CJK DOCP Recommendation: CJK multilingual linguistic markup
CJKDOCP Text Corpus Exchange Formats

Date: 1995-12-06
Author : Jing-Shin Chang
(SIGMT, ROCLING)
Revision : 1.2.3
File: CJKDOCP.xf.1.2.3.doc

0. Purpose of the Exchange Formats

0.1. Motivation

The purpose of the current recommended exchange formats is to propose a set of simple formats for corpus encoding and exchange. The design goal of the encoding formats is to use a very restrict subset of the SGML (*) markup tags to create corpora for common text types of interest to the CJKDOCP members. (* SGML: "Standard Generalized Markup Language".)

The intention is to make the encoding scheme as conformant as possible to the TEI (Text Encoding Initiative) P1 guidelines, possibly with some degree of extension and simplification on the TEI.1 encoding scheme to meet the special requirements of the CJKDOCP community.

For example, the header part of a TEI.1 document is simplified extensively so that only essential information is retained. As another example, since the encoding scheme for the proper name of a person is not well defined in the DTDs for TEI.1, we make use of some tags and attributes for encoding proper names. The degree of conformant, however, should enable us to convert a corpus easily when a more precisely specified TEI guideline is released.

In addition, easy encoding, decoding and immediate availability are also important principles to the recommended formats. Therefore, we will not assume the availability of a general purpose SGML parser for the corpus users.

In designing the formats, we hope that all the available information in the existing corpora can be carried over to the exchange formats as much as possible to avoid duplicate works; however, since corpus tagging could be time consuming for the corpus contributors, we hope that the corpus users can share the load of corpus tagging if some of the tagging information is not available in the original corpora. We also encourage the release of any derived works from the original works that contain extra information after further processing.

0.2. Simple Encoding Convention Used for Corpus Tagging

An "element" (e.g., a paragraph, a sentence, or a word) is, in general, enclosed by a pair of "start tag" and "end tag" in an SGML text. For example, a book title "Advanced Unix Programming" can be tagged as:

```
<title>Advanced Unix Programming</title>
```

where "<title>" is the "start tag" for the element of type "title", and </title> is the corresponding "end tag". In SGML convention, the end tag is acquired by adding a '/' mark immediately preceding the name of the element type of the start tag. That element names are case-insensitive by convention. For instance, <S> is the same as <s>.

Note that an element can contain other elements as its contents in a prescribed (and possibly recursive) way. For example, we may have a title page of the following form:

```
<title.page>
    <title>Advanced Unix Programming</title>
    <subtitle> -- A Global View</subtitle>
```

</title.page>

The element names (e.g., "title.page", "title", "subtitle") and their structures (e.g., ‘a "title.page" consists of a "title" followed by a "subtitle"’) are defined in a Data Type Definition (DTD) file. The DTD for the current recommendation is shown at the Appendix of this draft.

If no ambiguity will be introduced by omitting the end tag, and the end tag is declared as ’optional’ in the DTD file, then the end tag can be omitted in the document. This normally occurs when the end tag is followed by another tag which unambiguously indicates the end of the preceding element. We will use as few end tags as possible in designing the exchange format to reduce the labor of the corpus contributors. This policy may, however, slightly increase the programming efforts of the corpus users. See the DTD in the appendix for the full list of tags whose end tags are declared as optional.

Any comment can be enclosed with the string "<!--" and "-->" anywhere in the encoded text. (However, it should NOT appear within a start tag or an end tag.) For example, we may have the following comments:

```
<!-- Source description:
        version         = 1.0
        date            = 1992-02-29 12:25 GMT+8
        compiled by     = Jing-Shin Chang
        No. sentences   = 2,543
        No. Words       = 12,599
        verification    = OK
-->
```

for describing the characteristics of the current text corpus. (The syntax for comments is actually more flexible in SGML than the above mentioned one. For simplicity in decoding, however, it is not allowed to use the more general SGML syntax for comments in the corpora.)

If the original file contains the symbol "<" or ">", they should be represented as "<" ("less than" symbol) or ">" ("greater than" symbol), respectively, to avoid confusion with the open delimiter ("<") and close delimiter (">") used for enclosing the tag names. (A string enclosed with "&" and ";" in the text file, like the "<" and ">" examples, is called an entity in SGML term.)

Note that TABs and new lines (CR or NL) can be used to improve the readability of the encoded text. Any number of new lines immediately following a start tag or immediately preceding an end tag is interpreted as a null string. Any number of other new lines, tabs within a tag pair is interpreted as a single SPACE.

0.3. Attributes of an Element

An element can be associated with a set of attributes. The attributes are specified within the start tag as a set of "feature=value" pairs. For example,

<sentence id=s120 charset=BIG5>.....</sentence>.

In this example, the <sentence> has an ’id’ of ’s120’ and it is encoded with a ’charset’ (character set) called ’BIG5’. The "value" should be enclosed with a pair of double quotation marks (") when the value contains spaces, such as:

<book title="Advanced Unix Programming">...</book>

Alternatively, if the attribute value contains double quotation marks, the apostrophe mark (’) can be used instead to quote the value, and vice versa.

If an attribute is not specified, a default is usually given by the DTD file unless it is explicitly requested in the DTD. In addition, the attributes can be arranged in any order.

In the current recommendation, all element names, attribute names and attribute values must be encoded with the subset of the printable ASCII characters. The other character sets can be used in a comment (with a potential risk of code conflict).

0.4. Other Tagging Conventions

For convenience, the filename of a file conformant to the current recommendation should be given a filename extension of ".mxf" (CJKDOCP Multilingual eXchange Format). For instance, we may use "news.mxf" for such a purpose. To enable easy inspection on a conventional terminal, it is suggested that a text line in a file be limited to 80 bytes or less.

In many cases, a CJK text file may contain ASCII symbols. These symbols are regarded as ‘foreign’ symbols for CJK text. In other words, they are regarded as being languages from another character set. To avoid the need to add a <foreign> tag, as will be described in later sections, it is desirable to change such symbols into their equivalent symbols in the CJK character set before the file is encoded in the current exchange formats.

1. General Skeleton of a Corpus File in the CJKDOCP Recommendation

A general skeleton for a corpus file encoded with the recommended guidelines will look like this:

```
<!doctype CJKDOCP.corpus system "CJKDOCP.dtd">
<CJKDOCP.corpus id=manual-mac-sys.eng.000 n=001>
        <corpus.header>
                <refname><!-- a formal reference name --></refname>
                <source><!-- name of the source corpus --></source>
                <!-- optional COMMENTS:
                        #bytes, #words, #sent, #entries, #defs,
                        #rules, domain code, etc.
                -->
        </corpus.header>
        <corpus.text>
                <corpus.type>
                <!-- encoded text -->
                <!-- encoded text -->
                <!-- encoded text -->
                <!-- encoded text -->
                </corpus.type>
        </corpus.text>
</CJKDOCP.corpus>
```

The keywords "doctype", "system" as well as "CJKDOCP.corpus", "CJKDOCP.dtd", "corpus.header", "corpus.text", "refname" and "source" are reserved words either for SGML or for the current recommendation; they should be entered as shown in the above example.

The string "corpus.type", on the other hand, should be replaced with an appropriate string which corresponds to the type of corpus being encoded. For instance, it could be "name.list.0" for a list of names. In the current revision (ver 1.2.3) of this draft, only two corpus types, namely "name.list.0" and "text.0" are defined. (The number ’0’ is used here only for version control of the types.) See later sections for the formats of such recommended corpus types. This tag could be used by the decoding programs to identify the appropriate format.

1.1. Document Type Definition

The first line, enclosed with "<!" and ">", is an SGML declaration for the document type at hand; the keyword ’doctype’ specifies the ’root’ element of such a document type. The generic name of the document type for corpora encoded with the recommended scheme is "CJKDOCP.corpus". It can be used as a keyword by the decoding programs for the encoding scheme of the current recommendation.

The structures of the various type of corpora are formally defined in a Data Type Definition (DTD) file called "CJKDOCP.dtd". The current version of the CJKDOCP.dtd file is attached as an appendix in this draft. Its electronic version is also avail upon request.

Such structures will be described later in the following sections and will be announced for each new corpus type registered or requested. Therefore, most users need not to know the existence of this file.

1.2. Corpus Identification, Corpus Registration and Header Information

Each ’CJKDOCP.corpus’ has a required ’id’ attribute (referred to as the ’corpus id’) and an optional ’n’ attribute (referred to as the ’serial number’). The former is used to uniquely identify the corpus, which may physically consist of several files; the latter is used to identify the file ID (default 0) of the current file within the corpus.

The corpus id, in general, takes the form of "type.lang.NNN". The ’type’ part is a hierarchical ’type code’ for the corpus such as ’news-sport-baseball’, ’manual-pc-sys’. The ’lang’ part is a language code for the language(s) used such as ’eng-chn’ (bilingual). The ’NNN’ part is a non-negative number, indicating the ordinal number of the current corpus among all corpora with the same type code. The starting number for NNN is ’0’. This part can be padded with leading 0’s without changing its semantics. The type code need not have exactly the same number of fields as shown here so long as it is registered as an unique code. In addition, the ’type’ and ’lang’ parts should not contain the ’.’ mark so that the id values can be used for other purposes.

To ensure uniqueness, the unique type code should be registered to the working group of the CJKDOCP committee for corpus exchange. The registration will be pre-registered or registered upon request. (See the last section for the contact address of the working group.)

The file IDs for a corpus with multiple files can be assigned by the corpus contributor. The convention is to start the file IDs from 0 and to increment it by one for each extra file.

The <corpus.header> element is intended to contain required messages about the characteristics of the corpus. For the present, only a <refname> (reference name) element and an optional <source> (source corpus) element need to be specified.

The <refname> is a uniquely defined string used to refer to this file. For instance, we can use:

<refname>Name-List-Taiwan-1992-Part-I-of-10</refname>

to refer to a corpus file containing personal names. Only the first 512 bytes of the <refname> will be used for reference in the recommendation. Also, there should be only ONE <refname> per <corpus.header>. The characters used in the reference name must be within the code space of the printable ASCII for easy identification.

The symbolic <refname> is used for identifying the current file and providing a quick reference for finding files that are derived from the same source. It could be different from the filename of the file. In fact, it is recommended NOT to use a filename as the reference name since the filename may be system dependent and may vary with time.

Since the <refname> is a symbolic name for unique identification of a file and for human inspection. Therefore, it should be unique among all files too. (But no registration of such a name is required.) A systematic way for naming the files in the <refname> element is desirable. For instance, we may add something like "file: 1/20" for a number of files belonging to the same corpus, and increment the file count for each file in the <refname> element.

The optional <source> element should contain the reference name of the "source" file from which the current file is derived. For example, if the sentences are translated (or derived in any way, or analyzed by any means) from a source file with the reference name "Advanced-UNIX-Programming", then the content of the <source> element should be "Advanced-UNIX-Programming".

If it is an original text corpus, or a multilingual text which contains the original text, then the content of <source> should be the same as the one in the current <refname>. The default is the same as the content of the current <refname> if it is not specified or if it contains a null string.

It is also strongly encouraged to put some informative messages as a form of comments within the <corpus.header>. For example, the corpus users are generally interested in the number of sentences, the number of words, the number of records, the number of dictionary entries, and so on. Version control information may be important in many cases too.

The format inside the comments is undefined to allow any useful information. It is recommended, however, to express each item with a "feature=value" format and to enter each such item line-by-line, whenever possible. The comments are, of course, optional if it requires excessive efforts of the corpus contributors. Finally, the "corpus.text" part in the previous skeleton is used to contain the encoded text part of the corpus. In the following sections, we will propose the formats for some corpus types, and give each format a formal name for the "corpus.type" string. These formats are defined specifically for some interested topics of the current CJKDOCP members.

2. Recommended Exchange Formats for Some Text Types

The following formats are designed for corpora that were committed to be released for research purposes by some of the research groups of the CJKDOCP members. Therefore, the tags may not include general text elements that have too much subtle details yet uninterested to the research groups. Availability and easy encoding and decoding is one major concern. In the following sections, we will use Chinese language in the examples and illustrations; the examples, however, should also be applicable to Japanese and Korean in many cases. Features and conventions specific to a particular CJK language will be established step-by-step as this recommendation evolves toward newer revisions.

2.1. Name List: <name.list.0>

The general skeleton for a simple name list will look like this:

```
<!doctype CJKDOCP.corpus system "CJKDOCP.dtd">
<CJKDOCP.corpus id=...>
<corpus.header>
        <refname>Name-List-Taiwan-1992</refname>
        <source></source>
        <!-- comments:
                no. of entries          = 12000
                date                    = 1993-01-22
                other information about this corpus ...
        -->
</corpus.header>
<corpus.text>
<name.list.0>
        <propname><name.person>Jing-Shin<name.family>Chang
        <propname><name.person>Roy S.<name.family>Brown
        <propname><-- name2 -->
        <propname><-- name3 -->
        <propname>...
        <propname>...
        <propname>...
        <propname>...
        <propname>...
        <propname><-- nameN -->
</name.list.0>
</corpus.text>
</CJKDOCP.corpus>
```

The corpus type for such a format is "name.list.0". The <corpus.header> part is encoded as described in Section 1.2. In particular, "Name-List-Taiwan-1992" is used as the reference name of the corpus file in this example. The <source> element can be omitted in this example, since the null string means that it is the same as the <refname>. The <name.list.0> element contains a set of <propname> (proper name) elements. The <propname> tag, when used to encode a personal name (the default type for proper names), can be followed by a <name.family> (last name or family name), and a <name.person> (first name) if it is in the Chinese language (character set) or other languages with the same convention for names.

If the name is in the English language or other languages, which have the same conventions for expressing names, the sequence should be a <name.person> (including the first name and the middle name) followed by a <name.family> (last name). It remains true even if the name refers to a Chinese such as 'Jing-Shin Chang'.

If possible, the <name.person> element can be replaced by or be further divided into the <name.first> (first name) and <name.middle> (middle name) elements.

The <name.title> tag can be used to enclose the title of a person such as 'Mr.', 'Dr.' or its Chinese (or Japanese, Korean) counterpart. The <name.suffix> tag, on the other hand, can be used to enclose a generational suffix like 'Jr.' and 'III'. For instance, we may have the following example:

<propname><name.title>Mr.<name.person>John<name.family>Smart

See the DTD for the various conventions for personal names.

The <propname> tag can also be followed by a full name without splitting it into parts if the boundary information is not available. In this case, a name written in English is interpreted as a sequence of first-(middle-)last names even though the name might refer to a Chinese. Alternatively, if it is written in Chinese, the last name should be followed by the first name. To make decoding easier, Chinese names and English names should be separated into different files.

If it is known that some names belong to female (F) and the others belong to male (M), then the 'sex' attribute can be used to start the list of names having a particular sex attribute as follows:

```
<name.list.0>
        <propname sex=F>name0
        <propname>name1
        <propname sex=M>name2
        <propname>name3
        <propname>...
        <propname>...
        <propname sex=F>name6
        <propname>...
        <propname>...
        <propname>nameN
</name.list.0>
```

The 'sex' attribute value is assumed to inhere from the previous <propname>. Therefore, name0, name1, and name6 through nameN all have the attribute value of F (female); the remainder belongs to M (male). If the sex attribute is unknown, use the attribute value 'U' (Unknown, i.e., sex=U) to avoid ambiguity.

2.2. Simple Text Corpus: <text.0>

Most of the text corpora with simple annotated information in the research organizations can be encoded with the following skeleton:

```
<!doctype CJKDOCP.corpus system "CJKDOCP.dtd">
<CJKDOCP.corpus id=...>
<corpus.header>
        <refname>.....</refname>
        <source>.....</source>
        <!-- header information as described in Section 1.2. -->
</corpus.header>
<corpus.text>
<text.0 lang=...>
        <p  id=p00 lang=...>
                <s id=s00 lang=...>...<!-- sentence #000 -->
                        <ling.analysis>
                        <unit>
                        <level type=pos>
                        ...     <!-- pos annotation for #000 -->
                        <level type=pron>
                        ...     <!-- pronunciation for #000 -->
                        <!-- other types of annotation -->
                        </unit>
                        </ling.analysis>
                <s id=s01>... <!-- sentence #001 -->
                <s id=s02>... <!-- sentence #002 -->
        </p>
        <p  id=p01>
                <s id=s03>... <!-- sentence #003 -->
```

```
            <s id=s04>... <!-- sentence #004 -->
            <s>.....<uko><!-- unknown object --></uko>...
            <!-- uko: unknown or unclassified objects -->
            <s>...
            <s>...
        </p>
        <!-- other <p>... -->
        <!-- repeat any number of <p> elements as shown -->
</text.0>
</corpus.text>
</CJKDOCP.corpus>
```

The corpus type for such a corpus is 'text.0'. The <corpus.header> is specified as in Section 1.2. The <text.0> element consists of ONE OR MORE paragraphs ( ); each , in turn, consists of ONE OR MORE sentences (<s>). An <s> can be followed by ZERO OR MORE types of annotated information like parts of speech ("pos") and phonetic transcriptions ("pron"); such annotated information is introduced by a <ling.analysis> tag.

We have omitted the end tags for the <s> elements since there is no ambiguity on the boundaries in the current example. An ending tag can also be omitted if the omission does not introduce ambiguity on the ending paragraph boundary. If the paragraph boundaries are unknown or undefined, then the tag pairs can also be omitted.

2.2.1 Token Delimiters, Unknown Objects and Annotated Information

In general, an <s> consists of a number of tokens, such as words or compound words. The words can be delimited with white space characters (e.g., spaces, newlines or TABs). For special tokens, like compound words, the tokens should be enclosed with a <t>...</t> (token) tag pair if the boundaries are known to the corpus tagger. An example for using the <t> (token) tag pair is shown as follows:

```
<s>One of the most popular <t>operating systems</t> is UNIX.
```

For languages, such as Chinese and Japanese, which do not have 'natural' delimiters between the words, the words can also be delimited by white spaces or by the <t> tag pairs. Since the white spaces can be regarded as an implicit <t> tag, the decoding programs should be careful when decoding a sequence of spaces followed by a <t> or a </t> followed by a sequence of spaces; the former is equivalent to a <t> tag while the latter is equivalent to a </t> tag.

We use a paragraph-sentence two-level model to simplify the encoding of a simple text with possible annotations. Any object that cannot be classified with these two levels should be quoted with a <uko></uko> (unknown object) tag pair.

For example, since we do not define a "list" element for the frequently used list structure as shown below:

```
1. This is item-1 of a list.
2. This is item-2 of a list.
        #-some-uninterested-text-#
3. This is item-3 of a list.
```

we may comment out the item numbers ('1. ', '2. ', etc.) and the uninterested part with the <uko> tags as follows:

```
<s><uko>1. </uko>This is item-1 of a list.
<s><uko>2. </uko>This is item-2 of a list.
        <uko>#-some-uninterested-text-#</uko>
<s><uko>3. </uko>This is item-3 of a list.
```

The headings in a general text can also be encoded either as an <s> or as an <uko> in quite the same way. Similarly, uninterested part of the text corpus can be marked out with the <uko> tag pair as shown in the 3rd line of the above example. Logically, this <uko> will be interpreted as a character string at the end of the 2nd <s>.

The <uko> tag pairs should be used in a consistent way so that the application programs can skip this part without affecting the processing of the remaining text. Logically, an unknown object is the same as a comment to the application programs. However, it is part of the corpus.

Each <s> element can be followed by zero or more types of annotation for the current sentence. The annotated

information will be enclosed by a <ling.analysis> tag pair, which consists of one alignment <unit> (**); the <unit> tag pair, in turn, consists of one or more <level> of annotated information. The <level> tag can be used to encode a level of linearized annotated information, whose type is identified by the 'type' attribute of the <level> tag.

(** The unit-level mechanism is used in the TEI P1 encoding scheme for implementing a general purpose alignment mechanism; in such a mechanism, a <ling.analysis> (linguistic analysis section) may consist of several alignment units, each of which has its own annotated information. In the current application, we need to use ONE and only ONE <unit> tag corresponding to the current <s> element, which is the unit in question.)

Legal types for annotation in the current draft are:

pron: pronunciation, (e.g., Chinese bopomofo phonetic transcription)
pos: parts of speech (e.g., 'noun', 'verb')
sem: semantic tags or semantic attributes (e.g., animate, objects)
anno: unspecified annotation

For example, if we are tagging the sentences with their parts of speech, then we may have:

```
<s           >This is a   book.
<ling.analysis><unit>
<level type=pos >det  be art n.
</unit></ling.analysis>
```

In the current draft, the tag sets (e.g., the phonetic transcription symbols or parts of speech) in the various annotation schemes are not formally defined. The contributors are responsible to make appropriate declarations on the set of legal tags in an appropriate place, say in the comment part of the <corpus.header>, or make it available through another file.

We have shown some attributes for the various element types in the general skeleton. In the following sections, their functions will be further illustrated by inspecting the exchange formats.

2.2.2 Encoding Monolingual Text

With the above skeleton, a typical monolingual Chinese text corpus with annotated information will be encoded as follows:

```
<!doctype CJKDOCP.corpus system "CJKDOCP.dtd">
<CJKDOCP.corpus id=...>
<corpus.header>
        <refname>.....</refname>
        <source>.....</source>
        <!-- header information as described in Section 1.2. -->
</corpus.header>
<corpus.text>
<text.0 lang=CHN>
        <p>
                <s>.....<!-- sentence #000 -->
                <ling.analysis><unit><level>...
                        <!-- annotation for #000 -->
                        ...</ling.analysis>
                <s>.....<!-- sentence #001 -->
                <ling.analysis><unit><level>...
                        <!-- annotation for #001 -->
                        ...</ling.analysis>
                <s>.....<!-- sentence #002 -->
                <ling.analysis><unit><level>...
                        <!-- annotation for #002 -->
                        ...</ling.analysis>
        </p>
```

```
        <p>
                <s>.....<!-- sentence #003 -->
                <ling.analysis><unit><level>...
                        <!-- annotation for #003 -->
                        ...</ling.analysis>
                <s>.....<!-- sentence #004 -->
                <ling.analysis><unit><level>...
                        <!-- annotation for #004 -->
                        ...</ling.analysis>
        </p>
</text.0>
</corpus.text>
</CJKDOCP.corpus>
```

If the <ling.analysis> elements are deleted, a skeleton for a pure text corpus with known sentence boundaries is acquired. If the sentence boundaries are not clear, the corresponding skeleton can be acquired by simply removing the <s> tags too.

The 'lang' attribute in <text.0> shows that this is a 'CHN' (Chinese) corpus. It could be 'ENG' for English, 'JPN' for Japanese and 'KOR' for Korean. Each 'lang' attribute have a corresponding 'charset' (character set) attribute; the default 'charset' for traditional Chinese is 'BIG5', and the default for English is 'ASCII'. Since the 'charset' attribute is not specified here, it is assumed to be encoded in the BIG5 code. Simplified Chinese characters should use 'GB2312-1980' as the character set value. The recommended character set for Japanese is 'JISX0208-1983'. Other lang values and charset values which are not specified in this draft will be defined later upon request.

The 'lang' attribute (and the 'charset' attribute) for an element, if not specified, will be the same as its parent element. If the 'lang' attribute of the parent element is not specified, the last value specified by an element of the same element type will be used.

The 'lang' attribute used in the various elements indicates the 'major' language(s) used in the elements. If only a few words in the elements are written in another language, they could be enclosed by a <foreign> tag pair (possibly specified with the 'lang' attribute) to accomplish the language shift operation in the documents. The following sentence shows a typical such example:

```
<s>sho2 wei4 de1 <foreign lang=ENG>PC</foreign> shi4 ji3 gei4 jen2
        den4 nao3 <foreign lang=ENG>(Personal Computer)</foreign>
```

At the end of the </foreign> tag, the language attribute will shift back to the major language specified in its parent element.

2.2.3 Encoding Aligned Multilingual Text

I. Aligned Sentences within Multilingual Paragraphs

There are two common skeletons for encoding an aligned multilingual text. In the first format, the corresponding sentences in different languages are put together and enclosed in the same <p>.

In this case, the <text> part for a typical English-Chinese translation pair will look like this:

```
... ...
<corpus.text>
<text.0>
<p>
        ... ...
        <var>
        <rdg><s lang=ENG>This is a book.            <!-- ENG --></rdg>
        <rdg><s lang=CHN>je4 sh4 i4 ben3 shu1.      <!-- CHN --></rdg>
        </var>
        <var>
        <rdg><s lang=ENG><!-- English Sentence --></rdg>
```

```
        <rdg><s lang=CHN><!-- Chinese Sentence --></rdg>
        </var>
        ... ...
</p>
<p>
        ... ...
        ... ...
</p>
</text.0>
</corpus.text>
</CJKDOCP.corpus>
```

The <var> (text variant) tag pair is used to enclose the variant readings of a base text; each reading (including the base text) is introduced by the <rdg> tag pair. Such a variant encoding scheme can thus be used to align a number of parallel segments which are in different languages.

In some cases, the sentence alignment may not be 1-to-1. For alignment that involves multiple sentences, we can also enclose all the sentences in one alignment units with a <rdg> tag pair. For example, an English sentence which is aligned with two Chinese sentences may be encoded in the following way:

```
... ...
<text.0>
... ...
<p>
        ... ...
        ... ...
        <var>
        <rdg>  <s lang=ENG>This is a computer manual, which describes
               the operations of a PC.
               </rdg>
        <rdg>
               <s lang=CHN>je4 sh4 i4 ben3 den4 nao3 shou1 che4.
               <s lang=CHN>shu1 jon1 miao2 shu4 <foreign lang=ENG>PC</foreign> de5
                     chao1 jo4 fung1 shi4.
               </rdg>
        </var>
        ... ...
        ... ...
        <var>
        <rdg><s lang=ENG><-- ENG --></rdg>
        <rdg><s lang=CHN><-- CHN --></rdg>
        </var>
        ... ...
        ... ...
</p>
```

With such an encoding scheme, an ENG sentence (or ENG sentence group) is followed by its CHN counterpart (a CHN sentence or a CHN sentence group) and the alignment is implicitly implemented with the <var> and <rdg> tag pairs. In other words, the basic units for alignment in a parallel multilingual corpus can be a sentence or a sentence group.

Note that the 'lang' attribute of the <p> element (or its parent, say <text.0>) can be specified so that only the <s> elements with different 'lang' values need to be specified. The same is true for specifying the 'lang' attributes of the <var> or <rdg> elements. This could save substantial tagging efforts if it is done by hand. For automatic tagger, however, it is encouraged to specify the 'lang' attributes explicitly in each <s> element.

The same encoding method is also used when the unit for alignment is a paragraph as described in the following section.

II. Aligned Paragraphs

In the second format, the sentences in the same language are enclosed in the same paragraphs, and the languages for adjacent paragraphs change sequentially; the language used is specified in the 'lang' attribute of the <p> element. In such a paragraph-aligned correspondence, the skeleton may look like:

```
... ...
<corpus.text>
<text.0>
        <var><-- first aligned paragraph -->
        <rdg>
        <p lang=ENG><s>This is a book.
                        <s>I like it very much. ....
                </rdg>
        <rdg>
        <p lang=CHN><s>je4 sh4 i4 ben3 shu1.
                        <s>wuo2 hen3 shi3 huan1 ta1. ...
                </rdg>
        </var>
        <var><-- second aligned paragraph -->
        <rdg>
        <p lang=ENG><s>Its name is ...
                </rdg>
        <rdg>
        <p lang=CHN><s>ta1 de5 ming2 tz5 jiau4 tzuo4 ...
                </rdg>
        </var>
        <!-- an ENG paragraph is followed by a CHN paragraph -->
        ... ...
</text.0>
</corpus.text>
</CJKDOCP.corpus>
```

In this example, each paragraph (<p>) is enclosed by the <rdg> tag pair, and an English reading is followed by a Chinese one. Note that the <s> tags can still be added within the <p> elements if the sentence boundaries are available. In addition, each <s> can be followed by any number of annotations as usual. If such sentence boundaries are not available, the <s> tags need not be tagged.

III. Cross References among Different Files

Besides the above two formats, explicit link or alignment between a source corpus file and a derived corpus file can be easily established by using an external cross reference mechanism. This is necessary when a derived work is acquired from a source corpus.

The cross link is done by assigning an unique character identifier to the 'id' attribute of the source <p> or <s> element of interest, and by associating the same 'id' value to the 'x.target' (external target) attribute for the corresponding <p> or <s> elements. A 'sys.id' must also be specified to point to the external source corpus file.

Alternatively, a <p> or <s> element can be assigned an unique numerical serial number instead to its 'n' attribute, and we can associate the same 'n' to the 'x.target' attribute of the corresponding element. Both the 'id' and the 'n' attributes have the same function in terms of cross reference. The differences between these two kinds of reference scheme will be clear shortly.

The 'id' (or 'n') of the source <p> or <s> is associated with the derived <p> or <s> element by adding an <xref> tag, with the proper 'sys.id' and 'x.target' specified, immediately after the derived <p> or <s> element. The tagging process is otherwise the same as in tagging a monolingual (or multilingual) text. The following example shows such a link:

[File #1]

```
... ...
<CJKDOCP.corpus id=manual-os-unix.eng.000 n=001>
... ...
<refname>UNIX-OS-2-of-3</refname>
... ...
<s id=s12 n=2.3 ...>....</s>
<s id=s13 n=2.4 ...>....</s>
... ...
```

[File #2]

```
... ...
<CJKDOCP.corpus id=manual-os-unix.chn.012 n=001>
... ...
<refname>UNIX-OS-2-of-3 (Chinese Edition)</refname>
<source>UNIX-OS-2-of-3</source>
... ...
<s>
<xref sys.id="manual-os-unix.eng.000 001" x.target="id s12">
This is the 12th derived sentence...
</s>
<s>
<xref x.target="id s13">
This is the 13rd derived sentence...
</s>
... ...
```

The ’sys.id’ is the same as the CORPUS ID (e.g., "manual-os-mac.eng.022") of the external source corpus file if the corpus consists of only one file; if the corpus file belongs to a multi-file corpus, then it must also be followed by a BLANK and the FILE ID for the particular source file (e.g., "manual-os-unix.eng.000 001"). If the ’sys.id’ is not specified, it will be the same as the most recent ’sys.id’ of the <xref> tags.

Almost all elements defined in the current recommendation has an ’id’ attribute and a serial number (’n’) attribute for cross reference. These two types of attributes must be unique within a corpus file. The convention is to count the <p> (<s>, <var> or <rdg>) elements from 0 and precede the number with the element name to acquire an unique character id for the value of the ’id’ attribute. This means that ’p10’, ’s12’, ’var3’, ’rdg2’ refer to the 11th paragraph, the 13th sentence, the 4th variant and the 3rd reading, respectively.

The convention for assigning the ’n’ attribute is to express it as ’PPP.SSS’ where ’PPP’ is the paragraph count as acquired in the above manner and ’SSS’ is the sentence count within the current paragraph. This relative addressing method is more robust in that the deletion of sentences in other paragraphs does not affect the serial numbers of the current paragraph. On the other hand, the ’id’ attributes provide a convenient way for sequentially searching a particular type of elements. Therefore, the taggers and the users can choose one of the two reference schemes for their particular applications.

These conventions can also be used to calculate the ’id’ or ’n’ attributes when the source corpus file does not specify such reference IDs explicitly.

When specifying the value of the ’x.target’ attribute, the referenced ids must be preceded by the string "id" or "ref" to specify whether the id refers to the ’id’ attribute or the ’n’ attribute of the source sentence (or paragraph). For instance, if we want to use the ’n’ attribute for cross reference, we may specify ’x.target="ref N.N"’ as in the following example:

```
<s>
<xref sys.id="manual-os-unix.eng.000 001" x.target="ref 2.3">
...
</s>
<s>
<xref x.target="ref 2.4">
```

...
</s>
... ...

Note that the formal DTD for the above skeletons has more freedom in the syntax due to the intrinsic expressive power of the SGML markup language. However, all corpus contributors and users should only follow the formats and constrains described here.

For example, a DTD which can express the formats described in this draft may also allow a mix-up of paragraphs, some of which are assigned with the <s> tags and some of which are not; there is no explicit mechanisms for constraining the paragraph types to be 'uniform' in the DTD syntax. This situation, however, is not allowed in the exchange formats; we require that the <s> tags be either tagged or discarded in all paragraphs for easy decoding.

This means that a corpus file that is conformant to the current recommendation is always SGML conformant with respect to the DTD, but NOT vice versa.

3. Request for Comments

Any question, correction, suggestion and recommendation are highly appreciated. Please forward your messages to:

Jing-Shin Chang (SIGMT, ROCLING)
Fax: 886-35-770459
Tel: 886-35-770243 Ext. 247
Email: cjkxf@bdc.com.tw, rocxf@bdc.com.tw

The registration of any new corpus types for traditional Chinese corpora and corpus ids should also be directed to the above channels. Coordinator for corpus registration for languages other than (traditional) Chinese is under arrangements. Code conversion softwares for the various Chinese character sets (traditional or simplified) are also available upon request.

References

Eric van Herwijnen, *Practical SGML*, Kluwer Academic Publishers, 1990.

Martin Bryan, *SGML -- An Author's Guide to the Standard Generalized Markup Language*, Addison-Wesley Publishing Company, 1988.

C. M. Sperberg-McQueen and Lou Burnard (eds.), *Guidelines For the Encoding and Interchange of Machine-Readable Texts* (TEI P1), ACH, ACL, ALLC, 1990.

Appendix

This appendix contains some technical details which might be useful as a supplement to the above draft for the corpus tagger and the application programmers.

A. 1 The Document Type Definition for the Current Recommendation

The formal definition of the text elements described in the current recommendation is shown here for reference. The electronic version of this DTD, the "CJKDOCP.dtd" file, is available upon request.

The DTD contains two types of declarations, namely the ELEMENT declaration and the ATTLIST declaration. An ELEMENT declaration is used to specify the sub-elements contained in the element, the linear order of these sub-elements, and the number of times they can repeat in the element. The ATTLIST, on the other hand, specifies the attributes of an element, their types and their default values.

An element declaration has the form of:

<!ELEMENT GI Ms Me (content_model) +(inc_exception) -(exc_exception)>

The keyword ELEMENT designates that this is an ELEMENT declaration. The GI (generic identifier) field is used to specify the name of the element (or the names of a group of elements). The "minimization rule" for the start tag (Ms) and the minimization rule for the end tag (Me) are used to specify whether the corresponding tag is required or optional; if the tag is required, its value is a '-', otherwise, it is a big 'O' (optional or omitable). The content_model specifies the

contents (i.e., the sub-elements) of the element.

Note that the 'Me' for an element must be declared as 'O' if it is to be omitted in the corpus; an omission, of course, should not introduce ambiguity in the element boundary.

The 'inclusion exception' part (inc_exception), which is introduced by a '+', specifies the elements which can appear ANYWHERE within in the current element; on the contrary, the exclusion exception (exc_exception) part introduced by a '-' sign specifies which elements CANNOT appear within the current element. The inclusion exception and the exclusion exception parts are optional to an element declaration.

To allow easy extension, many elements have been specified in the 'inclusion exception' part of high-level elements. The use of such tags, however, should be restricted to the locations described in the draft.

For example, the <CJKDOCP.corpus> element can be specified as follows:

<!ELEMENT CJKDOCP.corpus - - (corpus.header, corpus.text) >

This means that we have an element called <CJKDOCP.corpus>, which contains a <corpus.header> element and a <corpus.text> element. The two '-' signs means that both the start tag (<CJKDOCP.corpus>) and the end tag (</CJKDOCP.corpus>) must be specified when encoding a text.

The ',' punctuation mark between <corpus.header> and <corpus.text> means that the former must be followed by the latter in that order. In SGML, three punctuation marks are used to specify the way of connection between sub-elements:

A , B , C :	A, B, C must appear and must appear in that order
A & B & C :	A, B, C must appear and can appear in any order
A | B | C :	only one of A, B, C can appear

The parentheses surrounding <corpus.header> and <corpus.type> means that these two elements compose of a group of elements. Note that only one of the above three "ordering connectors" can be used within a group of element; different groups, which need different ordering connectors can be enclosed with parenthesis pair.

If a sub-element can appear more than once, we have another three punctuation marks, called the "occurrence indicators", to specify the following facts:

E* :	E can appear ZERO or more times
E+ :	E can appear at least ONE time
E? :	E can appear ZERO or ONE time (i.e., E is optional)

For example, the following declaration shows that a sentence group contains ONE or more sentence(s):

<!ELEMENT ss - - (sent+) >

When the content model is an SGML reserved word, like EMPTY or #PCDATA, the element contains no further sub-elements except for certain character data; the element can thus be regarded as a terminal node of the document tree. (See any SGML bibliography for the definition of such reserved words.)

The attributes of an element is declared by the ATTLIST keyword as follows:

<!ATTLIST GI attr TYPE default ...>

Again, the generic identifier(s) (GI) is the name of the element. The 'attr' field is an attribute name of the element, the TYPE field is the data type of the attribute; the 'default' field is the default value of the attribute or the data type of the default.

Note that the 'attr TYPE default' triple can be repeated for each attribute of the element. For instance, a paragraph element (<p>) may have the attributes:

```
<!ATTLIST p  id      ID          #IMPLIED
             n       CDATA       #IMPLIED
             type    CDATA       #IMPLIED
             lang    (CHN|ENG)   ENG>
```

This example shows that the <p> element has 4 attributes, namely the 'id', 'n', 'type' and 'lang' attributes. Except for enumerating all the possible attribute values explicitly like in the last line of the above example, the TYPE field may have the following implicit types:

ID: a unique identifying value for the current element
IDREF: a pointer to some other element
CDATA: any valid character data, including tags
NUMBER: composed of numeric values only
NMTOKEN: any string of alphanumeric characters ('name token')

The most common default types of values are:

#REQUIRED: a value must be specified
#IMPLIED: a value need not be supplied
#CURRENT: if no value is supplied, the last specified value should be used.

For a long string in the declaration, we can define an entity (more or less a 'macro definition') as the short hand of this string. For instance, we may define the long string in the above ATTLIST as follows:

```
<!ENTITY % common.attrs
        "id     ID          #IMPLIED
        n       CDATA       #IMPLIED
        type    CDATA       #IMPLIED
        lang    (CHN|ENG)   ENG" >
```

and share this entity definition among several elements when declaring the attributes of these elements:

```
<!ATTLIST (par|sent|text) %common.attrs; >
```

The entity reference (i.e., the 'macro call') '%common.attrs;', which is enclosed by the '%' and ';' pair, will be replaced by the string "id ID #IMPLIED ... ENG" when processed by an SGML parser. The '"' marks is required in this entity declaration since the string contains spaces. An entity of this kind, which is preceded by an '%' mark in the entity name, is called a 'parameter entity'; it is used specifically in the DTD. Another type of entities, called the 'general entities', is declared as in the above ENTITY declaration except that the '%' mark is removed. Such kind of entities are used in the TEXT part of the document, and is enclosed by an '&' mark and a ';' mark like '&common.attrs;'. With these notations, the syntax for the following CJKDOCP.dtd should be easy to understand.

[File "CJKDOCP.dtd"(Rev. 1.5)]

```
<!-- ================================================================== -->
<!-- @(#)CJKDOCP.dtd by Jing-Shin Chang 1995/12/06                      -->
<!-- @(#)$Id: CJKDOCP.dtd,v 1.5 1995/12/05 16:11:27 shin Exp shin $     -->
<!--                                                                    -->
<!-- Data Type Definition for corpus exchange among CJKDOCP members     -->
<!--                                                                    -->
<!-- ================================================================== -->

<!-- ****************************************************************** -->
<!-- Macro definitions                                                  -->
<!-- ****************************************************************** -->

<!-- entity definitions for escaping the tag open/close delimiters      -->
<!ENTITY        lt      "<"                                             >
<!ENTITY        gt      ">"                                             >

<!-- corpus types currently defined in the CJKDOCP recommendation       -->
<!ENTITY % corpus.types "name.list.0 | text.0"                          >

<!-- comments: unknown objects, etc.                                    -->
<!ENTITY % comments " uko "                                             >

<!-- empty elements or optional elements                                -->
<!ENTITY % f.empty " xref "                                             >

<!-- ****************************************************************** -->
<!-- common attributes                                                  -->
<!-- ****************************************************************** -->

<!ENTITY % global.attrs
          "id       ID        #IMPLIED
           n        CDATA     #IMPLIED
           type     CDATA     #IMPLIED
           lang     NAME      #IMPLIED
           charset  CDATA     #IMPLIED"                                 >

<!-- the xref attributes are likely to be global among common elements  -->

<!ENTITY % xref.attrs
           "sys.id  CDATA     #IMPLIED
           x.target CDATA     #IMPLIED"                                 >

<!-- ****************************************************************** -->
<!-- Top Level Element: corpus                                          -->
<!-- ****************************************************************** -->

<!ELEMENT CJKDOCP.corpus - - (corpus.header, corpus.text)               >
<!ATTLIST CJKDOCP.corpus
           id       ID        #REQUIRED
           n        CDATA     #IMPLIED
           type     CDATA     #IMPLIED
           lang     NAME      #IMPLIED
           charset  CDATA     #IMPLIED                                  >

<!-- ****************************************************************** -->
<!-- Header                                                             -->
<!-- ****************************************************************** -->

<!ELEMENT corpus.header - - (refname, source?)                          >
<!ELEMENT refname    - -  (#PCDATA)                                     >
<!ELEMENT source     - -  (#PCDATA)                                     >

<!ELEMENT corpus.text     - -  ( %corpus.types; ) +( %f.empty;
                                                   | ling.analysis
                                                   | var
                                                   | %comments; )       >
<!ATTLIST corpus.text     %global.attrs;                                >
<!-- For simplicity, empty elements and optional elements              -->
```

```
<!-- are all embedded at the highest level of the DTD.                  -->
<!-- ********************************************************************* -->
<!-- Name List                                                          -->
<!-- ********************************************************************* -->

<!ENTITY % chn.name "(name.family,name.person,name.suffix?,name.title?)"  >

<!ENTITY % eng.persname " (name.person|(name.first,name.middle?)) "       >
<!ENTITY % eng.name "(name.title?,%eng.persname;,name.family,name.suffix?)">

<!ELEMENT name.list.0     - -  ( propname+ )                               >
<!ELEMENT propname        - O  ( #PCDATA | %chn.name; | %eng.name; )       >

<!ELEMENT name.title      - O  ( #PCDATA )                                 >
<!ELEMENT name.person     - O  ( #PCDATA | (name.first,name.middle?) )     >
<!ELEMENT name.family     - O  ( #PCDATA )                                 >
<!ELEMENT name.suffix     - O  ( #PCDATA )                                 >

<!ELEMENT name.first      - O  ( #PCDATA )                                 >
<!ELEMENT name.middle     - O  ( #PCDATA )                                 >

<!ATTLIST propname  %global.attrs;
          sex       CDATA     #IMPLIED                                     >

<!-- ********************************************************************* -->
<!-- Simple Text: mono lingual or aligned multi-lingual text            -->
<!-- ********************************************************************* -->

<!ENTITY % pars     "(p*|s*)"                                              >

<!ELEMENT text.0    - -  ( %pars; )                                        >

<!-- ********************************************************************* -->
<!-- Paragraphs & Sentences & Tokens                                    -->
<!-- ********************************************************************* -->

<!-- ENTITY % p.seq "(p|s)*"                                            -->
<!ELEMENT p         - O  ( s+ | #PCDATA ) +(foreign)                       >
<!ELEMENT s         - O  ( #PCDATA ) +(t | foreign)                        >
<!ELEMENT t         - -  ( #PCDATA )                                       >
<!ELEMENT foreign   - -  ( #PCDATA )                                       >

<!ATTLIST (p|s|t|foreign) %global.attrs;                                   >

<!-- ********************************************************************* -->
<!-- Comment Stuff                                                      -->
<!-- ********************************************************************* -->

<!ELEMENT uko       - -  ( #PCDATA )                                       >

<!-- ********************************************************************* -->
<!-- Empty elements                                                     -->
<!-- ********************************************************************* -->

<!ELEMENT xref      - -  ( #PCDATA )                                       >
<!ATTLIST xref
               sys.id    CDATA     #IMPLIED
               x.target  CDATA     #IMPLIED                               >

<!-- ********************************************************************* -->
<!-- Annotated Information                                              -->
<!-- ********************************************************************* -->

<!ELEMENT ling.analysis - - (unit*)                                        >
<!ELEMENT unit          - - (level+)                                       >
<!ELEMENT level         - O ( #PCDATA )                                    >
<!ATTLIST level %global.attrs;                                             >

<!-- type=pron|pos|syn|sem|anno                                        -->

<!-- ********************************************************************* -->
<!-- Tags for parallel segmentation method                              -->
```

```
<!-- ********************************************************************** -->
<!ELEMENT var        - -   (rdg+)                                             >
<!ELEMENT rdg        - -   (#PCDATA|p|s)*                                     >
<!ATTLIST (var|rdg)  %global.attrs;                                           >
```

A. 2 Attributes for the Elements

Many elements in the current recommendation have the following global attributes, which is defined in the '%global.attrs;' entity declaration. See the DTD for a full list of the elements which have the following global attributes:

- type: the type of the element. It is a generic attribute for describing the characteristics of the element. For example, we have the level of annotation ('pos', 'pron', 'syn', 'sem', 'anno') as used in the <level> tag.

For the textual elements, like <text.0>, <p> and <s>, it can be 'mono' for a monolingual text element, 'bi' for a bilingual text element, or a number greater than 2 for a multilingual corpus. The default is 'mono'. Its use in such application is, however, temporarily left unused in the current draft.

- lang: the major language for the text element. Currently, 'ENG' is used for English, 'CHN' for Chinese, 'JPN' for Japanese and 'KOR' for Korean. The default is 'CHN'. Other language values should be the standardized named defined in ISO 639.

- charset: the character set used to encode the text corpus. Currently, 'ASCII' is used for the printable ASCII characters, 'BIG5' is used for the Big5 code. The default is 'BIG5' for traditional Chinese text corpora if 'lang=CHN'; for simplified Chinese text corpora, 'GB2312-1980' must be sepcified. 'ASCII' is the default for 'lang=ENG'. For 'lang=JPN', the default character set is 'JISX0208-1983'.

Note: If the 'lang' attribute or the 'charset' attribute is not specified, the defaults is inherited from the parent elements of the current element. If the parent element does not specify (or inherit) the value, then the latest element, whose 'lang' (or 'charset') attribute is specified, should be used as the default.

- id: a character identifier which is unique for the current text element within the current corpus file. See the draft on the convention for acquiring an unique id. If this attribute is not specified, its default count is one plus the count for the previous id of the previous element which has the same element as the current element. The count of the id starts from '0' (e.g., 'p0', 's0') if not explicitly specified in the first element.

Note: Since the ID's are used mainly for cross reference in or multiple file applications, the id's should NOT be changed once they are defined either implicitly or explicitly. Changing the id's in an arbitrary way will make it necessary to change all the cross reference id's in the other derived files so that all the cross references remain consistent. To avoid accidental deletion or insertion of entries, it is recommended to specify all the id's explicitly. At least, the id of the element immediately following the deleted element(s) should be specified explicitly so that the cross reference links are retained.

- n: a numeric identifier which serves as the serial number of an element. It has the same function as the 'id' attribute. However, it uses a relative addressing method for specifying the identifier.

The following attributes are used, say in the <xref> tag, for cross reference among different files:

- sys.id: an identifier which points to the source file from which the current element is derived. See the draft for the convention for specifying such an attribute.

- x.target: an cross reference id which points to the element from which the current element is derived; the corresponding corpus file is designated by the 'sys.id' as described above. For instance, if an <s> element is translated from an <s> in "Advanced-UNIX-Programming" whose id is s12, then the value of 'x.target' for the current <s> is x.target="id p12". (The <source> element for the current corpus would be "Advanced-UNIX-Programming".) If 'x.target' is not specified, its default value is derived from the current 'id' for the current element.