Beyond MS Word: Alternatives and Developments

Christian Rapp, Till Heilmann, and Otto Kruse

Abstract Microsoft Word, the word processing software developed by Microsoft in 1983, established itself as the market leader in the 1990s and 2000s and remained the gold standard for many years. Despite its obvious benefits, it always faced criticism from various quarters. We address the persistent criticism that MS Word is overloaded with features and distracts from writing rather than facilitating it. Alternatives, mainly distraction-free editors and text editors for use with a markup language, are briefly reviewed and compared to MS Word. A serious challenger emerged in 2006 with Google Docs, a cloud-based writing software that has moved text production into the platform era, enabling files to be shared and creating collaborative writing spaces. Even though Google Docs failed to break the dominance of MS Word, it became the trend-setter in online writing. Microsoft and Apple soon followed by designing complex web environments for institutions and companies rather than individual writers. We give an overview of technologies that have evolved to challenge the supremacy of MS Word or compete for market share. By this, we hope to provide clues as to the future development of word processing.

Keywords Alternative word processors · Beyond MS-Word · Distraction-free writing

C. Rapp (✉)
School of Management and Law, Center for Innovative Teaching and Learning, Zurich University of Applied Sciences, Winterthur, Switzerland
e-mail: Christian.Rapp@zhaw.ch

T. Heilmann
Institute of Media Studies, Ruhr-University Bochum, Universitätsstraße 150, 44801 Bochum, Germany
e-mail: till.heilmann@ruhr-uni-bochum.de

O. Kruse
School of Applied Linguistics, Zurich University of Applied Sciences, Winterthur, Switzerland
e-mail: otto.kruse@gmx.net; xkso@zhaw.ch

O. Kruse et al. (eds.), *Digital Writing Technologies in Higher Education*,
https://doi.org/10.1007/978-3-031-36033-6_3

1 Overview

While, for a certain time, MS Word appeared to be the ideal writing tool and was the unchallenged market leader (Bergin, 2006), it had several shortcomings (see Bray, 2013; Johannsen & Sun, 2017; Sharples & Pemberton, 1990; Wilson, 2012), which motivated the search for viable alternatives.

– MS Word is tied to the paper world in several ways. It relies on pages as physical units of text and on the WYSIWYG (what you see is what you get) principle linking the text editor to the paper format. The programme mimics the format of a paper page and enables writers to create layouts and produce text. This direct connection is not inherently necessary as digital word processors can do without pages and, unlike a typewriter, can create the page design in a second step.
– For a long time, MS Word was limited in its capacity to present the mathematical equations and formulae needed in writing about science. Other software, for example, LaTeX, is more flexible in this regard. Similar specializations were created to accommodate specific domains or genres.
– Many writers have found MS Word to be too overloaded with functions for their purposes. Their needs led to the idea of plain-text tools allowing users to fully concentrate on content production while suppressing or masking all other functions.
– Synchronous collaboration of different authors in the same text was impossible in the desktop version of MS Word. This situation only changed when it became accessible as a web service with Office 365.

In the next section, we map the alternatives to MS Word and analyse how the word processing field has developed. We identify drivers for future developments and discuss their meaning for writing practice and the teaching of (academic) writing.

2 Core Idea, Functional Specifications, and Main Products

Several alternatives to MS Word are briefly described below, including the basic idea they follow and their main features.

(1) Office suites such as OpenOffice/LibreOffice include a word processor with similar features to those of MS Word. They are usually free of charge and open source.
(2) Google Docs is also part of an office package. It breaks new ground with a new way of software delivery accessed via a browser and running on a server rather than locally.
(3) Other word processors try to surpass MS Word in certain features, such as right-to-left writing support. MS Word usually incorporates these features over time.

(4) Distraction-free writing software does away with unnecessary functions claiming to help writers focus on the writing process itself.
(5) Markup editors separate text production from formatting and layout to give the user better control of both functions.
(6) Desktop publishing (DTP) programmes supplement rather than replace MS Word but may have a pivotal role in printing.

In this chapter, rather than adding a separate chapter on research, we integrated information on the literature, where available, into the description of the technology. Google Docs is covered in a separate chapter by Castelló et al. ("Synchronous and Asynchronous Collaborative Writing"), and research on word processors is reviewed by Kruse and Rapp ("Word Processing Software: The Rise of MS Word").

2.1 Parallel Solutions to MS Word

One of the word processors developed more or less at the same time as MS Word was StarOffice, which later became OpenOffice (now Apache OpenOffice) with the fork LibreOffice (for a comparison of OpenOffice and LibreOffice see Möhring, 2020). While this Microsoft competitor did not do as well as a business model, it was technically on par with the Microsoft Office Suite (cf. https://wiki.documentfoundation.org/Feature_Comparison:_LibreOffice_-_Microsoft_Office).

The precursor to StarOffice was StarWriter, which was released in 1985 and developed on the OS of Schneider/Amstrad CPC, after which it was exported to DOS. In 1993, it became available in Windows. The programme was developed by the Germany-based company Star Division, which added a complete office suite in 1992 called "office pack 2.0" (see StarOffice, 1998). The suite was expanded several times, and more than 20 million copies were sold. In 1999, Sun Microsystems bought StarOffice and released Version 6 of the suite under an open-source license as free software in 2000. This version can still be downloaded today from the openoffice.org website.

While StarOffice is no longer maintained, OpenOffice and LibreOffice were continuously advanced by a large community of developers. They are available for various operating systems such as Windows, MacOS, Linux, FreeBSD, NetBSD, OpenBSD, and Haiku. Both have no significant shortcomings compared to Microsoft Office. Like Microsoft's products, desktop, mobile, and online versions are available. Neither could, however, ever really compete with Microsoft Office financially, even if Open-/LibreOffice was, for a long time, one of the few office suites running on Linux. Incidentally, Open-/LibreOffice was and still is a feasible choice for public administrations, educational institutions, and companies looking for a free, open-source alternative to Microsoft Office.

2.2 *Writing in the Cloud: Google Docs*

The invention of, and advances in, cloud computing laid the foundations for a new wave of alternatives to MS Word, of which Google Docs is the best known and most used. As these solutions are discussed in depth by Castelló et al. ("Synchronous and Asynchronous Collaborative Writing") in the chapter on collaborative software, only key points are addressed here.

The technology for Google Docs was developed by Tom Schillace, who had co-programmed a word processor called Writely (Hamburger, 2013). Writely was not run locally on a conventional operating system such as Windows or Linux but on a web server; it was implemented to be used remotely via a web browser. It was acquired and adapted by Google in 2005 (McHugh-Johnson, 2021). Within less than a year, Google developed a version it called Docs, along with its online spreadsheet "Sheets". The beta versions of Google Docs and Sheets had many shortcomings compared to the sophisticated, convenient MS Office solutions. However, Google established a collaborative writing feature that allowed synchronous writing as an integral part of a freely available word processor. More importantly, this development opened a door to platform technology that all other providers of writing software had to take: Microsoft did so with Office Online in 2010, and Apple with its iWork apps in 2013 (see Ingraham, 2021). The announcement of Google Docs read as follows:

> With Google Docs & Spreadsheets, Google is taking a set of important tasks and offering an online solution to completing them individually or with a broader group. With a Google Account, a compatible web browser, and an Internet connection, users will now easily be able to:
>
> - Create documents and spreadsheets, and then manage and access them in a single, secure location
> - Easily collaborate with others, online and in real time
> - Export to and import from a wide variety of file formats
> - Share them with others as view-only
> - Publish them to a blog or as an HTML page
>
> Simply put, Google Docs & Spreadsheets is focused on providing users with an innovative and efficient way to create and share information on the Web. (Googlepress, 2006)

It is instructive to see Schillace's perspective on this from an interview with Oliver Burkeman:

> Word processors today were invented 20 years ago, when the endpoint of the document was usually print, so they were very focused in that direction,' Schillace says. 'But nowadays the endpoint of a document is usually communicating [online]: you're posting to a blog or a website, or you're emailing a document around. (Burkeman, 2006)

With the new browser-based word processors, software no longer had to be installed and continuously updated on a local computer but could be executed on a server and accessed through the internet (i.e., software as a service, SaaS). Saving documents was no longer necessary as the cloud-based software stored every input immediately.

In addition, by preserving the text's history, any former version could be restored. However, it became necessary for Google to create an offline function to make writing possible when an internet connection was unavailable or had broken down.

Another implication of platform-based software is that the documents, too, are stored on the server rather than locally. Along with online editors, cloud-based document structures were needed. Dropbox, Google Drive, and One Drive offered such a service with a vast storage capacity. These solutions became the basis for a large-scale file-sharing ability, a prerequisite for collaboration across larger teams or companies.

It is instructive to see what it took for Google to develop its software package beyond its beta status and integrate it into the emerging platform structures of communication, messaging, and networking. This process did not run smoothly but had severe drawbacks. One of the problems Google encountered was the need for synchronization of the online text with the locally stored text, a topic that is all but trivial technologically. In 2007, Google Gears was introduced, a browser extension for Mac, Windows, and Linux. It proved unstable and was dropped again in 2009 in favour of HTML 5 (Ingraham, 2021). Another failure was the introduction of Google Wave in 2009, a web-based platform meant to merge computational, communicative (email, instant messaging, wikis, social networking), collaborative, and writing software. Additional software such as automatic translation, spelling, and grammar checking was added or planned (see Google Wave, 2009). After only two years, it was abandoned, however, and sold to the Apache Software Foundation. Ingraham (2021) suspected that it happened "because it felt like even less of a finished product than most of Google's 'beta' launches."

Still, Google Wave anticipated developments that, ten years later, resulted in Google's "Workspace" (May 2021), previously called "G Suite" and "Google Apps" (a free version for private use with limited features exists as Google Docs Editors, 2022). Microsoft issued MS Teams in 2017 (followed by a free but limited version in 2018). On its part, Apple launched a version of iWork Apps in 2013, with a fully collaborative version to follow only in 2016. All three new platforms are not primarily aimed at individual users but at corporations and institutions that want to help their staff collaborate across the organization. It includes phone, video, messaging, email, text collaboration, translation, and more. For a short time, Google was the market leader in offering these platforms. With a market share of roughly 48 percent, the Microsoft Office package (Office 365) won back the pole position from Google Apps (46,44%) in February 2022 (Vailshery, 2022).

When it adopted cloud computing as the new technology, Google changed word processing forever by enabling truly synchronous writing and, even more, by turning the internet into the place where writing happens. Writing spaces shifted from local computers to the internet and the cloud accessed via the webbrowser, rather than a word processing software. The impact on writing in different contexts and the related research is discussed in depth in the respective chapter by Castelló et al. ("Synchronous and Asynchronous Collaborative Writing").

2.3 Outdoing MS Word

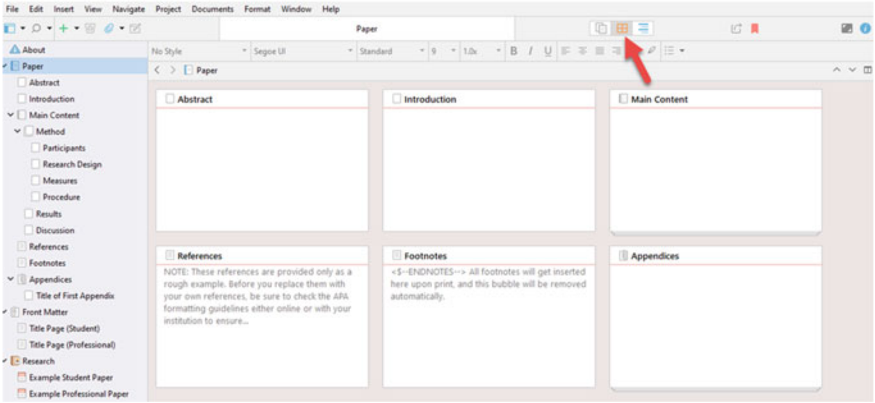
There are only a few applications on the PC word processing market that try to beat MS Word at its own game by offering a better word processor with regard to text editing or formatting capabilities. From the dozens—if not hundreds—of competitors in the 1980s and 1990s, no more than a handful remain today, the most popular one being WordPerfect. WordPerfect was created by Satellite Software International (SSI) and is today developed and distributed by Corel (see https://www.wordperfect.com). WordPerfect is a true WYSYWIG processor which was popular when DOS was the dominating operating system and it lost ground when Windows was introduced (see Bergin, 2006). For a long time, it was operated by key strokes only before it optionally integrated a menu band with key commands. Different from MS Word, control characters were visible within the text indicating what would be a headline or what would be printed in bold. The decline of WordPerfect, which for a period of time in the 1980s was the markt leader, seems to be owed to the increasing unpopularity of DOS, not to the unpopularity of the word processor itself.

Other programmes typically offer features that are—or were, at least—missing or more basic in MS Word and are geared towards audiences with particular needs. A good example of such a feature is support for right-to-left (RTL) writing in scripts like Arabic, Hebrew, or Sindhi. During the 1980s and 1990s, only a handful of PC word processors could handle RTL scripts and text. Even today, a lot of software from the western world still struggles to process non-Roman writing systems correctly (see Stanton, 2021).

By addressing otherwise neglected aspects of writing, competitors to MS Word have highlighted important characteristics and differences between various techniques and practices of writing across cultural, geographical, and linguistic boundaries. Catering to specific requirements and tasks, these programmes question the idea of a universal model for digital writing, a general-purpose word processor, or a one-size-fits-all technological solution to writers' wants and needs. It has to be said, however, that MS Word has always caught up with its competitors by incorporating features such as RTL and reference management.

In addition to the aforementioned WordPerfect, the most notable WYSIWYG alternatives to MS Word are probably Nisus Writer Pro, Mellel, Scrivener, and Storyist. Tellingly, perhaps, all of these, except for Scrivener, are macOS/iOS applications. While the programmes look and feel very much like MS Word and mostly implement near-identical GUI menus and commands for editing and formatting text, they nevertheless seek to differentiate themselves through distinctive functionality.

Nisus Writer Pro, for example, claims superior multilingual text support for writing in nearly any language and script. Similarly, Mellel provides multilingual support and commends itself for academic writing with its advanced bibliography and outlining tools. Storyist, on the other hand, is made explicitly for novelists, playwrights, and screenwriters with templates and formatting tools tailored to the respective literary genres.



NOTE: These references are provided only as a rough example. Before you replace them with your own references, be sure to check the APA formatting guidelines either online or with your institution to ensure...

<$--ENDNOTES--> All footnotes will get inserted here upon print, and this bubble will be removed automatically.

Fig. 1 GUI of Scrivener, storyboard view

Scrivener offers unique modes and features for planning, outlining, and organizing large writing projects in a modular structure. According to Bray (2013), it can be seen as a combination of a distraction-free tool (as discussed in more depth in the next section), creative writing software, and document management. Figure 1 shows one example of an unconventional view provided by Scrivener, in comparison to other word processors like MS Word or Google Docs, the story board view. Three alternative views (text, outline, storyboard) are easily provided by one click (red arrow). Bray (2013, p. 205) pointed out that

> These three types of alternative writing software have inspired Scrivener's key features: its support of nonlinear and distraction-free composing processes, the ability to view one's document in several modes, and the means to manage research and writing documents in one file. Indeed, it was the failure of standard software like Microsoft Word to support nonlinear composing processes and document management strategies that led Keith Blount to develop Scrivener.

Quite another idea is pursued by Thesis Writer (Rapp & Kruse, 2016, 2020; Rapp et al., 2015), a writing platform tailored to dissertation writing. At any level, dissertations and theses are writing situations or writing assignments with similar needs and demands. Thesis Writer uses an editor that is less elaborate than MS Word but adds specific functions such as tutorials, a proposal wizard, outline structures, sample phrases, corpus search tools, a project management tool, and more. At current (2023), Thesis Writer is available at the authors' Swiss university only.

2.4 *Distraction-Free Tools*

As alternatives to full-featured word processor systems, so-called "distraction-free" writing apps were created. Two examples of this type of software, which has gained some prominence in recent years, are iA Writer and Ulysses. Rather than adding more features or specializing in particular domains of writing, distraction-free tools emphasize ease of use. As such, they are the antithesis of the GUI and WYSIWYG models of writing embodied by MS Word.

Distraction-free word processors downplay the visual appearance of text on the screen and the possibilities for changing that appearance in favour of a much-simplified presentation and interface. They reject the logic of the printed page and conventional typography. Instead, they use the computer screen as a writing space “abstracted” from specific dimensions and materialities of paper and particular typographic realizations of text. Following the terminology of Bolter and Grusin (1999), distraction-free writing tools seek to replace the (simulated) immediacy of the printed page and the hypermediacy of the modern GUI with the immediacy of disembodied, “purely” digital writing.

Consequently, the options for formatting text are few and, typically—except for italics and boldface—restricted to semantic styles (i.e., section headings, block quotes, lists, etc.). Changing a text’s physical aspects (e.g., font and size, indentation of individual paragraphs, and exact line-spacing) is usually impossible. Text is presented and processed as a construct of logical pieces rather than a primarily visual phenomenon laid out on the page.

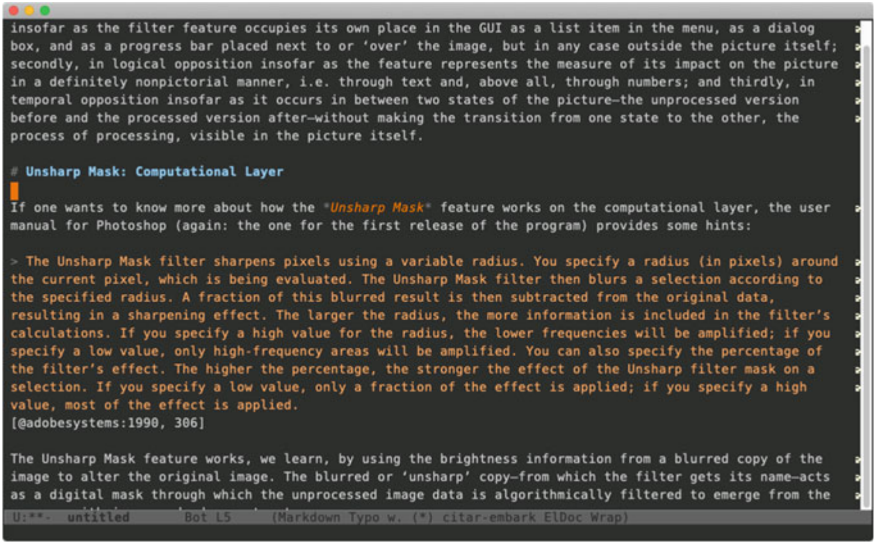
As the name suggests, distraction-free tools promise to divert an author’s attention as little as possible from the actual process of writing and the written text. To this effect, some programmes employ special features. iA Writer’s “focus mode”, for instance, keeps the sentence under the cursor always centred on the screen and dims all other visible text.

Of course, most regular word processors allow their interfaces to be customized by the user and thus can be made less intrusive or cluttered. Many programmes (MS Word among them) also offer a “distraction-free” modus. And some applications (e.g., Scrivener) could even be considered distraction-free out of the box as their graphical interface is relatively minimal.

However, actual distraction-free writing tools like iA Writer are built on the philosophy of decreasing functionality—and, by consequence, minimizing distraction—by giving authors only a restricted set of word processing options. Writing happens only at the level of entering and editing text in ‘plain text’ characters. This is achieved by replacing WYSIWYG processing capabilities with lightweight markup languages like Markdown, which is discussed in the following subsection.

2.5 Text Editors and Markup Languages

At the opposite end of the scale to graphical word processing with WYSIWYG is a return to the beginnings of digital writing. Using a markup language and a processor like Markdown (see Fig. 2 for an example), one can restrict oneself to a simple editor like Windows Notepad. Documents can be written and formatted as ‘plain-text’ files from which the processor generates ‘output’ files for printing or distribution, typically



```
insofar as the filter feature occupies its own place in the GUI as a list item in the menu, as a dialog
box, and as a progress bar placed next to or ‘over’ the image, but in any case outside the picture itself;
secondly, in logical opposition insofar as the feature represents the measure of its impact on the picture
in a definitely nonpictorial manner, i.e. through text and, above all, through numbers; and thirdly, in
temporal opposition insofar as it occurs in between two states of the picture–the unprocessed version
before and the processed version after–without making the transition from one state to the other, the
process of processing, visible in the picture itself.

# Unsharp Mask: Computational Layer

If one wants to know more about how the *Unsharp Mask* feature works on the computational layer, the user
manual for Photoshop (again: the one for the first release of the program) provides some hints:

> The Unsharp Mask filter sharpens pixels using a variable radius. You specify a radius (in pixels) around
the current pixel, which is being evaluated. The Unsharp Mask filter then blurs a selection according to
the specified radius. A fraction of this blurred result is then subtracted from the original data,
resulting in a sharpening effect. The larger the radius, the more information is included in the filter’s
calculations. If you specify a high value for the radius, the lower frequencies will be amplified; if you
specify a low value, only high-frequency areas will be amplified. You can also specify the percentage of
the filter’s effect. The higher the percentage, the stronger the effect of the Unsharp filter mask on a
selection. If you specify a low value, only a fraction of the effect is applied; if you specify a high
value, most of the effect is applied.
[@adobesystems:1990, 306]

The Unsharp Mask feature works, we learn, by using the brightness information from a blurred copy of the
image to alter the original image. The blurred or ‘unsharp’ copy–from which the filter gets its name–acts
as a digital mask through which the unprocessed image data is algorithmically filtered to emerge from the
```

U:**- untitled   Bot L5   (Markdown Typo w. (*) citar-embark ElDoc Wrap)

Fig. 2 Example of text formatted with Markdown language in Emacs editor

as PDFs. In theory, any WYSIWYG word processor application can be used like a simple text editor to write documents in markup languages.

Since RUNOFF, the first implementation of a digital markup language in the early 1960s, text markup technology has developed considerably. Today, there are almost as many different languages and processors for markup as there are editors. However, the idea has remained unchanged: Formatting and structuring text is achieved not by manipulating it on the visual level of WYSIWYG but by ‘marking it up’ with control characters and words that are constructed from ordinary characters and signs. In Markdown, for example, text can be *enclosed in asterisks* to emphasize it, or a # sign can be added to a line of text to denote it as a section heading. Only in the resulting output file produced by the markup processor will the corresponding text be italicized (for emphasis) or rendered in a larger font and possibly with automatic numbering (for a section heading). In addition to basic text formatting and structuring, modern markup languages also support procedures and practices necessary for academic writing, such as the handling of notes, tables, and figures, automatic citation, and reference lists.

The separation of content and style enforced by markup languages helps authors concentrate on the text without having to deal with matters of appearance and graphic design while writing and editing. Therefore, as in distraction-free writing tools, markup should be as minimal and unobtrusive as possible. More complex markup, such as in LaTeX, a user-friendly derivative of the typesetting language TeX, can easily get in the way of writing and make text files look cluttered and more like computer code than ordinary prose. This is the reason why distraction-free writing tools rely almost exclusively on the lightweight solution Markdown or one of its many variants. Other popular languages include BBCode, Textile, and reStructuredText. (La)TeX, arguably the most versatile and powerful digital markup system, is used primarily in the sciences to produce documents with complex mathematical expressions and graphics.

Once early markup languages like roff and (La)TeX had been relegated to special domains and niche audiences by the success of WYSIWYG word processors in the 1980s, the advent of the World Wide Web with its HyperText Markup Language (HTML), along with blogging in the 1990s, led to a flowering of new languages and processors. It is no coincidence that Markdown, probably the most popular markup language today, was explicitly developed "to make writing simple web pages, and especially weblog entries, as easy as writing an email" (Swartz, 2004). Yet, the separation of content from style in Markdown makes it possible to produce output in multiple document formats from one and the same 'plain text' source file. With a processor like pandoc, text written in Markdown (or a comparable markup language) can be converted not only to HTML, but also to EPUB, PDF, RTF, or even MS Word docx.

A not insignificant benefit of using a markup language to write and format text is that authors are free to choose whatever editor they consider best. Even the most rudimentary text editor application will do. More powerful programmes such as Notepad++, Sublime Text, Atom, vi(m), and Emacs offer advanced text editing capabilities and can often be customized to a user's needs and preferences.

2.6 Desktop Publishing

A final alternative to MS Word must be mentioned briefly: desktop publishing (DTP) programmes. Although not designed for writing and editing text, applications like Adobe InDesign nevertheless play a pivotal role in the digital production of printed text. DTP programmes are used to generate digital files for professional print publications.

While there is considerable overlap between the functionality of digital word processing and DTP, DTP applications are more robust in handling page layouts and offer more typographical control. And although editing text in DTP programmes is possible, this is not what the programmes are meant for. Typically, documents are written and edited by authors with standard word processing software first, then imported into DTP by the publisher and prepared for printing by typographers and graphic designers. As an author writing a text on your computer (even if the text is to be published professionally later on), you will probably never use a DTP programme yourself. Of course, word processors also do page layouts and typography. And some publishers will even demand camera-ready PDFs generated from the original MS Word manuscript (or comparable word processing programmes). Additionally, some markup languages and processors like DocBook and (La)TeX can produce high-quality output files suitable for professional printing.

Adobe InDesign has been the de facto standard for DTP since the early 2000s, taking over from Quark XPress. The commercial software Affinity Publisher and the free open-source programme Scribus are noteworthy competitors.

3 Conclusions

The monopoly position of MS Word as the dominating writing software has been dissolved mainly since Google Docs moved word processing into the cloud and forced all competitors to follow. Google Docs has been the gamechanger. Therefore, it is no longer the writing software itself at the centre, but the platform into which it is integrated. The new platforms host far more functionalities than the former Office solutions to act as working environments for companies or institutions. They are extendible, it seems, ad libitum. The creation of mega-platforms bundling a whole range of office software appears to be the current developmental trend. It is unclear whether this downgrades writing, but it certainly changes its position in social contexts and organizations.

Writing in word processors has lost some of its exclusiveness since writing has become part of almost all communication and learning media (learning platforms, blogs, email, chat, social media, calendars, mobile phones, etc.). The question arises as to what the role of the word processor in this orchestra might be or, to use another metaphor, how the role of word processors in a literate landscape hosting such a media ensemble should be specified.

The professional contexts of word processing have to be monitored more closely as the interconnectedness with domain-specific communication and design media is pushing writing into new directions. This generates activities for which the term “text work” (Bazerman, 2018) might be more apt than simply “writing”. Also, new working spaces are being created that “invade” word processors, reducing their spatiality to a subsection of, for instance, MS Teams.

In addition to the greater variability of writing tools, the ability of word processors (and most tools contained in the Office packages) to enable collaboration reconnects writing and communication in new ways. Although synchronous collaboration seems widely accepted and is used routinely, there is little reflection on the changes this imposes on writing (see Castelló et al., “Synchronous and Asynchronous Collaborative Writing”, for a deeper analysis).

With the arrival of alternatives to MS Word, a discussion has started about the most useful and most appropriate technology for writing. It seems that the one-fits-all era is over and that writing will have to be (or will be able to be) selective. Writers will soon be faced with the challenge of choosing the right tool for the right task. We have discussed a range of alternatives to MS Word that occupy different niches and serve specific writers’ needs. Bray (2013) showed in her study about Scrivener how writing support for nonlinear composition can be connected with better options for outlining and synthesizing materials. In academic writing, we have very little knowledge about how students or researchers use their word processors and how linear or nonlinear their writing is (see Kruse & Rapp, "Beyond MS Word: Alternatives and Developments").

Other questions to address in this context include: Who supports students in their choice of writing tools? And can we assume that they can find the best tools by themselves? The more specialized the writing tools and the more numerous the solutions on offer, the less likely it is that students will make appropriate choices. The same goes for decisions such as whether to use online or offline processors and whether a large platform is preferable to self-organisation of the writing software.

The future of writing is hard to predict. Still, for the writing sciences, it will be important to understand and keep close track of developments which are too important to leave up to computer scientists and programmer communities. As we have seen, writing software will increasingly assume the role of a co-author, not only by supporting and guiding writers but also by co-producing and co-evaluating the texts that are written.

4 List of Tools

Name (alphabetically)	Category	URL
Adobe InDesign	Desktop publishing	https://www.adobe.com/products/indesign.html
Affinity Publisher	Desktop publishing	https://affinity.serif.com/
Apache OpenOffice	Parallel offers to MS Word	https://www.openoffice.org/
Atom	Text editors and markup languages	https://atom.io/
BBCode	Text editors and markup languages	https://www.phpbb.com/community/help/bbcode
DocBook	Text editors and markup languages	https://docbook.org/
Emacs	Text editors and markup languages	https://www.gnu.org/software/emacs/ https://emacsdocs.org/
Google Docs	Writing in the cloud	https://docs.google.com/
iA Writer	Distraction-free Tools	https://ia.net/
iWork	Writing in the cloud	https://www.apple.com/iwork/
LaTeX	Text editors and markup languages	https://www.latex-project.org/
LibreOffice	Parallel offers to MS Word	https://www.libreoffice.org/
Mellel	Outplaying MS Word	https://www.mellel.com/
Microsoft 365	Writing in the cloud	https://www.microsoft.com/en/microsoft-365?rtc=1
Nisus Writer Pro	Outplaying MS Word	https://www.nisus.com/pro/
Notepad++	Text editors and markup languages	https://notepad-plus-plus.org/
pandoc	Text editors and markup languages	https://pandoc.org/
Quark XPress	Desktop publishing	https://www.quark.com/

(continued)

(continued)

Name (alphabetically)	Category	URL
reStructuredText	Text editors and markup languages	https://docutils.sourceforge.io/rst.html
Scribus	Desktop publishing	https://www.scribus.net/
Scrivener	Outplaying MS Word	https://www.literatureandlatte.com/scrivener/overview
Storyist	Outplaying MS Word	https://storyist.com/
Textile	Text editors and markup languages	https://textile-lang.com/
Ulysses	Distraction-free Tools	https://ulysses.app/
Vi(m)	Text editors and markup languages	https://www.vim.org/

Acknowledgements The research of Christian Rapp was financially supported by the ZHAW DIZH Fellowship Call 2020.

References

Bazerman, C. (2018). What do humans do best? Developing communicative humans in the changing socio-cyborgian landscape. In S. Logan & W. Slater (Eds.), *Perspectives on academic and professional writing in an age of accountability* (pp. 187–203). Southern Illinois University Press.

Bergin, T. J. (2006). The proliferation and consolidation of word processing software: 1985–1995. *IEEE Annals of the History of Computing, 28*(4), 48–63. https://doi.org/10.1109/MAHC.2006.77

Bolter, J. D., & Grusin, R. (1999). *Remediation.* MIT Press.

Bray, N. (2013). Writing with Scrivener: A hopeful tale of disappearing tools, flatulence, and word processing redemption. *Computers and Composition, 30*(3), 197–210. https://doi.org/10.1016/j.compcom.2013.07.002

Burkeman, O. (2006, November 4), Writely, Sam Schillace, *The Guardian*, https://www.theguardian.com/technology/2006/nov/04/news.weekendmagazine7

Google Docs Editors. (2022, July 27). In *Wikipedia.* https://en.wikipedia.org/wiki/Google_Docs_Editors

Googlepress. (2006, October 11). Google announces google docs & spreadsheets. *News from Google*. https://googlepress.blogspot.com/2006/10/google-announces-google-docs_11.html

Google Wave. (2009, May 27). In *Wikipedia.* https://de.wikipedia.org/wiki/Google_Wave

Hamburger, E. (2013, July 3). Google Docs began as a hacked together experiment, says creator. Interview with Sam Schillace. *The Verge.* www.theverge.com/2013/7/3/4484000/sam-schillace-interview-google-docs-creator-box

Ingraham, N. (2021, October 11). 15 years of Google Docs, and where the next 15 might take us. *Engadget*. https://www.engadget.com/google-docs-turns-15-190050626.html

Johannsen, J., & Sun, Y. (2017). An intelligent and semantics-aware distraction-free writing system. In *Proceedings IEEE 11th International Conference on Semantic Computing (ICSC)* (pp. 465–468). IEEE. https://ieeexplore.ieee.org/abstract/document/7889580

McHugh-Johnson, M. (2021, October 11). 15 milestones, moments and more for Google Docs' 15th birthday. *The Keyword*. https://blog.google/products/docs/happy-15-years-google-docs/

Möhring, C. (2020, June 22). OpenOffice & LibreOffice im Vergleich. *Heise online*. https://www.heise.de/tipps-tricks/OpenOffice-LibreOffice-im-Vergleich-4156436.html

Rapp, C., & Kruse, O. (2016). Thesis Writer (TW)—Tapping scale effects in academic writing instruction (demonstration). In *Proceedings of the Third (2016) ACM Conference on Learning @ Scale* (pp.325–326). ACM. https://doi.org/10.1145/2876034.2893400

Rapp, C., & Kruse, O. (2020). Thesis writer 2.0: A system supporting academic writing, its instruction and supervision. In C. Müller Werder & J. Erlemann (Eds.), *Seamless Learning: Lebenslanges, durchgängiges Lernen ermöglichen [Seamless learning—Enabling lifelong continuous learning]* (pp. 235–240). Waxmann. https://www.waxmann.com/waxmann-buecher/?tx_p2waxmann_pi2%5bbuchnr%5d=4244&tx_p2waxmann_pi2%5baction%5d=show#:~:text=Das%20Konzept%20des%20Seamless%20Learning,lebenslanges%2C%20nahtloses%20Lernen%20zu%20erm%C3%B6glichen

Rapp, C., Kruse, O., Erlemann, J., & Ott, J. (2015). Thesis Writer: A system for supporting academic writing. In *Proceedings of the 2015 ACM International Conference on Computer-Supported Cooperative Work and Social Computing* (pp. 57–60). ACM. https://doi.org/10.1145/2685553.2702687

Sharples, M., & Pemberton, L. (1990). Starting from the writer: Guidelines for the design of user centred document processors. *Computer Assisted Language Learning (CALL), 2*(1), 37–57. https://doi.org/10.1080/0958822900020104

Stanton, A. L. (2021). Broken is word. In T. Mullaney, B. Peters, M. Hicks, & K. Philip (Eds.), *Your computer is on fire* (pp. 213–230). MIT Press.

StarOffice. (1998). In *Wikipedia*. https://de.wikipedia.org/wiki/StarOffice

Swartz, A. (2004, March 19). *Markdown*. http://www.aaronsw.com/weblog/001189

Vailshery, L. S. (2022). Office productivity software global market share 2021. *Statista*. https://www.statista.com/statistics/983299/worldwide-market-share-of-office-productivity-software/

Wilson, M. (2012, April 16). Why Microsoft Word really sucks: it was invented in a paper-powered world. *Fastcompany*. https://www.fastcompany.com/1669530/why-microsoft-word-really-sucks-it-was-invented-in-a-paper-powered-world

Christian Rapp heads the Educational Technology Team at the Center for Innovative Teaching and Learning, School of Management and Law, Zurich University of Applied Sciences (ZHAW). He has coordinated various international R&D projects, among others "Seamless Writing: Expanding Technologies to Support Thesis Writing" (EU-Interreg). He is a fellow of the Digitalization Initiative of the Zurich Higher Education Institutions (DIZH) and a board member of the European Association for the Teaching of Academic Writing (EATAW). Together with Otto Kruse, they established "Thesis Writer", an online platform that supports students with writing their theses and dissertations (www.thesiswriter.eu).

Till A. Heilmann is a researcher at the Department of Media Studies at Ruhr University Bochum. He studied German, media studies, and history. Research associate at the University of Basel (2003–2014), University of Siegen (2014–2015) and University of Bonn (2015–2021); doctorate for a thesis on computers as writing machines (2008); visiting scholar at the University of Siegen (2011); Fellow-in-Residence at the Obermann Center for Advanced Studies, University of Iowa (2012); acting professor of Digital Media and Methods at the University of Siegen (2020–2021).

Otto Kruse is a retired professor of the Department of Applied Linguistics at the Zurich University of Applied Sciences in Winterthur, Switzerland. He was the head of his department's writing center. A psychologist by education, he worked in clinical psychology, social work and applied linguistics. His expertise in the field of writing is connected to the teaching of writing, intercultural aspects of writing, critical thinking, and to the development of writing technologies. Together with Christian Rapp, he created “Thesis Writer”, a writing platform supporting students with their dissertations.