XLingPaper's use of TEX Technologies

H. Andrew Black, Hugh J. Paterson III

Abstract

We discuss the use of TEX technologies by XLingPaper, an authoring tool for producing academically oriented publications with features required for linguistic publishing. We present the TEX modules used and the rationale for the history of XLingPaper development.

1 Introduction

Within the publishing industry there are several notable products for producing complex documents in beautiful formats. TEX [20] [21] is one of the well known publishing technologies used to meet these needs. Since 2000, XML-based technologies such as XSL-FO[1] or the TEXML[2] project [25] have been used to integrate content and compose complex documents such as textbooks and maintenance manuals. Requirements for composing these large, interlinked documents birthed the development of tools like XMLmind[3] and Xpublisher.[4] These tools can be used to compose content within predefined XML structures. XLingPaper, as discussed in [6] [7] [8], seeks to provide a constrained environment in which authors of complex works dealing with language descriptions and linguistic analyses can focus on content structure independently from the styling requirements of documents. In this way the underlying design principle of XLingPaper maximizes the SGML design practice of separating content from presentation. With XLingPaper, authors can keep content structure independent from page layout information and thereby provide maximal transfer-ability between publishing styles. The software does this while providing authors a clear structured interface for authoring content.

The XLingPaper software has a growing number of users who have successfully typeset complex documents including:

- master theses [44] [22] [30],
- doctoral dissertations [14] [34],
- textbooks [27],
- linguistic grammars [10],
- books [1] [35],
- journal articles [9], and
- bilingual software documentation [2] [3].

[1] www.w3.org/TR/xsl11
[2] getfo.org/texml
[3] www.xmlmind.com/xmleditor
[4] www.xpublisher.com/products

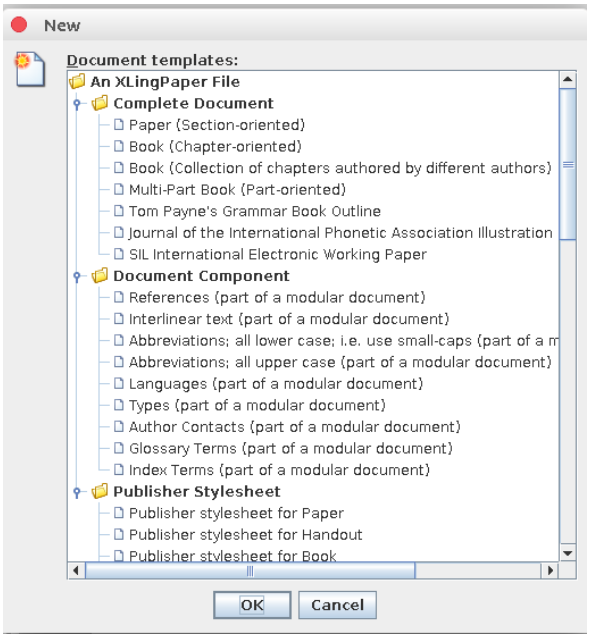

Figure 1: XLingPaper predefined document types via DTD

2 What is XLingPaper?

XLingpaper[5] is a plug-in to the XMLmind XML Editor. XLingPaper benefits from the XMLmind XML Editor's Java-based implementation which allows it to be used on Mac OS X, Windows, and Linux. Via a DTD, XLingPaper defines several document classes (articles, books, chapters, etc., as illustrated in Figure 1), in each case providing document layout sections (paragraphs, examples, endnotes, etc.). By working within the user-interface of the XMLmind XML Editor, as shown in Figure 2, formatting errors are reduced because users are constrained on where in the document flow they can introduce block and line level document elements. That is, first, authors cannot input page layout instructions directly into the document and second, the introduction of layout sections within the document flow is constrained via the DTD.

XLingPaper is designed to reduce friction in the process of writing, reading, composing, and publishing linguistic papers, grammars, and books by removing common time-sinks related to inconsistent formatting (especially citations, references, and numbered element like examples). A full list of benefits to all parties in the publishing work flow is available [8]. For many, the PDF format is the quintessential file format for final distribution of publishing outputs. XLingPaper supports PDF production but

[5] software.sil.org/xlingpaper

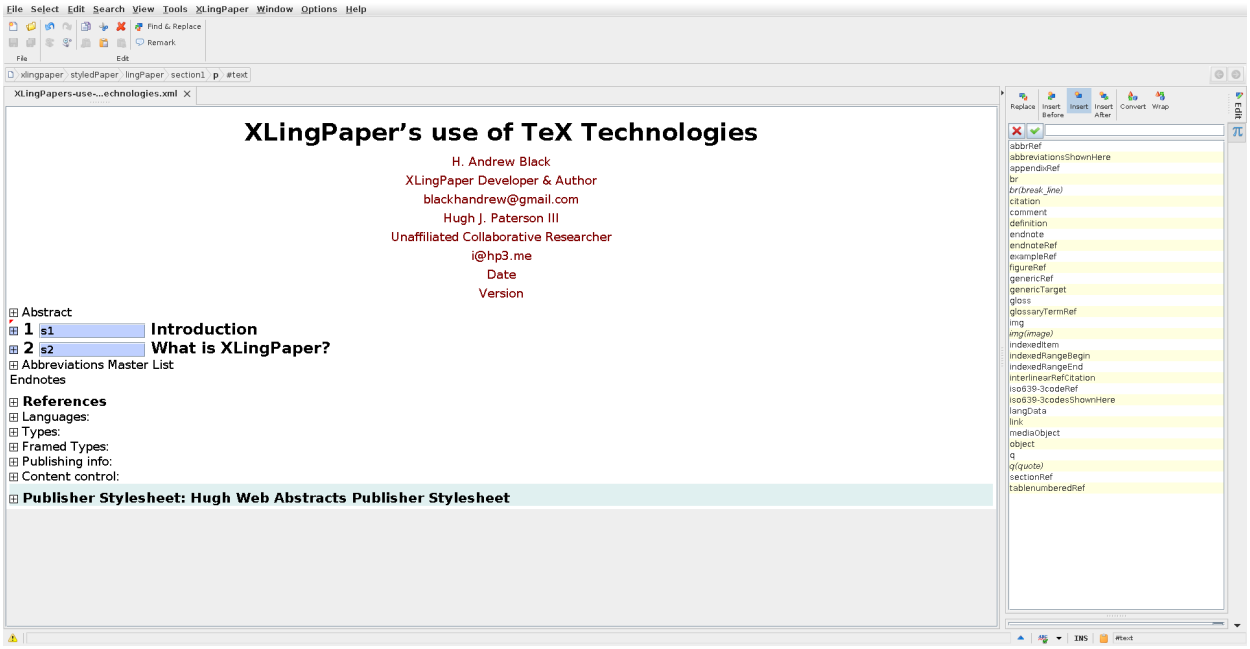

Figure 2: XLingPaper's user interface. Left side: document content editing. Right side: block and line level units available for use at the cursor location.

as illustrated in Figure 3, XLingPaper can produce documents with at least five outputs, all from the same source document:

- PDF (version 1.5),
- Web pages (HTML 4),
- Microsoft Word (.doc),
- Open Office Writer Document (.odt), and
- ePUB.

XLingPaper automatically numbers tables, examples, figures, and sections. It keeps track of internal references to these entities along with citation references, abbreviations, and gloss abbreviations. This keeps numbering and reference links dependable and automated. XLingPaper also automatically generates indexes, a table or list of abbreviations used, and a section for references cited (using a custom references implementation).

Unlike most editing programs which are based on either the WYSIWYG paradigm or as text editors used to code or produce Markdown, XLingPaper (via the XMLmind XML Editor) is a structured editor. Rather than visually structuring the document to look the way it is to be formatted, the author "marks up" the items in the document according to their kind. One of the many benefits that using a DTD provides is that there is a "grammar" of what a well-formed linguistic document looks like. This makes moving, replacing, switching, or reordering sections, chapters, tables, figures, and examples less error prone because it prevents users from inadvertently creating ill-formed documents. The following sections of this paper discuss the TEX technologies used by XLingPaper.

3 XLingPaper and TEX

Linguistic publishing has unique requirements when compared to general publishing. The following sections provide more detail on the linguistic publishing context, design requirements and LATEX packages used by XLingPaper.

3.1 TEX and Linguistic document production

TEX has long been embraced by linguists. Peter [32] writes of a personal communication with Don Knuth where Knuth suggests that linguists were some of the earliest adopters outside of mathematicians. Thiele [39] in an interview given in 2007 states that she was typesetting linguistic journals via TEX in 1983—a date prior to the release of Knuth's book on using the TEX typesetting system [20]. Thiele [38] gives an early overview of TEX use in linguistics with mention of significant repositories outside of CTAN. A slightly more recent (2004) update by Peter [32] provides some additional tips and tools for typesetting common information structures in linguistic publishing. The TEX community has produced many packages which have shaped the visual face of publishing

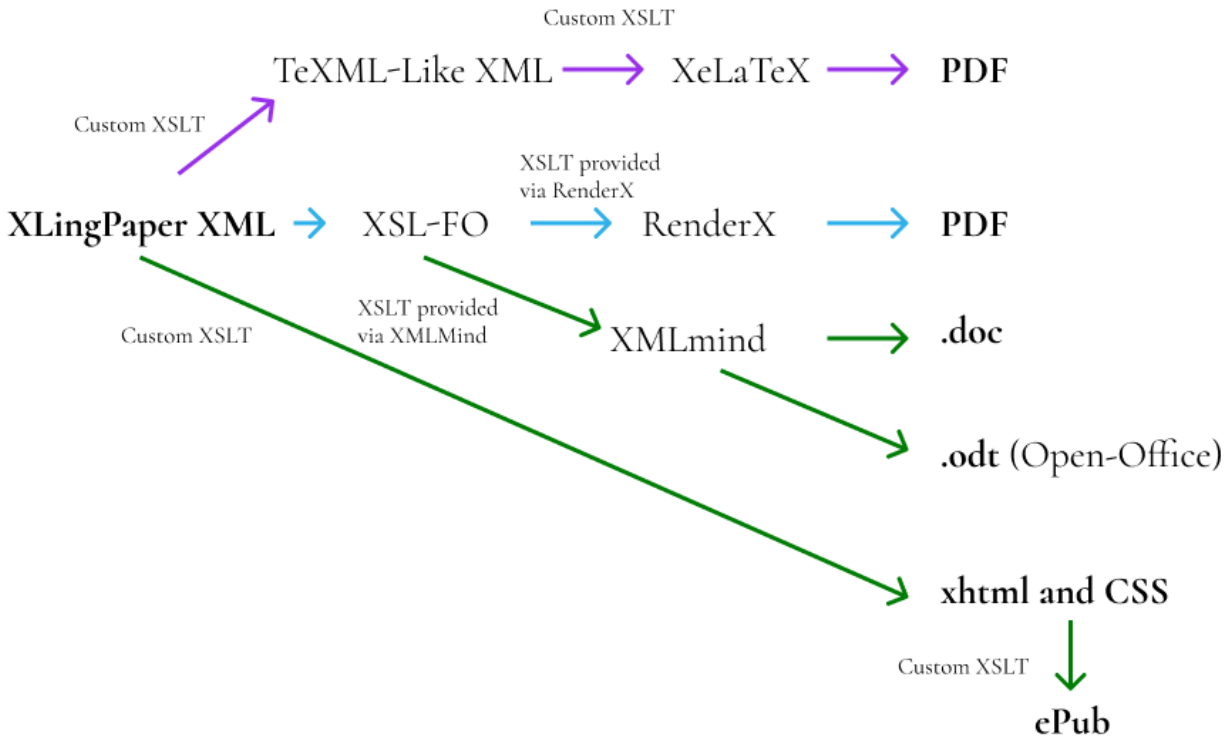

Figure 3: XLingPaper's data processing pipeline to multiple formats

in linguistics, including `tipa`[6] by Rei [36] which provided access to an excellent typeface for phonetic transcriptions and `pst-asr`[7] by Frampton [15] for autosegmental representations. Some packages used in linguistic publishing are not exclusive to linguistics. For example, Donnelly [13] describes how to use various packages to draw phonetic pitch traces using TEX. Peter [32] and Thiele [38] list out and review (through 2004) various packages across several areas of lingustics, including several packages used to draw syntax trees such as `qtrees`[8] and `forest`[9] to specialized packages for presenting examples and interlinear glossed texts such as `expex`.[10] In addition to packages which provide specific functionality, there are packages which are essentially collections of macros such as `covington`[11] and `gb4e`[12] which serve a variety of page layout functions targeted at publishing in linguistic topics.

The CTAN repository currently lists fifty-four different TEX packages for linguistic typesetting,[13] though some of these packages also include capabilities targeted as multi-lingual or multi-script publications or are specific style sheet implementations for publications at linguistic programs at institutions of higher education (there may be more packages which are not tagged but should be). Several of the packages tagged "linguistic" pre-date Unicode [40] but still see significant use. Sometimes it is the case that secondary packages are developed in an attempt to "fix" publishing outputs in different ways to bring Unicode features along with the features of the original package. For example, `tipa` is not Unicode compatible, but the packages `unitipa`[14] and `tipauni`[15] seek to address different implications of not publishing with Unicode while giving access to the beautiful typeface of `tipa`. Understanding the long history of publishing and the interdependency that packages have (including the order of loading packages) presents additional barriers of adoption to new TEX users.

[6] `ctan.org/pkg/tipa`
[7] `ctan.org/pkg/pst-asr`
[8] `ctan.org/pkg/qtrees`
[9] `ctan.org/pkg/forest`
[10] `ctan.org/pkg/expex`
[11] `ctan.org/pkg/covington`
[12] `ctan.org/pkg/gb4e`
[13] `ctan.org/topic/linguistic`
[14] `ctan.org/pkg/unitipa`
[15] `ctan.org/pkg/tipauni`

TEX barriers of adoption are important to the XLingPaper discussion for two reasons. First, it exemplifies some of the complexities that XLingPaper seeks to simplify as it presents authors not just a visual environment for document composition, but also a cohesive output solution. Second, it speaks to the software design process in finding the minimal viable product. That is, how much of a software stack is needed to make a useable software product for lingustic publishing? The TEX community is divided on this. While the diagrams in linguistic books and journals since the 1980's exemplify many beautiful, sharp, crisp, illustrations created directly in TEX, many trainers of TEX tools,[16] but not all,[17] have steered authors towards a more generic set of packages which do not include specific diagram creating macros. Rather, they suggest that authors use secondary illustration tools to generate illustrations and then include them as vector PDFs or images. In fact this second method is the document production path that the XLingPaper philosophy follows. That is, XLingPaper reduces the complexity of the typesetting task for authors by requiring complex visualizations to be produced via graphical tools. We have found tools like Figma[18] and Inkscape[19] very helpful in the graphic production task. The XLingPaper product seeks to lower barriers of entry, only produce valid documents, and keep the code base to a minimum.

As mentioned in the discussion of `tipa`, Linguistic documents have not always been typeset with Unicode. Unicode was introduced in 1991 and by the early 2000's Unicode along with document and data storage in XML formats were being heralded in academic linguistics as a best practice in order to avoid vendor lock-in, increase interoperability across use cases, and to separate data life-cycles from encoding or software life-cycles [4] [5] [41]. Due to the heavy reliance on Unicode by today's practitioners of language documentation and linguistic work, XLingPaper specifically uses XELATEX and compatible packages to produce PDF outputs. This brings continuity to the text input process for users across their workflows. It also makes importing and using language or phonetically transcribed examples simpler by removing the need to use macros to derive characters.

[16] Among others, see the Linguistics Dissertation guide for the University of Hawai'i at Mānoa [18], University of Pennsylvania [12], and Language Science Press Guidelines [29].

[17] For counter examples see [24] [37] [17] and [33].

[18] `www.figma.com`

[19] `inkscape.org`

3.2 Design desiderata for XLingPaper outputs via TEX

There have been three goals which have influenced the development of XLingPaper:

- separation of content and style,
- the software needed to be accessible (license and size), and
- beautiful multi-format outputs.

Deciding how TEX technologies fit within the project has been a journey. XLingPaper development started in 2001. In 2006, it added XSL-FO for PDF production. Prior to 2009, XLingPaper used RenderX[20] to produce PDF documents. However, in 2009 plans were made to add XELATEX-based output to XLingPaper because, while there was a free version of RenderX, the output contained a watermark. By implementing the ability to export to PDF via XELATEX, watermarks could be avoided all together. The XELATEX method of PDF production is now the default method to produce PDF documents, although the RenderX method is still possible.

Maintaining a separation of content and style in the XLingPaper envionrment was a key design requirement. When the XELATEX method of PDF production was introduced, XLingPaper already had a way to format output per a user-created publisher style sheet—allowing great flexibility due to the separation of style and content. Using TEX technologies meant the developer (Andrew Black) needed to be able to map from an XLingPaper publisher style sheet to XELATEX. It was known that LATEX was the ideal TEX implementation to target. However, pure LATEX came with predefined output formatting for front matter, chapters, sections, back matter, etc. Pure LATEX, then, would not allow direct control of formatting of all of these per an XLingPaper user-defined publisher style sheet. This required overriding these standard features of LATEX with a custom implementation of the TEX commands needed to control formatting. XLingPaper takes a custom approach in implementing flexibility here. The programmerrecently discovered the `memoir`[21] [42] [43] package, which accomplishes many of the same tasks and could be considered to replace some of the custom code if it were shown to be easy to implement and that the size of the total XLingPaper code base would be reduced.

The distributability of the software was also seen as a design requirement. Distributability is understood to have two components: license and accessability, including size.

[20] `www.renderx.com`

[21] `ctan.org/pkg/memoir`

From the outset, XLingPaper was designed to be costless to the end user. It is licensed under the MIT license, and its code is currently available on Github.[22] The XMLmind XML Editor had a Personal Use License that met this requirement for the vast majority of the target audience of XLingPaper. The few XLingPaper users who did not meet the terms of that license most likely would be able to afford to purchase (or have their organization purchase) a professional version of the XMLmind XML Editor. The actual XLingPaper plug-in has always been free.

The software size of XLingPaper is a major design influencer. Many of the expected users of XLingPaper live and work in places around the world where Internet connections are characterized by high costs, low bandwidth capacity, and general unavailability. Therefore, the download required to install XLingPaper needed to be as small as possible. On Windows the current full XLingPaper installer is 146MB, and the XMLmind XML Editor installer is 116MB. Both are required. This stands in contrast to the TeXLive 2010 installer which is about 1.2GB download and 2.38GB uncompressed. The size constraint impacts XLingPaper because its distribution must be independent of larger mainstream TEX distribution solutions such as TEXLive which have a large footprint. Therefore the developer determined which LATEX packages and binaries were needed and created a custom installation package which met the required specifications.

XLingPaper currently uses the following LATEX packages (in alphabetical order):

`attachfile2`	`lineno`
`booktabs`	`longtable`
`calc`	`lscape`
`color`	`mdframed`
`colortbl`	`multirow`
`etoolbox`	`normalem`
`fancyhdr`	`polyglossia`
`fontspec`	`setspace`
`footmisc`	`tabularx`
`hyperref`	`xltxtra`

The twenty LATEX packages that are part of the custom XLingPaper distribution are still rather large (29MB) for someone for whom Internet bandwidth is an expensive and inconsistent commodity. To reduce bandwidth requirements two assumptions were made which have more or less proven to obtain. The first assumption that the developer made was that the twenty packages and binaries would not need to change over time; in contrast, the second assumption was that XLingPaper would acquire new features and need bug fixes. These assumptions resulted in an architecture where page layout information expressed in XML is translated via custom TEX commands to either TEX directly or to commands understood by LATEX packages distributed with XLingPaper. This abstraction layer was then executed when the XELATEX file was processed. This middle layer has granted XLingPaper flexibility in adding new code and capabilities while keeping the "heavy" LATEX packages stable. The net result is a "heavy" first install package, but light-weight upgrade packages (6.21MB). In the thirteen year history of development, there have been a few occasions where upgrades have required the download of new "heavy" packages. One such case was when the ability to use framed units was added. These elements depend on the `mdframed`[23] package [11]. The architecture separating stable packages from custom code, however, has generally worked out well and kept update sizes low.

3.3 PDF production

There are two known pathways for converting XML content into PDFs. The first is via XSL-FO, and the second is via TEXML which converts XML content to TEX formatted documents. Given certain limitations in both XSL-FO and TEXML, XLingPaper uses a custom (or third) method. When an author instructs XLingPaper to produce PDF output via XELATEX, XLingPaper produces a TEXML-like XML file. This is then converted into a LATEX formatted document via a set of XSLT transforms and processed via XELATEX to produce the PDF. Figure 3 contains a diagram of the data handling process.

3.4 TEXML

TEXML was discovered in the process of planning for the transition of the default PDF renderer from RenderX (an XSL-FO processor) to XELATEX. Initial analysis conducted in 2009 understood TEXML to have two infelicities for use-cases required in linguistic publishing with XLingPaper:

1. TEXML has Python as a dependency and the developer did not want to require XLingPaper users to install a version of Python specifically for TEXML. This is especially the case since that version of Python might conflict with other installed versions of Python on their operating systems. Moreover, this approach would make the installation package for XLingPaper much larger due to the inclusion of Python.

[22] `github.com/sillsdev/XLingPap`

[23] `ctan.org/pkg/mdframed`

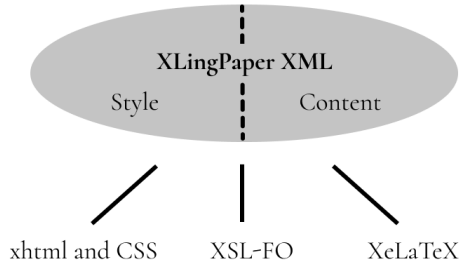

Figure 4: XLingPaper combines style and content information contained in its custom XML and then exports it into three different formats for further processing.

2. XLingPaper users require a high degree of control for white space. The fine grain control of whitespace was not clearly possible via TEXML.

3.5 Control characters

Even with the use of Unicode in the text of documents, there are some features of typesetting with TEX-based implementations which require the use of control characters. Additionally, XML also has control characters. In TEX these include `[`, `]`, `<`, and `>`. When transforming data between XML and TEX, TEX control characters and commands need to be escaped to ensure proper data processing. This has been implemented via Java since Java was already present in the dependency stack due to XMLmind XML Editor requiring it. Additionally, some small methods have been written in Java to provide additional access to features via the graphical user interface.

3.6 Ling-TEX

One might ask, "Why not add more linguistic related TEX packages to the available stack, or use those instead of creating custom code?" The answer has two simple parts: First, in 2009 the linguistic capabilities of TEX packages were different than they are today. Second, XLingPaper is more than a TEX document producer.

Besides TEX, XLingPaper also produces XSL-FO and XHTML/CSS outputs. When new features are considered for typesetting design, all output formats need to be considered.

When the developer began to implement the XELATEX-based output, he discovered the Ling-TEX group[24] which also ran the Ling-TeX mailing list from 1995–2018.[25] Ling-TeX seemed to be the locus of activity in linguistic typesetting even though other web pages discussing linguistics and TEX also existed, e.g., Essex,[26] UPenn.[27] Today, now that the mailing list is no-longer in operation, many of the mailing list participants can be found interacting on the TEX stackexchange.[28]

State-of-the-Art for TEX-based linguistic publishing in 2009, as recommended by the Ling-TEX website, suggested using `covington` and *ling-mac*—the list of macros discussed in [38]. These macros were used to solve similar use cases, among others, to those already implemented by XLingPaper. The more commonly implemented typesetting tasks are outlined in Section 4. Initial analysis of `covington` and other packages revealed limitations in the number of rows an interlinear text example could display, a typesetting capability XLingPaper had already overcome via XSL-FO processing using RenderX. XLingPaper had the following capabilities for typesetting interlinears:

- no limits on the number of lines within an interlinear grouping;
- no limits on the number of free and/or literal lines;
- the ability to include a source reference within the interlinear; and especially
- the ability to tag interlinear items with an ISO 639-3 code for the language used in the interlinear.

At the time the best solution given the state of the TEX packages available was custom TEX scripts, although now similar features may be possible via other packages. See Pellard [31] for discussion of approaches in TEX and his solution `typgloss`.[29] Figures 5–7 contain example output illustrating some of the special capabilities XLingPaper offers.

4 Typesetting tasks XLingPaper users often encounter

Linguistic documents have several formatting needs that other kinds of documents do not. This section discusses some of them.

[24] `web.archive.org/web/20150702123633/http://heim.ifi.uio.no/~dag/ling-tex/`
[25] `ling-tex.ifi.uio.narkive.com`
[26] `www.essex.ac.uk/linguistics/external/clmt/latex4ling`
[27] `www.ling.upenn.edu/advice/latex.html`
[28] `tex.stackexchange.com`
[29] `github.com/tpellard/typgloss`

Una frase cuantificadora puede acompañar al sustantivo (véanse Los Cuantificadores y Los Números Cardinales). Cuando se presenta esta frase, siempre va delante del núcleo de la frase nominal, como en los ejemplos en (2).

(2) a. [tcf-Zila]
Náa majñuu nákhu iduu iya'
náā māhjūù^{n} nákù īdūū ījā?
LOC entre TOT.cuatro ojo.3SG agua
'De entre los cuatro manantiales[5]' [Smajiin:6]

b. [tpl-Tlac]
Gí'doo witsu rakhóó mikhúdú
EST.tener.3SG cinco nariz.3SG (EST).picud@
'Tiene cinco esquinas picudas' [FC:5.1]

El cuantificador puede presentarse en construcciones donde no hay sustantivo expreso, como se explica en Los Cuantificadores. Un ejemplo se incluye aquí.

Figure 5: Interlinear example from [26]. Note the example numbers on the left followed by example groups (a) and (b). Each interlinear then also has a language indicator in square brackets. Customization allows for as many rows per group as is required. Finally, on the right the hyperlinked reference to the source text is indicated.

4.1 Numbered example layouts

Linguistic documents usually have many numbered examples. The prose often refers to examples near the material or to previous examples. XLingPaper automatically keeps track of the numbers. Authors, and publishing style sheets often make use of table-like design layouts including: lists of words along with their glosses (as shown in Figure 6) and interlinear clauses (as shown in Figure 5), with some cases even having headings in portions of the example.

4.2 Automatically wrapping interlinear texts

Many linguists want to include interlinear glossed text in their document. XLingPaper allows these to be wrapped automatically which makes the author's job much easier. Figure 5 shows one such text portion.

4.3 Gloss abbreviations

Linguists standardly use glosses for indicating the meaning of pieces of words (morphemes). XLingPaper allows the author to define a set of abbreviations and their definitions. When producing the output, XLingPaper creates hyperlinks between the abbreviation and its definition.

5 Outputs LaTeX allow that others do not

While XLingPaper has a large array of linguistically-oriented formatting capabilities, there are some that only the XeLaTeX output can produce. This is, of course, due to the formatting power of TeX and XeLaTeX.

5.1 Automatically wrapping interlinears

One of the most popular features of XLingPaper is its ability to automatically wrap long interlinear examples and lines in interlinear texts. It does so by formatting each aligned word in an hbox and then having XeLaTeX put them together in a hanging indent paragraph. This is based on the work of Kew & McConnel 1990 [19].

5.2 Font rendering

XeLaTeX renders fonts extremely well. It can even handle special features requiring Graphite[30] processing. For other outputs, some fonts (such as Charis SIL) may not line up vertically as expected due to them having different ascender and descender values. One has to add custom commands to deal with these. In the case of Graphite, they may not be able to be done at all. The RenderX way of producing PDF cannot handle stacked diacritics, but the XeLaTeX way does it very well.

5.3 Hyphenation for non-English languages

Since we use the `polyglossia` package, one can write an XLingPaper document in a non-English language and XeLaTeX will hyphenate according to that language's hyphenation rules.

[30] `graphite.sil.org`

Bantu D30 canonical infinitive verb pattern is exemplified in the Mbo data in (11):

(11)
a. [ex[ko-sis-o]ex] [ex[[– ¯ ¯]]ex] move forward
b. [ex[kɔ-kɨj-ɑ]ex] [ex[[– ¯ ¯]]ex] act
c. [ex[ko-ɓund-o]ex] [ex[[– ¯ ¯]]ex] break
d. [ex[kɔ-ɓʉt-ɑ]ex] [ex[[– ¯ ¯]]ex] become long
e. [ex[ko-ɓeŋ-o]ex] [ex[[– ¯ ¯]]ex] wink
f. [ex[kɔ-kɛk-ɑ]ex] [ex[[– ¯ ¯]]ex] decorate
g. [ex[ko-sok-o]ex] [ex[[– ¯ ¯]]ex] cackle
h. [ex[kɔ-mvɔɗ-ɑ]ex] [ex[[– ¯ ¯]]ex] suck
i. [ex[kɔ-bɑb-ɑ]ex] [ex[[¯ – ⁄]]ex] carry

Figure 6: List of words as seen in [34]

Rikha[2]

FC:1

Rikha	rígi'	najmąą	náą	yúoo'	rą'khą ká',[3]	rą'khą suan'[4]
flor.de.calabaza	INAN:PROX	IMPF.producirse	LOC	guía.3SG	calabaza.especie	calabaza.especie

khamí	náą	yúoo'	rą'khą' májin'.[5]
y	LOC	guía.3SG	chilacayote

'La flor de calabaza se da en la guía de la calabaza de Castilla, de la "calabaza espina" y del chilacayote.'

FC:2

Rí	rikhoo	rą'khą suan',	nagí'dụụ	namịdị	rí
SBD:INAN	flor.de.calabaza.3SG	calabaza.especie	IMPF.empezar.3SG.FM ±	IMPF.florear	SBD:INAN

gụn'	agóstó.
luna	agosto*

'La flor de la "calabaza espina" empieza a abrir en el mes de agosto.'

FC:3

Mbą'jụ,	mụjmụ'	rị'jiụụ.
(EST).grande:PL	(EST).amarill@	flor.3SG

'Sus flores son grandes y amarillas.'

Figure 7: Wrapped interlinear text as seen in [28].

5.4 Author contact information

XLingPaper allows one to define a set of contact information for authors. Only the XeLaTeX output is able to format them correctly.

5.5 Vertical fill

For title page material, only the XeLaTeX output allows using vertical fill between items. The other outputs require using overt, fixed spacing values.

5.6 Blank page

When one wants a totally blank even-numbered page between a final odd-numbered page and the next odd-numbered page which begins, say, a chapter or appendix, only the XeLaTeX approach is able to do this.

6 Features other outputs have that the LaTeX output does not

XeLaTeX does not allow for custom table cell padding and spacing. Having said that, Andy cannot remember any XLingPaper user ever asking for a way to do this for the XeLaTeX output. It just looks great.

Background color is not available for section titles.

Section 11.17.1.1 "Known limitations of using XeLaTeX" in the XLingPaper user documentation lists known problems.

6.1 Custom TeX commands

XLingPaper has a number of custom commands that enable it to handle various tasks in a way that is consistent with our desired outcomes. The following lists some of them in a schematic way:

Command for	Purpose
Table of contents	Store and retrieve page numbers; format the contents.
Lists	Numbered and bulleted lists with control over indents, etc.
Examples	Example number and example content, where the content can be a line, a list of lines, a set of words, a list of a set of words, interlinear, a list of interlinears, etc.
Indexes	Handle keeping track of XLingPaper's indexing capability, including page numbers.
Interlinears	Handle lines in an interlinear text or example, including dealing with an ISO 639-3 code in an interlinear example.
Block quotes	Handle special cases needed for block quotes.
Table headers	Attempt to calculate a column's width via its contents.

7 Conclusion

While the XLingPaper approach to composing documents via DTD controlled user interface limitations has great value in and of itself, the fact that it can produce great looking output via XeLaTeX makes the learning curve rewarding. We feel that being able to produce PDF via XeLaTeX has made XLingPaper a fantastic tool for linguists.

References

[1] Bartholomew, Doris A, and Louise C Schoenhals. 2019. *Bilingual Dictionaries for Indigenous Languages*. Edited by Thomas L Willett. 2nd ed. Tlalpan, Ciudad de México, México: Instituto Lingüístico de Verano, A. C. [SIL International in Mexico]. `www.sil.org/resources/archives/80401`.

[2] Beadle, Jennie, and Matthew Lee. 2020a. *Paratext 9 Manual – in English*. SIL International. `lingtran.net`.

[3] Beadle, Jennie, and Matthew Lee. 2020b. *Paratext 9 Manual – in French*. SIL International. `outilingua.net`.

[4] Bird, Steven, and Gary Simons. 2002. Seven Dimensions of Portability for Language

Documentation and Description. In ISCA SALTMIL SIG: "Speech and Language Technology for Minority Languages," 23–30. Las Palmas, Canary Islands, Spain: ELRA. www.lrec-conf.org/proceedings/lrec2002/pdf/ws15.pdf#page=29.

[5] Bird, Steven, and Gary Simons. 2003. Seven Dimensions of Portability for Language Documentation and Description. *Language* 79 (3):557–82. www.jstor.org/stable/4489465.

[6] Black, Cheryl A., and H. Andrew Black. 2012. *Grammars for the People, by the People, Made Easier Using PAWS and XlingPaper*. In Electronic Grammaticography, edited by Sebastian Nordoff, 103–28. LD&C Special Publication 4. Honolulu, Hawai'i: University of Hawai'i Press. hdl.handle.net/10125/4532.

[7] Black, H. Andrew. 2009. Writing Linguistic Papers in the Third Wave. *SIL Forum for Language Fieldwork* 2009 (004): 11 pages. www.sil.org/resources/publications/entry/7790.

[8] Black, H Andrew. 2017. *Why Learn to Use XLingPaper*. Dallas, Texas: SIL International. software.sil.org/downloads/r/xlingpaper/resources/documentation/WhyUseXLingPaper.pdf.

[9] Brownie, John. 2013. Adverbs in the Mussau-Emira Verb Phrase. *Language & Linguistics in Melanesia* 31(1):1–11. www.langlxmelanesia.com/issues.

[10] Buck, Marjorie J. 2018. *Gramática del amuzgo Xochistlahuaca, Guerrero*. (Serie de gramáticas de lenguas indígenas de México No16.) Tlalpan, Ciudad de México, México: Instituto Lingüístico de Verano, A.C. [SIL International in Mexico]. www.sil.org/resources/archives/75518.

[11] Daniel, Marco, and Elke Schubert. 2013. *The `mdframed` Package: Auto-split Frame environment* version 1.9b.

[12] Dimitriadis, Alexis. 2016. *TeX/LaTeX Information*. Web page. www.ling.upenn.edu/advice/latex.html.

[13] Donnelly, Kevin. 2013. Representing Linguistic Pitch in XeLaTeX. *TUGboat* 34 (2): 223–27. tug.org/TUGboat/tb34-2/tb107donnelly.pdf.

[14] Ebarb, Kristopher J. 2014. *Tone and variation in Idakho and other Luhya varieties*. University of Indiana Ph.D. dissertation. pqdtopen.proquest.com/doc/1625743679.html?FMT=ABS.

[15] Frampton, John. 2006. *Pst-Asr: Tex Macros for Typesetting Autosegmental Representations*. Version:1.1. CTAN. www.bakoma-tex.com/doc/generic/pst-asr/pst-asr-doc.pdf.

[16] Frampton, John. 2012. *ExPex for Linguists: Example Formatting, Glosses, and Reference*. Version: 4.1. mathserver.neu.edu/~ling/tex/expex/base/doc/expex-doc.pdf.

[17] Freitag, Constantin and Antonio Machicao y Priemer. 2019. *LaTeX-Einführung Für Linguisten*. Berlin, Germany: Humboldt-Universität zu Berlin. doi.org/10.13140/RG.2.2.29299.27682.

[18] Holton, Gary. 2021. *Writing You Dissertation with LaTeX*. Typescript. Hawai'i. Github.com. gmholton.github.io/files/DissertationWriting.pdf.

[19] Kew, Jonathan and Stephen McConnel. 1990. *Formatting Interlinear Text.* Occasional Publications in Academic Computing, Number 17. Dallas, Texas: Summer Institute of Linguistics.

[20] Knuth, Donald Ervin. 1984. *The TeXbook.* A. Computers & typesetting. Reading, Massachusetts: American Mathematical Society; Addison-Wesley.

[21] Knuth, Donald Ervin. 1986. *TeX: The Program.* B. Computers & typesetting. Reading, Massachusetts: Addison-Wesley.

[22] Lamicela, Andrew Charles. 2020. *Distinguishing Passive from MP2-marked Middle in Koine Greek*. University of North Dakota M.A. thesis. commons.und.edu/theses/3277.

[23] Lehmann, Christian. 2004. *Interlinear morphemic glossing*. In Morphologie: Ein internationales Handbuch zur Flexion und Wortbildung Morphology: an international handbook on inflection and word-formation, edited by Geert E Booij, Christian Lehmann, Joachim Mugdan, and Stavros Skopeteas, 2:1834–57. Handbücher zur Sprach- und Kommunikationswissenschaft Handbooks of Linguistics and communication science 17. Berlin ; New York: Walter de Gruyter. doi.org/10.1515/9783110172782.2.20.1834.

[24] Liter, Adam. 2017. *LATEX Workshop (for Linguists).* adamliter.org/content/LaTeX/latex-workshop-for-linguists.pdf.

[25] Lovell, Douglas. 1999. TeXML: Typesetting XML with TeX. *TUGboat*. 20 (3): 176–183. tug.org/TUGboat/tb20-3/tb64love.pdf.

[26] Marlett, Stephen A, (compiler). 2012. La Frase Nominal. In Stephen A. Marlett (ed.) *Los Archivos Lingüísticos Me'phaa*. Instituto Lingüístico de Verano, A.C. [SIL International in Mexico].

mexico.sil.org/publications/i-wpindex/work_papers_-_mephaa_grammar_files.

[27] Marlett, Stephen A. 2019. *Phonology From the Ground Up: The Basics*. Dallas, Texas: SIL International. www.sil.org/resources/archives/79207.

[28] Neri Méndez, Emilia and Stephen A. Marlett. 2011 (Nov). Presentación Analítica del Texto "Flor de Calabaza". In Stephen A. Marlett (ed.) *Los Archivos Lingüísticos Me'phaa* (versión preliminar). Instituto Lingüístico de Verano, A.C. [SIL International in Mexico]. mexico.sil.org/publications/i-wpindex/work_papers_-_mephaa_grammar_files.

[29] Nordhoff, Sebastian, and Stefan Müller. 2020. *Language Science Press Guidelines*. Berlin, Germany: Language Science Press. langsci.github.io/guidelines/latexguidelines/LangSci-guidelines.pdf.

[30] Paterson III, Hugh J. 2021. *Language Archive Records: Interoperability of Referencing Practices and Metadata Models*. University of North Dakota M.A. thesis. commons.und.edu/theses/3937.

[31] Pellard, Thomas. 2019. Automatic formatting of interlinear glosses with LaTeX. *Cipanglossia* cipanglo.hypotheses.org/1221.

[32] Peter, Steve. 2004. TEX and Linguistics. *TUGboat* 25 (1): 58–62. tug.org/TUGboat/tb25-1/peter.pdf.

[33] Machicao y Priemer, Antonio, and Constantin Freitag. 2019. *LaTeX-Einführung Für Linguisten*. Presentation at the MGK Workshop – SFB 1412, Berlin. doi:www.linguistik.hu-berlin.de/de/staff/amyp/latex20sfb/07-l4l-math2-trees-handout.pdf.

[34] Rasmussen, Kent. 2018. *A Comparative Tone Analysis of Several Bantu D30 Languages (DR Congo)*. University of Texas Arlington Ph.D. dissertation. hdl.handle.net/10106/27483.

[35] Rastorgueva, V. S., A. A. Kerimova, A. K. Mamedzade, L. A. Pireiko, and D. I. Edel'man. 2012. *The Gilaki Language*. Edited by Ronald M. Lockwood. Acta Universitatis Upsaliensis; Studia Iranica Upsaliensia 19. Uppsala, Sweden: Acta Universitatis Upsaliensis. urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-182789.

[36] Rei, Fukui. 1996. TIPA: A System for Processing Phonetic Symbols in LATEX. *TUGboat* 17 (2): 102–14. www.tug.org/TUGboat/tb17-2/tb51rei.pdf.

[37] Smith, Zac, Todd Snider, and Mia Wiegand. 2016. *LaTeX and Linguistics - How to Make Your Research Pretty*. Presentation at the Cornell Linguistics Circle, Cornell, New York. conf.ling.cornell.edu/miawiegand/Latex_Slides.pdf.

[38] Thiele, Christina. 1995. TEX and Linguistics. *TUGboat* 16 (1): 42–44. tug.org/TUGboat/tb16-1/tb46ling.pdf.

[39] Thiele, Christina. 2007. *Christina Thiele Interview by Dave Walden for the TEX Users Group*. Transcript. tug.org/interviews/thiele.html.

[40] Unicode Consortium, ed. 1991. *The Unicode Standard: Worldwide Character Encoding*. Version 1.0. Reading, Massachusetts: Addison-Wesley. www.unicode.org/versions/Unicode1.0.0.

[41] Ward, Monica. 2002. Reusable XML Technologies and the Development of Language Learning Materials. *ReCALL* 14 (2): 285–94. doi.org/10.1017/S0958344002000629.

[42] Wilson, Peter. 2007. The Memoir Class. *TUGboat* 28 (2): 243–46. www.tug.org/TUGboat/tb28-2/tb89wilson.pdf.

[43] Wilson, Peter. 2021. *The Memoir Class for Configurable Typesetting: User Guide*. version 3.70. Normandy Park, WA: The Herries Press. texdoc.org/serve/memoir/0.

[44] Wood, Joyce Kathleen. 2012. *Valence-Increasing Strategies in Urim Syntax*. Graduate Institute of Applied Linguistics M.A. thesis. www.diu.edu/documents/theses/Wood_Joyce-thesis.pdf.

[45] Beier, Christine, and Lev Michael. 2022. *Managing Lexicography Data: A Practical, Principled Approach Using FLEx (FieldWorks Language Explorer)*. In The Open Handbook of Linguistic Data Management, edited by Andrea L. Berez-Kroeker, Bradley McDonnell, Eve Koller, and Lauren B. Collister, 301–14. Open Handbooks In Linguistics. Cambridge, Massachusetts: The MIT Press. doi.org/10.7551/mitpress/12200.003.0029.

[46] Conzett, Philipp, and Koenraad De Smedt. 2022. *Guidance for Citing Linguistic Data*. In The Open Handbook of Linguistic Data Management, edited by Andrea L. Berez-Kroeker, Bradley McDonnell, Eve Koller, and Lauren B. Collister, 143–55. Open Handbooks In Linguistics. Cambridge, Massachusetts: The MIT Press. doi.org/10.7551/mitpress/12200.003.0015.

[47] Schenner, Mathias, and Sebastian Nordhoff. 2016. *Extracting Interlinear Glossed Text from*

LATEX Documents. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), edited by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, et al., 4044–48. Portorož, Slovenia: European Language Resources Association (ELRA). www.aclweb.org/anthology/L16-1638.pdf.

◇ H. Andrew Black
blackhandrew (at) gmail dot com

◇ Hugh J. Paterson III
i (at) hp3 dot me
http://hp3.me