ARCHIVING AND SHARING LANGUAGE DATA USING XML

Simon Musgrave
Linguistics Program, Monash University

The reasons for using XML as the preferred format for archiving text data are powerful and have been clearly articulated in various places. Yet the rate of adoption of this strategy by linguists is low. In this paper, I suggest that one means of persuading linguists to change their practice may be to demonstrate that having well-defined XML data models can be of great advantage for current work, as well as for archiving purposes. Once a decision has been made to use XML, this implies that some means of automating the transfer of text to XML formats will be needed. With a small amount of additional work, this process can also be used as a means of transferring data between different software tools, allowing researchers within a team, or more widely, to share data and still use whatever software they individually prefer. I will illustrate these points with an example from my own work, where lexicon files from two tools currently being used (Toolbox, and the Spinoza Catalogue of Areal Linguistic Analyses) as well as heritage data in a FileMaker database have been integrated using XML formats and XSL transformations. I also consider what type of model might be appropriate for a lexicon for archiving purposes.

Introduction[1]

Language documentation produces large quantities of text data of various types, for example:

- transcribed language events
- associated annotations
- lexica / dictionaries
- analyses
- ethnographic notes

Not surprisingly, given this range of possibilities, there is no single standard software tool used by linguists, and there is probably not even a single tool used by any one linguist for all these tasks. Some of the tools used are proprietary software, and these produce file formats with limited portability. The use of Extensible Markup Language (XML, see

http://www.w3.org/XML/) for producing portable text data has been advocated by various authors and institutions (Barnes, 2006; Bird & Simons, 2003; Library of Congress, n.d.; Sperberg-McQueen & Burnard, 2002). There are clear advantages to using a format such as XML for the archiving of such data: XML is Unicode compatible, it stores data in a transparent and portable file format, and it forces the use of explicit markup to code information about the structure of the data.

However, these clear advantages have not led to a wide acceptance of such archiving practice in the linguistic community. Several reasons might be suggested that may have contributed to this situation, such as the lack of incentive (and time) for researchers to learn to use another piece of technology and the associated software tools. Another reason may be that researchers see any benefit that derives from the use of XML as being altruistic; that is, others may benefit from this additional work at some future time, but there is no benefit to a researcher in using XML now. In this paper, I would like to suggest that this position is not correct and that sensible use of XML can have immediate benefits for researchers, and, that if this is demonstrated to them, they may well be more inclined to include such desirable practice in their work.

The line of reasoning is as follows. Once a decision has been made to use XML as an archiving format, then some procedure to automate the addition of data to the archive is assumed. One option is to accept the XML output generated by various different software tools, but in some cases, such output is not wholly satisfactory (although the XML generated from, for example, an Access table is a significantly more portable than the Access table itself). A second option is to create transformation scripts to move data from the output formats to specially designed archive formats. In cases where a researcher (or a team of researchers) is using a variety of software tools, the second option may be unavoidable. This option allows the possibility that the archiving format can also be used as an interchange format. Scripts that extract data from that format can also be created thus enabling the sharing of data between different software, between different computing platforms and between researchers. In the remainder of the paper, I present a case study from my own research on the languages of Central Maluku (Indonesia), ending with some comments on the possibility for field linguists to work towards a common standard for modeling lexicons.

A case study

Three data models

One of the Central Maluku languages on which I have been collecting data is the language spoken on Nusalaut Island. Nusalaut Island is the smallest and most easterly of the Lease Islands, which are situated immediately to the south east of Ambon Island. Lexical data for this language are currently stored in two different formats associated with two different software tools. I also wish to be able to use a third software tool, with another data model, in my research on this language.

I collected a small amount of data myself on a brief trip to the island in 2003. In addition, I have tapes and transcripts of several interviews conducted by a member of the Nusalaut community with some of the few surviving speakers. This data has been entered into a text database and a lexical database in the Toolbox database software made available by the Summer Institute of Linguistics (SIL).[2] Toolbox uses a non-relational model for database files and all fields apart from the record marker can appear more than once in a record. This feature is valuable for representing data such as multiple alternative forms of a morpheme, or for giving more than one example of the use of a word.

A second source of data on the Nusalaut language is a series of word lists collected by Dutch colonial officials in the late nineteenth and early twentieth centuries. Three versions of this list exist for Nusalaut, collected at three different locations (Stokhof, 1980). These lists were entered into a FileMaker Pro database by Margaret Florey. FileMaker Pro supports relational models, but in this case a flat database structure was used. Repeating units are again exploited (for multiple examples), but a limit (two instances) is enforced by the data model.

Both of the data models introduced above contain many fields. In contrast, the data model associated with the third piece of software is rather simpler. This software tool is the Spinoza Catalogue of Areal Linguistic Analyses (SCALA), which is a Microsoft Access application. Musgrave (2002) describes a prototype of the application. The application uses a relational design, and therefore data stored in the morpheme lexicon table are much more restricted than that in the two sources introduced above; data included in a single table in the other two sources is split across several tables in SCALA. Across all three data models, some fields correspond (for example: \lx – lexeme [Toolbox], Headword

[FileMaker Pro] and Morpheme [SCALA]), and these provide the basis for constructing a unified data model.

A unified data model

A data model for archiving should be able to hold all the information coded in all the possible input formats. There should be no loss of data, or, at least, where data is not transferred, this should be the result of a deliberate decision. In other words, the structure of the archive format should subsume the structure of all the other formats being used.

I have initially used a model that is based on the structure of the Toolbox database and was developed as part of another project investigating techniques for displaying richly annotated language data in a web browser.[3] The significant difference in the data structure from the Toolbox model is that all elements or groups of elements that can be repeated in the Toolbox structure occur as children of wrapper elements in the unified format. For example, example sentences appear as groups of several fields (example text, glosses, text reference), and each of these is grouped as an <example> child within the <examples> element. Where the Toolbox model has separate fields for glosses and definitions in different languages (for example, \de – English definition, \dd – Dutch definition), the unified format has a single <gloss> element with a language attribute. The <gloss> element also has an additional attribute, type. This is used because it is desirable to restrict the language attribute to taking ISO 639 three-letter codes as its value, but if that was the sole attribute then information from two of the source formats would be lost. The FileMaker database has a field Morph-morph_Gloss associated with example sentences to store a morpheme-by-morpheme gloss, and SCALA has a field ContGloss for glossing grammatical morphemes using a controlled vocabulary.

Also, an additional level of structure is modeled in this lexicon. Each record contains groups of fields under the labels form, grammar, semantics, examples, cross-references, metadata, and notes.

Importing data

All three source applications have an XML export function; therefore importing data to the archival format involves only XML-to-XML transformations. FileMaker Pro is the least problematic source for this processing; the problems encountered in mapping this data model to the unified model, such as accommodating the Morph-morph_Gloss data and

modelling the information about occurrences in the Holle lists, were design problem rather than processing problems. However processing problems do arise with both of the other sources.

Because of its non-relational design, the fields in a Toolbox record do not have a fixed order. This does not cause problems in general given that the processing sequence of XSL transformations follows the tree structure rather than linear sequence. But in the case of sequences of associated records, such as examples or cross-references, problems can arise if the internal ordering of the group of records is not consistent. Toolbox has a feature that allows for the marker of a following field to be specified when designing a database, and using this provides some control over the order in which grouped fields will appear. My implementation takes advantage of this, but errors in handling grouped fields might still occur. The transformation used assumed that, if a full group of fields was used, they appeared in an assumed order. Processing began with, for example, an example text field, and then tested whether the following three fields matched the expected ones, testing one by one. If there was no match, data from the non-matching field was not processed as part of the group. If a full group in non-standard order was present, the data would also not be processed. An alternative approach would be to test each of the following three fields for a match to any of the expected fields. This would ensure that a full group out of order would still be processed, but would have the significant disadvantage that, where an isolated text example (that is, without translations or a reference) was immediately followed by another example group, data from the second group would be processed as associated with the first example. Gibbon and others (2004) also discuss this problem.

The problems that arise in processing the Toolbox format are a result of the non-relational design of the database. In contrast, the problems that arise with SCALA are a result of its rigorous relational design. Several pieces of information that are part of the archival data model do not occur in the Morpheme_Lexicon table of SCALA. The information is accessible via a foreign key in that table, the Language_ID number. For present purposes, this problem has been handled by hard-coding the necessary information. A more satisfactory solution will be to have a separate export from the SCALA table, which holds general information about languages, with an associated transformation that outputs the data needed as a set of <xsl:variable> elements. This set of variables can then be included in the stylesheet that defines the transformation from SCALA's XML output to

the archival format. The same problem, accessing information about language name and SIL code, occurs also in the case of the other two transformations, but as both of them deal only with data from a single language at a time, encoding the information as a variable is unproblematic. Two fields included in the SCALA table are not transferred to the archival format as their value is specific to the SCALA application. These are Characterization, which is a concatenation of data held in other fields stored redundantly as a convenience for SCALA, and AlikeButDifferent, which is used in SCALA to help the user eliminate duplicate entries from the lexicon.

Exporting data

Exporting data from the archival format to any of the other three formats is possible, but I concentrate here on the two transformations that are of practical value: from the archival format to Toolbox and from the archival format to SCALA. The FileMaker Pro format is treated here as a heritage format; new data will not be added to that database.

There is a clear benefit in being able to add all the data from the Holle wordlists to my Toolbox lexicon for Nusalaut. Any words that occur both in the lists and in the texts that I have collected will immediately be available for interlinear glossing, giving a very useful saving of time, and additional data will be included in any dictionary printed from the lexicon database. All Toolbox files are text files, with new fields marked by carriage returns followed by backslash characters, and new records marked by the occurrence of a record marker field. The export transformation for Toolbox therefore outputs text. Some data imported into the archival format has no corresponding field in the Toolbox format; for example, the Source_contact and PAGE fields. Three options are available for handling such data when exporting to Toolbox. Firstly, the data can be omitted from the exported file, although this is unsatisfactory, as data is lost. Secondly, new fields can be created in the export file and Toolbox will read these and automatically amend the database definition. Thirdly, the data can be exported as generic notes to \nt fields. Neither of these last two solutions is ideal; I prefer the third option as information can be recovered easily by searches restricted to \nt fields, and the database definition is not altered.

The possibility of exporting lexical data to SCALA is also useful. It is possible to carry out syntactic analysis in SCALA, which is not possible in Toolbox, but Toolbox's handling of morphological parsing is superior to

the SCALA capability in that area. SCALA has the possibility for importing Toolbox data from interlinearised texts, but after that import, it is still necessary to build the lexicon for the new material. This means editing a list of form-meaning pairings that the application detects while parsing the Toolbox records. Importing the Toolbox lexicon entire is much quicker. Access has the capability to import from an XML format, therefore the transformation in this case outputs XML to the same format that was the source format for importing data from SCALA. There is a potential processing problem in this transformation, as SCALA uses a very specific system for numbering morphemes in the lexicon table dependent on Language_ID numbers. For the moment, I assume that any import to SCALA will involve data from a single language only, and the numbering can be accomplished by making the Language_ID number available as a variable, along with the start number for the numbering of the new records. If data from multiple languages were to be handled, it would be necessary to use a two stage transformation process, with the first stage outputting a set of records sorted by language, and the second stage adding numbers based on a set of variables such as those mentioned previously.

Towards an archival data model

In the preceding section, I described working with an *ad hoc* data model for archiving. But as Barnes (2006) points out, individual users creating their own XML formats is a dangerous practice with serious drawbacks for maintenance and interoperability. Also, where widely accepted data models are used, the possibility that useful tools may become available increases. Therefore it is desirable to work towards a model of the lexicon that can be described within a standard framework. Several such frameworks are available for the description of lexicons, and I briefly consider two of these here, the Lexical Markup Framework proposed as a part of the work of ISO Technical Committee 37(SC 4) (ISO TC37, 2005), and the Open Lexicon Interchange Format (OLIF 2.0/2.1 – see http://www.olif.net).

The design of OLIF is more restrictive than that of LMF, which is possibly a reflection of its history. Its development as a tool for interchange of material for translation and localisation has led to cross references and transfer information being accorded a relatively privileged position. A specific disincentive to using the OLIF format comes from the structure of the key data categories required in every entry. Five of these must be present, and three are unproblematic: a canonical form, a

language specification and a part of speech label. However, the remaining two items of key data do raise problems. The first is called *Subject Field*, and situates each entry in a knowledge domain. The preferred list is one that is not suitable for work on languages from indigenous cultures; there is a possibility for the user to expand the list, but the value of the key data category is then lessened. The final key data category is *Semantic Reading*, intended to disambiguate entries that are identical for the other key data categories. Data to be entered here are to be taken from a designated standard for each language (such as *Roget's Thesaurus*). In the case of the data most field linguists would be coding, this procedure would not be independent from the data being encoded in the lexicon. The LMF format does not have these drawbacks, and I would suggest that it is a preferable framework for linguists to use in working towards more standard lexicon models.

The data model I used in my work was based on a Toolbox lexicon, and I suggest that one particular Toolbox format can usefully be taken as a point of departure. The lexicon structure associated with the Multi-Dictionary Formatter (MDF) in Toolbox is quite comprehensive and is also widely used by linguists; the possibility of easily producing a well formatted dictionary attracts many users to this lexicon model. The full list of fields in the database definition associated with MDF contains more than 100 fields. This list can be compressed in three ways. Firstly we can eliminate a set of specific fields intended for storing morphological paradigms (with labels such as singular and plural); these can be replaced by an already existing group of more general fields such as paradigm form and paradigm label. Secondly, many near-duplicate fields can be discarded by using a language attribute on a more general field (such as gloss) as described previously. A similar strategy is applied to the notes fields, of which the MDF model has seven, by providing a subject attribute. Finally, several fields can be omitted that are handled as part of the structure of an LMF lexicon entry. These include fields such as homonym number and sense number.

The result of this process is the list of forty six fields seen in the middle column of Figure 1. (Fields with a language attribute are marked with an asterisk, the notes field with a subject attribute is marked with a dagger.) The left-hand column of Figure 1 shows labels for the grouping of fields, used here, corresponding more or less to the groupings used in my first unified model. The right-hand column shows a tentative mapping to the high-level structural elements of an LMF lexicon entry. Figure 1 suggests

that it will not be too difficult to map the Toolbox MDF lexicon structure to an LMF lexicon, and that such an approach is worth pursuing.

Groups	*MDF Field names*	*LMF Structure*
Form	Lexeme	**FORM**
	Citation form	
	Alternate form	
	Underlying form	
	Phonetic form	
	Variant form(s)	
	Variant comment	
Morphology	Morphology	
	Paradigm form	
	Paradigm form gloss*	
	Paradigm	
	Paradigm label	
	Reduplication form(s)	
Grammar	Part of speech*	
	Restrictions*	
	Usage*	
Metadata	Source	
	Bibliography	
	Status	
	Date	
Cross Reference	Cross-reference	**LINGUISTIC FRAMES**
	Cross-reference gloss*	
	Lexical function value	
	Lexical function label	
	Lexical function gloss*	
	Main entry cross reference	
	Picture	
Semantics	Gloss*	**SENSE**
	Word-level gloss*	
	Definition*	
	Semantic Domain	
	Literally	
	Reversal	
	Index of semantics	
Examples	Example text	
	Example translation*	
	Example reference	
Additional Information	Encyclopaedic information	
	Scientific name	
	Notes†	
Etymology	Proto-form	
	Etymology gloss*	
	Etymology source	
	Etymology comment	
	Borrowed word	

Figure 1: First proposal for mapping Toolbox MDF lexicon to LMF lexicon.

Conclusion

While consensus over archival formats would be very desirable, it is also desirable to persuade linguists to adopt XML as a storage format for data. The example described above suggests that the expenditure of relatively little effort can give benefits even in the short term. Firstly, interoperability within the project is improved when data can be imported to the archive file from one format and exported to another format. Secondly, possibilities for interoperability outside the project are also improved as there is a well-defined data model that is available as a target for people who wish to share data: such people will be able to define transformations to and from their own data formats. Thirdly, the fact that linguists will be conscious of data design issues may help provide impetus for moves towards the creation of standards that may be adopted consensually by the community.

Endnotes

[1] Earlier versions of this material were presented to the Technology for Endangered Languages Forum at the University of Melbourne (August 2003) and the Digital Resources in the Humanities Conference (Cheltenham, September 2003). I am grateful to both of those audiences for helpful comments, to Margaret Florey for sharing her Nusalaut database with me, and to three anonymous reviewers for suggestions that led to an improved paper.

[2] SIL website: http://www.sil.org/computing/catalog/index.asp (Summer Institute of Linguistics, n.d.).

[3] This work is being carried out in collaboration with John Hurst (Computer Science, Monash University).

References

Barnes, I. (2006). *Preservation of word-processing documents*. Retrieved October 2, 2006, from http://www.apsr.edu.au/publications/preservation_of_word_processing_documents.html

Bird, S., & Simons, G. (2003). Seven dimensions of portability for language documentation and description. *Language, 79*, 557-582.

Gibbon, D., Bow, C., Bird, S., & Hughes, B. (2004). Securing Interpretability: The Case of Ega Language Documentation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, 1369-1372. Lisbon, Portugal

ISO TC37. (2005). *Working draft of ISO WD 24613:2005*. Retrieved October 16, 2006, from http://www.tc37sc4.org/new_doc/ISO_TC37-4_N130_rev5_LMF_19March05.pdf

Library of Congress. (n.d.). *Preferences in Summary for Textual Content.* Retrieved October 2, 2006, from http://www.digitalpreservation.gov/formats/content/text_preferences.shtml

Musgrave, S. (2002). Inducing Typological Generalizations in a Cross-Linguistic Database. Paper presented at *International Workshop on Resources and Tools in Field Linguistics, LREC 2002*, Las Palmas.

Sperberg-McQueen, C. M., & Burnard, L. (2002). *TEI P4: Guidelines for Electronic Text Encoding and Interchange*. Oxford, Providence, Charlottesville; Bergen: Text Encoding Initiative Consortium (XML Version).

Stokhof, W.A.L. (1980). *Holle lists, vocabularies in languages of Indonesia, vol. 3/4: Central Moluccas: Ambon (II), Buru, Nusalaut, Saparua (D-50)*. Canberra: Pacific Linguistics.

Summer Institute of Linguistics (SIL) (n.d.). Retrieved October 26, 2006, from http://www.sil.org