III International Conference LREC 2002

Building an Integrated Environment for Field Data Creation-Maintenance-Analysis

Shoju CHIBA
Reitaku University, Japan

On ELPR Project

- "Endangered Languages of the Pacific Rim" (ELPR, 1999-2003)
  - 4 year-long academic project supported by the Japan Ministry of Education
  - Over 100 researchers in Japan are participating in the project
  - includes 7 research Units, (of which 4 [A01-A04] are regional studies, 3 [B01-B04] handle theory & information processing matters)
  - Research Unit B03 "Digitization of linguistic data and information retrieval for the study of endangered languages"
    - A project on the prototyping of field linguists' toolkit (*fwtk*)

Aims of *fwtk* Project

- developing a fieldworkers' toolkit (thus *fwtk*) for the text corpora of endangered languages
  - Enable to handle various linguistic annotations, including grammatical or semantic description, phonetic transcription
  - Focusing on portability, usability, and field linguists' needs:
    - easy to install to a laptop PC
    - tools for rapid data creation
    - assisting data exchange: import/export formatted textual data; Web-based publishing
    - search function and tools for basic textual analysis

Roadmap

- Prototype software under development
  - Windows2000/XP as a target environment
  - Programming language: Tcl/Tk 8.3
- Project URL (now under construction):
  http://www.fl.reitaku-u.ac.jp/~schiba/fwtk/
- Research workshop by B03 Research Unit will be held in this autumn.
- A Manual (together with the software) will be published as a publication of ELPR project in the spring of 2003

Motivation **(WHY "UNIFIED"?)**

- Word processor vs. plain text
  - Word processor: best for printing, but not "processing"
    - Application-dependent format
  - Avoiding Word processor?
    - How to put IPA or other symbols? *Encode*, *Input*
    - How to distinguish different levels of description? *Text*, *gloss*, *transcription*, *translation*
- Enriching editing environment for field linguists
  - Rapid data creation
    - Uniform way to store meta-data (grammatical/semantic/phonetic…)
    - Using IPA transcription in the text
      - Relevant encoding scheme
      - Input support
  - Handling structured data
    - Needs of a structure description framework
    - Structure-sensitive search: searching sentences/paragraphs on the basis of meta-linguistic annotation

Why "integrated"?

- Integrated Toolkit: Tools for
  - Data creation
  - Data management
  - Data analysis
- Rapid: one package is enough to do basic tasks
- Tools working within the
  - IPA softkeyboard is available anytime
    - Editing
    - Searching

Technical Specifications

- Implementing Unicode (utf-8 transformation format is supported by the programming language Tcl/Tk)
- Input support for IPA transcription
- Using XML (eXtensible Markup Language) for structured data description
  - Distinguishing *Phrasal*, *Sentence*, *Word* Level
  - Enabling to add various linguistic annotations on each structural level (e.g. grammatical, phonetic, semantic description, or simple memo)
- Tool designed for textual analysis
  - Output format: KWIC (KeyWord In Context) and grep (show search result line by line)
  - Structure-sensitive search: specifying the field by main text or/and annotation(s) specified

7

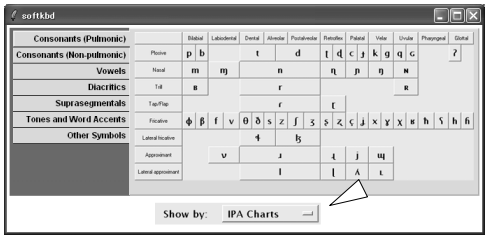
Using Unicode

- Advantages of Using Unicode (utf-8)
  - Multilingual
  - Full IPA symbols are includes
- Potential problems: Unicode incompatible with the other encoding schemes
  - Using Unicode incompatible tools
  - Publishing data on WWW
  - Converting Unicode characters with Numeric Character Reference (&#x*nnnn*;)
  - Handling Unicode and its numeric reference in searching tools

8

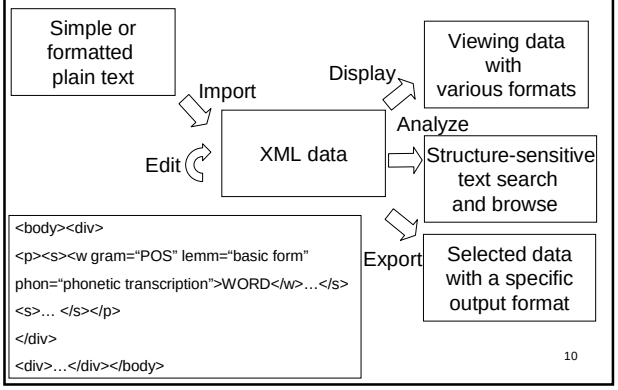
Soft keyboard for inputting IPA symbols

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



9

Using XML (Extensible Markup Language)



10

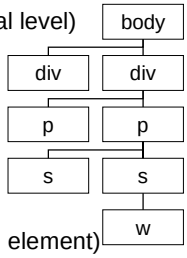
Nature of Field Data and its structure

- Field data: structured
  - Word list
    - ← 2-dimensional table: spreadsheet program
  - Phrase list or narrative text
    - ← more complex: simple table-like format is not enough
- Complex data
  - Paragraph, Sentence, Word Level
  - Annotation
    - grammatical (Part of Speech, Lemma (basic form), Stem/Root) annotation
    - phonetic transcription
    - memo

11

Target Data Structure

- Elements (indicate structural level)
  <body>**text body**</body>
  chapter
  paragraph
  <s>**sentence**</s>
  <w>**word**</w>



- Attributes (Annotation to an element)
  <w phon="phonetic transcription" lemm="basic form" gram="grammatical description" memo="a note">**WORD**</w>

12

Target Data Structure: Sample

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

13

Structure-Sensitive Data View

3 display modes

1. XML Source Editor
   - Show the full XML data
   - Edit Text
   - Add Element(s)/Attribute(s)
2. Detailed Data View
   - Select a Level
   - List the Elements and Their Attributes under the Level selected
3. Main Text View
   - Show the Text without XML tags/Attributes
   - Use mouse to browse Word-level attributes

14

Structure-Sensitive Search/Textual Analysis

- Basic requirements for text search
  - Implement Regular Expression
  - Design a elaborated search menu for each display mode
    - Keep the format identical
    - Enable to specify the search field by Element/Attribute
- *grep* extended
  - Specify a string and select an Element/Attribute where it occurs.
  - Search the string and show the sentence(s) which include it
- *kwic* extended
  - Search a string (with a particular attribute value) and show it with a range of context

15

Remaining Problems

- Full Implementation of Unicode is still under way
  - Varying levels of implementation (OS, programming language): Developing Unicode-ready programming is still a highly complex task.
    - Tcl/Tk program featuring Unicode slows down on **Windows9x** platform
  - "Dynamic Composition" challenge: combined characters with multiple diacritical marks are open-ended.
    - Printing/displaying
    - Searching
  - Dazzling principles of Unicode
    - Unification
    - Visual ambiguity (*Unicode Standard* p.17)
    - equivalent character sequences
- Refining algorithm: search mechanism and XML parsing process still need to be improved.

16

Things to Do

- Extensions of the current software design
  - Customize data structure or Implement popular data formats: TEI on XML, for example
    - Conversion
    - Data creation
    - Export
  - Uniform ways to access XML data
    - Using XML parser for a more efficient data parsing
    - Using XSLT to extract/search data
  - Establish a way to add new tools/dialogues or register plug-in tools

17

Conclusion

18