

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 XLingPaper development.

1 Introduction

Within the publishing industry there are several notable products for producing complex documents in beautiful formats. T_EX [23] [24] is one of the well known publishing technologies used to meet these needs. Since 2000, XML-based technologies such as XSL-FO¹ or the T_EXML² project [28] have been used 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 tools can be used to compose content within predefined XML structures. XLingPaper, as discussed in [7] [8] [9], 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 [50] [25] [33],
- doctoral dissertations [16] [37],
- textbooks [30],
- linguistic grammars [11],
- books [1] [38],
- journal articles [10], and
- bilingual software documentation [2] [3].

¹ www.w3.org/TR/xsl11

² getfo.org/texml

³ www.xmlmind.com/xmlmind

⁴ www.xpublisher.com/products

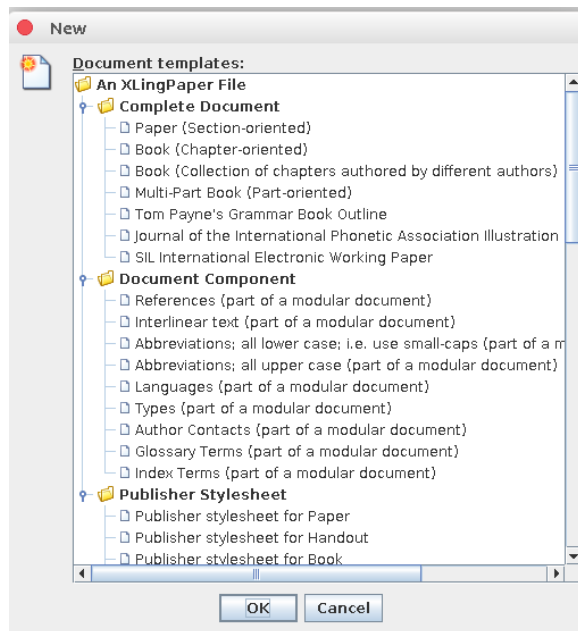


Figure 1: XLingPaper predefined document types via DTD

2 What is XLingPaper?

XLingpaper⁵ 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, 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 [9]. For many, the PDF format is the quintessential file format for final distribution of publishing outputs.

⁵ software.sil.org/xlingpaper

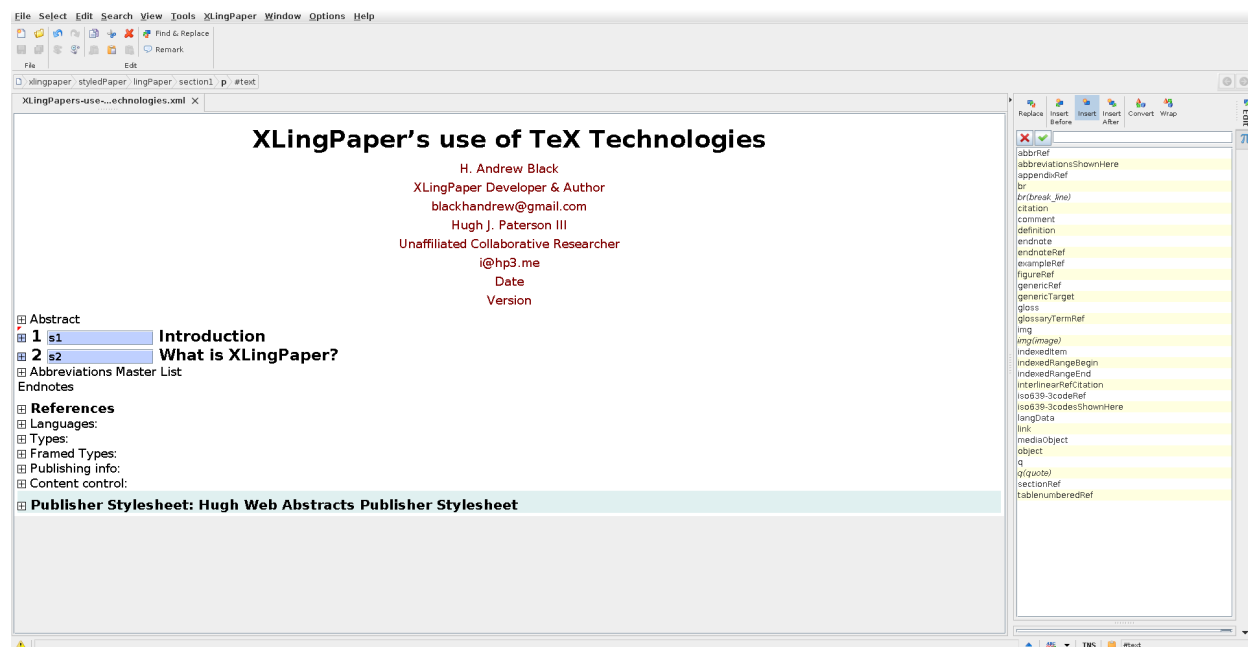


Figure 2: XLingPaper’s user interface. Left side: document content editing. Right side: block and line level units available for use at the cursor location.

XLingPaper supports PDF production but 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 are unconstrained text editors such as those used to code or produce Markdown, XLingPaper (via the XMLmind XML Editor) is a structured editor much more like the block editors we see in tools like MailChimp or WordPress’s Gutenberg editor, albeit without the drag-and-drop features. 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 [35] writes of a personal communication with Don Knuth where Knuth suggests that linguists were some of the earliest adopters outside of mathematicians. Thiele [45] 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 [23]. Thiele [44] 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 [35] provides some additional tips and tools for typesetting

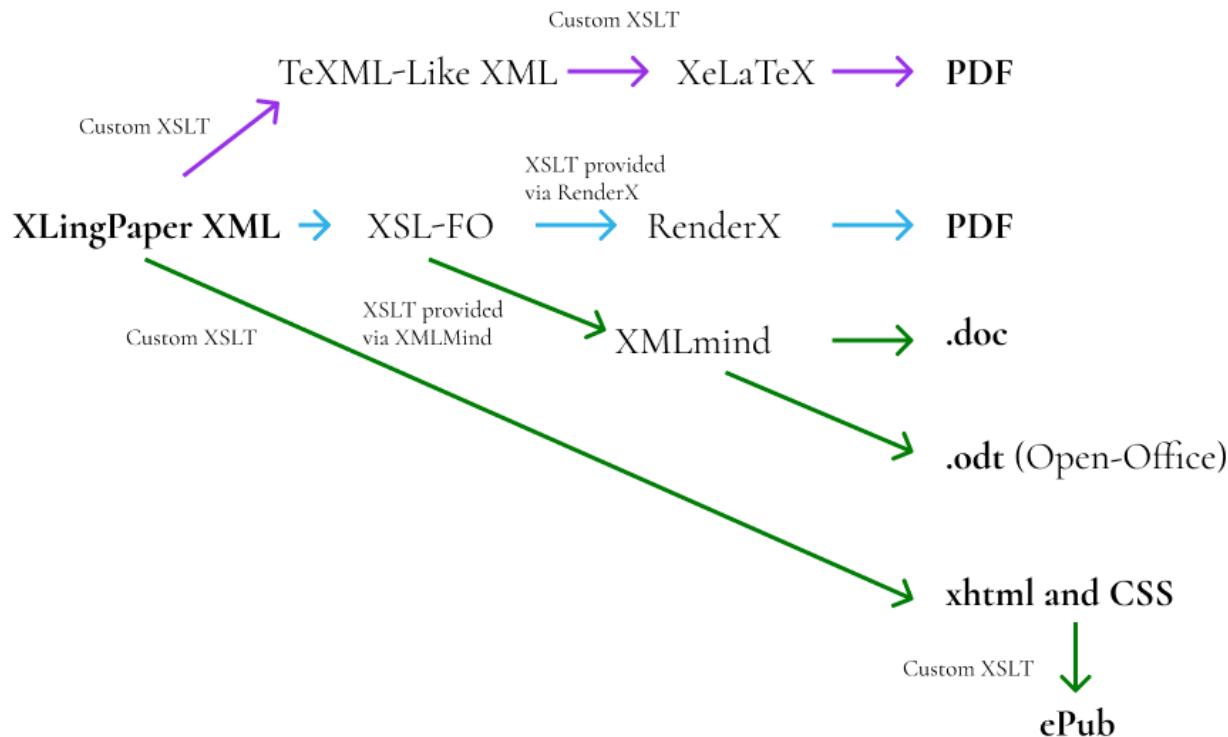


Figure 3: XLPaper’s data processing pipeline to multiple formats

common information structures in linguistic publishing. The T_EX community has produced many packages which have shaped the visual face of publishing in linguistics, including `tipa`⁶ by Rei [39] which provided access to an excellent typeface for phonetic transcriptions and `pst-asr`⁷ by Frampton [17] for autosegmental representations. Some packages used in linguistic publishing are not exclusive to linguistics. For example, Donnelly [15] describes how to use various packages to draw phonetic pitch traces using T_EX. Peter [35] and Thiele [44] list out and review (through 2004) various packages across several areas of linguistics. Among others, they discuss several packages used to draw syntax trees such as `qtrees`⁸ and `forest`⁹ and specialized packages for presenting examples and interlinear glossed texts such as `expex`¹⁰. Their reviews also discuss 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 T_EX 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 [46] 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 T_EX 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 X $\text{\text{LingPaper}}$ discussion for two reasons. First, it exemplifies some of the complexities that X $\text{\text{LingPaper}}$ 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 linguistic 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 X $\text{\text{LingPaper}}$ philosophy follows. That is, X $\text{\text{LingPaper}}$ reduces the complexity of the typesetting task for authors by requiring complex visualizations to be produced via graphical tools. We have found tools like Figma¹⁸ and Inkscape¹⁹ very helpful in the graphic production task. The X $\text{\text{LingPaper}}$ 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 [5] [6] [47]. Due to the heavy reliance on Unicode by today's practitioners of language documentation and linguistic work, X $\text{\text{LingPaper}}$ specifically uses X \LaTeX 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 [20], University of Pennsylvania [14], and Language Science Press Guidelines [32].

¹⁷ For counter examples see [27] [42] [19] and [36].

¹⁸ www.figma.com

¹⁹ inkscape.org

3.2 Design desiderata for X $\text{\text{LingPaper}}$ outputs via \TeX

There have been three goals which have influenced the development of X $\text{\text{LingPaper}}$:

- separation of content and style,
- software accessibility (license and size), and
- beautiful multi-format outputs.

Deciding how \TeX technologies fit within the project has been a journey. X $\text{\text{LingPaper}}$ development started in 2001 without any use of \TeX technologies. In 2006, X $\text{\text{LingPaper}}$ added XSL-FO for PDF production. Prior to 2009, X $\text{\text{LingPaper}}$ used RenderX²⁰ to produce PDF documents. XSL-FO processors can have various degrees of implementation of the XSL-FO standard. RenderX has decent coverage, but also has some limitations which limits page layout. These limitations are discussed in Section 6. In 2009 plans were made to add X \LaTeX -based output to X $\text{\text{LingPaper}}$ because, while there was a free version of RenderX, the output contained a watermark. By implementing the ability to export to PDF via X \LaTeX , watermarks could be avoided all together. The X \LaTeX 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 X $\text{\text{LingPaper}}$ environment was a key design requirement. When the X \LaTeX method of PDF production was introduced, X $\text{\text{LingPaper}}$ 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 X $\text{\text{LingPaper}}$ publisher style sheet to X \LaTeX . 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 X $\text{\text{LingPaper}}$ 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. X $\text{\text{LingPaper}}$ takes a custom approach in implementing flexibility here. The programmer of X $\text{\text{LingPaper}}$ recently discovered `memoir`²¹ [48] [49]. As a package, `memoir` 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 X $\text{\text{LingPaper}}$ code base would be reduced.

²⁰ www.renderx.com

²¹ ctan.org/pkg/memoir

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

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 costless 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 license 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 identified which L^AT_EX packages and binaries were needed and created a custom installation package which met the required specifications.

XLingPaper currently uses the following L^AT_EX 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 L^AT_EX 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 L^AT_EX packages distributed with XLingPaper. This abstraction layer was then executed when the X_QL^AT_EX file was processed. This middle layer has granted XLingPaper flexibility in adding new code and capabilities while keeping the “heavy” L^AT_EX 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 mdframed²³ package [12]. The architecture separating stable packages from custom code, however, has generally worked out well and kept update sizes low. Appendix A lists the custom commands used.

3.3 PDF production

We know of two 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 X_QL^AT_EX, XLingPaper produces a TeXML-like XML file. This is then converted into a L^AT_EX formatted document via a set of XSLT transforms and processed via X_QL^AT_EX 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 X_QL^AT_EX. 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

²² github.com/sillsdev/XLingPap

²³ ctan.org/pkg/mdframed

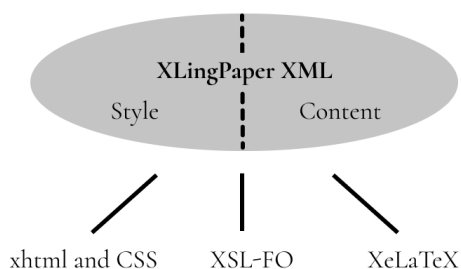


Figure 4: XLPaper combines style and content information contained in its custom XML and then exports it into three different formats for further processing.

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 XLPaper much larger due to the inclusion of Python.

2. XLPaper users require a high degree of control for white space. The fine grain control of whitespace was not clearly possible via \TeX XML.

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, XLPaper is more than a \TeX document producer.

Besides \TeX , XLPaper 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 \TeX -based output, he discovered the Ling- \TeX group²⁴ 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 [44]. These macros were used to solve similar use cases, among others, to those already implemented by XLPaper. 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 XLPaper had already overcome via XSL-FO processing using RenderX. XLPaper 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 [34] for discussion of approaches in \TeX and his solution `typgloss`.²⁹ Figures 5–7 contain example output illustrating some of the special capabilities XLPaper offers.

²⁴ web.archive.org/web/20150702123633/http://heim.ifi.uio.no/~dag/ling-tex/

²⁵ ling-tex.ifi.uio.no/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- NÁa majñuu nákhü iduu iya'
Zila] náā māhjūùⁿ nákù idūū ījā?
LOC entre TOT.cuatro ojo.3SG agua
'De entre los cuatro manantiales⁵¹ [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 5: Interlinear example from [29]. 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 citation to the reference for the source text is indicated.

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 exemplar material (Figures, Tables, Examples, etc.) or to previous examples. XLingPaper automatically keeps track of the ordinal indicators while allowing for different systems of numeration (1,2, 3; a,b,c; i, ii, iii, etc.). 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 Interlinear glossed texts

There is a long tradition within linguistics and language study of presenting phrases containing different languages (but the same content) as interlinear texts. Di-Biase-Dyson et al. [13] trace the practice back as far as the 1652 publication of Kircher [22]. More recent publications display significant variation in page layout related to interlinear glossed texts and interlinear examples. Variation exists in three dimensions:

- content,

- data-structure of the encoding,
- page layout.

A full demonstration of the variation in content and its positioning across common style sheets in linguistics is beyond the scope of this paper. Significant variations include the presence or absence of the following elements:

- index elements such as example numbers or sub-numbers (as shown in Figure 5),
- headings to the interlinear,
- speaker indicator,
- language indicator,
- citation indicator pointing to the larger text from which the exemplar element is taken (see Figure 5 for an example), and
- limits on the number of rows in the original, gloss, translation, and free translation tiers.

Existing T_EX packages approach these content requirements in different ways. As far as we can tell the following commonly used packages for interlinear glossing all have limitations to some degree. The **expex** package does not offer a content solution for the language code or the citation. The package **langsci-gb4e**,³⁰ a fork of **gb4e**,³¹ supports the *Leipzig Glossing Rules*,³² a commonly adopted set of linguistic typesetting conventions. However, while

³⁰ ctan.org/pkg/langsci-gb4e

³¹ ctan.org/pkg/gb4e

³² eva.mpg.de/lingua/pdf/Glossing-Rules.pdf

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-ʒund-o]ex] [ex[[- - -]]ex] break
 d. [ex[kɔ-ʒut-a]ex] [ex[[- - -]]ex] become long
 e. [ex[ko-ʒep-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[[- - ʌ]]ex] carry

Figure 6: List of words as seen in [37]

the *Leipzig Glossing Rules* do call for the language name or identifier to appear on the right hand side of the interlinear glossed text, it does not have a place for the citation. The package `linguex` does not have either language or citation content places built in. With these considerations XLingPaper offers more to authors than other packages in the \TeX ecosystem, even though this meant creating custom \TeX scripts to implement interlinear texts.

There are still some reasons related to data structure for considering XLingPaper. Interlinear glossed texts are often stored in one of a few formats: ELAN files,³³ FLEEx Text files,³⁴ Standard Format files,³⁵ \LaTeX files,³⁶ or custom project-specific XML files. Moving content from analysis and markup tools to typesetting tools is an ever present need for linguists. Several tools such as ELAN and FLEEx have well established workflows for data transfer [40]. FLEEx is often considered the tool of choice for many field linguists, language documentors, and lexicographers. For many linguists entering the field it is the tool of choice over older tools like Toolbox due to built in collaborative features and grammar parsers [4]. Interlinear text in FLEEx can be exported directly into XLingPaper documents.. This presents FLEEx users the opportunity to typeset their texts rather easily. Using the XML document referencing strategy allows authors to reflow typesetting outputs easily if they make changes in their FLEEx environment. We have also had reports of linguists using the FLEEx-XLingPaper bridge to typeset ELAN texts in \LaTeX documents. They capture the \LaTeX document prior to rendering and then copy the relevant \TeX sections to their primary document.

Still finally, there is the matter of page layout. The main types of variation in page layout we have seen include the grouping of lines into sets or subsets (see Figure 5 for example), the labeling of sets and subsets, wrapping of interlinear glosses across lines (recall that these may themselves include three or more lines), and the alignment of the various elements of content within the interlinear glosses. We have seen word and morpheme aligned interlinears. XLingPaper automatically wraps interlinears which makes the author’s job much easier. Figure 7 in FC:1 and FC:2 demonstrate the wrapping of interlinear glossed texts. It does so by formatting each aligned word in an `hbox` and then having \LaTeX put them together in a hanging indent paragraph. This is based on the work of Kew & McConnel 1990 [21].

4.3 Gloss abbreviations

Linguists standardly use glosses for indicating the meaning of pieces of words (morphemes). One common set of glosses is the Leipzig Glosses. However, Leipzig Glosses are not universally used for several reasons including:

- some authors have established their own tradition within their works which they started prior to the release of the Leipzig Glosses,
- they are not comprehensive, and
- they are not theoretically sufficient for some linguists.

XLingPaper approaches this by providing built-in access to Leipzig Glosses, but also allowing the author to define a set of abbreviations and their definitions. When producing the output, XLingPaper creates hyperlinks between the abbreviation and its definition. This allows for the automatic generation of a table or list of abbreviations used.

³³ archive.mpi.nl/tla/elan

³⁴ software.sil.org/fieldworks

³⁵ software.sil.org/toolbox

³⁶ For examples see [41] and [43].

Rikha²

FC:1

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

khamí náā yúoo' rā'khā' 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'khā 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

Mbā'ju, mujmū' rī'jiūu.
 (EST).grande:PL (EST).amarill@ flor.3SG

'Sus flores son grandes y amarillas.'

Figure 7: Wrapped interlinear text as seen in [31].

4.4 Bibliographies

For better or worse XLingPaper has rolled its own bibliography solution. Import solutions are provided for MODS and EndNote XML formats. This enables users to import from tools like Endnote, Zotero, and JabRef. XLingPaper uses custom \TeX scripts to output \TeX code for final rendering. It does not rely on $\text{BIB}\TeX$ or $\text{BIB}\LaTeX$.

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 $\text{X}\LaTeX$ output can produce. This is, of course, due to the formatting power of \TeX and \LaTeX .

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.

5.2 Font rendering

$\text{X}\LaTeX$ renders fonts extremely well. It can even handle special features requiring Graphite³⁷ 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.

³⁷ graphite.sil.org

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 $\text{X}\LaTeX$ 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 $\text{X}\LaTeX$ 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 $\text{X}\LaTeX$ output is able to format them correctly.

5.5 Vertical fill

For title page material, only the $\text{X}\LaTeX$ 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 $\text{X}\LaTeX$ approach is able to do this.

Chao, Yuen Ren. 1930. ə sistim əv "toun-letəz" [A system of "tone-letters"]. *Le Maître Phonétique (Troisième Série du Le Maître Phonétique)* 30. 24–27.

赵元任 [Chao, Yuen-Ren]. 1980. 一套标调的字母 (英文). 方言 1980(2). 81-83.

Chelliah, Shobhana Lakshmi, Willem Joseph de Reuse. 2011. *Handbook of descriptive linguistic fieldwork*. Dordrecht, Netherlands; New York: Springer. doi:[10.1007/978-90-481-9026-3](https://doi.org/10.1007/978-90-481-9026-3)

Chen, Yiya & Carlos Gussenhoven. 2015. Shanghai Chinese. *Journal of the International Phonetic Association* 45(3). 321-37. doi:[10.1017/S0025100315000043](https://doi.org/10.1017/S0025100315000043)

Cheung, Kwan-hin [張群顯]. 2016. Chao Tone Letters: Original theory Versus Current Practice. In 錢志安, 郭必之 and 鄒嘉彥, *Commemorative Essays for Professor Yuen Ren Chao: Father of Modern Chinese Linguistics* 現代漢語語言學之父 — 趙元任先生紀念論文集, 65-76. 臺北市 [Taipei City]: 文鶴出版有限公司 [Crane Publishing Company].

Figure 8: An XLPaper bibliography demonstrating Latin and Chinese scripts.

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

X_QL^AT_EX does not allow for custom table cell padding and spacing. Having said that, the developer cannot remember any XLPaper user ever asking for a way to do this for the X_QL^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_QL^AT_EX” in the XLPaper user documentation lists known problems.

7 Conclusion

While the XLPaper 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 X_QL^AT_EX makes the learning curve rewarding. We feel that being able to produce PDF via X_QL^AT_EX has made XLPaper a fantastic tool for linguists.

A Custom T_EX commands

XLPaper 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 XLPaper’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.

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] 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.
- [5] Bird, Steven, and Gary Simons. 2002. Seven Dimensions of Portability for Language Documentation and Description. In ISCA SALT MIL 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] Black, H. Andrew. 2017. *Why Learn to Use XlingPaper*. Dallas, Texas: SIL International. software.sil.org/downloads/r/xlingpaper/resources/documentation/WhyUseXlingPaper.pdf.
- [10] Brownie, John. 2013. Adverbs in the Mussau-Emira Verb Phrase. *Language & Linguistics in Melanesia* 31(1):1–11. www.langlxmlanesia.com/issues.
- [11] 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.
- [12] Daniel, Marco, and Elke Schubert. 2013. *The mdframed Package: Auto-split Frame environment* version 1.9b.
- [13] Di-Biase-Dyson, Camilla, Frank Kammerzell, and Daniel A Werning. 2009. Glossing Ancient Egyptian. Suggestions for Adapting the Leipzig Glossing Rules. *Lingua Aegyptia. Journal of Egyptian Language Studies* 17: 343–66. www.user.gwdg.de/~lingaeg/lingaeg17.htm.
- [14] Dimitriadis, Alexis. 2016. *TeX/LaTeX Information*. Web page. www.ling.upenn.edu/advice/latex.html.
- [15] Donnelly, Kevin. 2013. Representing Linguistic Pitch in X_gLaTeX. *TUGboat* 34 (2): 223–27. tug.org/TUGboat/tb34-2/tb107donnelly.pdf.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] Holton, Gary. 2021. *Writing Your Dissertation with LaTeX*. Typescript. Hawai’i. Github.com. gmholton.github.io/files/DissertationWriting.pdf.
- [21] Kew, Jonathan and Stephen McConnel. 1990. *Formatting Interlinear Text*. Occasional Publications in Academic Computing, Number 17. Dallas, Texas: Summer Institute of Linguistics.

- [22] Kircher, Athanasius. 1652. *Edipus Ægyptiacus, hoc est Vniuersalis Hieroglyphicæ Veterum Doctrinæ temporum iniuria abolitæ Instauratio*. Opus ex omni Orientalium doctrina & sapientia conditum, nec non viginti diuersarum linguarum autoritate stabilitum, Romæ: Ex Typographia Vitalis Mascardi.
- [23] Knuth, Donald Ervin. 1984. *The T_EXbook*. A. Computers & typesetting. Reading, Massachusetts: American Mathematical Society; Addison-Wesley.
- [24] Knuth, Donald Ervin. 1986. *T_EX: The Program*. B. Computers & typesetting. Reading, Massachusetts: Addison-Wesley.
- [25] 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.
- [26] 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.
- [27] Liter, Adam. 2017. *LATEX Workshop (for Linguists)*. adamliter.org/content/LaTeX/latex-workshop-for-linguists.pdf.
- [28] Lovell, Douglas. 1999. T_EXML: Typesetting XML with T_EX. *TUGboat*. 20 (3): 176–183. tug.org/TUGboat/tb20-3/tb64love.pdf.
- [29] 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.
- [30] Marlett, Stephen A. 2019. *Phonology From the Ground Up: The Basics*. Dallas, Texas: SIL International. www.sil.org/resources/archives/79207.
- [31] 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.
- [32] Nordhoff, Sebastian, and Stefan Müller. 2020. *Language Science Press Guidelines*. Berlin, Germany: Language Science Press. langsci.github.io/guidelines/latexguidelines/LangSci-guidelines.pdf.
- [33] 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.
- [34] Pellard, Thomas. 2019. Automatic formatting of interlinear glosses with LaTeX. *Cipanglossia* cipanglo.hypotheses.org/1221.
- [35] Peter, Steve. 2004. TEX and Linguistics. *TUGboat* 25 (1): 58–62. tug.org/TUGboat/tb25-1/peter.pdf.
- [36] 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/amp/latex20sfb/07-141-math2-trees-handout.pdf.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] Salfner, Sophie, and Tim Gaved. 2014. Working with ELAN and FLEx Together: An ELAN-FLEx-ELAN Teaching Set. Electronic Manuscript. SOAS, London, England. soas.ac.uk/elar/helpsheets/file122785.pdf.
- [41] 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.
- [42] 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.
- [43] So Miyagawa, and Vincent W.J. van Gerven Oei. 2021. Building Web Corpus of Old Nubian with Interlinear Glossing as Digital Cultural Heritage for Modern-Day Nubians. In *The Proceedings of the 11th Conference of Japanese Association for Digital Humanities*, vol. 2021. 144–147. Tokyo: Historiographical Institute, The University of Tokyo. hi.u-tokyo.ac.jp/JADH/2021/Proceedings_JADH2021_rev0905.pdf.
- [44] Thiele, Christina. 1995. TEX and Linguistics. *TUGboat* 16 (1): 42–44. tug.org/TUGboat/tb16-1/tb46ling.pdf.
- [45] Thiele, Christina. 2007. *Christina Thiele Interview by Dave Walden for the TEX Users Group*. Transcript. tug.org/interviews/thiele.html.
- [46] Unicode Consortium, ed. 1991. *The Unicode Standard: Worldwide Character Encoding*. Version 1.0. Reading, Massachusetts: Addison-Wesley. www.unicode.org/versions/Unicode1.0.0.
- [47] Ward, Monica. 2002. Reusable XML Technologies and the Development of Language Learning Materials. *ReCALL* 14 (2): 285–94. doi. [org/10.1017/S0958344002000629](https://doi.org/10.1017/S0958344002000629).
- [48] Wilson, Peter. 2007. The Memoir Class. *TUGboat* 28 (2): 243–46. www.tug.org/TUGboat/tb28-2/tb89wilson.pdf.
- [49] 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.
- [50] 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.

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

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