DEVELOPING TOOLS FOR CREATING – MAINTAINING – ANALYZING FIELD DATA [*]

Shoju CHIBA
College of Foreign Languages, Reitaku University
schiba@reitaku-u.ac.jp

Abstract

Field linguists encounter various problems when they create their own electronic text data and try to utilize them for their research. This paper tackles two of the most serious difficulties, namely phonetic transcription and structured data description, and shows that introducing XML and Unicode may best promote the integration of fieldwork and data creation.

1. The *fwtk* Project

This paper reports an on-going project to develop a fieldworkers' toolkit (*fwtk*, in short) for the textual study of endangered languages. *fwtk* has two main features described below:

- **Annotation compatible:** *fwtk* enables you to handle various linguistic

[*] This project is a sub-project of the Research Unit B03 of ELPR project ("Digitalization of linguistic data and information retrieval for the study of endangered languages", Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Scientific Research on Priority Areas (KAKENHI) No. 12039213). The project URL is: http://www.fl.reitaku-u.ac.jp/~schiba/fwtk/, from which the update information of the tools is available.

annotations, e.g. grammatical, semantic or pragmatic descriptions and phonetic transcriptions using IPA symbols.

- **Portability and Usability:** *fwtk* is easy to use and covers the basic tasks to handle field data. It enhances portability and usability of textual data to meet the special needs of field linguists, for example:
  - Rapid data creation and exchangeability: deriving formatted data from raw text; converting formatted data for Web publishing (HTML) or printing (RTF) or to other data formats
  - Advanced search function and tools for basic textual analysis

The prototype *fwtk*, which is now under development, is written in *Tcl/Tk* 8.3 and Microsoft *C#* and the target environment is Microsoft Windows2000/XP.

Before proceeding to the details of the *fwtk* project, let us briefly review the state of the arts of the current computer-aided field research.

2. Field Linguists' Dilemma

For researchers working with endangered languages the collection and the documentation of the materials constitute the most essential part of their study. As it has become very common that field researchers go for their fieldwork with a laptop computer, the working environment for this documenting task has changed dramatically: now they can make their electronic data in their very field, or they can even set to work on analyzing them immediately after their fieldwork. This change has helped to skip the most steps of digitalizing analog data, and led to a growing interest in publishing resources on the Web or on CD-ROM.

It is also worth mentioning that emerging machine-readable media include not only textual data but also multimedia data: there is even a trend to publish multimedia resources of field data on CD-ROM (Nathan 1999) or to construct a large-scale multimedia database of endangered languages with a series of multimedia annotation tools (*DOBES* Project, since 2000). Making machine -readable data for endangered languages and publishing them on the Web or on CD-ROM are thus becoming more and more essential among researchers' tasks.

Even if multimedia data is becoming popular, though, textual data yet remains to be the main resources of field linguistics. Indeed, many of the publications of ELPR project comprise solely textual data. Curiously enough, however, there have been few studies which published textual data in machine-readable form. Why?

Though the plain text is the simplest and the most popular machine-readable format, making a field-linguistic textual database (i.e. a corpus) in text format is not actually an easy task. The main difficulty comes from the fact that transcribing the language involves the use of special characters like IPA symbols or diacritical conventions, which few encoding systems are implementing. [1]

This difficulty leads to a use of word processor, where various characters (including phonetic symbols) can be stored by choosing different fonts. You can also indicate different levels of description easily by different font styles, for example, or you can tabulate the gloss and arrange it nicely beside the original text. This would be OK, as far as you intend to publish the data on *paper*. But once you wish to construct a textual database on the basis of your data or if you want to put the data into various textual analyses or publish it on the Web or on CD-ROM, it causes serious problems that the files are "domain-specific" (Antworth & Valentine 1998:172) and thus not application-free.

Linguistic analysis of field data yields various descriptive information or transcriptions. Word processor can arrange data fragments visually, but it doesn't help to distinguish consistently the levels of description (for example, gloss and transcription). Therefore the conversion from word processor file to text file mixes linguistic description with the original data and simply messes up the whole material.

Here is the dilemma: A word processor has powerful visual facilities for printing, but it lacks flexibility and re-usability that are typical of machine-readable data. Text data is application-free and has thus maximum flexibility but it suffers from the lack of expressive power. Worse still, both formats lack an explicit way to store structured data, i.e. data with different levels of descriptive notations (see Figure 1).

---

[1] There have been proposed alternative ways to represent IPA in plain text (SAMPA project since 1987; Kirshenbaum 2001, for example), but they require additional training to read and write the data and skills in processing texts.

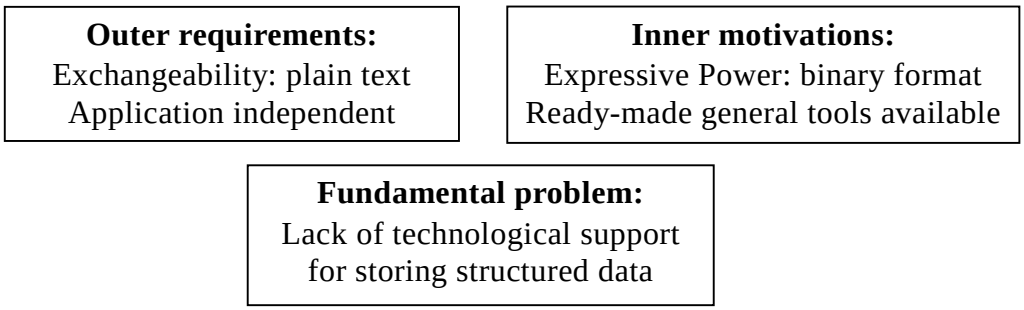

Figure 1: Digitalization Dilemma of Field Linguists

3. Designing *fwtk*

3.1. Motivations and Solutions

As we have seen in the previous section (§2), there are two major problems that field linguists may encounter when they create textual data, namely, (1) there are few straightforward ways to transcribe phonetic symbols; and (2) once you decide to add some data to the texts, for example phonetic transcription, grammatical explanation, notes on semantics, etc., the original texts may easily be intermingled with the data you have added. It is often true that the more annotations you add the more difficult it becomes to retrieve original lines.

The prototype *fwtk* implements two recent technologies to solve the problems mentioned above: Unicode and XML.

3.2. Using Unicode

Unicode (Graham 1999; Unicode Consortium 2000) is a new character encoding standard published by the Unicode Consortium and is "designed to include all of the major script of the world in a simple and consistent manner" (Graham 1999:75). Unicode is thus fully multilingual and includes full IPA symbols.

fwtk takes full advantage of Unicode and store the data in **utf-8**, a transformed format of Unicode suitable for network exchange.

Because the structure of Unicode is fundamentally different from the existing

encoding systems, the compatibility issue may be a potential problem for the use of Unicode. (For example, there are application programs that are localized and can't properly handle Unicode.) To avoid any loss of the data in an Unicode-unaware program, *fwtk* has an export tool which converts Unicode characters to their Numeric Character Reference equivalents (notation: &#x*nnnn*; where *n* stands for hexadecimal number of the Unicode codepoint). Furthermore, in searching tools and text view function, *fwtk* treats Unicode characters and their numeric reference counterparts as equivalent expressions.

The function of IPA input of *fwtk* is described in § 3.6.1.

3.3. Using XML

XML (eXtensible Markup Language) is a modern markup method to express data structure in plain text format, which is intended to be simple and explicit to process. It is *extensible* in the sense that one can define his/her markup tags according to his/her needs. XML is proposed by World Wide Web Consortium and the current version is 1.0 (2. edition, issued in October, 2000).

fwtk implements XML so that the data is processed and stored in XML format. It also provides import/export functions, which are summarized below (Figure 2):

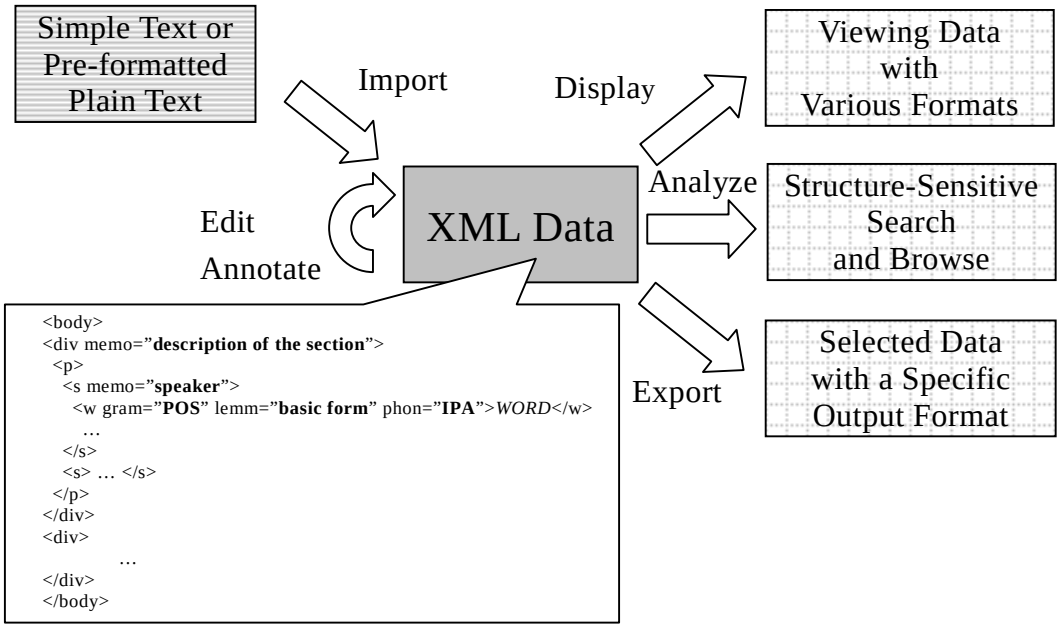

Figure 2: Target Data Structure of *fwtk*

3.4. Structural Features of Field Data

Once you decide to implement XML, then you can freely define, how complex the data will be: XML is *eXtensible*. Structure of field data varies greatly according to the contents, but as far as the textual data is concerned, it falls into two main categories:

- **Word list**, which can be expressed in simple 2-dimensional table
- **Phrase list or sequential text** (narrative text, for example), which is more complex and can't be expressed by a simple table-like format

Because a spreadsheet-type program (like Excel) is well applicable to the former type of data, it is the latter type of textual data that *fwtk* project is particularly focusing on.

Now the question is: to what extent a normal field-linguistic data should be complex. We need to distinguish two levels of complexity:

- **Structural level**: how each text element is arranged and grouped. Usually, one wishes to distinguish, for example, *Paragraph*, *Sentence*, and *Word* level.
- **Descriptive level**: how each text element (*Word*, for example) can be linguistically described. For linguistic purposes, we need to annotate *grammatical description* (e.g. Part of Speech, Stem/Root, basic form (Lemma), Syntactic Role), *phonetic transcription*, and other descriptive memos, *etc*.

With the distinctions mentioned above in mind, in the prototype program we restrict the target data structure as simple as possible and represent it in XML format as follows:

- Elements (which indicate the structural level information)
  - <body>*text body*</body> … shows the range of main text
  - *section* ... shows the main divisions that the main text embodies
  - *paragraph*
  - <s>*sentence*</s>
  - <w>*word*</w>

- Attributes (which indicate the descriptive level information and annotate elements)
  <w phon="*phonetic transcription*" lemm="*basic form*" gram="*grammatical description*" memo="*descriptive note*">*word*</w>

The following Figure 3 shows a sample of the target data structure of *fwtk*:

```
<body>
<div memo="description of the section">
    <p>
        <s memo="speaker">
            <w gram="POS" lemm="basic form" phon="IPA">WORD</w>
            …
        </s>
        <s> … </s>
    </p>
</div>
<div>
    …
</div>
</body>
```

Figure 3: Target Data Structure of *fwtk*

3.5. Maximizing integrity

Field linguists, especially working with endangered languages, make a great effort at managing the data they collect: actually many of them document, analyze, and publish the data by their *own* efforts. This means that once the data is digitalized, they will have to master different tools according to the different phases of their job. This often causes difficulties, as the programs differ in their user interface and functionalities.

Here is the reason why an *integrated* workbench designed for field linguists is needed: there should be a package which covers most of their basic tasks and shows the maximum integrity with regard to the user interface and functionality.

Specifically, our *fwtk* includes the following tools:

- Basic tools for different phases of the field study
  - A Tool for Data Creation (Editing)
  - A Tool for Data Management (Maintenance)
  - A Tool for Data Analysis (Search and Export)
- Tools repeatedly used with the toolkit
  - IPA Soft Keyboard for Data Creation and Data Analysis
  - Basic Search function which implements Unicode characters and their character reference counterparts

The basic tools should be maximally *integrated*, so that one can call any basic tools when you want and proceed swiftly from one tool to another. This is essential, for fieldworkers often do the three tasks (editing, analyzing, maintaining) either in succession or even simultaneously.

3.6 Functions and Interfaces

Now let us examine closely the functions the individual tools included in *fwtk*. Three functions will be pointed out here: (1) IPA software keyboard, (2) structure-sensitive text view, and (3) structure-sensitive search.

3.6.1. Software Keyboard for inputting IPA symbols

The IPA inputting system of *fwtk* has 3 ways of character display, namely:

- Tables implementing IPA charts (IPA 1999)
- Character list sorted by:
  - IPA number, or
  - Unicode order

This IPA keyboard dialogue can be run anytime from any main tools of *fwtk*. The Figure 4 shows how the software keyboard looks like:

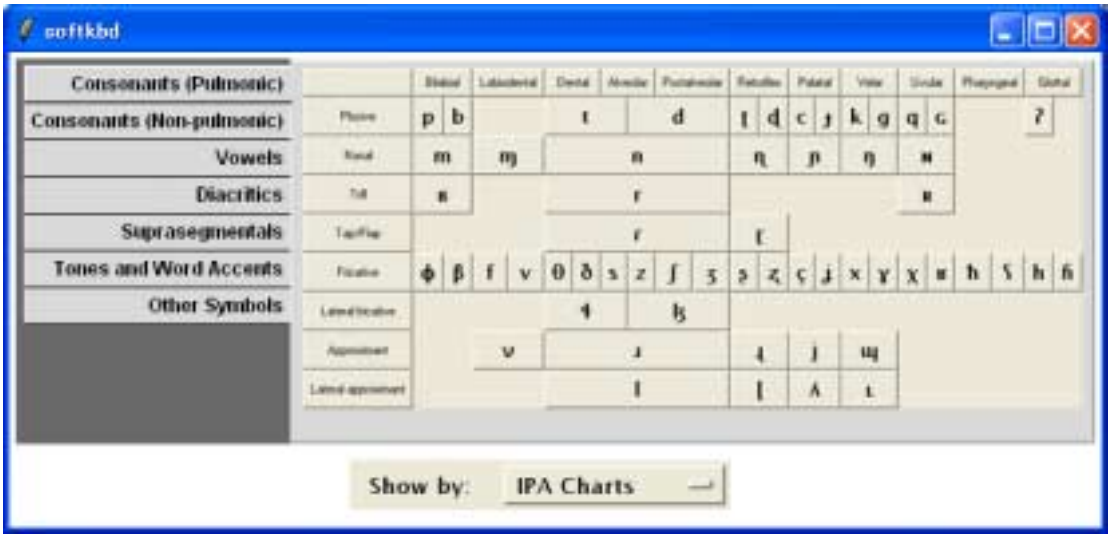

Figure 4: IPA Software Keyboard (Prototype *Tcl/Tk* Version)

3.6.2. Structure-Sensitive Data View

The tool for editing data in *fwtk* has the following 3 display modes:

1. XML Source View, which
   - shows the full XML data, and
   - is suitable for editing text and XML tags (including elements for structural level information and attributes for descriptive level information)
2. Data View, which
   - selects a level (paragraph, sentence, or word level), and
   - lists the elements and their attributes on the level selected
3. Main Text View, which
   - shows the text without XML tags, and where
   - by using mouse one can browse Word-level attributes

Note that the user interface for editing XML is now experimental and the most part of the XML Source View will be rewritten in the future version.

3.6.3. Structure-Sensitive Search and Textual Analysis

As is briefly mentioned in §3.2, the search function of *fwtk* supports Unicode and its character reference notations. However, this isn't actually sufficient for linguistic analysis. *Fwtk* enriches the search function with the following features:

- Implementation of regular expression
- Structure-sensitive search method designed for each display mode
  - to keep the output format identical with the original text displayed
  - to enable to specify the search field by Element/Attribute

The latter feature is particularly important to successfully skim off a pattern on a specific descriptive level and arrange it for display.

Further, on the Main Text View two types of data display method are available.

- Enhanced *grep*, which
  - specifies a string and the descriptive level where it occurs, and
  - searches the string and shows the sentence(s) which include it
- Enhanced *kwic*, which
  - searches a string (with a particular attribute value), and
  - shows it with a range of context

3.7. Future Development

Because *fwtk* is still under development, there still remain many functions unincorporated into it. There are also several technical/general problems to be solved during the development of the application. Among such basic problems, Unicode and XML related issues and language-specific customization problem may be particularly worth mentioning here.

Firstly, full implementation of Unicode is technically very requiring. Here is a list of the technical problems we are confronting:

- Because the implementation level of Unicode is varying between OS versions or between programming languages, developing an Unicode-ready program is still a highly complex task. For example, A *Tcl/Tk* program featuring Unicode slows down on older versions of Microsoft Windows family (Windows9x).
- In Unicode, combined characters with multiple diacritical marks are open-ended.
  - Software must incorporate some “dynamic composition” functionality to display/print the Unicode combined letters properly.
  - There are difficulties to search characters with a particular diacritical mark, because any numbers of other diacritics can break in between the target string.
- Unicode has several special conventions to assign a code to glyphs, so that one code can represent different glyphs/string sequences.
  - Several letters are squeezed into one code and can’t be differentiated in terms of the character code (“Unification”).
  - There can be multiple ways to express a particular character: one character can be represented by a single character or a combined character string.
  - There are wide varieties of symbols that have separate character codes assigned and look nevertheless similar. This easily leads to the inconsistency of the characters used in the data.

Secondly, the prototype *fwtk* treats XML on rather unsystematic basis and doesn’t support any customized tags. Though XML is in itself a simple text data, the resulting XML data is structurally very intricate, hence it takes longer time to process XML data than to handle plain unstructured texts. Thus there is always a room for improving efficiencies of XML data parsing by implementing DOM, for example. Parsing algorithm is to be refined particularly in the following terms:

- Search mechanism
- XML validation (structure check)
- Conversion to other data formats

Finally, many language-specific customizations remain unimplemented. For example, a sub-program that sorts the search results by a language-specific order would be desirable to analyze the field data.

4. Conclusion

This paper introduced an Unicode & XML compliant toolkit designed for field linguists and examined how these two technologies, when tightly united, can facilitate the creation, maintenance and the analysis of field linguistic data. The Figure 5 below sums up how Unicode and XML cooperate in doing different field-linguistic tasks.

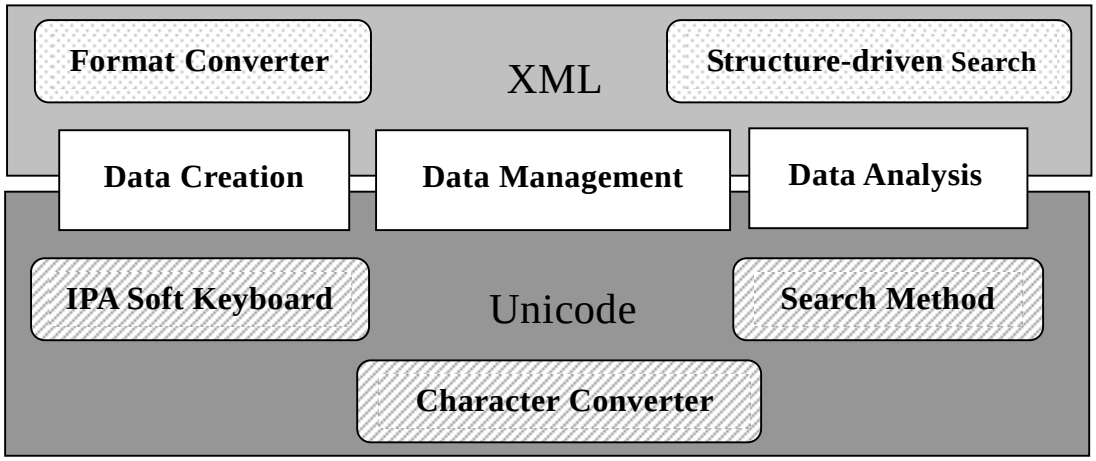

Figure 5: How two key technologies facilitate the field study

Field linguists have suffered from the lack of good methodology how to make digital resources: they needed a variety of special characters/symbols; they also needed to distinguish texts from glosses, phonetic transcriptions or other descriptive notations. Many of the researchers had no choice but to satisfy themselves with the use of word processors, while understanding well the importance of adhering to plain text format, because using word processor decreased the flexibility and (re-)usability of the data they created.

Now once you choose Unicode & XML, the non-binary plain text format will become finally a good alternative to a binary word processor format. Bringing

together the various functions needed for field linguists, a small but well field-oriented software toolkit like *fwtk* will make the individual researches more efficient, simplify the process of the publication of the data on various media (on the Web or CD-ROM), hence facilitate the vigorous exchange of the data between researchers.

Reference

Antworth, Evan L. and J. Randolph Valentine 1998 “Software for doing field linguistics.” Lawler, John and Helen Aristar Dry (eds.) *Using Computers in Linguistics: A Practical Guide.* London: Routledge. pp. 170—196.

DOBES (Dokumentation der bedrohten Sprachen) Project: http://www.mpi.nl/DOBES

Graham, T. 1999 “Unicode: what is it and how do I use it?” *Markup Languages: Theory and Practice* 1: 75-102.

IPA (International Phonetic Association) 1999 *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge: Cambridge University Press.

Kirshenbaum, Evan 2001 “Representing IPA phonetics in ASCII.” Manuscript available at: http://www.kirshenbaum.net/IPA/ascii-ipa.pdf

Nathan, David 1999 “Tools for communities, tools for linguists: new technologies for endangered languages.” Paper read at the ICHEL Colloquium on Endangered Languages, Oct. 27, 1999, Tokyo University, Japan.

SAMPA (Speech Assessment Methods Phonetic Alphabet): Computer Readable Phonetic Alphabet. http://www.phon.ucl.ac.uk/home/sampa/home.htm

Unicode Consortium 2000 *The Unicode Standard Version 3.0.* Reading, MA: Addison Wesley Longman. (For update information, consult: http://www.unicode.org)