Grammars for the people, by the people, made easier using PAWS and XLingPaper

Cheryl A. Black
H. Andrew Black
SIL International and University of North Dakota

The task of documenting the minority languages of the world, many of them endangered, is daunting. Further, it is most likely impossible to expect that linguists can go to every language and write a reference grammar for it. At the same time, the indigenous people are becoming more educated and more interested in working on their own languages. This paper describes a computational tool that teaches native speakers about various linguistic constructions, has them enter data from their language and answer simple questions about it, and then produces a draft of a practical grammar of the language. This grammar can be edited for publishing electronically and/or on paper and is useful for the people themselves as well as by linguists.

The underlying XML technology allows much of the complexity to be hidden from the user, while providing multiple views and outputs possible from the same data. The marked-up XML files are archivable and usable by many XML editors. Localization and customization are also possible.

1 Introduction. Linguists are scrambling to try to meet the need of documenting and describing the endangered languages of the world, as well as many of the other minority languages. Further, it is fair to assume it would be impossible for linguists to go to every language and write a reference grammar for it. The task simply takes too much time and there are not enough trained linguists available. Even if the linguists could accomplish the task of language documentation and description, current methods would not be productive enough, since documents written in English for linguists do little to help preserve a language.

At the same time, the indigenous people want to be involved as they are becoming more educated and more interested in working on their own languages. A different type of grammar is needed: one that serves the language community, describes the language in general terms, and is also useful to linguists for extracting data for analysis. This type of grammar has the potential to revitalize the use of a language as the people realize their language is a real language worthy of use because it has a grammar and a dictionary like the national language.

This paper describes a computational tool called PAWS (Parser and Writer for Syntax) that can be used, especially in a workshop setting, to teach native speakers about various linguistic constructions, have them enter data from their language and answer simple questions about it, and then produce a draft of a practical grammar of the language. Currently PAWS only runs on Windows operating systems. It is available at http://carla.sil.org/paws.htm.

The practical grammar style is illustrated in section 2. Section 3 details the user interface for input and how to edit the output using the XLingPaper authoring tool (H. A. Black 2009).[1] Section 4 then explains how it all works computationally.

2 Practical grammars. Practical grammars, also known as popular grammars, are designed for use by the native speakers in the language community. As such, the grammars should be written using the national language for the explanations and glosses. The version described here includes additional material to provide some pedagogy for the reader. Moreover, numerous tables and data in interlinear format and description of the constructions make it useful to linguists and bilingual teachers as well.

[1]For more on XLingPaper, see http://www.xlingpaper.org/.

2.1 General structure. A practical grammar consists mostly of data with some prose explanation. Information about single words or morphemes is usually presented in tables, but all longer examples are given in interlinear format. This format is a bit different than that found in most linguistic publications in order to make it most useful to and understandable by native speakers of the language. Four lines are used: the first line gives the vernacular words, without breaking them into morphemes, as morpheme breaks could be very confusing to the native speaker. The second line is the gloss of the word, with any additional words needed in the gloss language separated by periods. The third line gives the morpheme gloss with normal linguistic abbreviations and conventional symbols like hyphens separating the glosses for each morpheme. This third line is especially for linguists, but is given lower than the word gloss to make it easier for the speakers of the language and bilinguals to skip over if they so choose. The individual morphemes will usually be listed in separate tables to enable the linguists to parse the words, as exemplified in (3)-(4). (It would also be possible for the author to add a line to the grammar output giving the vernacular morphemes between hyphens, but this should come after the word gloss line and before the morpheme gloss line if included.) Finally, a free translation is given on the fourth line of the interlinear. This four-line structure is illustrated schematically in (1).

A completed interlinear example from Isthmus Zapotec is shown in (2):

One or more tables listing the dependent pronoun forms, as illustrated in (3) for Isthmus Zapotec, document this information in a central place and aid the linguist in parsing the words in the interlinear examples.[2]

[2]First and second person singular forms are not listed in the table because they cause a change in the noun root. Such details need to be explained separately while editing the grammar output.

The PAWS interface also asks the user to check off the inflection features used in their language. This is output in a table such as shown in (4) for Isthmus Zapotec.

Note that judgements on the grammaticality of data, usually noted by * and ?, are not used in the practical grammar in order to avoid confusion for the language community. Instead, the prose description of the construction is used to make the distribution clear. The prose is meant to be a-theoretical, but complete enough to allow linguists to apply their theory to the data. A descriptive style grammar also has a longer and wider useful life since it is not limited to the applicability of any particular linguistic theory.

At the end of the grammar, we suggest the addition of several native texts from various genres. This not only allows the language community to identify with the grammar, but also provides examples of sentences in a larger discourse context for the linguist. We suggest a three part presentation of each text to best meet the needs of each audience:

The output from PAWS does not include any such texts, but it does have a section where the user is encouraged to add them.

2.2 Impact on a language community. We have mentioned the importance of completing language description for the endangered languages. Even when a language has been described in English and it is not in imminent danger, a practical grammar written for the language community can have a profound impact. This was clearly demonstrated when the first edition of the practical grammar for Isthmus Zapotec was dedicated in January, 1999 (Pickett, Black & Marcial 1998).

While Zapotec speakers in general do feel inferior to Spanish speakers, Isthmus Zapotec is the prestige variety of Zapotec. Further, the grammar of the language had been described quite well by Velma Pickett in her dissertation (Pickett 1959) and in a number of other articles. Still, Zapotec speaker Vicente Marcial Cerqueda came to Velma and to Cheryl Black and asked for help in writing a grammar in Spanish for his people. He wanted his people to realize that they did not have to abandon their language and only teach their children Spanish. The grammar needed to be presented in a simple, clear and correct form, so that they could understand that while

their mother tongue is distinct from Spanish in many ways, it has a rich and complete structure just like all the other languages of the world.

We are happy to say that Vicente's goals were substantially realized. When the practical grammar was dedicated, there was a big celebration in Juchitn, Oaxaca, Mexico. Over 100 Zapotecs in full native dress crowded into the auditorium of the Casa de la Cultura. The top mariachi band came out of retirement to play for the dedication. The whole program was videotaped and televised later in its entirety throughout the region. The people stood in a long line afterward to purchase their copy of the grammar and have it autographed by all three coauthors. The next day, there was a radio talk show about the grammar. There were speakers from four different varieties of Zapotec on the air. They would open to a particular page and discuss the construction described there, saying for example, "It says on page 20 that Isthmus has a plural marker *ca* before the noun. In my language we say..." At the church service we attended on Sunday, people were happy that the grammar was presented because they could now use their Zapotec New Testaments in the public church service instead of just reading it at home. Interest and pride in their language was clearly restored.

A further encouragement came the following week when a dedication service was held at the Universidad Nacional Autnoma de Mxico in Mexico City: One of the speakers from the university commented that this grammar, even though meant for the people, is useful for the Mexican linguists as well because the data is presented in interlinear format and the IPA charts are included. Our goal of producing a single grammar that could meet both needs was accomplished.

3 What the user sees. Turning now to the implementation of PAWS, when the practical grammar option is selected, only the teaching and questions relevant to writing the grammar are presented to the user. There is a series of fifty-seven interactive web pages for the user to complete. The output is a draft of a practical grammar for their language based on their answers and sample data, which is intended to be further edited and enhanced for publishing.

3.1 Interactive pages. The PAWS program contains a series of interactive pages which teach some linguistics and then ask questions about the various constructions. The program initially assumes default answers based on the

word order typology of the language, but it allows for exceptions.

On most pages, there is a brief instruction on the construction with illustrative examples, then a series of multiple-choice questions about that construction for the language the user is working on. The number of questions asked depends upon how previous questions were answered. These answers are recorded and later used to give the prose explanation about the distribution of the construction in their language. It also asks the user to enter sample data for the various constructions, and records this data for the examples in the grammar.

Examples (59) show one such page for how possessors are handled within nominal phrases. We show what the user sees when they have chosen to only produce a practical grammar output. (The PAWS program can also produce a PC-PATR grammar output.[3] See McConnel (1995) for more on PC-PATR.)

Example (5) below shows an instruction section along with a request to enter sample data.

Example (6) below shows two sets of multiple choice questions, followed by more instruction and a third multiple choice question.

If a user clicks on the second "Yes" answer shown in (6), then more questions are asked (due to the fact that more information is needed about the nature of the phrasal clitic). Example (7) below shows the relevant portion.

If the user instead chooses the option to say that the phrasal clitic is written as a separate word, then even more questions appear as shown in example (8) below.

[3]PC-PATR is an implementation of the unification-based PATR-II computational linguistic formalism (Shieber 1986). In addition, it is augmented with logical constraints on feature nodes and a priority union operation. The "PC" part of the name reflects the fact that it is designed to be used on personal computers (as opposed to mainframe or other large computers common at the time it was written). It is available for MS-DOS, Microsoft Windows, Macintosh, and Unix.

Example (9) shows the bottom portion of this page. It has some instruction, a question, and then two buttons, one for going back to the previous page and the other for going forward to the next page. It also has a link to jump to the main contents page.

3.2 Grammar draft to edit. As the user works his/her way through these interactive pages, s/he can save their work and return later for another session. Whenever the work is saved, the output is produced based on the answers given so far. Naturally, we recommend that the user not look at the generated output until s/he has completed all the interactive pages.

Depending on the complexity of the language and how much data is entered, the draft of the practical grammar which is output could be about 60-90 pages if printed out. This includes the prose description and tables and interlinear data.

To illustrate the coverage of the practical grammar, the table of contents for the initial output of one grammar is shown in examples (1011).[4]

3.2.1 Use of XLingPaper. The practical grammar is output in XLingPaper format, which can then be edited as described in sections 3.2.23.2.3. Before discussing this, we give a brief sketch of the advantages of XLingPaper.

Linguistic documents are by nature complex and also have many conventions. Using WYSIWYG editors like Microsoft Word and Open Officer Writer work but are not very convenient. As Simons & Black (2009) point out, WYSIWYG editors are second wave technology. XLingPaper, on the other hand, uses third wave technology and is designed specifically for linguistic documents.

Linguists commonly face three obstacles in formatting papers. First, all examples are numbered in a paper. If during the writing process the author discovers a need to insert an example, then the numbering of all following examples and all references to those examples within the text need to be re-adjusted. This mechanical change can be both time-consuming and prone to error. Similarly, if the author decides to reorder some examples, then

[4]In addition to the first and second level sections shown in (1011), there are twenty four third level sections.

the numbering needs to be adjusted appropriately. XLingPaper provides an automatic way to facilitate such numbering and renumbering.

Secondly, linguists cite the work of other researchers using a standard citation format. This format functions essentially as an abbreviation or reference to the full citation entry which appears in the references section of the paper. The burden of maintaining consistency between citation and reference typically falls totally on the author. Many a reader has been disappointed to find a citation to a paper in the body of a paper for which there is no entry in the references section. XLingPaper provides an automatic means for a writer to maintain consistency; all citations in the text must have a corresponding entry in the references. Conversely, XLingPaper will include only those entries in the references section which are cited in the text. This latter characteristic implies that one can maintain one master list of references and merely include it in any given paper. Only those references actually cited in the given paper will appear in the references section.

Thirdly, linguists commonly use a set of abbreviations while glossing examples. They usually include either a list of the abbreviations and their definitions in a footnote, in a special front-matter page, or in a back-matter page. As for citations and references, the burden of maintaining consistency between the abbreviations used in the text and the abbreviations defined in the list typically falls totally on the author. Many a reader has been disappointed to find an abbreviation in a gloss for which there is no corresponding entry in the list of abbreviations. XLingPaper provides an automatic means for a writer to maintain consistency; the author can make it so all abbreviations in the text must have a corresponding entry in the list of abbreviations. Conversely, XLingPaper will include only those abbreviations in the list of abbreviations which are actually used in the text. This latter characteristic implies that one can maintain one master list of abbreviations[5] and merely include it in any given paper. Only those abbreviations actually cited in the given paper will appear in the list of abbreviations. By the way, XLingPaper also creates a hyperlink between the abbreviation in the text and the abbreviation in the list of abbreviations. Thus, a reader can click on the abbreviation and see what it means.

Furthermore, XLingPaper uses actionable data by marking-up linguistic documents in XML so one can produce multiple outputs from a single input.

[5]The starter master list which comes with XLingPaper is based on the Leipzig conventions given in Leipzig (2011).

As a brief example, the short section shown in example (12) is how a portion of one XLingPaper document appears in the XMLmind XML Editor.

When this portion is formatted using the default PDF output format of XLingPaper, it looks like what is given in example (13).

When this sample document is associated with a publisher style sheet designed for submissions to the *International Journal of American Linguistics* (see IJAL 2011), this portion will be formatted as in example (14). Notice that it is double-spaced and that the section numbers are in bold.

When this sample document is associated with a publisher style sheet designed for the journal *Language* (see Language 2011), this portion will be formatted as in example (15). Notice that it is single-spaced in a smaller font size, the section numbers are in regular type face, and that rather than using the word section, it uses the section symbol .

The three different outputs shown in examples (1315) are all produced without any changes to the main content of the XLingPaper document. This shows the power of using actionable data.[6]

XLingPaper works natively on Windows, Mac OS X, and Linux operating systems.[7] An XLingPaper document can be output in any of five formats:

Those who have used XLingPaper have said things like the following quotes:

To see some sample papers produced via XLingPaper, see Working Papers # 1, 3, 4, 7, 8, 9, 10, and 13 at http://www.sil.org/mexico/workpapers/WPindex.htm.

While XLingPaper is a powerful authoring tool for linguistic documents,

[6]For more on this, see the demonstration movie The Power of Actionable Data" on the XLingPaper web site, http://www.xlingpaper.org/?page_id=14.

[7]This is largely due to the fact that the XMLmind XML Editor runs natively on these operating systems.

the user of PAWS does not have to learn everything about XLingPaper before s/he can begin editing. This is because PAWS already formats the sections, interlinear examples, and tables so that the user only has to key in the glosses or other additional information requested. Further, adding additional description or interlinear examples or tables can be done by simply copying and pasting similar ones already in the document.

3.2.2 XLingPaper outputs. As we just mentioned, the output is in XLingPaper format, which allows editing within XML editors, and produces outputs in both HTML (W3C 1998) and PDF (ISO 2005) formats. PAWS automatically generates the HTML output for the user's convenience.

The HTML output page corresponding to the input page from examples (59) of section 3.1 is shown in examples (1617).

While PAWS only generates the HTML output, XLingPaper allows for multiple possible PDF outputs.

3.2.3 XMLmind XML Editor. We have found that using XLingPaper with the XMLmind XML Editor is the most convenient.[8] This is because the XMLmind XML Editor hides the XML from the user. In addition, the XLingPaper configuration files for the XMLmind XML Editor provide many other capabilities that makes it convenient for the author.

What the user sees within the XMLmind XML Editor for the first part of the same page we illustrated above in examples (59) is given in example (18):

The section level 2 item begins the portion, followed by several paragraphs of descriptive prose. It ends with two sets of interlinear examples. These have the four lines discussed in section 2.1. The editor allows the user to type in the information asked for in blue, as well as to add additional data and/or prose as appropriate.

[8] See http://www.xmlmind.com/xmleditor/ for more on this exceptional structured editor.

4 How it works. Having described what PAWS entails, we turn now to describing how it works.

4.1 The configuration files. The PAWS program consists of a shell or host program (called cabhab) which has an embedded web browser and also processes a set of configuration files. These files determine the user interface such as menu items, define sets of transforms to apply to the answer file, and also determine what shows in the embedded web browser.

There are several sets of configuration files.

4.1.1 Controlling the shell. There is a configuration file that controls what the cabhab shell shows the user and also how the answer file is transformed into various outputs.

For example, the menu items are defined as shown in (19).

The commands referred to by <item> elements are defined in the configuration file as given in example (20).

The message attribute refers to code in the cabhab program which is run when that command is invoked.

The set of transforms which are used to produce the various outputs are defined as illustrated in example (21). Only the one for the practical grammar output (in English) is shown. The others are similar.

4.1.2 Web page description. The second set of configuration files are the web page descriptions. The development of each web page used in PAWS was done by a non-programmer linguist, who wrote XML files describing the content of the page. Example (22) shows a portion of the XML description used to produce the page shown in (59).

An XSLT transform then produces the web page itself on the fly while the user is running PAWS.

4.1.3 Writer output description. The next set of configuration files

describes a given writer output.

The writer output was developed similarly: the linguist described the desired output in XML and then this XML was transformed into the XSLT that PAWS uses. For example, the portion of XML shown in (23) shows part of what ends up producing the kind of output given in examples (1617).[9]

4.2 Process for producing writer output. The overall process for producing the writer output is illustrated in example (24).

The linguist writes several writer description XML files, each of which must conform to a Document Type Definition (also known as a DTD; see W3C 2008). Each such file typically describes a section or a sub-section of the intended output. Each of these writer description files is then passed through an XSLT transform to produce the corresponding writer XSL file.[10] These are combined together in the master XSLT writer transform. The PAWS program then applies the answer file to this combined transform to produce the XLingPaper document output.

4.3 Choice of outputs. As noted above in section 3.1, through the use of actionable data, a single set of answers to questions and example data that the user enters can provide either a PC-PATR grammar file and test data for syntactic parsing or a choice of two styles of a grammar write-up.

This grammar write-up is an XML file in XLingPaper format, so it is ready for electronic publishing.[11] See examples in section 3.2.1 above.

[9]The use of the <content> element instead of plain PCDATA is a result of how the XSLT to transform this XML to XSLT was written: if one used PCDATA, the resulting XSLT would be incorrect.

[10]This is done only once during the development process. The individual writer XSL files are then included in the installation package as part of PAWS.

[11]One reviewer of an earlier version of this paper mentioned the Mukurtu repository system (see http://www.mukurtu.org/). One could easily put the output of the edited XLingPaper result of PAWS on such a site.

Another reviewer noted that a wiki rather than XML would be a possible target for a language community. We acknowledge this possibility with thanks. Since XLingPaper data is actionable, it is at least conceivable that one could create an XSLT transform that would convert the XLingPaper XML document to wiki pages. One could then post these pages to a wiki site, enabling the language community to refine the result.

4.4 Localization. The original grammar write-up included in PAWS closely follows the order and style of the user interface pages within PAWS, so it is a descriptive, pedagogical grammar comparative to English. This has been customized with the addition of the option of producing a practical grammar style write-up. Currently, the practical grammar is available in English and in Spanish. The process of translating the practical grammar output involves making a new copy of the XML writer description files, such as the one shown in (23), and translating the text.

SIL International uses such practical grammars in Mexico, written in Spanish. (See Hollenbach (1999) as well as Pickett, Black & Marcial (2001) and D. Persons, Black & J. Persons (2009) for examples.) Therefore, the practical grammar option has been translated into Spanish and the translation of the user interface is in process, to allow greater use throughout the Spanish-speaking world. Similar localization for other languages could be done in the future.

Additional customization is possible by the user (with configuration files to generate a transform), since PAWS employs XML technologies.

5 Conclusion. This paper has shown how using XML technologies to produce an expert system allows PAWS to meet multiple needs and produce multiple outputs. While originally developed for syntactic parsing, it can also be used to produce practical grammars, with the more complex instructions hidden from the user. This practical grammar option, especially coupled with localization into the national language, allows linguistically-aware native speakers of indigenous languages to partner with linguists in the task of documenting and describing the minority languages of the world and providing useful grammars for each language community.

Workshops on grammar writing are much more productive by beginning with PAWS. In such workshops, the participants all complete PAWS and then are taught how to edit the output using XLingPaper. From then on, the workshop can move to dealing with any language-specific phenomena not covered in PAWS. The participants can be taught how to search for the needed data, how to analyze it and then how to write it up in the grammar.

Though the grammar drafts output by PAWS have a template-like quality, they are simply a big head start on writing the grammar. After editing and enhancing the grammar using XLingPaper, a unique description of the particular language can be produced.

References

Black, H. Andrew. 2009. Writing linguistic papers in the Third Wave. *SIL Forum for Language Fieldwork* 2009-04. http://www.sil.org/silepubs/abstract.asp?id=52286 (30 November, 2011)

McConnel, Stephen. 1995. PC-PATR reference manual. SIL International. http://www.sil.org/pcpatr/manual/pcpatr.html (30 November, 2011)

Hollenbach, Barbara E. 1999. *Elaboracin de gramticas populares de lenguas indgenas: una breve guia (con referencia especial a las lenguas otomangues).* Summer Institute of Linguistics. http://www.sil.org/americas/mexico/ling/E001-GramaticasPopulares.pdf (30 November, 2011)

International Journal of American Linguistics. 2011. Style sheet. http://www.jstor.org/page/journal/intejamerling/style.html (1 December, 2011)

International Organization for Standardization. 2005. ISO 19005-1:2005. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38920 (23 June 2011)

Linguistic Society of America. 2011. *Language* style sheet. http://www.lsadc.org/info/pubs-lang-style.cfm (1 December, 2011)

Max Planck Institute for Evolutionary Anthropology, Department of Linguistics. 2011. Leipzig glossing rules: Conventions for interlinear morpheme-by-morpheme glosses. http://www.eva.mpg.de/lingua/resources/glossing-rules.php (1 December, 2011)

Persons, David A., Cheryl A. Black and Jan A. Persons. 2009. *Gramtica de zapoteco de Lachixo.* Mexico City: Instituto Lingstico de Verano. http://www.sil.org/mexico/zapoteca/lachixio/G040-LachixioGram-zpl.htm (30 November, 2011)

Pickett, Velma B. 1959. *The grammatical hierarchy of Isthmus Zapotec.* University of Michigan Ph.D. dissertation.

Pickett, Velma B., Cheryl Black & Vicente Marcial Cerqueda. 1998. *Gramtica popular del zapoteco del Istmo.* Juchitn, Oaxaca and Tucson: Centro de Investigacin y Desarrollo Binniz A.C. and Instituto Lingstico de Verano A.C.

Pickett, Velma B., Cheryl Black & Vicente Marcial Cerqueda. 2001. *Gramtica popular del zapoteco del Istmo.* 2nd edn. Juchitn, Oaxaca and Tucson: Centro de Investigacin y Desarrollo Binniz and Instituto Lingstico de Verano. http://www.sil.org/mexico/zapoteca/istmo/G023a-GramaticaZapIstmo-zai.htm (30 November, 2011)

Shieber, Stuart M. 1986. *An introduction to unification-based approaches to grammar.* CSLI Lecture Notes, 4. Stanford, CA: Center for the Study of Language and Information.

Simons, Gary F. and H. Andrew Black. 2009. Third Wave writing and publishing. *SIL Forum for Language Fieldwork* 2009-005. http://www.sil.org/silepubs/abstract.asp?id=52287 (30 November, 2011)

World Wide Web Consortium. 1998. HTML 4.0 specification. http://www.w3.org/TR/1998/REC-html40-19980424/ (23 June 2011)

World Wide Web Consortium. 2008. Definition of the XML document type declaration

from Extensible Markup Language (XML) 1.0 (Fifth Edition) on W3.org.
http://www.w3.org/TR/REC-xml/#dt-doctype (23 June 2011)