

XLingPaper's use of T_EX Technologies

H. Andrew Black, Hugh J. Paterson III

Abstract

We discuss the use of T_EX technologies by XLingPaper, an authoring tool for producing academically oriented publications with features required for linguistic publishing. We present the T_EX 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. T_EX[10],[11] is one of the well known publishing technologies used to meet these needs. Since 2000, XMLbased technologies such as XSL-FO¹ or the T_EXML project[13]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 XMLmind³ and Xpublisher.⁴ These can be used to compose the content within predefined XML structures. XLingPaper[3],[4],[5] 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 publishers. The software has a growing number of users who have successfully typeset complex documents including:

- master theses[17],[12],[15],
- doctoral dissertations[8],[16],
- textbooks[14],
- linguistic grammars[7],
- journal articles[6], and
- bilingual software documentation[1],[2].

XLingpaper⁵ is a plug-in to the XMLmind XML Editor. XLingPaper 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

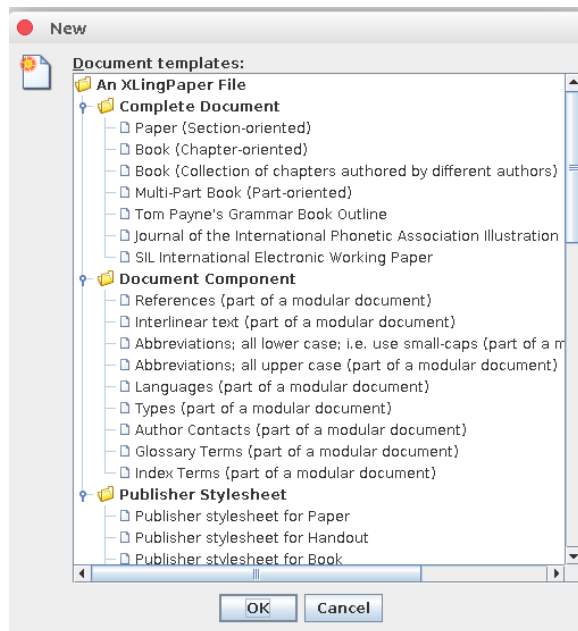


Figure 1: XLingPaper predefined document types via DTD

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 X_LAT_EX code directly into the document and second, 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. A full list of benefits to all parties in the publishing workflow is available.[5] 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 entities 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 implementation).

¹ <https://www.w3.org/TR/xsl11>

² <http://getfo.org/texml>

³ <https://www.xmlmind.com/xmlmind>

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, XLPaper (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 values this gives is that there is a “grammar” of what a well-formed linguistic document looks like. This makes moving, replacing, switching, or reordering sections, chapters, and examples less error prone because it prevents users from inadvertently creating ill-formed documents.

The following sections of this paper discuss the T_EX technologies used.

3 XLPaper and T_EX

Due to the heavy reliance on Unicode in modern language documentation and linguistic work, XLPaper specifically uses X_QL_AT_EX and compatible packages to produce PDF outputs. The following sections provide more detail on the design requirements and packages used.

3.1 Design desiderata for T_EX with XLPaper

From the outset, XLPaper 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 XLPaper. 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 XLPaper plugin has always been free.

Prior to 2009 XLPaper used RenderX⁶ to produce PDF documents. However, in 2009 plans were made to add X_QL_AT_EX-based output to XLPaper because, while there was a free version of RenderX, the output contained a watermark. By implementing the ability to export to PDF via X_QL_AT_EX, there would be no water marks in PDF documents.

At the time, XLPaper 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 XLPaper publisher style sheet to X_QL_AT_EX. This was the second criterion. He knew that he wanted to use L_AT_EX but that pure L_AT_EX came with predefined output formatting for front matter, chapters, sections, back matter, etc. Pure L_AT_EX, then, would not allow direct control of

formatting of all of these per an XLPaper 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 XLPaper. Many of the expected users of XLPaper live in places around the world where Internet connections are poor. Therefore, the download required to install XLPaper needed to be as small as possible. This meant requiring users to use something like T_EXLive was out of the question. Andy determined which L_AT_EX 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 XLPaper was expected to include new features and need bug fixes. In fact, for any custom commands, he did not create a separate XLPaper 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 XLPaper-specific commands would be needed than new L_AT_EX 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 PDF production

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

3.3 TeXML

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

1. TeXML required Python and he did not want to force XLPaper 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.

⁶ <http://www.renderx.com>

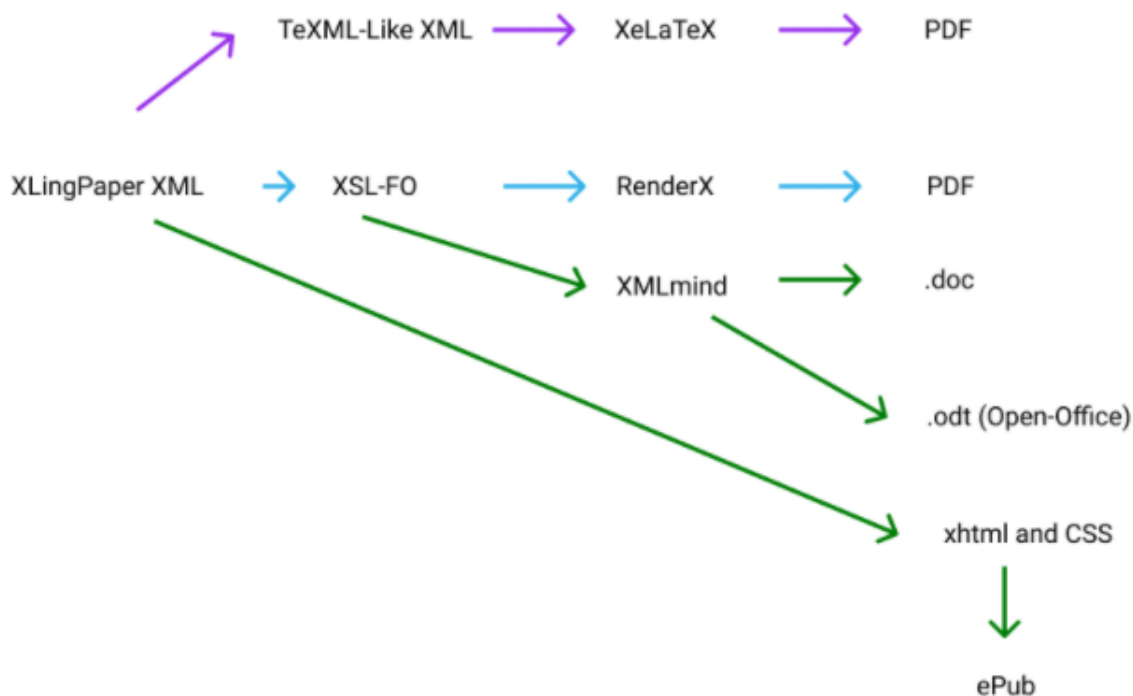


Figure 2: XLPaper output process

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 XLPaper 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 \XeLaTeX -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 XLPaper already dealt with. So he rolled his own. Figure 3 contains an example output with some of the special capabilities XLPaper offers.

4 Typesetting tasks XLPaper 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. XLPaper 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. XLPaper 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). XLPaper allows the author to define a set of abbreviations and their definitions. When producing the output, XLPaper creates hyperlinks between the abbreviation and its definition.

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- NÁa majñuu nákhū iduu iya'
Zila] náā māhjūùⁿ nákù īdūū ījā?
LOC entre TOT.cuatro ojo.3SG agua
'De entre los cuatro manantiales'⁵ [Smajiin:6]
- b. [tpl- Gí'doo witsu rakhóó mikhúdú
Tlac] 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-kij-a]ex] [ex[[- - -]]ex] act
c. [ex[ko-ɓund-o]ex] [ex[[- - -]]ex] break
d. [ex[kɔ-ɓut-a]ex] [ex[[- - -]]ex] become long
e. [ex[ko-ɓep-o]ex] [ex[[- - -]]ex] wink
f. [ex[kɔ-kɛk-a]ex] [ex[[- - -]]ex] decorate
g. [ex[ko-sok-o]ex] [ex[[- - -]]ex] cackle
h. [ex[kɔ-mvɔd-a]ex] [ex[[- - -]]ex] suck
i. [ex[kɔ-bab-a]ex] [ex[[- - -]]ex] carry

Figure 4: List of words

Rikha²

FC:1

Rikha rígi' najmaā náā yúoo' rā'kha ká',³ rā'kha suan'⁴
 flor.de.calabaza INAN:PROX IMPF.producirse LOC guía.3SG calabaza.especie calabaza.especie

khamí náā yúoo' rā'kha' 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 rā'kha suan', nagí'dūu namídi rí
 SBD:INAN flor.de.calabaza.3SG calabaza.especie IMPF.empezar.3SG.FM ± IMPF.florear SBD:INAN

gūn' agóstó.
 luna agosto*

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

FC:3

Mba'ju, mujmu' rí'jiyu.
 (EST).grande:PL (EST).amarill@ flor.3SG

'Sus flores son grandes y amarillas.'

Figure 5: Wrapped interlinear text

5 Outputs L^AT_EX allow that others do not

While X_LLingPaper has a large array of linguistically-oriented formatting capabilities, there are some that only the X_LL^AT_EX output can produce. This is, of course, due to the formatting power of T_EX and X_LL^AT_EX.

5.1 Automatically wrapping interlinears

One of the most popular features of X_LLingPaper 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 X_LL^AT_EX put them together in a hanging indent paragraph. This is based on the work of Kew & McConnel 1990 [9].

5.2 Font rendering

X_LL^AT_EX renders fonts extremely well. It can even handle special features requiring Graphite⁷ 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 X_LL^AT_EX 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 X_LLingPaper document in a non-English language and X_LL^AT_EX will hyphenate according to that language's hyphenation rules.

5.4 Author contact information

X_LLingPaper allows one to define a set of contact information for authors. Only the X_LL^AT_EX output is able to format them correctly.

5.5 Vertical fill

For title page material, only the X_LL^AT_EX 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 X_LL^AT_EX approach is able to do this.

6 Features other outputs have that the L^AT_EX output does not

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

Background color is not available for section titles.

Section 11.17.1.1 “Known limitations of using X_YL^AT_EX” in the XLingPaper user documentation lists known problems.

6.1 List of L^AT_EX packages used

XLingPaper currently uses the following X_YL^AT_EX packages (in alphabetical order):

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

6.2 Custom T_EX 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 X_YL^AT_EX makes it very worthwhile learning to use. We feel that being able to produce PDF via X_YL^AT_EX has made XLingPaper a fantastic tool for linguists.

References

- [1] Beadle, Jennie, and Matthew Lee. 2020a. *Paratext 9 Manual – in English*. SIL International. <https://lingtran.net>.
- [2] Beadle, Jennie, and Matthew Lee. 2020b. *Paratext 9 Manual – in French*. SIL International. <https://outilingua.net>.
- [3] 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>.
- [4] 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>.
- [5] Black, H Andrew. 2017. *Why Learn to Use XLingPaper*. Dallas, Texas: SIL International. <http://software.sil.org/downloads/r/xlingpaper/resources/documentation/WhyUseXLingPaper.pdf>.
- [6] Brownie, John. 2013. Adverbs in the Mussau-Emira Verb Phrase. *Language & Linguistics in Melanesia* 31(1): 1–11. <https://www.langlxmelanesia.com/LLM%20Vol.%2031%20Adverbs%20Mussau.pdf>.
- [7] 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]. <https://www.sil.org/resources/archives/75518>.
- [8] Ebarb, Kristopher J. 2014. *Tone and variation in Idakho and other Luhya varieties*. University of Indiana Ph.D. dissertation. <https://pqdtopen.proquest.com/doc/1625743679.html?FMT=ABS>.
- [9] Kew, Jonathan and Stephen McConnel. 1990. *Formatting Interlinear Text*. Occasional Publications in Academic Computing, Number 17. Summer Institute of Linguistics. Dallas, Texas.

- [10] Knuth, Donald Ervin. 1984. *The T_EXbook*. A. Computers & typesetting. American Mathematical Society; Addison-Wesley. Reading, Massachusetts.
- [11] Knuth, Donald Ervin. 1986. *T_EX: the program*. B. Computers & typesetting. Addison-Wesley. Reading, Massachusetts.
- [12] Lamicela, Andrew Charles. 2020. *Distinguishing Passive from MP2-marked Middle in Koine Greek*. University of North Dakota M.A. thesis. <https://commons.und.edu/theses/3277>.
- [13] Lovell, Douglas. 1999. T_EXML: Typesetting XML with T_EX. *TUGboat*. 20 (3): 176–183. <https://tug.org/TUGboat/tb20-3/tb64love.pdf>.
- [14] Marlett, Stephen A. 2019. *Phonology From the Ground Up: The Basics*. Dallas, TX: SIL International. <https://www.sil.org/resources/archives/79207>.
- [15] Paterson III, Hugh J. 2021. *Language Archive Records: Interoperability of Referencing Practices and Metadata Models*. University of North Dakota M.A. thesis. <https://commons.und.edu/theses/3937>.
- [16] Rasmussen, Kent. 2018. *A Comparative Tone Analysis of Several Bantu D30 Languages (DR Congo)*. University of Texas Arlington Ph.D. dissertation. <http://hdl.handle.net/10106/27483>.
- [17] Wood, Joyce Kathleen. 2012. *Valence-Increasing Strategies in Urim Syntax*. Graduate Institute of Applied Linguistics M.A. thesis. https://www.diu.edu/documents/theses/Wood_Joyce-thesis.pdf.

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

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