Building a Domain-Specific French-Korean Lexicon

Aesun YOON
Language Information Lab., Dept. of French & Dept. of Cognitive Science, Pusan National University
San 30, Jang-jeon-dong, Geum-jeong-gu, Pusan, 609-735, Rep. of Korea
asyoon@pusan.ac.kr

Abstract

Korean government has adopted the French TGV as a high-speed transportation system and the first service is scheduled at the end of 2003. TGV-relevant documents are consisted of huge volumes, of which over than 76% has been translated in English. A large part of the English version is, however, incomprehensible without referring to the original French version. The goal of this paper is to demonstrate how DiET 2.5, a lexicon builder, makes it possible to build with ease domain-specific terminology lexicon that may contain multimedia and multilingual data with multi-layered logical information. We believe our wok shows an important step in enlarging the language scope and the development of electronic lexica, and in providing the flexibility of defining any type of the DTD and the interconnectivity among collaborators. As an application of DiET 2.5, we would like to build a TGV-relevant lexicon in the near future.

1 Introduction

Korean government has chosen the French TGV as a high-speed transportation system and the first service is scheduled at the end of 2003. TGV-relevant documents are consisted of 700 thousands pages of which 170 thousands pages are written only in French language and the rest has been translated in English. A few locomotive engineers or railroad officers have reached a certain level of proficiency in comprehending French. As a language itself, English has more advantages to be better understood than French in Korea. Unfortunately however, a large part of the English version is not comprehensible without referring to the original French version.

The successful translation of manuals of high-tech machines needs (1) the fluency in object and target languages, (2) and the expert knowledge in the specific domain in question. It is hard, however, to find out human translators who have both of those 2 totally different capacities. The ideal situation might be that a pair of the language expert and the domain expert work together. But it is certainly a time-consuming and costly job, if there are a great number of object documents. The time effectiveness is a very important factor in the translation of manuals, since the new technology described in the manuals might be no longer valuable some time after. One of the solutions for the efficient and effective translation consists in classifying those pair works in 3 steps, provided those experts are equipped with appropriate tools. In the first step, the language experts translate the target documents into the first and rough version of the object language, with the domain-specific terminology lexicon. In the second step, the domain experts, who have not a good mastery of the object language, can examine the correctness of contextual meaning of roughly translated documents, using the bilingual or multilingual lexicon. In the third and final step, both experts can focus on deciphering jointly the incomprehensible parts. These processes could be done manually. But it would be certainly less time-consuming and more efficient that there is a kind of network-based translator's workbench where

they can do differentiated jobs quasi-simultaneously. A domain-specific terminology lexicon is a prerequisite for these processes (Zinglé, 1999).

However, there are some technical problems in developing domain-specific terminology lexicon, especially when it contains European languages in Korea. In order to support the input/output of diacritics used in most European languages, a professional knowledge of computation is highly required in Korean computational circumstances (Jeong & Yoon 1998, Jeong & Yoon 1999, Yoon & alii 1999, Yoon & alii 2000). Coupled with the technical problems, there are no developmental tools to be easily used. Some SGML (Standard General Markup Language)-Based tools, which have been developed so far by computer scientists, provide a fixed DTD for specific dictionaries and there are not concern about the solution for European languages (Choi & alii 1996, Kang 1996) Since developing a domain-specific terminology lexicon is a time-consuming work, it is likely that a widely separated group of language experts and domain knowledge experts work together. In order that one can communicate one's ideas and work with others effectively, the developmental tool can provide them the interconnectivity.

The goal of this paper to demonstrate how *DiET* (**Di**ctionary **E**di**T**or) 2.5 makes it possible to build with ease domain-specific terminology lexicon that may contain multimedia and multilingual data with multi-layered logical information. The functions of *DiET* 2.5[1] will be presented and described in detail as a powerful lexicon builder. *DiET* is a network-based developmental tool for multilingual lexica or corpora, which can support also multimedia data with multi-layered text data, so that domain experts who are not good at computation or computer use can easily construct their expert knowledge in structured information. Section 2 overviews the system architecture of *DiET* which enables developers to work jointly from different places, and shows how to define a DTD and to design a database for a selected domain-specific terminology lexicon. Section 3 introduces solutions for the multilingual and multimedia support. Then, we conclude with notes on the future work, in Section 4.

2 System Overview for *DIET*

DiET and *DiET-Web*, support the network environment, respectively the Intranet and the Internet.[2] Since developing a lexicon or a corpus is a time-consuming work, it is likely that a large number of developers collaborate but scatter in space. The availability to exchange up-to-date data in real-time is quite important, because such an ability facilitates the application interconnectivity so that the information can be exchanged during development of contents. The Internet/Intranet is a powerful infrastructure to distribute and share contents, because it is an effective and economical for making information available to the widely separated group of individuals. Within the Internet/Intranet environment, especially Web environment, one can represent, encode and ultimately communicate one's ideas and works with the others. *DiET* and *DiET-Web* allow lexicographers to collaborate efficiently on a dictionary without loosing the consistency of contents. Figure 1 shows the system architecture of *DiET-web*.

---

[1] *DiET* was developed and is maintained by the Language Information Laboratory at Pusan National University. (Its program registration number is 99-01-12-5270.) Its capability is not limited only to building the dictionaries of which the entries are words, but extended eventually to facilitating the construction of a corpus composed of parallel phrases or sentences.

[2] *DiET* is a groupware, in the sense that it can be used by a group of people who are working on the same information but may be distributed in space.

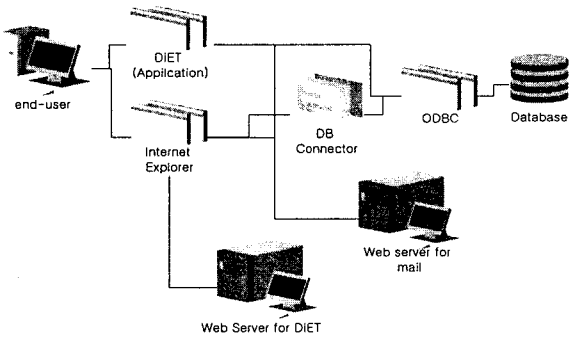

Figure 1: System architecture o f *DiET-Web*

DiET and *DiET-web* are developed with MS SQL Server 2000 that can support for XML (eXtensible Markup Language) and XSL (eXtensible Style Language). In this section, the database management and data handling functions of *DiET* will be described.

2.1 Database Management

DiET can create a database in which the administrator can handle with ease the developer management, the language registration, the language properties specification of sub-lexica.
Since developing a lexicon or a corpus is a time-consuming work, it is likely that several groups of developers differ in jobs. There should be a decision making group, such as system administrators or project initiatives, who take in charge of defining the structures or properties of the database, or handling numerous developers. The job of another group could be limited merely to inputting data. The third group might be needed to examine the validity or the correctness of input data, and so on. In this case, those groups should be differentiated in terms of authority for accessible data and functions. Figure 2 shows the developer managing function with which detailed authorizations are specified for each developer.

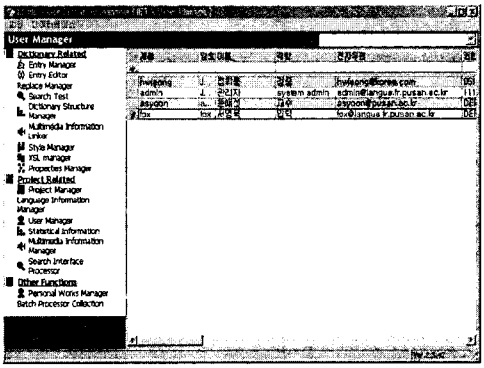

Figure 2: Developer Managing Function

DiET can support the input/output of special characters adopted in the 20 European languages[3] and the IPA (International Phonetic Alphabets), in Korean computational environment. Since each language uses its own diacritics for the special characters, *DiET* admits only a valid set of the special characters according to the languages, and invalid characters are prevented from being typed in.[4] For example, á is a valid character in Spanish but invalid in French. In creating a multi-lingual database, it is necessary to register the language(s) in use in that database. Figure 3 demonstrates the language registering function. Closely relevant to the language registering function, the administrator can also

[3] They are Ancient Greek, Danish, Dutch, English, Finnish, French, German, Hungarian, Irish, Italian, Modern Greek, Norwegian, Polish, Portuguese, Rumanian, Russian, Slovenian, Spanish, Swedish and Turkish.

[4] The codes and the input methods will be explained in detail in the section 3.

specify the language pair(s) or set(s) that will be used in the sub-lexica, as shown in figure 4. This function allows the default language(s) or language pair(s) when developers type in text data through the input window without selecting the language they need among a long list of available ones.

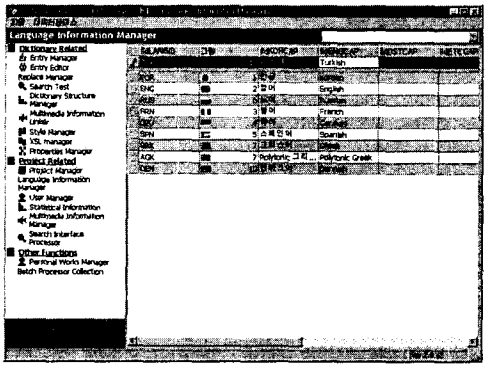

Figure 3: Language Registering Function

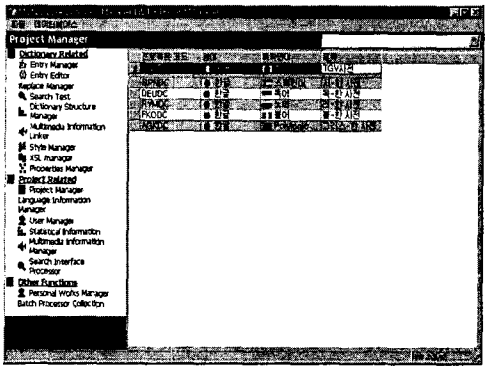

Figure 4: Language Specification of Sub-Lexica

2.2 Data Handling

Once developers are classified and languages are specified, the decision making group will (1) decide the number of sub-lexica that compose the database object, (2) determine the scope of lexical entries that will serve as sorting keys, and (3) define the document type of each sub-lexicon.
It is obligatory to differentiate the sub-lexica, if they use different DTD (Document Type Definition). The lexica using the same DTD may also be divided into sub-lexica, for practical reasons.
As for the determination of lexical entries, the scope of compound words and the expression of the homonymic relationship are issues in making domain-specific terminology lexicon. The domain-specific terminologies have more probabilities to be composed of more than 2 simple forms of word. In parsing domain-specific documents, difficulties arise in discriminating the compounds forms that do not contain any marks for compound words, such as an apostrophe or a hyphen. The registration of all possible compound words into a lexicon ameliorates certainly the success rate of exact searching. Another property of domain-specific terminology is that one lexical entry may have several different meanings (in monolingual lexicon) or translations (in bi- or multi-lingual lexicon) according to sub-domains. This property resembles very much to the homonymic relationship in language dictionary where the homonyms are determined on the basis of the etymology, the grammatical category or the meaning. In a domain-specific lexicon, sub-domains will be the most important criterion for the homonyms. That will offer a more proper and precise list of candidate translations to translators when they select a sub-domain, in using the translator's workbench.

The description of a lexical entry in a domain-specific lexicon depends largely on its objectives and its thesaurus that stands for the logical structure of the domain and its sub-domains. The TGV-relevant lexicon would need, at least, the grammatical category, the morphological notes, the syntactic

information as for verbs or adjectives, the sub-domain specifications, the translation in English and in Korean, and so on.

Segmenting and storing data as meaningful parts rather than as large clumps of text, dispense with full parsing every time it is accessed, thus allow for the creation of documents that can be easily searched. The first step in creating the DTD of the lexicon is investigating the logical structure of the documents, such as the technical manuals or the collection of regulations, and categorizing the meaningful information they contain (Boguraev 1994, Copestake 1990, Hausser 1999, Ide & Veronis 1995, Svensén 1993).

XML[5] is a data description language and it is designed to describe document data using arbitrary tags. As its name implies, extensibility is a key feature of XML; users or applications are free to declare and use their own tags and attributes. Therefore XML ensures that both the logical structure and content of semantics-rich information is retained. XML emphasizes description of information structure and content as distinct from its presentation. The data structure and its syntax are defined in a DTD specification, which is derivative from SGML and defines a series of tags and their constraints.[9]

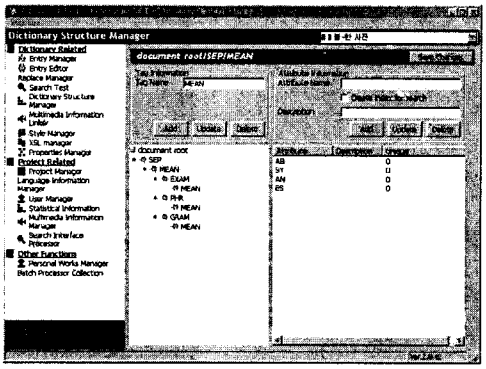

Figure 5: XML-Based DTD Manager

Based on the XML and XSL, *DiET* enables developers to define the DTD, as shown in Figure 5. Creating logical structures needs element and attribute declarations, which are the core of XML. A well-written set of elements and attributes will make it for programs easy to extract useful information. Based on the specification of the information structure, developers can define those segmented features into elements that can allow further embedded structure, and attributes that cannot possess child nodes. (Svensén 1993)

In contrast to information structure, the presentation issues are addressed by XSL (XML Style Language),[6] a language used to create style sheets for XML. similar to CSS (Cascading Style Sheets) that are used for HTML An XML document has to be formatted before it can be read, and the formatting is usually accomplished with style sheets. Style sheets consist of formatting rules for how particular XML tags affect the display of a document on a computer screen or a printed page. *DiET* can also support XSL-Based Style manager, as shown in figure 6.

---

[5] XML is a data description language standardized by the World Wide Web Consortium (W3C). Both HTML and XML are defined by SGML (ISO 8879). While HTML documents generated by documentation tools cannot be reused in other applications other than HTML browsers due to its fixed tag set, XML which is a sophisticated subset of SGML, is a semantic-rich mark-up language. One of the goals of XML is to be suitable for the use on the Web; thus to provide a general mechanism for extending HTML (Yoon & alii 2000).

[6] XSL is also W3C's emerging standard for expressing how XML-based data should be rendered. XSL is based on DSSSL (Document Style Semantics and Specification Language, ISO/IEC 10179) and interoperable with CSS (Cascading Style Sheet), which was originally a style definition language specific to HTML (Yoon & alii 2000).

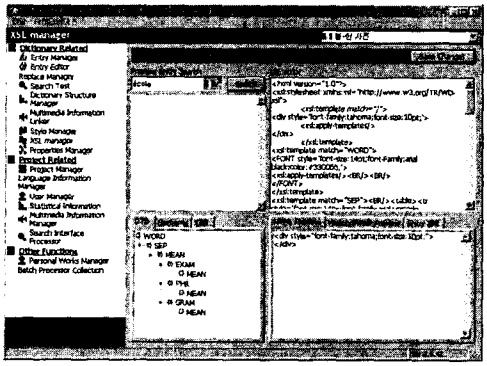

Figure 6: XSL-Based Style Manager

As such, XML has great potential as an exchange format for many kinds of structured data, and increases the productivity to author, maintain and view this data, together with the style sheet and linking mechanisms (Laurent 1998).

2.3 Text Data Building

When the DTD is determined, developers can start to build a lexicon by listing up lexical entries. Since a lexical entry serves as a key in sorting and retrieving, it has to be unique and does not allow any homographs. *DiET* provides developers with an entry managing function, as shown in figure 7.
An authorized developer can always add or find lexical entries, using the input window and function buttons that appear on the upper side of the screen. Input lexical entries show up on the middle column of the screen, in alphabetical order, when a developer selects one letter with which a lexical entry begins. If he clicks on an item, *DiET* calls the lexical descriptions of the selected entry on the right column, and he can manage the entries, by modifying or deleting them.

Once the list of lexical entries is set up, a developer can input needed lexical descriptions for each entry, in using the entry editor as shown in figure 8. The screen composition of the entry editor is similar to that of the entry manager. When the developer clicks on an entry by using the input windows or by selecting one among all input lexical entries on the middle column, the structure controller and the data input windows are activated on the right column. The structure controller enables the developer to construct the multi-layered information structure of the selected lexical entry, by adding a defined element as a sister node or a daughter node, or moving it to another mother node. The "control + s, a" creates respectively a sister node and an immediate daughter node of the current one he works on. The "control + <, >" alternates the property of elements. The developer can also move a node with its subordinate nodes by dragging and dropping it. The subordinate relationship among nodes is examined every time a node is created or moved by the validity check routine. The attributes that are dependent on elements are added on through the data input windows.

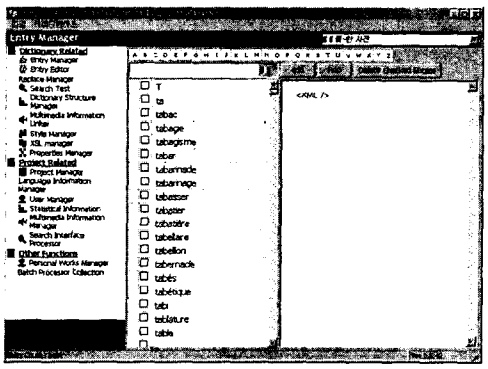

Figure 7: Entry Manager

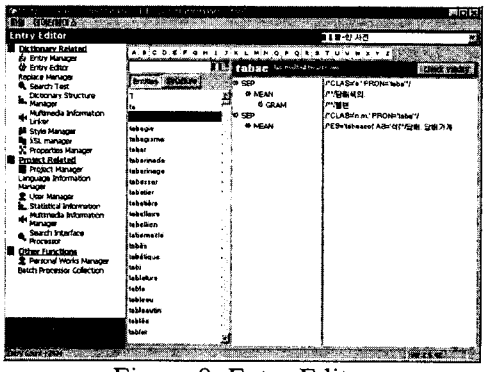

Figure 8: Entry Editor

3 Multilingual and Multimedia Support

French and many other European languages[7] have accent or special characters such as é, à, ñ, ß and ð. In addition to using Roman alphabet, most European Languages also contain diacritics, of which arise input/output problems in Korean environment.

Different from lexica or dictionaries printed on paper, electronic ones have more advantages in handling multimedia data. For the majority of end-users, the IAP is another jargon difficult to learn and decipher, but audio data will be intuitive and much informative from a phonetic point of view. As for TGV engineers, images will be much more helpful than simple translations of a terminology in text data format, when it refers to a particular part of a train.

In this section, the multilingual support and multimedia support that *DiET* can provide will be explained.

3.1 Input/output device

Since the local code pages of the Latin characters with diacritics are used to define Korean characters, the Korean versions of the operating systems can support neither their IME (Input Method Editor) nor APIs (Application Programming Interface) including displayed outputs and fonts. Moreover, each country tends to use its own keyboard of which the characters are differently displayed. Our study has redefined the code pages of European and Korean languages in the Unicode 2.0, and has recomposed 12 sub-sets of TrueType fonts.

Input methods have been also developed to solve these problems so that Latin characters can be made easily input with the Korean/English 101 keyboards. A particular input editor is implemented to support European characters with diacritics as well as Korean characters for *DiET* and for others application programs that we have developed. To allow normal European character input, we adopt the English/Korean 101 keyboard system, which is the most familiar with Korean users. The special characters of 20 European languages[8] and IAP (International Phonetic Alphabets) can be typed in by using the method of combination, as shown in table 1. Two or three keys are used to feed one character with diacritics. We choose carefully a symbol for a diacritic, on the basis of formal similarity between them, on the condition that the symbol should not be used for existing meta-languages such as tags (<, >) or parentheses ((,), [,], {, }). Our input system respects the order of "alphabet + diacritics" in hand writing, which is appropriate for human cognitive structure (Laurent 1998) It also supports not only the Russian and Greek characters but also that of the IPAs, by redefining whole the keyboard array, in theses cases.[9]

---

[7] The term *European languages* has a geographical ground rather than a linguistic ground, just like the term *Pan-European Windows*.

[8] They are Ancient Greek, Danish, Dutch, English, Finnish, French, German, Hungarian, Irish, Italian, Modern Greek, Norwegian, Polish, Portuguese, Rumanian, Russian, Slovenian, Spanish, Swedish and Turkish.

[9] In Ancient and Modern Greek, a character may contain double or triple diacritics within a character. The method

Table 1 : Input of special characters (*exerpt*)

Diacritics	Examples	Combination of key strokes
Acute	Á	O + /
Cedilla	Ç	O +]
Circumflex	Â	O + ^
Diaeresis, umlaut, tréma	Ä	O + "
Dot	Ż	O + *
Double Acute	Ű	O + ;
Grave	À	O + \
Haček or carom	Ǎ	O + #
Macron	Ē	O + _
Ogonek	Ą	O + `
Ring	Å	O + @
Slash	Ø	O + %
Tilde	Ñ	O + ~

3.2 Multimedia Support

DiET can create a sort of meta-database for multimedia data, which handles file name, data type, data property, key word(s), and so on, as shown in figure 9. When multimedia files are registered, a developer can link them to appropriate lexical entries of a selected lexicon, as shown in figure 10. These 2 processes are similar respectively to the entry managing function and the entry editing function explained in section 2.3, except one aspect that the meta-database is lexicon independent while the list of lexical entries is lexicon dependent.

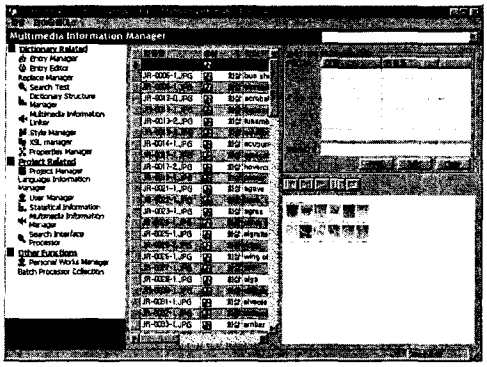

Figure 9: Multimedia Data Registration

of keystroke combination is also applied to a newly defined keyboard, in these cases. As for the Greek, we have adopted the combination method of key strokes proposed by Silver Mountain Company. The suggestion of the SIL (Summer Institute of Linguistics) has been borrowed, in *DiET*, to type in phonetic alphabets by combining key strokes, for examples "a, e" and "<" representing the open vowels of "a" and "e". [12]

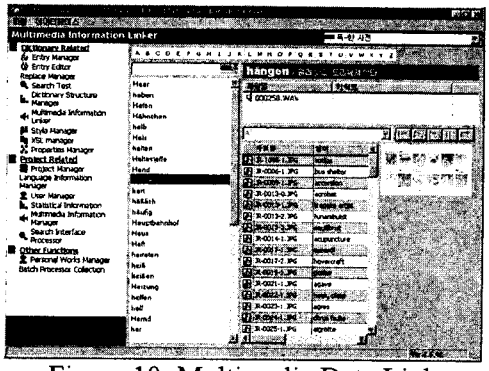
Figure 10: Multimedia Data Link

4 Conclusion and further studies

This paper addresses how *DiET* 2.5, a lexicon builder, makes it possible to build with ease domain-specific electronic lexicon that may contain multimedia and multilingual data with multi-layered logical information. The other developmental tools for building lexicon developed so far provide only a fixed DTD for a specific lexicon and they do not concern about the solution for European languages. The issues included are:

-. Definition of a DTD by analyzing the logical structure of documents in the specific domain in question;
-. Construction of a database;
-. Handling of various types of data
-. Multilingual support.

We believe our wok shows an important step in enlarging the language scope and the development of electronic lexica, and in providing the flexibility of defining any type of the DTD and the interconnectivity among collaborators. As an application of *DiET* 2.5, we would like to build a TGV-relevant lexicon in the near future and we are now analyzing the logical structure. As for the further studies in building TGV-relevant terminology lexicon, we have to (1) design multimedia and multi-lingual information structure of the TGV documents, based on the classification of multi-layered sub-domains, (2) demonstrate how to build collaboratively a multilingual TGV-relevant lexicon in a network environment, and (3) develop an extensible structure of language information in a lexical database so that the content will be easily reusable in future works, such as on-line translation or on-line education.

Acknowledgements

The author wishes to acknowledge the financial support of the Korean Research Foundation made in the program year of 2000 (Project number: 2000-A00359).

References

Boguraev, B. 1994. Machine-readable dictionaries and computational linguistics research, in A. Zampolli, N. Calzolari, and M Palmer(eds.) *Linguistica Computationale: Current Issues in Computational Linguistics: in Honor of Don Walker*, Vol. IX.X, Kluwer Academic Publishers, Dordrecht. 119-154.

Choi, B.J., W.J. Lee, J.S. Lee, K.S. Choi. 1996. The logical structure of lexical entries for the construction of a machine readable dictionary, *Korean Journal of Cognitive Science*, 7(2): 75-94. [in Korean]

Copestake, A. 1990. "An Approach to Building the hierarchical Element of a Lexical Knowledge base from a Machine Readabl Dictionary", in *Proceedings of the First International Workshop on Inheritance in Natural Language Processing, Tilburg*, The Netherlands, 19-29.

Hausser, R. 1999. Foundations of Computational Linguistics, Springer, Trento, Italy.

Ide, N. and J. Veronis. 1995. "Encoding Dictionaries", in Nancy Ide & Jean Veronis *Text Encoding Initiative*, Kluwer Academic Pub, 167-179.

Jeong, H.W. and A.S. Yoon. 1998. "Implimentation of multi-lingual Information Structure in Korean Environment ", *Proceedings of '98 Joint Conference on Korean Language Processing Celebrating the Han-Geul Day*, 198-203 [in Korean]

Jeong, H.W. and A.S. Yoon. 1999. Designing XML Document Structure for Electronic Dictionary developing, *Proceedings of '99 Spring Conference on Cognitive Science* , 172-177 [in Korean].

Kang, B. M. 1996. Using the TEI scheme in compiling a Korean dictionary, *ALLC-ACH 1996*, Joint International Conference.

Laurent, S. 1998. *XML: a primer*, Press: Forster City , California.

Svensén, B. 1993 *Practiced Lexicography - Principles and Methods of Dictionary making*. Oxford, Oxford University Press, New York.

The Unicode Consortium, *The Unicode Standard version 2.0*, Addison Wesley Pub., New York.

Yoon, A.S. and alii 1999. *Development of Internet-based multilingual multimedia electronic dictionaries*, Technical Report, Ministry of Information and Communication [in Korean].

Yoon, A.S. and alii 2000. Building reusable contents for electronic dictionaries, *Proceedings of SICOLC 2000*, 320-330.

Zinglé, H. 1999. *La modélisation des langues naturelles: Aspects théoriques et pratiques, Travaux du LILLA*, Numéro spécial, Publications de l'Université de Nice-Sophia Antipolis: Nice, France [in French].