. In addition, several other contributors help write certain chapters and/or exercises. They collaborate by syncing AsciiDoc and Jupyter notebook files using a Git repository.

On a regular basis, the main author converts the set of AsciiDoc files for a chapter into a single Word document and sends it to the editor. In parallel, a graphic designer creates the layout using InDesign, in accordance with specific textbook constraints set by the publisher. When it is ready, the designer sends the InDesign file to the typesetter, who also receives the Word files forwarded by the editor and uses them to populate the empty layout with the textbook's content, tweaking the layout as needed.

After each integration of the textbook's content, the typesetter exports the book as a PDF document and sends it to the editor, who annotates it and sends it back to both the main author, who must either approve it or update the content, and to the typesetter, who must integrate the changes from both the author and the editor into the InDesign document. Whenever the author or a contributor makes a change, they mark their edits in the AsciiDoc source files. These edits show up as highlighted text when converted to DOCX and sent back to the typesetter for integration into the InDesign document.

6.3 Patterns

We analysed the infosync graphs corresponding to the four workflows we modelled in order to identify patterns that are either recurring or unique. One author used the diagrams to reflect on different aspects of the graphs, such as the correspondence between the tasks in a workflow and the subgraphs that describe them and the nature, necessity and difficulty of synchronisation steps (or sequences of steps). After presenting their observations to a second author, they engaged in discussions to agree or disagree with the propositions and suggest other patterns. We group the resulting patterns into five categories:

- Pa1 Collaborators have unequal responsibilities.
- Pa2 Files with duplicated content serve different purposes.
- Pa3 Loops come in two types, preview and feedback.

Pa4 Syncing collaborators is easy but constraining.

Pa5 Syncing formats is hard, especially when the target file exists.

6.3.1 Pa1 — Collaborators have unequal responsibilities. All the workflows feature multiple collaborators (and in W2 and W3 a remote system acting as collaborator) with different characteristics. They form three distinct groups: those who create and update the content (authors, contributors, copy editors), those who create and populate the layout (graphic designer, typesetter, layout artist) and those who coordinate and guide the two other groups (editors). Each group has its own habits and work practices, especially in terms of software they use, which in turn constrain the file formats they can work with in order to collaborate with other groups.

In each workflow, one or two collaborators have a pivot role in maintaining connections amongst distinct groups. While this role traditionally falls to a dedicated editor (as in W1), it may also be embodied by another person, such as the author responsible for fixing the $\LaTeX$ code on behalf of the other authors in W2. In W4, this role forced the main author to write complex scripts to convert the content written in AsciiDoc into Python notebooks (for the contributors who corrected exercises) and into Word files (for the editor and the typesetter). Although this makes the transformation between these formats easier in the long run (simply by running the scripts again—hence the dashed arrows), it also induces a high initial cost that may not be reused across projects.

6.3.2 Pa2 — Files with duplicated content serve different purposes. All four workflows rely on multiple file formats—up to six different formats in W4—, let alone those of auxiliary files not shown in the graph (such as scripts to convert one format into another). In addition to including multiple *output* formats, these workflows also include multiple *authoring* formats for, e.g., using a specific piece of software (IPYNB in W4) and helping transfer information between different groups of collaborators (DOCX in W4).

As a result, different groups of collaborators may rely on different formats to represent the same information. In every workflow, the authors either use Word files (DOCX) or files written in a document description language (TEX, MD, ADOC) that must be compiled into a preview format (HTML, PDF), all of which contain the same text. Yet, some formats contain information unique to them. For example, only INDD files contain information about paragraph styles in W1 and only IPYNB files contain the solutions of the exercises in W4. On the contrary, two files that are supposed to be the same are sometimes different. For example, in W3, the graphic designer uses a specific operating system and a specific web browser to generate high-quality PDF files from HTML.

6.3.3 Pa3 — Loops come in two types, preview and feedback. Preview loops are small, local loops between two files formats: one that can be edited and one that can be visualised as a typeset document. They appear in W2 ($\LaTeX$ files visualised as PDF) and W4 (AsciiDoc files visualised as HTML). In each case, the forward transformation is seamless (it can be performed in the background by the text editor) but the backward transformation is entirely manual. Although some tools help users discover the mapping between the two files, such as Sync $\TeX$ [45] for PDFs generated from $\LaTeX$ files, making any modification, such as fixing a typo, requires searching for the corresponding piece of code in the source file.

Feedback loops are larger loops that are designed to refine an “output” file shared by at least two collaborators. For example, in W2, the PDF output is annotated to update the content of the $\LaTeX$ file that was used earlier in the workflow to author the text that generated the PDF. Similarly, the PDF exported from InDesign in W1 and W4 is duplicated and annotated by the editor before being sent back to the author and to the typesetter, resulting in an additional file in the workflow. Unless the PDF file is copy-edited in a very specific way (see footnote 19), this process is entirely manual.

6.3.4 Pa4 — Syncing collaborators is easy but constraining. All synchronisation steps between collaborators but one (vertical arrows) are either automatic or manually triggered. The only exception is the back-and-forth between the editor and the printer in W1, who must discuss and agree on a number of details regarding how the PDF they exchange should be printed.

While this makes synchronisation between collaborators look easy, the relatively low effort often comes at the cost of additional constraints. For example, using Overleaf lets authors write the article's content synchronously in W2, but requires them to use Overleaf's web-based editor and to stay connected to the internet to read and edit the manuscript. Conversely, in W3 and W4, Git lets collaborators work offline with the editor of their choice, but at the cost of manually committing changes and resolving conflicts when they finally synchronise.

The additional workload of using a Git repository is manageable for the contributors of workflow W4, who are trained computer scientists, but is too demanding for other collaborators such as the editor. As a result, the main author must regularly convert a snapshot of the Git repository into a DOCX file that they email to the editor. To work around this type of constraint, W3 includes a headless content-management system (CMS) that lets the author and the editor modify the catalogue's content by editing Markdown files with a rich-text editor. A button commits the changes to the Git repository. In case of a conflict, the last change silently wins.

6.3.5 Pa5 — Syncing formats is hard, especially when the target file exists. Fewer than half the steps between formats (horizontal arrows) involve executing a predefined command or a workflow-specific script after source file changes, either automatically (23%) or via a GUI application or command-line interface²⁴ (20%). Over half the other synchronisation steps (vertical arrows) are manual (57%). Further automating fully manual steps is difficult due to what Brooks refers to as *essential* and *accidental* complexities [12]. An essential complexity is the need to manually process certain changes, with a corresponding increase in cognitive effort that cannot be automated, such as deciding how to rephrase the content of an AsciiDoc file based on a PDF annotated by the editor in W4. An accidental source of difficulty is that most formats and programs do not include sufficient information for mapping a change in an "output" file back to the source file that generated it.

7 Discussion

The first study provides concrete examples of both successful and unsuccessful collaborations among independent publishing professionals, who often experience inefficient, error-prone communication loops that slow down their workflows. The second study uses *ecologies of formats* through the intermediary of *information synchronisation graphs* to illustrate the key role of file formats in understanding and addressing these issues. These results indicate that the lack of support for diverse file formats, both by tools and by social organisations, results in more errors, increased workloads, greater tension, and often-unsatisfactory workarounds.

7.1 Publishing relies on multiple file formats

Publishing workflows are composed of multiple steps that transform information from one format to another, effectively duplicating and scattering information across multiple files. We discuss how this process produces multiple communication breakdowns.

7.1.1 Mismatched file formats require duplicate effort. The wide range of software used in publishing leads to duplicating information across formats (T1, Pa2). For example, content can be authored using a word processor such as Microsoft Word (DOCX files) or with a markup languages such as

²⁴Most of these steps could be automated by executing a script whenever the source file changes, as in FileWeaver [28]. This is mostly a design choice, not an intrinsic feature of the transformations themselves.

Markdown (MD), L^AT_EX (TEX) or AsciiDoc (ADOC). Similarly, although Adobe InDesign (INDD files) dominate print layout, some alternatives rely on CSS for styling and on HTML for previewing.

While restricting software choices may help limit the number of formats at play, this top-down approach does not fit the reality of publishing: many authors prefer word processors [60] (T1), whereas designers often use InDesign for lack of a better alternative (T4). Some content-authoring tools are relatively easy to integrate into constrained workflows, both because certain formats—such as DOCX—are widely supported and because other formats can often be converted using existing tools such as Pandoc (T1, Pa5). However, this does not apply to graphic designers, who are trained for and restricted to the Adobe software ecosystem (T4; see also Blanc [8]). Sometimes a professional simply cannot work with a specific software application (and format) because they lack access to the appropriate device, licence or operating system (T4), which then requires creating additional, intermediate formats.

7.1.2 Collaborators struggle to synchronise information. Files with specific formats serve as boundary objects [69] across professionals from different communities (T1, Pa1, Pa4). Although typeset or annotated documents are often exchanged as PDF files, designers often use DOCX files to import content into other applications such as InDesign (W1) or to track changes within a feedback loop (T3, Pa3). Although many applications can export PDF and DOCX files, they often require additional tools (Pa5), as in W4, and may be difficult to import or re-import (T3, Pa5).

Different data-exchange techniques impose constraints on the data formats. For example, while email attachments and file sharing websites support binary formats such as DOCX and PDF, Git and collaborative text editors such as Overleaf use plain-text formats, such as MD and TEX. Workflows that require both formats (W3 and W4) must abandon either version control features, e.g. Git, or changes tracked in only one format, e.g. DOCX.

7.1.3 Challenges synchronising multiple outputs from a single source. Each media targeted by a multiformat publishing workflow requires specific content and/or layout (T2, Pa2), which must often be encoded in different files with output-specific formats. Common export formats, such as PDF for printing, HTML for websites and EPUB for ebooks, all differ from formats used to produce these outputs, such as DOCX, MD and INDD. Some publications also involve more exotic output formats, such as interactive notebooks (W4). In addition, some books created during live events—such as a party or radio show (T2)—are considered “experiential output” that cannot be captured by a digital format. Similarly, the crowd’s experience writing and playing during a *live coding* concert [7] differs greatly from reading and executing the code at home.

7.1.4 Format switching makes it harder to track progress. Publishing professionals sometimes use format changes to mark specific stages in the book production process. For example, when a graphic designer receives a manuscript written in Microsoft Word (W1) that has been validated by the editor, they usually switch to a non-modifiable format such as PDF to lay it out. This lets them control which content, if any, the author can change (T3). We found that many professionals explicitly avoid synchronous collaborative tools (T3) and include specific contract clauses to maintain control over further author modifications (T4). Several graphic designers provide early layout previews to the author, partly as motivation and partly to encourage them to address details, such as shortening a paragraph to make it fit on a single page (T3) that they would otherwise need to fix themselves. Some create duplicate or annotated files (T2, W1, W4) to track the document’s progress.

7.1.5 Collaborators must juggle competing obligations and preferences. Publishing workflows involve multiple contributors with different needs and preferences, which in turn affects the choice of formats and how software is (or is not) used. For example, after a graphic designer uses InDesign to create a PDF document, the editor may hand-annotate a printed version, mark it up on a tablet,

or use Adobe software to insert specially encoded annotations that can be incorporated directly into InDesign (T2, T3, Pa3). The choice of method depends not only on the editor's preferences, but also on hardware, software, time and financial considerations (T4).

Although alternatives such as $\text{\LaTeX}$ (W2) and CSS for paginated webpages (W3) do exist, they rely on formats that are incompatible with WYSIWYG software such as InDesign, and are rarely combined, despite their complementary qualities. Moreover, workflows that rely on $\text{\LaTeX}$ or CSS require designers who are willing and able to use such formats or else omit the designer, which increases work for the author and often results in lower quality output (T4, W2, Pa1).

7.2 Current work practices and tools lack support for multiple file formats

Unfortunately, neither current publishing tools nor the social organisations that use them provide adequate support for complex, multiformat publishing workflows.

7.2.1 Lack of adequate tooling. The technology that underlies the two types of information synchronisation at play in publishing workflows—between people and between formats—suffer from a number of limitations (T1, Pa4, Pa5). Collaborators already use asynchronous collaboration tools such as email (W1, W4) and Git (W3, W4), which may be combined within a single workflow (W4) or hidden behind simpler interfaces (W3) to make them more easily accessible to all collaborators. Others use synchronous collaboration tools, such as Overleaf, Etherpad and Google Docs, but these lack a number of critical features such as providing and restricting access to some contributors (T3).

Similarly, converting a file to another format may require using a compiler or converter that may not support the source or destination formats, destroy valuable information, or require project-specific scripts that can only be written by skilled professionals (T1, W4). In addition, the systematic presence of loops in workflows (T3, Pa3) implies that changes made to one file must be integrated into other files whose content has become out-of-date. Current tools cannot automate this process, which results in repetitive manual work (Pa5), often carried out by professionals who do not consider it their responsibility (T1, T4, Pa1).

7.2.2 Lack of education and planning. Many professionals do not benefit from state-of-the-art tools and techniques either because they are unaware of their existence or lack the necessary skills to use them. The technical skills required for tools that synchronise information, such as Git and Pandoc, often exceed those of non-technical professionals such as authors and graphic designers. For example, many authors prefer direct formatting tools such as Microsoft Word but are unable to use essential features such as paragraph styles (T1).

This lack of technical education often wastes time and causes extra work. For example, graphic designers may redo the orthotypography of an entire section because of improperly tracked changes. They may also strip direct formatting from an author's file and replace it with different semantics even though this is normally the author's job (T1, T3). This shift in responsibility is exacerbated by small team sizes and the disappearance of specialised roles such as editorial secretaries. The remaining professionals are expected to be skilled with software and processes that were not designed for them and for which they were not trained, resulting in overwork and challenging their own personal values (T4).

Better understanding and communicating such issues may help create better workflows. For example, deciding that authors must write content in a format such as Markdown automatically prevents them from using direct formatting to distinguish semantic units such as headings and quotes. However, this requires giving authors access to tools that support these alternative formats (W3) and/or (semi-)manually converting them to other formats (W4), which further burdens more technically skilled individuals (T1, Pa5). This also suggests that moving away from proprietary

software such as InDesign, which does not document its INDD format, would trade interactivity and output quality for more diverse and experimental formats and practices (T2, T4).

8 Recommendations

The following recommendations address format-related challenges in independent publishing workflows, based on the findings from the two studies and the discussion.

Prioritise open and extensible formats. Since collaboration in publishing depends upon conversions between formats, we advocate avoiding formats that are difficult to parse and convert in favour of simpler, documented alternatives. When working with complex formats is unavoidable, we further suggest inspecting the format in search of workarounds. For example, a seemingly binary format such as DOCX can be unzipped to reveal XML files that are easier to read and transform programmatically. Moreover, while every format constrains what information can be represented, some formats leave more space for idiosyncratic appropriations than others. For example, AsciiDoc and Djot²⁵ are natively extensible. Their compilers are immune to syntax errors, making them fit to progressively add custom markup as unplanned needs appear during the publication’s lifetime. Appendix A lists relevant properties of common publishing formats.

Track changes and provenance. Information about the history and the origin of files and their content must be preserved to better account for loops in workflows. We advocate tracking history with versioning systems such as Git or versioned formats such as CRDTs, as well as tracking provenance, either within the file itself, e.g., with invisible annotations in PDFs or attributes in HTML, or via auxiliary source map files²⁶. Note that standard converters may discard such information, which might require explicit encoding in the source file or recovered using additional files and programs, such as SyncTeX for $\text{\LaTeX}$ [45].

Broaden tools to support richer interaction. Many powerful tools that synchronise files within workflows, such as Git, Pandoc and Make, only exist as command-line utilities or complex, often limited, GUIs, making them out of reach for most publishing professionals. Using more familiar interfaces, such as a button to force-push changes to a Git repository, and gradually exposing their features would facilitate adoption by non-specialists. We therefore recommend using or configuring readily available tools, such as a headless CMS, and developing additional front-ends for programs such as compilers and converters.

Maintain and prioritise semantic consistency. Semantic structuring gives meaning to data, such as marking a line of text as a heading, and links files that contain related content together, such as citing a reference from a BibTeX file. We therefore advocate authoring content using formats and editors that prioritise meaning over form, such as rich text editors featuring structuring commands rather than direct formatting commands such as font size and weight. We also recommend preserving semantics across formats throughout a workflow, wherever possible, to minimise information loss between the first and final stages of the editorial process. When documents must be structured retroactively, tools such as large language models (LLMs) may help post-process unstructured documents submitted by authors²⁷, though these solutions come with significant ecological and ethical implications. In addition, we recommend teaching semantic structuring to a broader audience, including the general public, as greater digital literacy benefits all publishing professionals.

²⁵<https://github.com/jgm/djot>.

²⁶https://developer.mozilla.org/en-US/docs/Glossary/Source_map.

²⁷See, e.g., <https://azure.microsoft.com/products/ai-foundry/tools/document-intelligence> and <https://www.docling.ai>.

Advocate for format literacy. Designing collaborative workflows requires attention to how content evolves across formats: what information must be preserved, what may be lost, and where conversions introduce additional work. Attending to these interdependencies requires a practical awareness of file formats, their expressive limits, their conversion paths, the values they carry and the software ecosystems that surround them. We argue that such “format literacy” is an important aspect of digital literacy. While such specialised knowledge may be unnecessary for non-professional users, it is necessary for professionals engaged in collaborative book production. In practice, greater awareness of format-related constraints would help them better anticipate the downstream consequences of their format choices.

Design malleable workflows. Workflows must sometimes be adjusted to accommodate additional constraints, such as producing accessible content to comply with new legislation. Changing established professional and personal habits can be difficult or even damaging [14]. We thus recommend creating open, composable and malleable [71, 22] publishing workflows that can incrementally accommodate changes and adapt to each stakeholder’s needs, practices and preferences.

Share reusable features. Software that is customised for publishing can often benefit more than one workflow. We therefore recommend that such changes be integrated into existing software stacks. For example, the prototype presented in the first study injected provenance information into a popular, open-source HTML-to-PDF converter in the form of annotations on the generated PDF. This feature was officially incorporated into the main version of the software and is now available to all users. This strategy of reinforcing the links between source markup and generated PDFs can be applied to a variety of other applications, and was identified as a key need in a previous study of $\text{\LaTeX}$ users [27, p. 5].

9 Limitations and future work

This paper addresses a specific set of existing publishing practices. The first study focused on French independent publishing professionals engaged in multiformat publishing. As such, the findings may not generalise to different audiences beyond independent publishers, although some large-scale, non-French publishers already use web-to-print workflows similar to those we studied.²⁸ The second study partially addresses this limitation by exploring four additional real-world publishing workflows. However, a more extensive study of non-traditional publishing workflows, or workflows that exist at an organisational, macro level [52], is beyond the scope of this paper.

A deeper understanding of the sociological implications of power dynamics and distribution of responsibility would benefit any specific design solution, since challenging established behaviour can cause new problems [14]. This is especially important for independent publishing houses that lack the time and financial resources to develop custom toolchains.

Our results suggest that advances in format conversion software would increase both the breadth of supported formats and the depth and quality of the conversion, which often discards provenance information crucial for later stages. Future research could explore how schema change [46, 20], adversarial interoperability²⁹, and reactive file systems, such as FileWeaver [28] and Patchwork³⁰, could be used to improve publishing workflows.

Finally, ecologies of formats may also help researchers and practitioners in other fields, especially those where files act as boundary objects [31], including knowledge work [28], bio-informatics [70] and data science [40], or more generally any collaborative practice that requires conversion across

²⁸<https://www.xml.com/articles/2017/02/20/beyond-xml-making-books-html>.

²⁹<https://www.eff.org/deeplinks/2019/10/adversarial-interoperability>.

³⁰<https://www.inkandswitch.com/patchwork/notebook>.

multimedia file formats. For example, healthcare practitioners must perform invisible repair work across documents, systems, and people [33]. They re-enter, translate and reorganise information from diverse, incompatible sources to make it “fit” the appropriate electronic health record [10].

10 Conclusion

This paper investigates the challenges faced by independent publishing professionals as they collaborate on books with both print and digital formats. Although recent advances in web technologies have let them experiment with a variety of multiformat books, they remain tied to workflows that were developed for print-only publishing. This paper explores two key research questions: *How do independent publishing professionals collaborate across heterogeneous software environments to produce multiformat publications?* (RQ1) and *How can we enhance their workflows to help mitigate problems that arise from conflicting constraints and work practices?* (RQ2).

The first key contribution is an empirical participatory design study of 16 publishing professionals. We collected concrete examples of their daily practices, the breakdowns they encountered, and the workarounds they implemented. We found that the lack of synchronisation across data formats negatively affects collaboration: many participants struggle to transform a single manuscript into multiformat output, although some have developed compelling workarounds that reduce undesirable and inefficient publishing loops. Many of these publishing professionals said they must give up some of their personal values and autonomy in exchange for increased productivity. In contrast to the research literature, we find that the primary cause of collaboration breakdowns arises from interoperability problems across file formats rather than a lack of coordination across software applications.

The second key contribution is the concept of *ecologies of formats* which acts as an analytical lens for examining the interplay between formats and people in a highly collaborative workflow. We operationalise this theoretical framework through the introduction of an analytical tool: *information synchronisation graphs*, a novel graph notation that reveals recurring patterns in how publishing professionals exchange documents with diverse formats. Instead of focusing on different software applications, these graphs trace how publishing professionals create and exchange files with multiple formats and highlight inefficient or error-prone loops that may cause breakdowns. We use this graph notation to illustrate and analyse four real-world publishing workflows, and show that incompatible file formats often force inefficient and error-prone loops that disrupt the publication process.

Together, these two studies show that current work practices and toolsets fail to support the multiplicity of file formats that is both unavoidable and the source of multiple barriers to collaboration. We argue that addressing format conversion issues is essential for reducing friction in publishing workflows and suggest the following seven recommendations: publishing professionals should adopt open and extensible formats and strengthen their format literacy; more powerful tools should improve tracking of changes and provenance; provide richer interaction; prioritise and maintain semantic consistency; support more malleable workflows; and share reusable features. We conclude with directions for future research that adapts systems and organisations to better support both established and emerging work practices, while acknowledging the inherent constraints in multiformat publishing.

Acknowledgments

We thank the participants of the group discussions study for their time and valuable feedback. We are especially grateful to Julie Blanc for the time she spent clarifying W3 for us. Finally, we extend our deepest appreciation to the members of éditions Burn~Août, the PrePostPrint community, and the bureau 339 for their countless insights, which were instrumental in shaping this research. This work was partially supported by a French government grant managed by the Agence Nationale de la Recherche as part of the France 2030 program, reference ANR-22-EXEN-0003 (PEPR eNSEMBLE / PILOT), and by the ERC PoC grant № 101113339 OnePub “Single-source collaborative publishing”.

References

- [1] Janneke Adema. 2021. *Living Books: Experiments in the Posthumanities*. The MIT Press. doi:10.7551/mitpress/11297.001.0001.
- [2] Maristella Agosti and Nicola Ferro. 2007. A formal model of annotations of digital content. *ACM Transactions on Information Systems*, 26, 1, 3–es. doi:10.1145/1292591.1292594.
- [3] Thibaut Arribe. 2014. *Conception des chaînes éditoriales : documentariser l’activité et structurer le graphe documentaire pour améliorer la maîtrise de la rééditorialisation*. Ph.D. Dissertation. Université de Technologie de Compiègne. <https://theses.hal.science/tel-01127476>.
- [4] Elisa Bastianello, Christopher Tomlinson, and Alessandro Adamou. 2024. PubLink: editorial workflow for digital scholarly publications in the humanities. In *Proceedings of the 35th ACM Conference on Hypertext and Social Media (HT’24)*. ACM, 318–322. doi:10.1145/3648188.3677051.
- [5] Alison Baverstock, Richard Bradford, and Madelena Gonzalez, (Eds.) *Contemporary publishing and the culture of books*. Routledge Literature Companions. Routledge, Milton Park, Abingdon, Oxon New York, NY. 1 p. ISBN: 978-1-317-69630-8.
- [6] Alex Bigelow, Steven Drucker, Danyel Fisher, and Miriah Meyer. 2017. Iterating between tools to create and edit visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 23, 1, (Jan. 2017), 481–490. Conference Name: IEEE Transactions on Visualization and Computer Graphics. doi:10.1109/TVCG.2016.2598609.
- [7] Alan F. Blackwell, Emma Cocker, Geoff Cox, Alex McLean, and Thor Magnusson. 2022. *Live Coding: A User’s Manual*. The MIT Press. doi:10.7551/mitpress/13770.001.0001.
- [8] Julie Blanc. 2023. *Composer avec les technologies du web : genèses instrumentales collectives pour le développement d’une communauté de pratique de designers graphiques*. Ph.D. Dissertation. Université Paris-VIII-Vincennes-Saint-Denis. <https://theses.fr/2023PA080046>.
- [9] Julie Blanc and Nolwenn Maudet. 2022. Code ↔ design graphique : dix ans de relations. In *Création, outils, recherche*. Number 1286-2584 in Graphisme en France. Vol. 28. CNAP, Paris, 3–30. <https://www.cnap.fr/actualites/graphisme-e-n-france/revues/ndeg28-graphisme-en-france-creation-outils-recherche-2022>.
- [10] Claus Bossen, Lotte Groth Jensen, and Flemming Witt. 2012. Medical secretaries’ care of records: the cooperative work of a non-clinical group. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (CSCW ’12)*. ACM, 921–930. ISBN: 978-1-4503-1086-4. doi:10.1145/2145204.2145341.
- [11] Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health*, 11, 4, 589–597. _eprint: <https://doi.org/10.1080/2159676X.2019.1628806>. doi:10.1080/2159676X.2019.1628806.
- [12] Frederick P. Brooks. 1987. No silver bullet — essence and accidents of software engineering. *Computer*, 20, 4, 10–19. doi:10.1109/MC.1987.1663532.
- [13] Sankalan Pal Chowdhary, Dipendra Manocha, M. Balakrishnan, Akashdeep Bansal, and Himanshu Garg. 2019. Making legacy digital content accessible at source. In *Proceedings of the 16th International Web for All Conference (W4A ’19)*. ACM, 1–2. doi:10.1145/3315002.3332444.
- [14] Dave Clark. 2002. Rhetoric of present single-sourcing methodologies. In *Proceedings of the 20th annual international conference on Computer documentation (SIGDOC ’02)*. ACM, 20–25. doi:10.1145/584955.584959.
- [15] Stéphane Crozat. 2012. Structured and fragmented content in collaborative XML publishing chains. In *Proceedings of the 2012 ACM symposium on Document engineering (DocEng ’12)*. ACM, 145–148. doi:10.1145/2361354.2361388.
- [16] Robin de Mourat. 2020. *Le vacillement des formats*. Ph.D. Dissertation. Université Rennes 2. <http://www.these.robindemourat.com/>.
- [17] Angelo Di Iorio, Andrea Giovanni Nuzzolese, Francesco Osborne, Silvio Peroni, Francesco Poggi, Michael Smith, Fabio Vitali, and Jun Zhao. 2015. The RASH framework: enabling HTML+RDF submissions in scholarly venues. In *14th International Semantic Web Conferences (ISWC 2015)*. Bethlehem, PA. http://figshare.com/articles/Poster_of_t

- [he_ISWC_2015_demo_paper_The_RASH_Framework_enabling_HTML_RDF_submissions_in_scholarly_venues_/1572159.](#)
- [18] Dan Diaper and Martin Beer. 1994. Collaborative document annotation using electronic mail. *Computer Supported Cooperative Work (CSCW)*, 3, 3, 297–325. doi:[10.1007/BF00750744](#).
- [19] Tom Djajadiningrat, Bill Gaver, and Joep Frens. 2000. Interaction relabelling and extreme characters: methods for exploring aesthetic interactions. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques (DIS’00)*. ACM, 66–71. doi:[10.1145/347642.347664](#).
- [20] Jonathan Edwards, Tomas Petricek, Tijs van der Storm, and Geoffrey Litt. 2024. Schema evolution in interactive programming systems. *The Art, Science, and Engineering of Programming*, 9, 1, 2. eprint: [2412.06269\[cs\]](#). doi:[10.22152/programming-journal.org/2025/9/2](#).
- [21] Stephen E. Edwards. 2002. Newspaper editorial systems get ready for cross-media future. *Seybold Report: Analyzing Publishing Technologies*, 2, 16, 13. <https://research.ebsco.com/linkprocessor/plink?id=60bcb17-49dc-3667-bfd1-0dd8264a0931>.
- [22] Antoine Fauchié. 2024. *Fabriquer des éditions, éditer des fabriques : reconfiguration des processus techniques éditoriaux et nouveaux modèles épistémologiques*. Ph.D. Dissertation. Université de Montréal. <https://these.quaternum.net/>.
- [23] Antoine Fauchié and Yann Audin. 2023. The importance of single source publishing in scientific publishing. *Digital Studies / Le champ numérique*, Special DSCN Collection #9. Number: Special DSCN Collection #9. doi:[10.16995/dscn.9655](#).
- [24] Antoine Fauchié, Roch Delannay, Michael E. Sinatra, and Marcello Vitali-Rosati. 2023. Exploring new (digital) publishing practices with *le pressoir*. *Pop! Public. Open. Participatory*, 5. <https://popjournal.ca/issue05/fauchie>.
- [25] Lisa Formentini, François Charoy, and Matthieu Tixier. 2025. Studying digitally supported collaboration through Artifact Ecologies: a State of art. In *Proceedings of the 36th Conference on l’Interaction Humain-Machine (IHM ’25)*. ACM, 1–13. ISBN: 979-8-4007-1522-8. doi:[10.1145/3765712.3765731](#).
- [26] Francisco-Javier Garcia-Marco. 2011. Libraries in the digital ecology: reflections and trends. *The Electronic Library*, 29, 1, 105–120. doi:[10.1108/02640471111111460](#).
- [27] Camille Gobert and Michel Beaudouin-Lafon. 2022. *i- $\LaTeX$* : manipulating transitional representations between $\LaTeX$ code and generated documents. In number 554. CHI 2022 -CHI Conference on Human Factors in Computing Systems. ACM, 1. doi:[10.1145/3491102.3517494](#).
- [28] Julien Gori, Han L. Han, and Michel Beaudouin-Lafon. 2020. FileWeaver: flexible file management with automatic dependency tracking. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST ’20: The 33rd Annual ACM Symposium on User Interface Software and Technology. ACM, Virtual Event USA, 22–34. doi:[10.1145/3379337.3415830](#).
- [29] A. Groten. 2022. *Figuring things out together: on the relationship between design and collective practice*. Ph.D. Dissertation. Leiden University. <https://hdl.handle.net/1887/3487176>.
- [30] Karl A. Hakkarainen. 1999. Unifying documentation teams. In *Proceedings of the 17th annual international conference on Computer documentation (SIGDOC ’99)*. ACM, 176–184. doi:[10.1145/318372.318591](#).
- [31] Richard Harper, Siân Lindley, Eno Thereska, Richard Banks, Philip Gosset, Gavin Smyth, William Odom, and Eryn Whitworth. 2013. What is a file? In *Proceedings of the 2013 conference on Computer supported cooperative work (CSCW ’13)*. ACM, 1125–1136. doi:[10.1145/2441776.2441903](#).
- [32] Tamir Hassan and Niranjana Damara Venkata. 2015. The browser as a document composition engine. In *Proceedings of the 2015 ACM Symposium on Document Engineering (DocEng ’15)*. ACM, 3–12. doi:[10.1145/2682571.2797063](#).
- [33] Morten Hertzum, Gunnar Ellingsen, and Camilla Bjørnstad. 2025. Repair work in electronic medication management. *Computer Supported Cooperative Work (CSCW)*, 34, 3, 949–977. doi:[10.1007/s10606-025-09526-0](#).
- [34] Claudia-Lavinia Ignat and Moira C. Norrie. 2008. Multi-level editing of hierarchical documents. *Computer Supported Cooperative Work (CSCW)*, 17, 5, 423–468. doi:[10.1007/s10606-007-9071-2](#).
- [35] Heekyoung Jung, Erik Stolterman, Will Ryan, Tonya Thompson, and Marty Siegel. 2008. Toward a framework for ecologies of artifacts: how are digital artifacts interconnected within a personal life? In *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges (NordCHI ’08)*. ACM, 201–210. doi:[10.1145/1463160.1463182](#).
- [36] Peter Kent. 2004. *Word 2003 Bible*. In collab. with David Angell, Peter Kent, and Brent Heslop. OCLC: 899177878. John Wiley & Sons, New York, NY. ISBN: 978-0-7645-5573-2.
- [37] Hee-Cheol (Ezra) Kim and Kerstin Severinson Eklundh. 2001. Reviewing practices in collaborative writing. *Computer Supported Cooperative Work (CSCW)*, 10, 2, 247–259. doi:[10.1023/A:1011229212323](#).
- [38] Martin Kleppmann, Adam Wiggins, Peter van Hardenberg, and Mark McGranaghan. 2019. Local-first software: you own your data, in spite of the cloud. In *Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward! 2019)*. ACM, 154–178. doi:[10.1145/3359591.3359737](#).

- [39] Clemens Nylandstedt Klokmose, James R. Eagan, and Peter van Hardenberg. 2024. MyWebstrates: webstrates as local-first software. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (UIST '24). ACM, 1–12. doi:[10.1145/3654777.3676445](https://doi.org/10.1145/3654777.3676445).
- [40] Laura Koesten, Emilia Kacprzak, Jeni Tennison, and Elena Simperl. 2019. Collaborative practices with structured data: do tools support what users need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (CHI '19). ACM, 1–14. doi:[10.1145/3290605.3300330](https://doi.org/10.1145/3290605.3300330).
- [41] Henrik Korsaard, Peter Lyle, Joanna Saad-Sulonen, Clemens Nylandstedt Klokmose, Midas Nouwens, and Susanne Bødker. 2022. Collectives and their artifact ecologies. *Proceedings of the ACM on Human-Computer Interaction*, 6, 432:1–432:26, CSCW2. doi:[10.1145/3555533](https://doi.org/10.1145/3555533).
- [42] Klaus Krippendorff. 1989. On the essential contexts of artifacts or on the proposition that "design is making sense (of things)". *Design Issues*, 5, 2, 9–39. doi:[10.2307/1511512](https://doi.org/10.2307/1511512).
- [43] Anukriti Kumar and Lucy Lu Wang. 2024. Uncovering the new accessibility crisis in scholarly PDFs: publishing model and platform changes contribute to declining scholarly document accessibility in the last decade. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS '24). ACM, 1–16. doi:[10.1145/3663548.3675634](https://doi.org/10.1145/3663548.3675634).
- [44] Ida Larsen-Ledet, Henrik Korsaard, and Susanne Bødker. 2020. Collaborative writing across multiple artifact ecologies. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (CHI '20). ACM, 1–14. doi:[10.1145/3313831.3376422](https://doi.org/10.1145/3313831.3376422).
- [45] Jérôme Laurens. 2008. Direct and reverse synchronization with syntex. *TUGBoat*, 29, 3, 365–371.
- [46] Geoffrey Litt, Peter van Hardenberg, and Orion Henry. 2021. Cambria: schema evolution in distributed systems with edit lenses. In *Proceedings of the 8th Workshop on Principles and Practice of Consistency for Distributed Data* (PaPoC '21). ACM, 1–9. doi:[10.1145/3447865.3457963](https://doi.org/10.1145/3447865.3457963).
- [47] Geoffrey Litt, Sarah Lim, Martin Kleppmann, and Peter van Hardenberg. 2022. Peritext: a CRDT for collaborative rich text editing. *Proc. ACM Hum.-Comput. Interact.*, 6, 531:1–531:36, CSCW2. doi:[10.1145/3555644](https://doi.org/10.1145/3555644).
- [48] Patrice Lopez. 2009. GROBID: combining automatic bibliographic data recognition and term extraction for scholarship publications. In *Research and Advanced Technology for Digital Libraries*. Vol. 5714. Series Title: Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, 473–474. doi:[10.1007/978-3-642-04346-8_62](https://doi.org/10.1007/978-3-642-04346-8_62).
- [49] Gaëlle Lortal, Myriam Lewkowicz, and Amalia Todirascu-Courtier. 2005. Annotation: textual media for cooperation. In *IWAC*, 41–50. http://myriam.lewkowicz.free.fr/publi/Annotation_for_collaboration_Lortal_Lewkowicz_Todirascu.pdf.
- [50] Silvio Lorusso. 2016. *Extending Horizons - The Praxis Of Experimental Publishing In The Age Of Digital Networks*. Ph.D. Dissertation. University of Venice. doi:[10.13140/RG.2.2.13029.73443](https://doi.org/10.13140/RG.2.2.13029.73443).
- [51] Alessandro Ludovico. 2012. *Post-digital print: the mutation of publishing since 1894*. Number 77 in *Onomatopée*. Onomatopée, Eindhoven. ISBN: 978-90-78454-87-8.
- [52] Peter Lyle, Henrik Korsaard, and Susanne Bødker. 2020. What's in an ecology? a review of artifact, communicative, device and information ecologies. In *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society* (NordiCHI '20). ACM, 1–14. doi:[10.1145/3419249.3420185](https://doi.org/10.1145/3419249.3420185).
- [53] Michael S. MacFadden, Agustina Ng, Claudia-Lavinia Ignat, Ning Gu, and Chengzheng Sun. 2017. The fifteenth international workshop on collaborative editing systems. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (CSCW '17 Companion). ACM, 351–354. doi:[10.1145/3022198.3022653](https://doi.org/10.1145/3022198.3022653).
- [54] Wendy E. Mackay. 2023. DOIT: the design of interactive things. selected methods for quickly and effectively designing interactive systems from the user's perspective. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM. doi:[10.1145/3544549.3574172](https://doi.org/10.1145/3544549.3574172).
- [55] Wendy E. Mackay. 1990. Patterns of sharing customizable software. In *Proceedings of the 1990 ACM conference on Computer-supported cooperative work* (CSCW '90). ACM, 209–221. doi:[10.1145/99332.99356](https://doi.org/10.1145/99332.99356).
- [56] Wendy E. Mackay and Michel Beaudouin-Lafon. 2023. Participatory design and prototyping. In *Springer International Publishing*, 1. doi:[10.1007/978-3-319-27648-9_31-1](https://doi.org/10.1007/978-3-319-27648-9_31-1).
- [57] David W. McDonald, Chunhua Weng, and John H. Gennari. 2004. The multiple views of inter-organizational authoring. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (CSCW '04). ACM, 564–573. doi:[10.1145/1031607.1031703](https://doi.org/10.1145/1031607.1031703).
- [58] Bonnie A. Nardi and Vicki O'Day. 1999. *Information Ecologies: Using Technology with Heart*. The MIT Press. doi:[10.7551/mitpress/3767.001.0001](https://doi.org/10.7551/mitpress/3767.001.0001).
- [59] William Newman. 2012. Design case study: the bravo text editor. *interactions*, 19, 1, 75–80. doi:[10.1145/2065327.2065342](https://doi.org/10.1145/2065327.2065342).
- [60] Sylvie Noël and Jean-Marc Robert. 2004. Empirical study on collaborative writing: what do co-authors do, use, and like? *Computer Supported Cooperative Work (CSCW)*, 13, 1, 63–89. doi:[10.1023/B:COSU.0000014876.96003.be](https://doi.org/10.1023/B:COSU.0000014876.96003.be).

- [61] Midas Nouwens and Clemens Nylandsted Klokmoose. 2018. The application and its consequences for non-standard knowledge work. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18). ACM, 1–12. doi:[10.1145/3173574.3173973](https://doi.org/10.1145/3173574.3173973).
- [62] Pia Pol and Astrid Vorstermans, (Eds.) Futurebook(s): sharing ideas on books and (art) publishing = Toekomstboek(en): over (kunst)publicaties en uitgeven. Valiz, Amsterdam. 495 pp. ISBN: 978-94-93246-27-0.
- [63] Chiara Rossitto, Cristian Bogdan, and Kerstin Severinson-Eklundh. 2014. Understanding constellations of technologies in use in a collaborative nomadic setting. *Computer Supported Cooperative Work (CSCW)*, 23, 2, 137–161. doi:[10.1007/s10606-013-9196-4](https://doi.org/10.1007/s10606-013-9196-4).
- [64] Martina Angela Sasse, Mark James Handley, and Shaw Cheng Chuang. 1993. Support for collaborative authoring via email: the MESSIE environment. In *Proceedings of the Third European Conference on Computer-Supported Cooperative Work 13–17 September 1993, Milan, Italy ECSCW '93*. Springer Netherlands, Dordrecht, 249–264. doi:[10.1007/978-94-011-2094-4_17](https://doi.org/10.1007/978-94-011-2094-4_17).
- [65] Svante Schubert, Sebastian Rönna, and Patrick Durusau. 2014. Interoperable document collaboration. In *Proceedings of the 2nd International Workshop on (Document) Changes: modeling, detection, storage and visualization* (DChanges '14). ACM, 1–4. doi:[10.1145/2723147.2723155](https://doi.org/10.1145/2723147.2723155).
- [66] Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. 2011. Conflict-free replicated data types. In *Stabilization, Safety, and Security of Distributed Systems* (Lecture Notes in Computer Science). Springer, Berlin, Heidelberg, 386–400. doi:[10.1007/978-3-642-24550-3_29](https://doi.org/10.1007/978-3-642-24550-3_29).
- [67] Rick Sobiesiak. 1996. Towards a new generation of authoring tools. In *Proceedings of the 13th annual international conference on Systems documentation: emerging from chaos: solutions for the growing complexity of our jobs* (SIGDOC '95). ACM, 157–161. doi:[10.1145/223984.224013](https://doi.org/10.1145/223984.224013).
- [68] Volker Sorge, Akashdeep Bansal, Neha M Jadhav, Himanshu Garg, Ayushi Verma, and M Balakrishnan. 2020. Towards generating web-accessible STEM documents from PDF. In *Proceedings of the 17th International Web for All Conference (W4A '20)*. ACM, 1–5. doi:[10.1145/3371300.3383351](https://doi.org/10.1145/3371300.3383351).
- [69] Susan Leigh Star and James R. Griesemer. 1989. Institutional ecology, ‘translations’ and boundary objects: amateurs and professionals in berkeley’s museum of vertebrate zoology, 1907-39. *Social Studies of Science*, 19, 3, 387–420. doi:[10.1177/030631289019003001](https://doi.org/10.1177/030631289019003001).
- [70] Aurélien Tabard, Wendy Mackay, Nicolas Roussel, and Catherine Letondal. 2007. PageLinker: integrating contextual bookmarks within a browser. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '07). ACM, 337–346. doi:[10.1145/1240624.1240680](https://doi.org/10.1145/1240624.1240680).
- [71] Philip Tchernavskij. 2019. *Designing and Programming Malleable Software*. Ph.D. Dissertation. Université Paris Saclay. <https://theses.hal.science/tel-02612943>.
- [72] Ei Pa Pa Pe-Than, Laura Dabbish, and James D. Herbsleb. 2018. Collaborative writing on GitHub: a case study of a book project. In *Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing* (CSCW '18 Companion). ACM, 305–308. doi:[10.1145/3272973.3274083](https://doi.org/10.1145/3272973.3274083).

A Publishing file formats and their use in practice

Table 1. File formats mentioned in this paper and their uses in the two studies, in alphabetical order. $\emptyset$ indicates that the criterion is non-applicable, and $\cdot$ is short for “No”. *Encoding* indicates whether the format is encoded as plain text, as a zipped archive of plain text files, or if it is binary; *Edited* indicates whether participants edited files in this format (either using a text editor, personal scripts, or dedicated applications); *Converted* indicates whether the participants tried to convert the format into other formats they needed to work with, and if it was easy, hard, or resulted in failure; *Paged* indicates whether the format includes pagination data for its content; *Annotated* indicates whether the format was used to encode and share annotations amongst stakeholders. *Extended* indicates whether custom features were added to the formats by the participants to make it fit their needs.

Extension	Description	Encoding	Edited	Converted	Paged	Annotated	Extended
ADOC	AsciiDoc text	Text	Yes	Easy	$\cdot$	Yes	Yes
BIB	Bibliographic flat-file database	Text	Yes	$\cdot$	$\emptyset$	$\cdot$	$\cdot$
CSS	Webpage style rules	Text	Yes	$\cdot$	$\emptyset$	Yes	Yes
DOCX	Word document	Zipped	Yes	Easy	$\cdot$	Yes	Yes
EPUB	Electronic publication	Zipped	Yes	$\cdot$	Often	Yes	Yes
HTML	Webpage text	Text	Yes	Easy	$\cdot$	Yes	$\cdot$
ICML	InCopy Markup Language	Text	Yes	$\cdot$	$\cdot$	$\cdot$	$\cdot$
IDML	InDesign Markup Language	Zipped	$\cdot$	Failed	$\cdot$	$\cdot$	$\cdot$
INDD	InDesign document	Binary	Yes	Failed	$\cdot$	$\cdot$	$\cdot$
IPYNB	Python notebook	Text	Yes	Easy	$\emptyset$	$\cdot$	$\cdot$
JSON	JavaScript Object Notation	Text	Yes	Easy	$\emptyset$	$\cdot$	$\cdot$
MD	Markdown text	Text	Yes	Easy	$\cdot$	Yes	Yes
ODT	OpenDocument Text	Zipped	Yes	Easy	$\cdot$	Yes	Yes
PDF	Portable Document File	Binary	$\cdot$	Hard	Yes	Yes	$\cdot$
PS	PostScript code	Text	$\cdot$	$\cdot$	Yes	$\cdot$	$\cdot$
SGML	Standard Generalized Markup Language	Text	$\cdot$	$\cdot$	Yes	$\cdot$	$\cdot$
TEI	Text Encoding Initiative	Text	Yes	Hard	Yes	Yes	$\cdot$
TEX	L ^A T _E X text	Text	Yes	Easy	$\cdot$	Yes	Yes
TXT	Plain text without formatting	Text	Yes	$\emptyset$	$\cdot$	$\cdot$	$\cdot$
XML	Extended Markup Language text	Text	Yes	Easy	$\cdot$	Yes	$\cdot$

B Participatory design workshop details

Table 2. Summary of the participants’ characteristics from the first study. Participants are grouped by discussions, for example, P1 to P4 participated to the first group discussion. The *Exp.* column corresponds to the years of experience of the participants working in publishing; *Freq.* is the frequency of use of editorial software; *Multi.* stands for *Multiformat publishing* and acknowledges if the participant has already worked on a multiformat publication project; finally, *Soft.* corresponds to the number of pieces of software that the participant has already used to produce a book.

ID	Age	Genre	Study fields	Practices	Exp.	Freq.	Multi.	Soft.
P1	45–54	Man	Philosophy	Editor, graphic designer	>10	Everyday	Yes	15
P2	25–34	Woman	Law, art history	Editor, translator	1–5	Everyday	No	3
P3	35–44	Woman	Graphic design, cooking	Graphic designer	>10	Everyday	Yes	15
P4	35–44	Woman	Fine arts	Artist, researcher	1–5	Everyday	Yes	8
P5	35–44	Man	Product design, HCI, aesthetics	Designer, researcher	None	Monthly	Yes	7
P6	35–44	Man	Graphic design	Graphic designer	>10	Everyday	No	3
P7	25–34	Woman	Literature, publishing	Editor	1–5	Everyday	Yes	6
P8	25–34	Man	Graphic design	Editor, designer, teacher, UX/UI designer	5–10	Weekly	No	10
P9	45–54	Woman	Stylistics, linguistics, modern literature	Editor	>10	Everyday	Yes	4
P10	35–44	Man	Graphic design	Editor, designer, type designer	>10	Everyday	No	1
P11	25–34	Woman	Graphic design, ergonomics	Designer, researcher, web developer	None	Everyday	Yes	19
P12	35–44	Woman	Graphic design	Editor, designer	5–10	Everyday	No	2
P13	45–54	Man	Audiovisual, film, programming	Artist, director, designer, developer	5–10	Monthly	Yes	10
P14	35–44	Man	Data science	Web developer	5–10	Everyday	Yes	8
P15	45–54	Man	Comp. science, publishing	Publishing manager	>10	Weekly	Yes	11
P16	25–34	Man	Graphic communication	Designer, developer	1–5	Weekly	Yes	15

Received May 13, 2025; revised January 13, 2026; accepted March 17, 2026

Table 3. Details and counts of software used by the participants. Numbers in parentheses correspond to the number of participants who mentioned the corresponding piece of software in their answer to the following question: “What publishing software or workflow do you know? Include all software or system with which you already published documents.” Software are ordered by number of occurrences, then by alphabetical order.

Word processors	LibreOffice (7), Microsoft Word (5), Google Docs (4), Microsoft Office (2), LibreOffice Writer (1), OpenOffice Online (1)
Text editors	Etherpad (3), Sublime Text (3), TextEdit (2), Adobe InCopy (1), HackMD (1), iWork Pages (1), Typora (1), Vim (1), Visual Studio Code (1), VSCodium (1), Zettlr (1)
Composition languages	LaTeX (2), XeLaTeX (1)
Desktop Publishing	Adobe InDesign (13), Scribus (4), Adobe Creative Cloud (1), Aldus PageMaker (1), Microsoft Publisher (1), QuarkXPress (1)
Image editing	InkScape (2), Adobe Photoshop (1), GIMP (1)
Scripts	Node.js (3), Make (2), Python (2), Bash (1), JavaScript (1), Lua (1)
Conversion	Paged.js (7), Métopes (4), Pandoc (4), Prince XML (2), Adobe FrameMaker (1), Domus (now PomDoc) (1), html2print (1), Jouve (now Luminess) (1), Vivliostyle (1), XSLT (1)
Semantics	HTML (3), CSS (3), AsciiDoc (1), EPUB (1), Sigil (1), XML (1), XMLmind (1)
Publishing Platforms	11ty (3), Bookolab (2), Jekyll (2), Kirby CMS (2), Zotero (2), Collabprint (1), Decap CMS (1), Forestry (1), Hugo (1), Netlify (1), OpenEdition (1), Open Journal System (1), Ovide (1), Peritext (1), ProcessWire (1), Stylo (1), Wiki (1)
Collaboration	Git (1), GitHub (1), Mail (1)

Table 4. Details and counts of software used by the participants. Numbers in parentheses correspond to the number of participants who mentioned the corresponding piece of software in their answer to the following question: “What publishing software or workflow do you know? Include all software or system with which you already published documents.” Software are ordered by number of occurrences, then by alphabetical order.

ID	Text preparation & editing		Composition languages	WYSIWYG layout & design		Conversion & multiformat production		Publishing platforms & collaboration	
	Word processors	Text editors		Desktop publishing	Image editing	Scripts	Conversion	Semantics	Platforms
P1	2	.	1	5	.	.	4	2	1
P2	1	1	.	1	.	.	.	.	.
P3	.	1	.	2	2	2	2	2	1
P4	3	3	.	1	.	.	.	.	1
P5	1	.	.	.	.	2	1	.	3
P6	1	1	.	1	.	.	.	.	.
P7	3	.	.	1	.	.	2	.	.
P8	.	.	.	1	.	1	1	2	4
P9	1	.	.	1	.	.	1	1	.
P10	.	.	.	1	.	.	.	.	.
P11	3	3	.	.	.	2	4	.	6
P12	1	.	.	1	.	.	.	.	.
P13	.	1	2	2	1	3	1	.	.
P14	1	1	.	1	.	.	1	1	2
P15	1	1	.	1	.	.	4	3	1
P16	2	4	.	2	1	.	2	.	4