

Developing Tools for Creating-Maintaining-Analyzing Field Data

Shoju CHIBA
Reitaku University, Japan
schiba@reitaku-u.ac.jp



2002-11-25

<http://www.fl.reitaku-u.ac.jp/~schiba/fwtk/>

1

Topics

- ELPR Research Unit B03
- The *fwtk* Project
 - Backgrounds
 - Key Technologies
 - Overviews
 - Future Developments and Problems

2

ELPR and the Research Unit B03

- Research Unit B03
 - “Digitization of linguistic data and information retrieval for the study of endangered languages”
 - Visited numerous leading institutions (in Japan or abroad)
 - Gathered information on the current states of applying IT technologies to the linguistic researches
 - Formed several projects to design tools for field linguists

3

What we've learnt and realized

- Growing interests in
 - making Multimedia data
 - web-publishing data
 - sharing and data
- Urgent needs of
 - freely usable fonts
 - tools for data creation
 - tools for searching data

} *fwtk*

4

The *fwtk* Project

- Develop a fieldworkers' toolkit (*fwtk*) for the research of endangered languages
 - Enable to handle various linguistic annotations (grammatical description, phonetic transcription, etc)
 - Focusing on portability and usability

5

Roadmap

- Prototype software under development
 - Windows2000/XP as a target environment
 - Programming language: Tcl/Tk 8.3 and Microsoft C#
- Project URL:
<http://www.fl.reitaku-u.ac.jp/~schiba/fwtk/>
- A Manual (together with the software) will be published as a publication of ELPR project

6

Technical Specifications

- Implementing Unicode (utf-8 transformation format) which enables transcription using IPA
- Using XML (eXtensible Markup Language) to describe structured data
 - Distinguish *Phrasal*, *Sentence*, *Word Level*
 - Enable to add various linguistic descriptions on each structural level

7

Technical Specifications (continued)

- IPA input support (software keyboard)
- Tools designed for textual analysis
 - Output format: KWIC (KeyWord In Context) and grep (shows the sentence matched)
 - Structure-sensitive search: specifying the field by main text or/and annotation(s) specified

8

Backgrounds

- Growing interest in *multimedia*
- Textual data still very important
- Word processor vs. plain text
 - Word processor: best for printing, but not “processing”
 - Application-dependent (“domain-specific”) format

9

Backgrounds (continued)

- How can we avoid word processor?
 - How to express IPA or other symbols? – *encoding*, *input method*
 - How to distinguish different levels of description? – *original text*, *gloss*, *transcription*, *translation*

10

Field Linguists' Dilemma

Outer requirements:

Plain Text

Exchangeability
Application independence

Inner motivations:

Binary Format

Expressive power
Availability

Fundamental problem:

Lack of technological support
for making structural data
and utilizing it

11

Motivations

- Enriching editing environment for field linguists
 - Use of IPA symbols in the texts
 - Relevant encoding scheme
→ **Unicode**
 - Uniform way to store linguistic descriptions (grammatical/semantic/phonetic...)
 - Needs of a framework for structure description
→ **XML**

12

Solutions

- Using Unicode for storing various characters
- Using XML for storing data structure
- Providing tools for linguists
 - IPA Input support
 - Structure-sensitive search: searching sentences/paragraphs on the basis of meta-linguistic annotation

13

Using Unicode

- Advantages of Using Unicode
 - Multilingual
 - Full IPA symbols included
- Using utf-8 as the data format
 - Simple text format fully compatible with Unicode
 - Character codes of the basic characters preserved
 - suitable for data exchange via network

14

Using Unicode (continued)

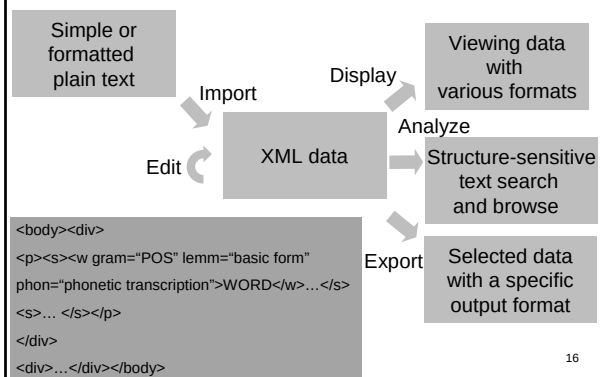
- Potential problems: Unicode incompatible with the other encoding schemes
 - Using Unicode incompatible tools

Solutions in *fwtk*:

- Converting Unicode characters with Numeric Character Reference (&#xnnnn;)
- Handling Unicode and its numeric references uniformly in search functions

15

Using XML (Extensible Markup Language)



16

Nature of Field Data

- Field data: structured
 - Word list
 - ← 2-dimensional: spread-sheet program (like Excel) available
 - Phrase list or narrative text
 - ← more complex: simple table-like format is not enough

17

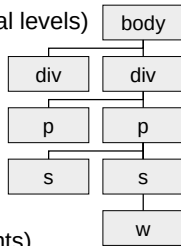
Target Data Structure

- Complex data
 - Structural Level: *Paragraph*, *Sentence*, *Word*, etc.
 - Descriptive Level: Annotations
 - grammatical annotation (*Part of Speech*, *Lemma* (basic form), *Stem*, *Root*, *Grammatical Role*)
 - phonetic transcription
 - descriptive memo, etc.

18

Target Data Structure (continued)

- Elements (indicate structural levels)
 - `<body>text body</body>`
 - `<div>chapter</div>`
 - `<p>paragraph</p>`
 - `<s>sentence</s>`
 - `<w>word</w>`
- Attributes (added to elements)
 - `<w phon="phonetic transcription" lemm="basic form" gram="grammatical description" memo="a note">WORD</w>`



19

Target Data Structure: Sample

```
<body>
<div>
  <p>
    <s memo="speaker A">
      <w gram="POS" lemm="basic form" phon="IPA">WORD</w>
    </s>
    ...
  </p>
</div>
<div>
  ...
</div>
</body>
```

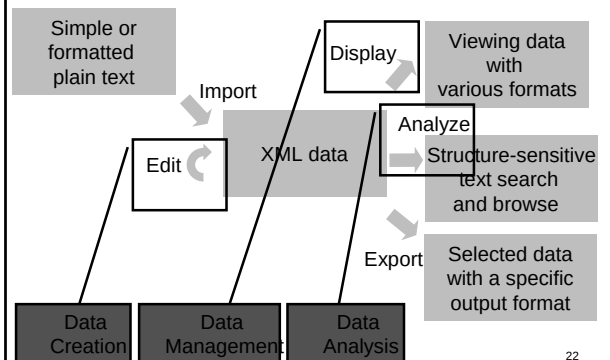
20

Enriching the Working Environment

- Basic tasks should be included in one package (as one *toolkit*)
- Basic tools should be "Integrated": each tool should include
 - same utilities
 - same user interface
- Utilities
 - IPA software keyboard available anytime
 - Editing
 - Searching

21

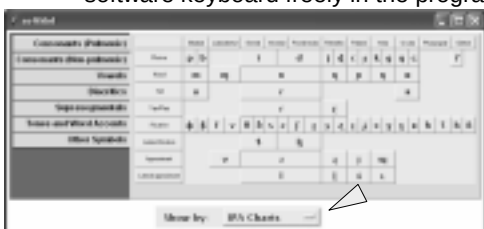
3 Basic Tools of *fwtk*



22

Software keyboard for inputting IPA symbols

- 3 Display modes
 - Tables implementing IPA Charts
 - Lists sorted by IPA number / Unicode order
- Run anytime, anywhere: you can call the software keyboard freely in the program



23

Structure-Sensitive Data View: 3 Modes

- XML Source View
 - Shows the full XML data
 - Suitable for editing text and XML tags (elements and attributes)
- Structural (Data) View
 - Selects a structural level
 - Lists the elements and their attributes on the level selected
- Main Text View
 - Shows the text without XML tags
 - Browses Word Level attributes with mouse

24

Structure-Sensitive Search/Textual Analysis

- Basic requirements for text search
 - Regular expression
 - Elaborated search method for each display mode
 - Keep the output format identical
 - Enable to specify the search field by Element/Attribute
- Tools for detailed textual analyses on the Main Text View
 - Enhanced *grep*
 - Enhanced *kwic*

25

Remaining Problems

- Full Implementation of Unicode is still under way
 - Implementation level of Unicode varies: Developing Unicode-ready programming is still a highly complex and demanding task.
 - “Dynamic Composition” challenge: combined characters with multiple diacritical marks are open-ended. → *Printing/Displaying/Searching*
 - Dazzling principles of Unicode
 - Unification
 - equivalent character sequences
 - Visual ambiguity
- Refining algorithm: search mechanism and XML parsing process still need to be improved.

26

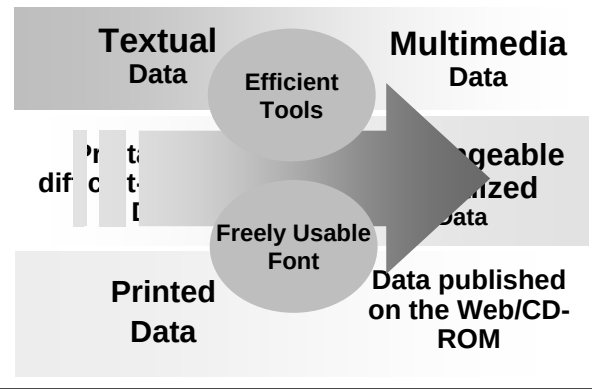
Things to Do

- Customize data structure or Implement popular data formats: *TEI on XML*, for example
 - Conversion
 - Data creation
 - Export
- Uniform ways to access data
 - Using XML parser for a more efficient data parsing
 - Using XSLT to extract/display data
- Establish a way to customize functions or to add plug-in tools

27

Perhaps
Most Important (?)
Give this software
a nice name!

Summing Up: a Major Drift



Conclusion

29