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 its development.

1 Introduction

Within the publishing industry there are several notable products for producing complex documents in beautiful formats. TeX[5],[6] is one of the well known publishing technologies used to meet these needs. Since 2000, XMLbased technologies such as XSL-FO[1] or the TeXML project[7][2] have also seen use to integrate content and compose complex documents such as textbooks and maintenance manuals. Requirements for composing these large, inter-linked documents birthed the development of tools like XMLmind[3] and Xpublisher.[4] These can be used to compose the content within predefined XML structures.

XLingPaper[1],[2],[3] is a plug-in to the XMLmind XML Editor. XLingPaper[5] benefits from XMLmind's Java-based implementation, which allows it to be used on MacOS, Windows, and Linux. XLingPaper, via a DTD, 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 XMLmind, 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, authors cannot input XeLaTeX code directly into the document and the introduction of layout sections within the document flow is constrained via the DTD.

2 What is XLingPaper?

As mentioned above, XLingPaper is an XML- and Java-based computer plug-in for the XMLmind XML Editor. It is designed to make writing, reading, and publishing linguistic papers, grammars, and books better and more consistent. XLingPaper can produce linguistic 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 (.odt), and ePUB. It automatically numbers sections, figures, tables, and examples. It keeps track of internal references to these entites along with citation references and gloss abbreviations. This keeps numbering and reference links dependable and automated. It also automatically generates abbreviations used and references cited (using a custom references implimentation).

[1] `https://www.w3.org/TR/xsl11`
[2] `http://getfo.org/texml`
[3] `https://www.xmlmind.com/xmleditor`
[4] `https://www.xpublisher.com/products`
[5] `https://software.sil.org/xlingpaper`

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 making the document 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 values this gives is that there is a "grammar" of what a well-formed linguistic document looks like. This makes it easy to move and shift sections and chapters and examples and prevents users from inadvertently creating ill-formed documents.

This paper is about the XeLaTeX-produced PDF output.

The official website for XLingPaper is `https://software.sil.org/xlingpaper`. A list of advantages for the author, the reader, and the publisher can be found at the "Why XLingPaper?"[3] portion of the documentation.

3 XLingPaper and TeX

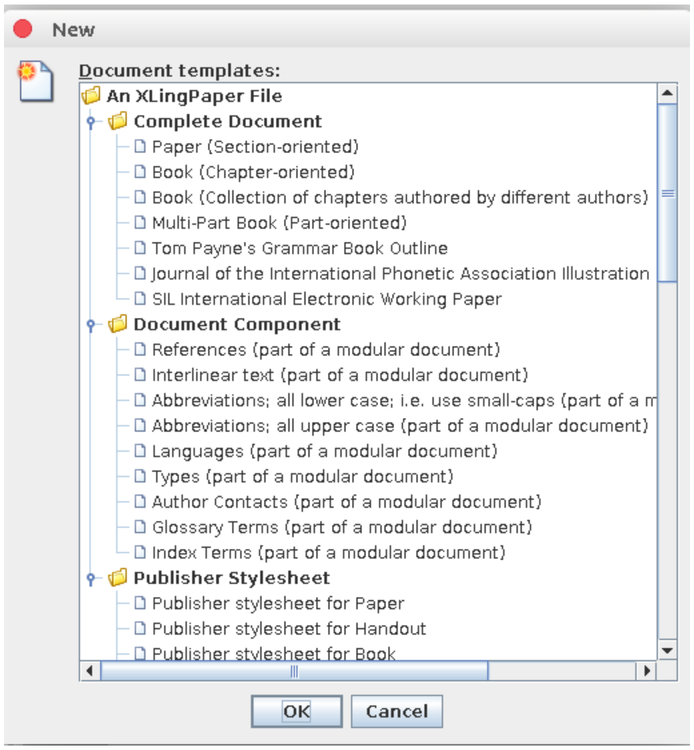
We now describe how XLingPaper uses TeX technologies.

3.1 Design desiderata for TeX with XLingPaper

From the outset, XLingPaper was designed to be free. The XMLmind XML Editor had a Personal Use License that fit the bill for the vast majority of the target audience of XLingPaper. The few that 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 plugin has always been free.

Prior to 2009 XLingPaper used RenderX[6] 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, there would be no water marks in PDF documents.

[6] `http://www.renderx.com`



An XLingPaper File
- Complete Document
  - Paper (Section-oriented)
  - Book (Chapter-oriented)
  - Book (Collection of chapters authored by different authors)
  - Multi-Part Book (Part-oriented)
  - Tom Payne's Grammar Book Outline
  - Journal of the International Phonetic Association Illustration
  - SIL International Electronic Working Paper
- Document Component
  - References (part of a modular document)
  - Interlinear text (part of a modular document)
  - Abbreviations; all lower case; i.e. use small-caps (part of a m
  - Abbreviations; all upper case (part of a modular document)
  - Languages (part of a modular document)
  - Types (part of a modular document)
  - Author Contacts (part of a modular document)
  - Glossary Terms (part of a modular document)
  - Index Terms (part of a modular document)
- Publisher Stylesheet
  - Publisher stylesheet for Paper
  - Publisher stylesheet for Handout
  - Publisher stylesheet for Book

Figure 1: XLingPaper predefined document types via DTD

At the time, XLingPaper had a way to format output per a user-created publisher style sheet. This meant the developer (Andrew Black) needed to be able to map from an XLingPaper publisher style sheet to XƎLaTeX. This is the second criterion. He knew that he wanted to use LaTeX but that 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. He would need to "roll his own" way of handling these. (Unfortunately, he did not learn about the `memoir` package until several years later; otherwise he might have used it.)

The third criterion concerned some of the target audience for XLingPaper. Many of the expected users of XLingPaper live in places around the world where Internet connections are poor. Therefore, the download required to install XLingPaper needed to be as small as possible. This meant requiring users to use something like TeXLive was out of the question. Andy determined which LaTeX packages and binaries were needed and created a custom installation package for just those items.

This was still rather large for someone with poor Internet. The hope was that this set of packages and binaries would not need to change over time, given that XLingPaper was expected to include new features and need bug fixes. In fact, for any custom commands, he did not create a separate XLingPaper package containing them and include them in this custom set of packages. Rather he generated them as commands in the file processed by XeLaTeX. The thinking was that it was more likely new XLingPaper-specific commands would be needed than new LaTeX packages. This has proven true over time. While it was necessary to create new versions of the packaging, such as when he added framed units via the `mdframed` package, generally it has been the case that adding items to the package has not been needed very much.

3.2 PDFx production

When an author has XLingPaper produce PDF output via XƎLaTeX, XLingPaper produces a TeXML-like XML file. This is then converted into LaTeX format via a set of XSLT transforms and given to XƎLaTeX which produces the PDF. Figure 2 contains a diagram of this process.

3.3 TeXML

When Andy began implementing the XƎLaTeX-based output, he found TeXML, but understood it to have two infelicities:

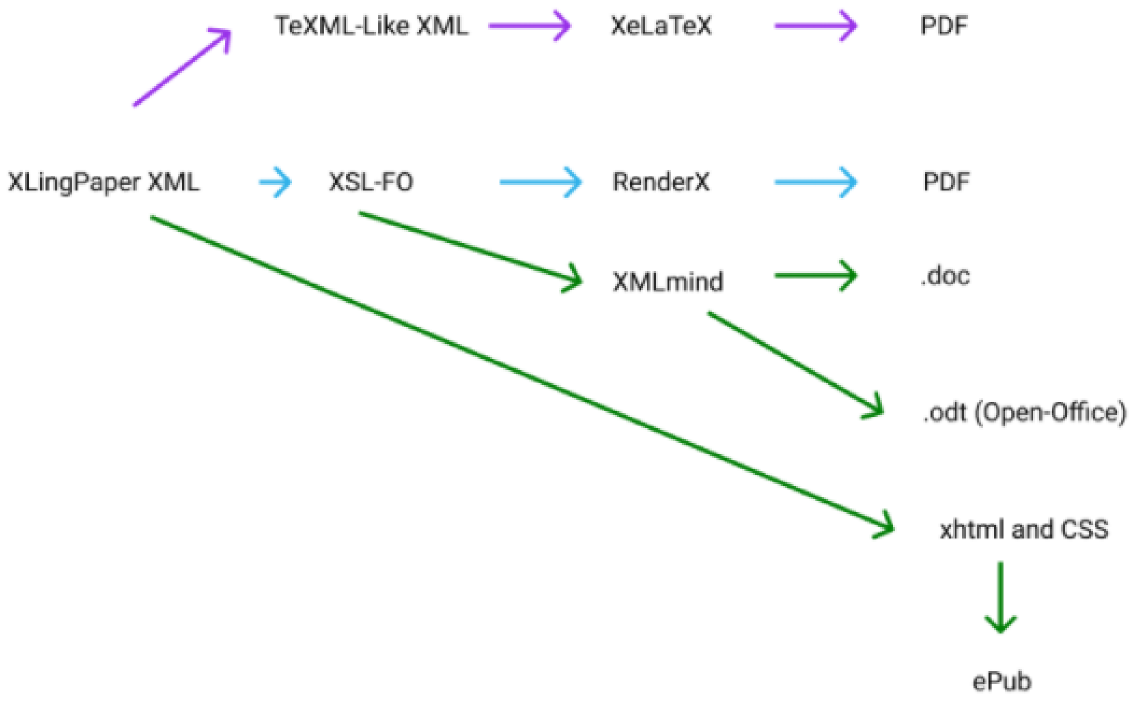


Figure 2: XLingPaper output process

1. TeXML required Python and he did not want to force XLingPaper users to have to install a version of Python for TeXML when that version may conflict with other versions of Python they might already have installed. Furthermore, this approach would make the installation package much larger because of needing to include Python.
2. He also needed some extensions for formatting white space (more finely as far as he could tell).

He did implement some Java code to deal with mapping certain characters used in TEX commands to their TEX equivalents. He used Java because the XMLmind XML Editor is written in Java and XLingPaper already used Java code to improve the user experience in the XMLmind XML Editor.

3.4 Ling-TeX

At the time Andy began implementing the XƎLATEX-based output, he discovered the Ling-TeX group. From what he could tell, the packages that help with interlinear texts did not allow for the larger number of capabilities XLingPaper already dealt with. So he rolled his own. Figure 3 contains an example output with some of the special capabilities XLingPaper offers.

4 Typesetting tasks XLingPaper users often encounter

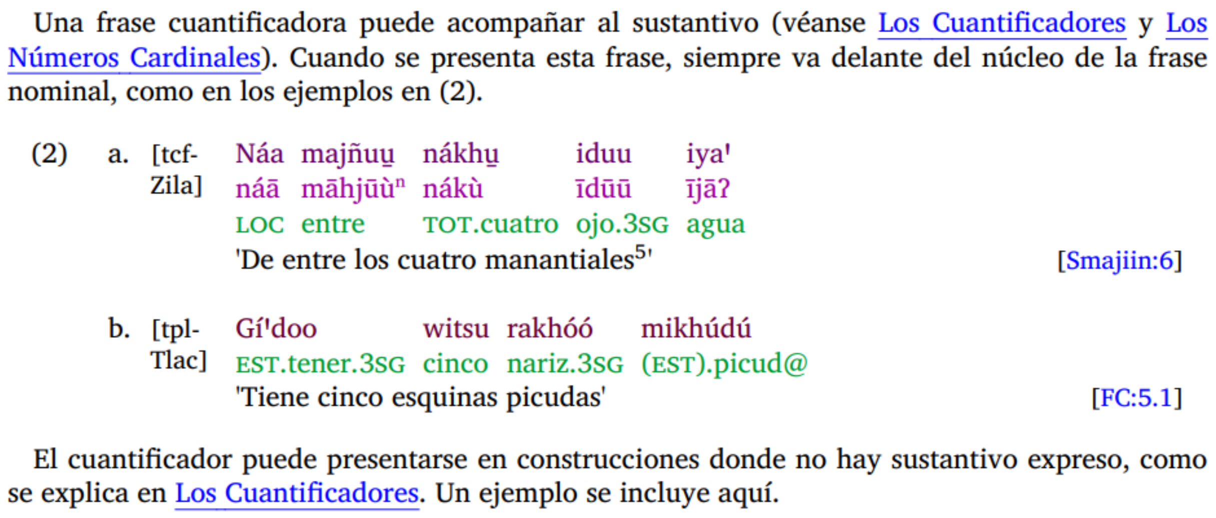
Linguistic documents have several formatting needs that other kinds of documents do not. This section discusses some of them.

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. Besides table-like layouts, linguists also need lists of words along with their glosses (as shown in Figure 4), interlinear clauses (as shown in Figure 3), and even having headings in portions of the example.

4.2 Automatically wrapping interlinear texts

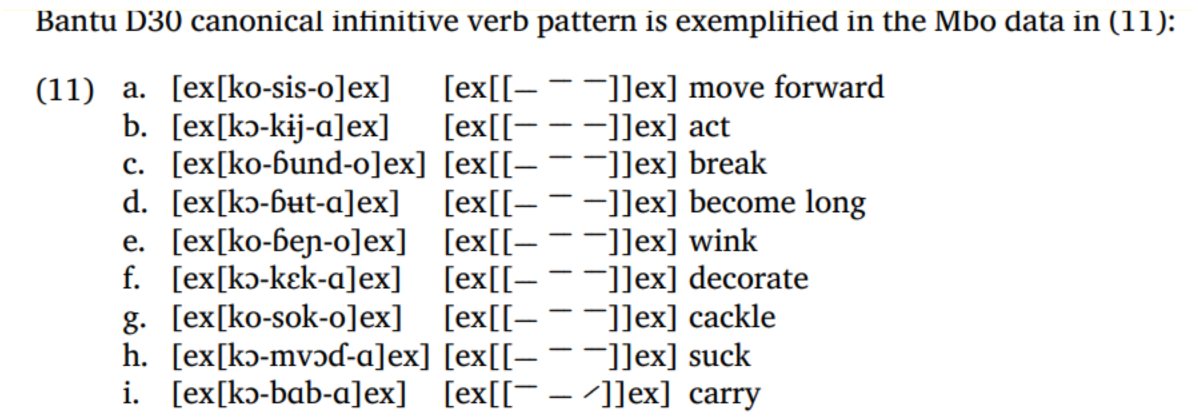
Many linguists want to include interlinear glossed text in their document. XLingPaper allows these to be wrapped automatically which makes the author's job much easier. Figure 5 shows one such text portion.


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	majñuu̱	nákhu̱	iduu	iya'	
		náā	māhjūùⁿ	nákù	īdūū	ījāʔ	
		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]

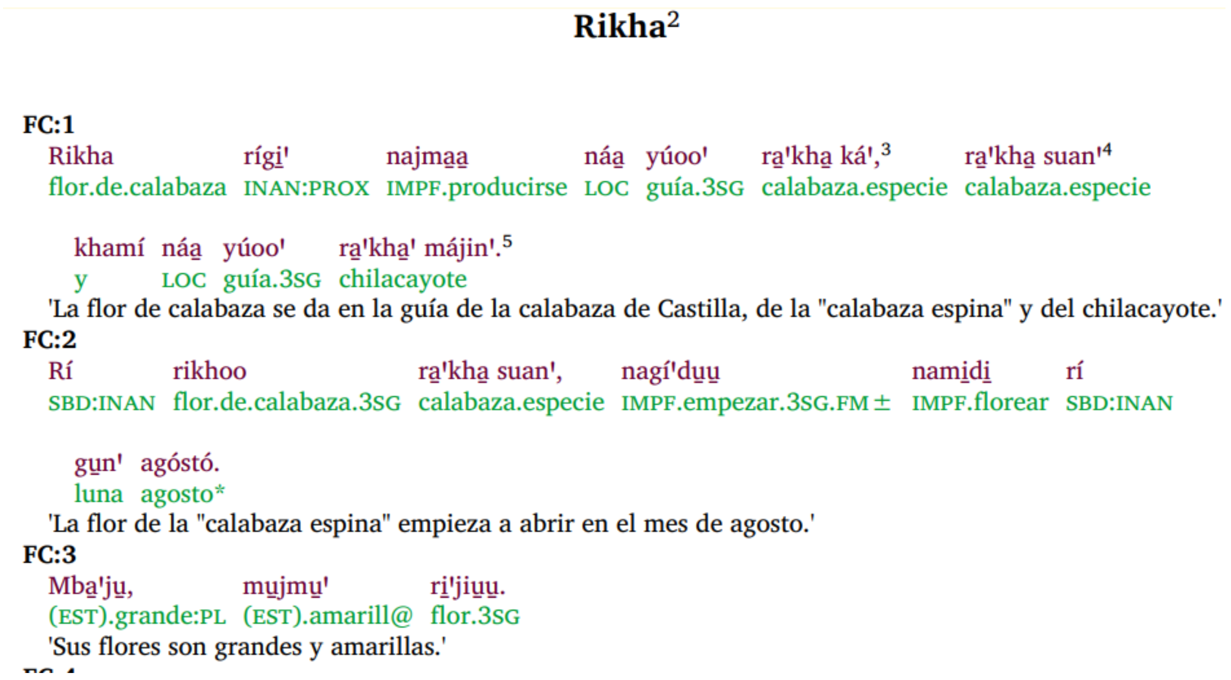
El cuantificador puede presentarse en construcciones donde no hay sustantivo expreso, como se explica en Los Cuantificadores. Un ejemplo se incluye aquí.

Figure 3: Interlinear example


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-ɓeɲ-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 4: List of words



Rikha[2]

FC:1

Rikha	rígi̱'	najma̱a̱	náa̱	yúoo'	ra̱'kha̱ ká',[3]	ra̱'kha̱ suan'[4]
flor.de.calabaza	INAN:PROX	IMPF.producirse	LOC	guía.3SG	calabaza.especie	calabaza.especie

khamí	náa̱	yúoo'	ra̱'kha̱' 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	ra̱'kha̱ suan',	nagí'du̱u̱	nami̱di̱	rí
SBD:INAN	flor.de.calabaza.3SG	calabaza.especie	IMPF.empezar.3SG.FM±	IMPF.florear	SBD:INAN

gu̱n'	agóstó.
luna	agosto*

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

FC:3

Mba̱'ju̱,	mu̱jmu̱'	ri̱'jiu̱u̱.
(EST).grande:PL	(EST).amarill@	flor.3SG

'Sus flores son grandes y amarillas.'

Figure 5: Wrapped interlinear text

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 [4].

5.2 Font rendering

XeLaTeX renders fonts extremely well. It can even handle special features requiring Graphite[7] 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.

[7] `graphite.sil.org`

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.

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 List of LaTeX packages used

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

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

6.2 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 writing linguistic documents has great value in and of itself, the fact that it can produce great looking output via XeLaTeX makes it very worthwhile learning to use. We feel that being able to produce PDF via XeLaTeX has made XLingPaper a fantastic tool for linguists.

References

[1] 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. `http://hdl.handle.net/10125/4532`.

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

[3] Black, H Andrew. 2017. *Why Learn to Use XLingPaper*. Dallas, Texas: SIL International. `http://software.sil.org/downloads/r/`

`xlingpaper/resources/documentation/WhyUseXLingPaper.pdf`.

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

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

[6] Knuth, Donald Ervin. 1986. *TEX: the program.* B. Computers & typesetting. Addison-Wesley. Reading, Massachusetts.

[7] Lovell, Douglas. 1999. TEXML: Typesetting XMLwith TEX. *TUGboat.* 20 (3): 176–183.

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

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