Resource Sharing System for Humanity Researches

Shoichiro Hara, Hisashi Yasunaga

National Institute of Japanese Literature
1-16-10, Yutaka-cho, Shinagawa-ku, Tokyo 142-8585, Japan
hara@nijl.ac.jp, yasunaga@nijl.ac.jp

Abstract

The NIJL has developed variety kinds of databases, i.e., catalogue databases, image databases, movie databases, and full text databases. As these systems have been developed under different backgrounds, users have to learn different command for each database. Furthermore, although some databases have similar contents, users cannot access related information unless they understand NIJL database system well. This paper describes NIJL's new resource sharing system, called "NIJL Collaboration System," to solve above problems.

The "NIJL Collaboration System" is an ongoing project involving data conversion to XML and developing platform independent data manipulation system for a distributed environment. The essential of the project is to introduce XML as a common data description, Dublin Core meta-data as a common access points to databases, and Z39.50 as a common searching protocol. This system enables users to access various sorts of multimedia data in distributed databases on the WEB seamlessly by a single graphical user interface.

1. Introduction

The National Institute of Japanese Literature (NIJL) was founded in 1972 (NIJL, 1997). Its mission is to survey the bulk of printed and handwritten texts from the beginning of the country, to collect both originals and microfilm reproductions for preservation, and also to provide public access to original documents. The NIJL has developed variety kinds of databases, i.e., catalogue databases, image databases, movie databases, and full text databases.

Early NIJL information system was composed of a mainframe computer and a network. A feature of the system was that all text processing — including compiling and editing — database services, and publishing were executed on a mainframe. Nowadays, integrated text processing is becoming the norm, but considering that the core of the system was designed more than fifteen years ago, it served its purpose remarkably well. However, over the years, we had encountered an increasing number of problems in the areas of software and hardware. For example, several programs had to be abandoned because they could not run on the latest computer systems owing to a dependence on old devices or lack of vendor support. Another problem was that multimedia databases and internet-based applications are not easy to construct on a mainframe. Furthermore, system maintenance and software development on such a mainframe had become too expensive for small institutes like NIJL.

As a major step toward circumventing these problems and paving the way for future development, NIJL initiated the project named "Digital Library System for Japanese Classical Literature (Adachi, Hara, Yasunaga, Hasebe, Ishizuka, 1999)." This project had taken several years, as it required migrating from the mainframe platform to a distributed system. The project adopted SGML (Standard Generalized Markup Language (ISO, 1986; JIS, 1992) as the basis of data description because SGML data serves as an excellent intermediate document format for storage, interchange, and retrieval. As the result of the project, all catalogue data, image data index, and some full-text data were converted to SGML data, and all database systems were reconstructed on distributed computer systems.

However as each database system has been developed under different backgrounds, users have to learn different commands for each database. Furthermore, although some databases have similar contents, users cannot access related information unless they understand NIJL database system well.

This paper describes NIJL's new resource sharing system, called "NIJL Collaboration System," to solve above heterogeneous data access problems. The NIJL Collaboration System is an ongoing project, involving data conversion to XML and developing platform independent data manipulation system for a distributed environment. The essential of the system is to introduce XML as a common data description, Dublin Core meta-data as a common access points to databases, and Z39.50 as a common searching protocol. This system enables users to access various sorts of multimedia databases in distributed computers on the WEB seamlessly by a single graphical user interface.

In the following discussion, section two presents digital resources in NIJL and describes our proprietary markup rules (KOKIN rules) for the full-text and catalogue databases. In section three, a project "Digital Library System for Japanese Classical Literature" that converted KOKIN-based texts to SGML-based texts is briefly presented. Section four describes the new project named "NIJL Collaboration System" that converts every text and index data to XML data, and unifies all digital resources by Dublin Core meta-data and Z39.50 protocol. Finally, some problems we are faced with are discussed in section six.

2. The Digital Resources in NIJL

The NIJL has digitized variety of sorts of literal resources during twenty years, and has organized as many databases, including catalogues, images, full texts, movies and so on.

2.1. Catalogue Database

At present, all catalogue data have converted to the SGML/XML compliance format for the portability of data transferring and document processing. Following are the lists of our catalogue databases:

1) Catalogue Database Japanese Manuscripts and Printed Books on Microform: bibliographic information about manuscripts and woodblock-printed books held in universities, libraries and archives throughout Japan.
2) Catalogue Database of Japanese Manuscripts and Old Printed Books: bibliographic information about manuscripts and woodblock-printed books held by NIJL.
(3) Catalogue Database of Research Theses in Japanese Literature: bibliographic information about periodicals and bulletins related to Japanese classical literature researches.
4) Union Catalogue Database of Japanese Old Books
5) Database of Historical Material Locations
6) Sharing Catalogue Database of Historical Material
7) OPAC

The record structure of above catalogue databases did not conform to standard catalogue formats such as LCMARC or JPMARC (JaPan MAchine Readable Cataloguing). The primary reason for this is that bibliographical descriptions of Japanese classical materials are not unified. For example, unlike modern publications, a title can appear anywhere (i.e., on the cover, on the first page, on the back cover, on the spine). There may even be multiple, different titles within the same book. A second reason is that most of classical materials are owned by universities, private individuals, temples, and shrines; this information is important. For these reasons, the catalogue format has been modified and expanded.

At the beginning, cataloguers read and extracted bibliographical descriptions from classical materials. The descriptions were inscribed on cards. Instead of the cataloguers inputting the data directly, these cards were sent to a processing company to compile the "Source Catalogue Data." The reason for this was that mainframes had not been designed to facilitate the input of ***kanji*** (Sino-Japanese ideographs), so direct data input by cataloguers was time consuming. Moreover, handwritten ***kanji*** characters are sometimes very difficult to identify; this is one of the reasons why editorial comments are so important. We introduced tags to insert editorial comments and to identify data elements in the Source Catalogue Data. That is, the Source Catalogue Data was marked up according to our own markup rules. These markup rules resembled the KOKIN rules explained in Section 2.3. Tags were used when converting the Source Catalogue Data for use in a relational database or for publishing.

2.2. Image Database for Japanese Classical Literature

Image data comes from "the Japanese Manuscripts and Old Printed *Books* microfilms." As they are held by NIJL, we are not faced with copyright problems. On the other hand, other materials, such as "Japanese Manuscripts and Printed Books on Microform," are not at NIJL. This situation will lead to time-consuming negotiations with the copyright-holders about making the images public on the Internet.

The images are 1-bit monochrome at 600dpi resolution, stored as G4-compressed TIFF files on CD-ROM. At present, we have digitized about 750,000 microfilm frames (about 1200 CD-ROMs), representing about 70% of our microfilms.

The image database is linked with the catalogue database. Database users first consult the catalogue database then access the actual image data in the image database by following the link between the two databases. This link is based on the "call-number" of the materials in both databases. Figure 1 is an example display of the image database system.

A special feature of our system is the way in which we have managed to link the databases for bi-directional searches. We inserted the "call-number" of the original material in each image file (tag 0x10d of "Document Name" in the TIFF specification is used for this purpose). Then, an image browser can access the corresponding catalogue information automatically by selecting the same "call-number" in the catalogue database. Using this, users can first examine the image database to find an interesting picture then they can access the relevant catalogue information by following the link.

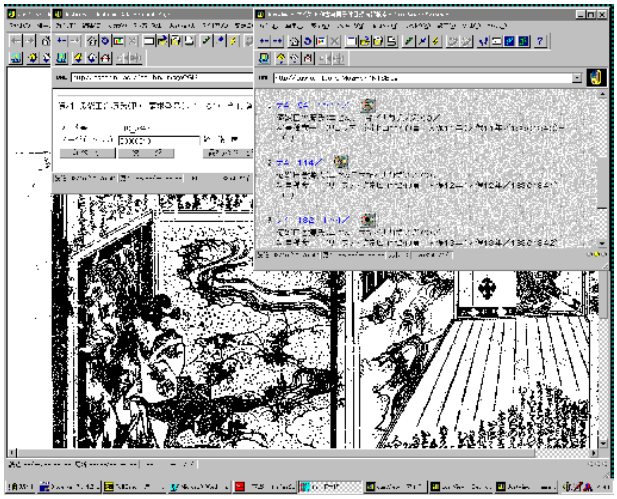


Figure 1: Example of the Catalogue-Image Database

2.3. Full-text databases

One of the major fields of study involving text data is perhaps vocabulary analysis, for which attribute information — such as part of speech and pronunciation — need to be attached to each word in order to create a useful database. There are many convenient lexical analyzers for European languages, but virtually none of these can be used for processing the Japanese language. There are no spaces between words in Japanese orthography, on top of which the prevalence of compounds results in a lack of consensus regarding where spaces might be introduced (for the purpose of analysis). It is thus very difficult to conceive of a system for the automatic division of a text into words, let alone the attachment of appropriate attributes to those words. Moreover, orthography varies from work to work, genre to genre, period to period. Consequently, our materials require different methods of preparation, management, and use of vocabulary indexes. Thus, for intensive computational analysis, manual preparation of a text is necessary.

To overcome these problems and to satisfy researchers' diversified needs, a database should be flexible and

amenable to different purposes and methods. We concluded that a full-text database was appropriate. An ordinary database with fixed word units cannot satisfy researchers' needs, but a full-text database can offer them sufficient flexibility provided that it is left up to the individual researcher to introduce any spaces. Of particular importance was the establishment of markup rules, both for the researchers' convenience and for maintaining data description consistency. We thus developed our own proprietary markup rules

2.3.1. NIJL Specific Markup Rules (KOKIN Rules)

The NIJL has transcribed classical texts in digital form. At the time we began constructing full-text databases, SGML was not popular in Japan, and unfortunately there were no SGML tools that could process Japanese text. For these reasons, we created our own text markup rules that resembled SGML in basic concept. These rules were designed for clarity and ease of use by researchers in the field of Japanese classical literature. We called them the "KOKIN rules" after KOKubungaku (means Japanese literature) INformation ("KOKIN" also suggests the title of a famous Japanese classical poetry anthology) (Yasunaga, 1992, 1996).

Texts contain various kinds of logical elements (i.e., titles, chapters, etc.). A tag is an identifier that marks logical elements of text, while "markup" expresses how a researcher analyses a text. We decided that the definition of text structure and the identification of logical elements should be performed by researchers themselves. However, to standardize the tag-setting for data distribution, we introduced the "**tag rule**" for defining the basic structure of a text. The following is a part of the tag rule definition:

<Logical Record>::=<Tag Begin><Tag><Tag End>|
<TagBegin><Tag><Data><Tag End>
<Tag Begin>::='Japanese-Yen-Mark'
<Tag End>::= 'Star-Mark'
<Tag>::= <Tag Symbol>|<Tag Symbol><Tag Attribute>
<Data>::= <Line>|<Original Data>|<Repeating Symbol><Original Data>
<Line>::= <Original Data>|<Serial Number><Original Data>
<Repeating Symbol> ::= ';'
<Original Data>::= defined in flag rule

The basic syntax of the tag rule is the 'Japanese Yen mark' followed by alphabetic characters (e.g., 'T' for title, 'P' for page, and 'G' for the insert position of a picture). We defined a physical line as the basic text structure and called it "logical record." We then defined a series of logical records as a "logical record set." A series of several lines determining the region of a story is an example of the logical record called "Story." Thus, the KOKIN rules can describe a hierarchical text structure. The tag rule is analogous to SGML elements.

The structure of Japanese classical texts can be thought of as occupying two dimensions — that is, the texts comprise a main text with supplementary text (i.e., annotations, side notes, etc.) placed parallel to the corresponding passages in the main text. The "**flag rule**" was introduced to indicate the starting and ending position of the supplementary texts. In other words, the flag rule was intended to convert a two-dimensional layout to a one-dimensional string by inserting supplementary text into the main text. The following shows a part of flag rule definition:

<Original Data>::= <Flag Begin><Data Element><Flag End><Supplement>|
<DataElement><SpaceFlag> <Supplement><Data Element>|<Data Element>
<Data Element>::= <String>
<Flag Begin>::= '/'
<Flag End>::= '/'
<Space Flag>::= '/'
<Supplement>::= <Right Supplement>|<Left Supplement>|<Bi-Supplement>
<Right Supplement>::= <Supplement Begin><Supplement Element>
<Supplement End>
<Left Supplement>::= <Left Supplement Begin><Supplement Element>
<Supplement End>
<Bi-Supplement>::= <Supplement Begin><Supplement Element> '|'
<Supplement Element><Supplement End>
<Supplement Element>::= <Single Supplement>|<Double Supplement>
<Single Supplement>::= <Supplement Element>
<Double Supplement>::= <Supplement Element><Supplement Separator>
<Supplement Element>
<Supplement Begin>::= '('
<Left Supplement Begin>::= "(|"
<Supplement End>::= ')'
<Supplement Separator>::= '#'
<Supplement Element>::= <String>|<String><String Separator><String>
<String Separator>::= '='
<String>::= Defined in Value Added Rule

The basic syntax of the flag rule starts with a string enclosed by a pair of '/' to indicate the region that is annotated; the following supplementary string is enclosed by '(' and ')'. This approach resembles the MECS system used for the Wittgenstein Archives (Robinson, 1994) and the TEI <app> element (McQueen and Burnard, 1994).

As mentioned above, researchers have to manually separate a text into discrete words before lexical studies can begin. The problem is that the criteria used to identify words differ among researchers. We believed that this difficult task should be performed by the researchers themselves. The "**value-added rule**" was intended to facilitate the process of dividing up a text into words and adding attributes (e.g. pronunciation) to the words in preparation for further analysis. The following is a part of the value-added rule definition:

<String>::=words|
<Value Added Begin>words<Value Added End><Value Added>
<Value Added>::= <Value Begin><Values><Value End>
<Values>::= <Value 1>|<Value 2>|<Supplement Value>|
<Value 1><Binding Symbol><Value 2>
<Value 1>::= Pronunciation of Sino-Japanese Ideographs
<Attribution 2 Begin> Chinese Ideograph<Attribution End>|
<Repeating Symbol><Value 1>
<Value 2> ::= <Attribution 1 Begin><Variation><Attribution End>
<Attribution 2 Begin> Information<Attribution End>|
<Repeating Symbol><Element 2>
<Value Supplement>::= Not Use
<Variation>::= Part of Speech | Name | Location | Position
<Value Added Begin> ::= ' '
<Value Added End> ::= ' '
<Value Begin> ::= '('
<Value End> ::= ')'
<Attribution 1 Begin> ::= ' ['
<Attribution 2 Begin> ::= " [,"
<Attribution End> ::= ']'
<Binding Symbol> ::= '!'
<Repeating Symbol> ::= ';'

The basic syntax of the value-added rule starts with a word enclosed by a pair of blank ‘ ’ and followed by attributes enclosed by ‘(’ and ‘)’. As the identification of words and their attributes depends on the purpose of research, it is impossible to take every possibility into account, and in this sense the value-added rule is incomplete.

Figure 2: KOKIN Document Example

To check the validity of KOKIN rules, we marked up many Japanese classical materials — such as the Anthology of Japanese Classical Literature and the Anthology of Comic Tales — totaling about 150 volumes and 42.5 million characters. As a result, we concluded that KOKIN rules were suitable for transcribing Japanese classical texts. Next, we constructed full-text databases to evaluate the usability of KOKIN documents. We examined three types of full-text databases: one was a CD-ROM database (Kitamura, 1991; Hara 1993); the second was an ordinary relational database; then the third was an SGML/XML (Hara, 1995; Hara, 1996).

Although the validity of KOKIN rules was confirmed, there were still some problems. As KOKIN rules are independent of other standards, there are no off-the-shelf tools to process KOKIN documents. For example, when we evaluated KOKIN documents, we had to construct our own check program (a kind of lexical analyzer to check the sequence of symbols in KOKIN documents). More complicated check programs, such as syntax parsers, were implemented while converting KOKIN documents to SGML documents.

The syntax-check programs revealed another problem: KOKIN rules were constructed on an ad-hoc basis: rules were modified and/or expanded when exceptions were found. This had resulted in some ambiguous symbol sequences in the flag rule and the value-added rule. Thus, while converting KOKIN documents to ordinary relational data, the conversion was restricted to the tag rule level, and symbols involved in the flag rules and value-added rules were left as mere text strings.

2.4. Other Multimedia Databases

A movie database and some utility databases such as ***Kanji*** dictionaries have been constructed.

3. Digital Library System for Japanese Classical Literature

We had been engaged in construction of catalogue databases and full-text databases. Both catalogue and full-text data were structured and marked up according to our proprietary markup rules (KOKIN rules) as described above sections. Thus, if we take “data retrieval” as a search for a string in documents, constructing a database system that uses a string-searching device will be possible. This idea was convenient for those like us with small computer systems who wanted to organize many kinds of databases on one data architecture. However KOKIN rules had some defects in the syntax. Furthermore, as KOKIN rules are NIJL generic rules, we had to create all tools to process KOKIN documents (Herwijnen 1994).

At that time, SGML has come to be widely accepted as an encoding scheme for the transmission of documents between systems. Given these facts, we decided that we should convert our specific-tagged documents to SGML to facilitate effective document management and distribution. As SGML had become popular in Japan, we set up a new project to construct SGML based multi-media database system named “Digital Library System for Japanese Classical Literature” (Hara, 1995; Hara, 1996).

3.1. Document conversion for the Digital Library System

The basic documents conversion procedure from KOKIN to SGML simply involves replacing a string starting with the ‘Japanese Yen Mark’ followed by a “Tag” with the corresponding sequence of START-TAG (<), GI (General Identifier), and END-TAG (>) of SGML. This involved a lexical process, which must also serve to generate SGML starting tags to identify elements marked by the flag rule. These processes result in pre-compiled SGML documents with many omitted tags. A syntactic process then checks the correctness of the pre-compiled documents before converting it to fully tagged SGML documents by referring to the DTD (Document Type Definition). Additional programs were needed to resolve the context-dependent features in the flag and value-added rules. These programs temporarily replace the symbols that cause ambiguity with intermediate non-ambiguous symbols.

The modern Japanese population uses about 2000 characters in daily life. Japanese writing is comprised of ***kanji***, two phonetic syllabaries (*hiragana* and *katakana*), the alphabet (*romaji*), and some symbols. As there are so many characters, 2 bytes are required for encoding (Lunde, 199). Thus, we had to modify the SYNTAX definition in SGML declarations (Bryan, 1988).

3.2. Data Searching

There are many problems in document searching. At first, we constructed full text database using a relational database system. Though a relational database has standard query languages based on elegant mathematical models (i.e., SQL, QBE), the relational database imposes fairly strict restrictions on data structure. In other word, a relational database for structured documents comprise many “pointer tables,” which decreases the searching

effectiveness. An object-oriented data model is suitable for complicate document structures, but it has neither standard search methods nor a standard language such as SQL.

By the way, document searching can be regarded as a "search for a string in a desired element in document." This means that a full text database system can be constructed on a string searching system. In the "Digital Library System for Japanese classical literature," searching a string of interest is a common task. There were fast string searching machines or software, and some of these products could handle Japanese SGML data. Consequently, we conducted new examination of constructing full text database system using string searching software.

3.3. Publishing

SGML documents can be seen as an intermediate documents for storage, conversion, interchange, database publishing, electronic publishing, and publishing. While we intended to realize in-house publishing system of the transcribed materials, there were no DSSSL (Document Style Semantics and Specification Language) compliant tools. Then, we had to create our in-house publishing system that converts SGML documents to LaTeX data to printout text.

4. Unification of Databases

As mentioned above, our databases have been created during twenty years under the different purposes and backgrounds, and each database system is constructed on the different hardware and software. Thus, the following problems have been indicated;

1) Users have to learn different commands for each database.

2) Several data items from the same topic are sometimes recorded on different databases, that is, users have to try to retrieve several databases.

3) Even if the information related to some materials exist in databases in the same institute, it will be difficult for users to find them.

The same situation will happen when users want to access resources distributed on networks. An Internet searching engine such as "YAHOO" is a strong tool to access information resources over the WEB. However, these search engines can only access "Home Pages" and cannot retrieve databases, and furthermore, search engines collect unnecessary data items, i.e., the "precision ratio" is very low.

The "NIJL Collaboration System" has been developed to overcome this ineffective data sharing circumstance. The essential of the system is to introduce XML as a common data description, Dublin Core meta-data and Z39.50 to theoretically unify different databases into as if one database system. This system enables users to access various sorts of multimedia data in distributed databases on the networks seamlessly by a single graphical user interface.

A scenario of the system is that once a user retrieves "GENJI story," he or she can access not only to its catalogue records but also to research papers, images and movie pictures simultaneously. If the same system is introduced to a different institute, users of the NIJL Collaboration System can access resources not only in the NIJL but also in other institute.

4.1. Creation of XML Documents

In the project of "Digital Library System for Japanese Classical Literature," we converted most of our propriety-marked documents into SGML documents. However, as SGML is very large and complex specification, there are no tools to fully process SGML data. For example, DSSSL is style sheet for SGML, but it is very complicate and there are no SGML packages that can support DSSSL. This is the reason we had to make in-house programs to convert SGML documents to LaTeX data for publishing.

XML (eXtensible Markup Language) is based on SGML, defined by the World Wide Web Consortium as a subset of SGML, and intended to be easier for programmers to implement. One good feature of XML for us is that it is compatible with SGML. This enables us to convert SGML documents to other type of documents, i.e., to HTML and the other type of XML documents. In the NIJL Collaboration System project, all data are converted from SGML documents to XML documents mainly by using XSL (eXtensible Stylesheet Language) and XSL Transformation tools. As source documents are described by SGML, conversions are relatively easy. A small numbers of preprocessing programs are necessary, such as to put end tags, to change empty tag format, and to escape characters that are not allowed to use in XML documents.

Figure 3: XML Document Example

While converting SGML documents, we try to introduce standard markups. For example, one peculiarity of Japanese classical texts is that they have many

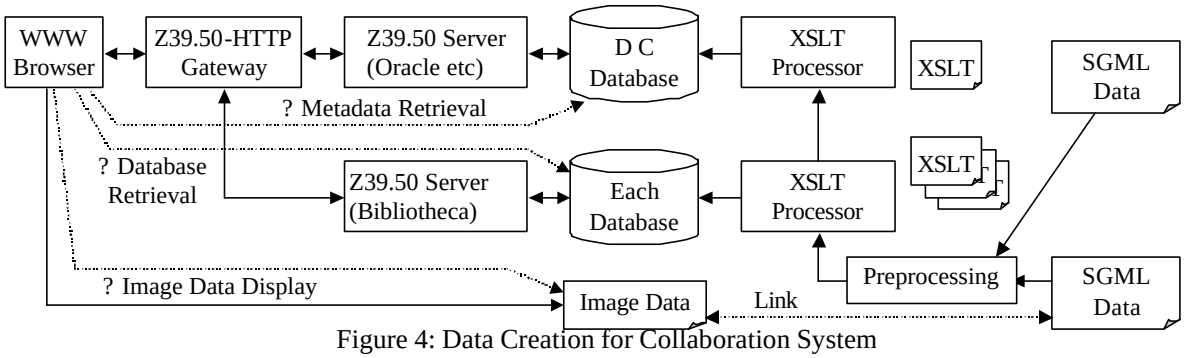


Figure 4: Data Creation for Collaboration System

annotations for the pronunciation of ***Kanji***, notes, and so on. Researchers consider that the digitally transcribed text should preserve all such original features because of their importance for study. KOKIN flag rule was introduced for this purpose. SGML documents created under "Digital Library System for Japanese Classical Literature" markup these annotations by ad hoc way. Recently, as JIS (JIS, 2000) and W3C (W3C, 2001) define these features as "Ruby Annotation," we apply these specifications to XML documents as shown in Figure 3.

In the NIJL Collaboration System, Dublin Core meta-data (Dublin Core Metadata Initiative, 1999) is used as a common access points to databases, which hides the difference of each database record structure. Appropriate record items from each database that have different record structure are extracted, and these items are mapped to Dublin Core meta-data elements. Thus, the Dublin Core meta-data can be used as a gateway to all record items in all databases in the NIJL.

However, Dublin Core meta-data defines only data elements but does not mention about its implementation. That is, Dublin Core meta-data cannot be used as a gateway to different information systems of outside the institute. The solutions might be;

1) data-clearing house
2) standard information retrieval protocol.

A data-clearing house is a repository about databases to accelerate data circulation on networks. Z39.50 is the only international standard protocol for information retrieval independent from information systems (ANSI/NISO, 95). We adopt Z39.50 as a common searching protocol, which hides the difference of each data base operations. Though these two solutions complement each other, we introduce Z39.50, because Z39.50 is only protocol, and this is easier solution than creating data-clearing house.

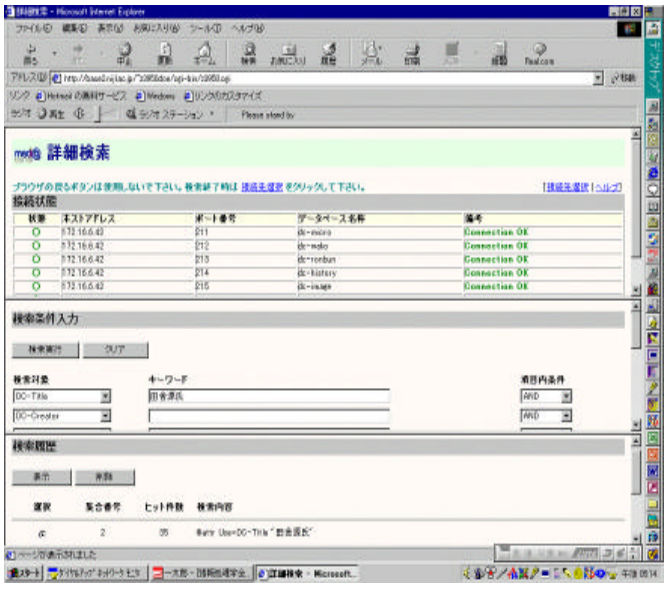
4.2. Structure of NIJL Collaboration System

The NIJL Collaboration System comprises of data conversion subsystem, meta-data generation subsystem, databases, Z39.50 server, and Z39.50-HTTP gateway as shown in Figure 4. The data conversion subsystem converts each SGML document to XML document according to appropriate XSL as mentioned above section. Each converted document is organized in each dull-text database system. The meta-data generation subsystem generates Dublin Core meta-data index for information retrieval and stores then in the Dublin Core meta-database (an ordinal relational database system). Extraction of Dublin Core meta-data index is also done according to XSL.

The Z39.50 server receives protocols, analyzes them, create appropriate query commands, then sent them to Dublin Core meta-database system. The Z39.50-HTTP gateway converts queries from WEB browsers to Z39.50 protocols then sends them to the Z39.50 server. On the contrary, the Z39.50-HTTP gateway converts responses from the Z39.50 server to appropriate HTML document then returns them to WEB Browsers. The peculiarity of the Z39.50-HTTP gateway is that it can reply to requests from more than one Z39.50 servers simultaneously, which enable the NIJL Collaboration System to access more than one database simultaneously. Each database stores document to be retrieved and acts as an independent database system. Document in each database can be also accessed by following links (broken lines indicated as 2 or 3 in the Figure 4) to each database, and these links are embedded in the document from Dublin Core meta-database (broken line indicated as 1 in the Figure 4). Figure 5 is an example of data retrieval by the NIJL Collaboration System.

At present, NIJL Collaboration System includes several catalogue data, images, and movie pictures in the NIJL. Recently, we ask some of the humanity institutes in Japan to construct nationwide resource sharing system based on our collaboration systems.

Figure 5: The NIJL Collaboration System



5. Discussions

We have just built the system and begin data conversion, thus the evaluation is the future task. However, some problems emerge to be solved.

5.1. Non-nesting Structure

The first problem is an illegal text structure to SGML/XML. For example, there are missing pages in the classical materials because of damaged by deterioration, worms and so on. These missing pages can be seen as layout information. In this case, the empty tag to indicate the missing pages is available. However, some researchers consider the missing pages as the essential information of the original materials. The problem is that if an ordinal tag is used to indicate the region of missing pages, this tag would override another region. Following is an example.

……<Chapter> ……. <MissingPahe>
..... *Recovered Text by using Another Materials …*
……
</Chapter>
<Chapter> .....
</MissingPage>
.....

In this case, region <MissingPage> overrides the region of two <Chapter>s. This is the illegal description of SGML/XML.

5.2. Non-standard Characters (Gaiji)

The second problem is ***Kanji***. Some researchers say over 50,000 ***Kanji*** characters are needed to describe Japanese classical texts. However, only 12546 characters are registered as a Japan Industrial Standard (JIS), and only 6355 characters among those are actually available on the computer. Thus, many users have defined their own character set — a so-called external (***Gaiji***) character set. NIJL also has made about 2,000 ***Gaiji*** and more than 10,000 fonts for displaying and publishing. The problem is that most of the client computers on the network cannot display ***Gaiji***.

At NIJL, ***Gaiji*** are registered and coded as 4 hexadecimal digits. We use this code to identify the ***Gaiji*** character in SGML/XML data, that is ***Gaiji*** are described as the External General Entity of SGML/XML — for example, "&K" followed by 4 hexadecimal digits. Thus, a ***Gaiji*** coded as "F4E4" is represented "&KF4E4;" in SGML data. The ***Gaiji*** Entity is processed in two ways. One is to display a ***Gaiji*** character on a WEB browser. In this case, a server-side CGI program used for converting SGML/XML data to HTML replaces the ***Gaiji*** Entity with an image file that contains font-image data in the GIF format. The other is to print ***Gaiji*** in journals and books. In this case, DTP programs, while converting SGML/XML data to LaTeX or other formats, replace the ***Gaiji*** Entity with PostScript data.

At first, the relation between ***Gaiji*** Entity (ex., &KF4E4;) and its image filename or PostScript filename was maintained by a simple table. Both CGI and DTP programs referred only to this table. This table was soon expanded in to the Kanji Server. The Kanji Server is a database that includes many attributes of ***Kanji***, such as the authority of a ***Kanji***, its pronunciation, its structure or shape, ideographic relation, the original image, and so on. Since handwritten characters can be difficult to identify, this database will provide useful information for identifying ***Kanji***.

5.3. Difficulty of Markup

It is indubitable that SGML/XML is a more appropriate document markup language than KOKIN rules from the point of data circulation. However, as SGML/XML description is rather complicate, many humanity researchers hesitate to use SGML/XML for their data construction. One solution is the SGML/XML editor, but we don't have any effective editors yet.

Now, we have the KOKIN-SGML-XML converter. This converter can check the KOKIN documents before converting to SGML/XML documents. In another word, KOKIN rules can be seen as the pre-tagging system of SGML/XML documents. Recently, we think that KOKIN rules should be a good solution to transcribe primary text.

5.4. Mapping Problem

While creating meta-databases, two mapping problems, i.e., mapping from each database to Dublin Core meta-database and mapping from Dublin Core meta-database to Z39.50 attribute set, emerged. At present, mapping rules are ad hoc. As the result, mapped results are different between resemble databases, which will be a risk of decreasing the quality of the NIJL Collaboration System.

One reason for this mapping dispersion is that Dublin Core meta-data has only a few data items, and hence there is more than one interpretation on mapping. To solve the risk, we are trying to introduce "intermediate" data model on each specific research field, and to put the data model between each database and Dublin Core meta-database. As an intermediate data model has enough data items to be mapped from many databases in the specific research field, mapping dispersion will be minimized. The candidates for intermediate data models are meta-data models proposed in each research field, i.e., ISAD(G) for historical archives, MARK for bibliography. Mapping will be done between each database and intermediate data model, and mapping from each intermediate data model to DC meta-data is determined beforehand.

There are two ways for mapping from Dublin Core meta-data to Z39.50 attributes, the one is to map 15 elements of Dublin Core meta-data to appropriate elements of Bib-1 attribute set of Z39.50, the other is to use expanded Dublin Core meta-data elements in Bib-1 attribute set. At present, we use the latter way (Dublin Core Metadata Initiative, 1998).

6. Summary

This paper describes our comprehensive research and development project "NIJL Collaboration System", from digitizing literal resources, text data description by SGML/XML, and to data unification.

We investigated the various functions for the document description by analyzing Japanese classical materials. As the result, we have defined and developed markup schema. Many Japanese classical documents have been electronically transcribed using this schema, and evaluated their availability especially for their application.

Simultaneously, not only text data but also catalogues, archives, images, moving pictures are unified Dublin Core meta-data and Z39.50 protocol. We have just built the

system and began data conversion, and the evaluation is the future task.

7. References

Adachi, J., Hara, S., Yasunaga, H., Hasebe, K. & Ishizuka, H. (1999). Academic Digital Library and Contents in Japan. Literary and Linguistic Computing, pp.131- 145.

ANSI/NISO (95). ANSI/NISO Z39.50-1995 Information Retrieval (Z39.50): Application Service Definition and Protocol Specification

Bryan, M. (1998). SGML: An Author's Guide to the Standard Generalised Markup Language. Addison-Wesley.

Dublin Core Metadata Initiative (1998). Dublin Core and Z39.50. http://purl.org/DC/documents/notes/notes-levan-19980202.htm

Dublin Core Metadata Initiative (1999). The Dublin Core Element Set Version 1.1. last update 1999-07-02",http://purl.org/dc/documents/rec-dces-19990702.htm

Hara, S. and Yasunaga, H. (1993). On the Full-text Database of Japanese Classical Literature. In Joint International Conference ALLC-ACH Conference Abstracts, pp. 61-63.

Hara, S. and Yasunaga, H. (1995). On the Text Based Database Systems for Public Service. In Joint International Conference ALLC/ACH Conference Abstract, pp. 43-45.

Hara, S. and Yasunaga, H. (1996) SGML Markup of Japanese Classical Text: A Case Study. In Joint International Conference ALLC/ACH Conference Abstract, pp. 131-134.

Herwijnen, Eric (1994). Practical SGML. Kluwer Academic Publishers.

ISO 8879 (1986). Information processing: Text and office systems: Standard Generalised Markup Language (SGML).

JIS X 4151-1992 (1992). Standard Generalised Markup Language' [Japanese].

JIS X 4052 (2000). Exchange Format for Japanese Document with Composition Markup

Kitamura, K. and Yasunaga, H. (1991). Data Base Delivery for Japanese Literature by CD-ROM. In Joint International Conference ALLC/ACH Conference Abstract, pp. 261-265.

Lunde, Ken (1993). Understanding Japanese Information Processing. O'Reilly & Associates.

NIJL (1997). National Institute of Japanese Literature 1997.

Robinson, Peter (1994). The Transcription of Primary Textual Sources Using SGML. Office for Humanities Communication Publications, No.6.

W3C (20001). http://www.w3.org/TR/2001/ruby.

Sperberg-McQueen, M. C. and Burnard, L. (1994). Guidelines for Electronic Text Encoding and Interchange (TEI P3).

Yasunaga, H. (1992). Data Description Rule and Full-text Database for Japanese Classical Literature. In Joint International Conference ALLC/ACH Conference Abstract, pp. 234-239.

Yasunaga, H. (1996). Text Data Description Rule for Japanese Classical Literature [Japanese]. In Natural Language, Vol. 3 No. 4, pp. 3-29.