For Internet Minority Languages

A metadata format to include all the language cultures

OHYA Kazushi
Maruzen Co.,Ltd.
ohya@maruzen.co.jp
Graduate School of Natural Science,
Chiba University
ohya@cogsci.l.chiba-u.ac.jp

Syun Tutiya
Faculty of letters
Chiba University
tutiya@chiba-u.ac.jp

Abstract

For the coming global library in the Internet, a metadata format should be provided with a field for reflecting multiple language cultures not to invite persecution for Internet minority languages. In this paper we propose adding a definition of a way of description into the existing metadata format, especially Dublin Core. If not, the coming metadata format will cause a failure that has been brought by MARC-style format.

Keywords: multiple languages, metadata, MARC, Dublin Core, URN, a way of description.

1 Introduction

In this paper we indicate that a definition of a way of description is necessary in a metadata specification, and propose the metadata format. First, we show disregard for minority languages in existing metadata formats. Next, we investigate difficulty in exchanging languages information, and necessities of it. Lastly, we propose our new metadata format and show the sample.

2 Lacks in present global libraries

These days the amount of information we can get from the Internet has increased steadily. It looks as if virtual libraries were realized. But, there is a serious problem of information retrieval. Since HTML is a tag set only for describing layout information, then we have no way to directly access the desired information except for a part of layout information or the whole opened contents. Some solutions to that have been proposed so far, and as concerns text-based data models, network-oriented catalogue data, i.e. metadata formats are viewed as a clue to the problem.

But all the present metadata formats define only what content to be described, does not define how to describe the content. This situation is similar to the case of MARC format. That is, the metadata formats can not be fully applicable to multiple language cultures.

2.1 Defects in MARC

MARC is, although old-fashioned, the only and most stable catalogue format adapted globally. But, MARC has a fatal defect that it is only a definition for content types of descriptions. As the result, many MARC-style metadata format have been generated according to each language culture. And, it has caused difficulty of exchanging the contents of MARC data with other type of MARC format in different languages.

The situation might be accepted in closed usage of local metadata services. But, in the case of global libraries where we can access all of the information resources connected on-line like the Internet, the language-closed metadata format can not ensure global exchange of metadata contents over a wall of language cultural differences.

2.2 Efforts toward multiple languages

If we continue to adopt MARC-style metadata formats including Dublin Core, there will be no choice but to use one specific common language as lingua franca for an information exchange[1].

As such a language, English is the most probable one. There may be no objection to what English is appropriate for such a common language. But, the question is not in that.

The point is that we contribute whether to unify a language culture or to reserve each one[2]. The answer is simple. We are not willing to abandon our mother tongue. Our researcher should endeavor not to invite persecutions that have been caused for native Americans, Siberians, and Ainus, in the Internet world again.

---

[1] One word can convey multiple concepts, and each of the concepts can be indicated by multiple words in different languages. Then, multi-directional translation can not ensure the permanent semantics under multiple steps of translation processes. To exchange and share information, we have to convert all the metadata into one format in one language.

[2] English could fill the role of lingua franca, but essentially it can not be that. Fundamentally all the people who use the same language must have the same right to the existence of the language. But English is not such a language, because it is not a death but a living language. Historically there was no case that living languages spontaneously became new pidgin or pivot languages. Although it may sound strange, living languages do not necessarily permit all the people who use them to decide the change and the existence by themselves. Then, essentially English can not be lingua franca because of its living.
However, in practice, English has already been a native language on computers, precisely native letters; letters in Latin alphabet and some symbols, called ASCII.

3 The way of exchanging information

The means to ensure exchange of information between multiple languages can be divided into the following types.

1. a common protocol.
2. translation from one to one.
3. a common way to describe information.

Whenever something is exchanged, the participants must share a common way to access the targets. Then, the first way is necessary. The second way requires individual operations between specific two languages. It means that the way is difficult to be adopted for multiple language systems. If we utilize the stock of MARC data, we can not help with adopting this way, or converting all the contents by using the last way. The last is a so-called pivot method. In this way, we have to specify what is described in the pivot table [3].

When we prepare a common way of semantic description to share meanings in MARC and Dublin Core, there is an advantage that almost all the people can retrieve the contents in metadata. But, there are disadvantages as follows.

1. There are meanings which can not be precisely represented by the common semantic description.
2. One item can be translated into multiple meanings. It means that the common semantic description does not necessarily represent a common semantic object.
3. If information retrievers know the target language in the above case, the retrieved result has more noise.
4. A metadata maker is burdened with mastering the new common semantic description in addition to the original language.

It means that even if we share one common language like English for semantic description, it is not sufficient for data exchange among multiple languages to share semantics.

Then, when we prepare a common representation of language, there is an advantage that all the languages can be represented. But, there are disadvantages as follows.

1. If information retrievers do not know the target language, they can not retrieve anything.
2. Even then, if the retrievers do not know metadata description styles specific to languages, they can not retrieve anything.
3. Then, it does not ensure the world wide retrieval.
4. In each language, there are problems of the way to apply the common description style to itself.

It means that in order to share information between multiple languages we need both a common semantic notation and a common language representation. As the former there is no choice but English. English is expected to work as a pivot language to represent semantics. A question is what is described as language representations.

3.1 Language representations

To identify languages used for descriptions requires explicating letters, the configuration, and the semantic world in the descriptions[4].

Letters are the primitive units shared on descriptive levels. A set of the letters is called a script. The configuration is a notation, i.e. a rule of letter combinations. And the resultant clusters of letters are called morphemes, which are basic components of the semantic world. Language names can be substituted for the semantic world[5].

In addition to those, in the case of computers, a code name of a script set is needed. To put in order, the required language features are codes, scripts, notations, and language names.

3.2 Unicode

We deplore that many people seem to regard Unicode specification as a breakthrough of script code problems especially observed beyond ISO8859. Unicode is not a coded character set for Digital Libraries but only for computer industries. For example, in the case of Japanese, Unicode can not fully cover the documents published at most one hundred years ago[6].

We are worried that Unicode will prevent ISO2022 from pervading. As tools for Digital Libraries we need a set of common scripts that can be used for an information exchange, and a set of powerful scripts that can be used for describing fully an individual language. ASCII is sufficient for the former. But Unicode is not suitable for the latter[7].

---

[3]Those who think that "semantic interoperability" can be realized by defining only what is described, or a set of meaning seem to misread the "interoperability" in information exchanges. To share semantics, we must define a way of semantic representation in addition to the meaning of the contents. This is not a protocol or a data form like a data representation model defined by MARC and RDF.

[4]Linguistically language can be defined by the five elements; phoneme sets, lexical sets, semantic sets, syntactic rule sets, and pragmatic constraint sets. And, in the case of operations on computers, the elements for detecting language types are primitive symbols, the occurrence rule, and a set of meaning that is used as information for top-down inference.

[5]For example, we can specify languages on descriptions as follows. Script:Kiril, Notation:Kulirof, Language:Yukagiel; Script:Latin, Notation:ISO3602, Language:Japanese; Script:Latin, Notation:Hepburn, Language:Ainu; and so on.

[6]For example, the average rate of uncovered character appearance in the book "Furansu siryaku(French history in brief)" published in 1874 is about 7%.
Here we would like to express our gratitude to Mr. Katano for helping this research.

[7]That is, Unicode is the new for the former.

4 Expansion of descriptive contents with characteristics of scripts

Linguistically scripts have two types; a phonogram and an ideograph. When we operate scripts, we have to consider the difference

Scripts have four features; forms, reading, pronouncing[8], and meanings. And, the features must be observed at the two levels; letter and morphemic levels.

At the letter level[9], in the case of phonogram languages, a relation between a script and its reading is one to one relation. On the other hand, in the case of ideograph languages, the relation can be one to many relation. Then, transliteration can not be made on the base of reading[10][11]. That is, the base for transliteration should be "pronouncing" [12].

Theoretically, pronunciation had better be indicated in IPA. But, without Unicode it would be difficult. Then, in principle, we have to use ASCII to share a description of pronunciation. And, whether we can use IPA or not, the way to translate each phoneme into IPA or ASCII must be shared with all the languages. However each language has a different phoneme system. Then, this can not be a conclusive solution to a pronouncing description.

It means that we need a common and an individual metadata format. The former is for global libraries and the latter for a specific information type [13]. In principle, in the case of the former, we should use ASCII for a pronouncing description, since IPA requires high-trained metadata makers. Then, the former must prepare a place for indicating a script system used to indicate pronunciation.

To put in order, the required descriptive contents for metadata are "meaning" in English, "scripts" in an original language, the "code name", the "notations", the "language name", and "pronunciation" in some script[14] specified by a script identifier.

At the morphemic level, we can handle fundamental units of meaning. Then, in order to exchange information of metadata, the format has to reflect the features observed at this level.

4.1 Observation of Japanese

As concrete examples of morphemic levels, we observe Japanese. In Japanese, relations between a word and its pronunciation are classified as in Table 1. The types below the fourth row are all about a proper noun.

From that, we can say as follows.

1. At least one value of the three features is unique except for a proper noun.
2. In the case of a proper noun the way of identifying indicated targets, or meanings is important.

Then,

1. All the features are permitted to occur multiple times in a metadata format.
2. In order to identity the meaning of a proper noun, a system of unique names is needed.

4.2 Unique names

Essentially a proper name has various meanings[15]. The uniqueness is ensured by human cognition out of the language world. Then, if we need to give an object a unique name, there must be a meta-language system to name it.

And, curiously, in computer science, although all the name of target objects must be unique to be operated, the uniqueness has not been ensured by any computational system but by a rule

---

[8]Phonemic values.

[9]At this level, we do not have to consider meaning of scripts even in the case of ideograph languages. Unicode seems to misread it. A difference of meaning is not necessary to explain unification of Han characters. Theoretically languages exist prior to scripts. That is a reason of being languages without letters or characters. In history many languages have borrowed script systems. Then, according to a rule that "湯" is regarded as the same character and unified within CJK characters(p6-106 in *Unicode Standard Ver.2*), naturally "A(0041, 0391, 0410 in Unicode)s" become to be unified as one letter. It is one of the theoretical inconsistencies in Unicode.
But it does not mean the unification is purposeless. A table of the unification is useful for inter-code retrieval.

[10]According to reading-based transliteration, if the target language has no writing system, we need, as metadata elements, the language name, the first script used to describe it, the notation name of the first description, the present script name(mostly ASCII), and the present notation name.

[11]However, "reading" is an important feature as one way of indicating a letter form in other scripts.

[12]In case of Japanese there can be a one-to-many relation between characters and pronouncing at both letter and morphemic levels, and at the morphemic level sometimes each phoneme does not take the corresponding character. Then, reading of scripts can not work as representation of words instead of pronouncing in Japanese.
One of the drawbacks of "pronouncing" is to easily reflect a difference of dialect or sub-languages. The inconvenience sometimes becomes a reason for language standardization. But, we think that it should be regarded as the benefit of reflecting features of living languages. If we adopt a way of "reading scripts", such features will be left out of metadata contents. For example, in the case of handling a language with a multiple description way, "reading scripts" does not work well. But, "pronouncing" can function as one of the identifiers with meanings and language names.

[13]We believe that the specific metadata format should be defined by those who use the language by themselves. Of course, we well understand that when each language society does not define the specification, the other societies can not start to describe the contents in the language without fears of the modification. Some language society could not define the specification because of lack of social or human resources. But, if we researchers have respect to each language culture, our efforts should be made toward not defining the specification but supporting the defining processes. Certainly it must be a time-consuming process, but, we believe it is a sound attitude.

[14]ASCII will be appropriate to it.

[15]Not philosophically but linguistically, proper names can be used to have various meanings. The nature is called arbitrariness of languages.

	Meaning	Scripts	Pronouncing	e.g.(meaning–scripts–pronouncing)
1	one	one	one	horses–馬–uma
2	one	one	many	bodies–身体–shintai, karada
3	one	many	one	boy's returing home–薮入り, 養父入り, 家父入り– yabuiri
4	one	many	many	(aliases)
5	many	one	one	Mr/s Suzuki–鈴木さん–suzuki san
6	many	one	many	Mr/s No[sz]aka–野坂さん–nosaka san, nozaka san
7	many	many	one	Mr/s Ooya–大矢, 大谷, 大屋, 大家–ooya

Table 1

of practical usage in the real world[16].

New attempts at investigating into unique name systems in computer science can be seen in URN research. We think that URN research is indispensable for defining metadata formats, and what unit a URN is assigned to is significant for metadata research.

According to RFC1737, URNs are identifiers of information units[17]. If the unit is an abstract meaning unit, metadata contents need using with it, because information on data types, dates of digitization, and the like are important for selections of data objects from URNs[18]. If the unit is not an abstract but a concrete data unit, metadata contents are also needed in this case, because they are necessary to identify an abstract meaning unit. That is, URNs are necessarily used with metadata, and what unit a URN is assigned to affects what function is required for metadata[19].

4.3 Distinctive feature sets as unique names

In this paper we propose a definite number of metadata elements to be used for indicating unique names.

Theoretically abstract names as proper names can have the ability to distinguish a specific object from others and to identify the object. In natural languages the two functions are ensured by human cognition. On the other hand, in artificial languages especially to be used on computers, the functions must be ensured by some system.

As we observed, a content that can be given a proper name can not be identified only by descriptions of language representation. The way of language representation ensures only discriminability of the content in multiple languages. Identifying the content requires meta-language systems to represent semantics. Such a content unit is an independent data unit that can take own metadata. That is, in principle, a system of abstract names as unique names should be provided with a function of identification primarily.

It is possible for a set of metadata elements to work as distinctive features of URNs. For example, an element set of Dublin Core can be used for that, and, in principle, the contents are described freely without reserved names or pre-defined glossaries. This way does not necessarily ensure name resolutions, but ensure distinctive ability. The name resolution can be left to URN systems[20].

5 Proposals

In this paper we propose that a metadata specification including Dublin Core should define not only the content type but also the way of the description to reflect multiple language cultures. In such a metadata format, there are attributes for language features(namely languages, script codes, and notations) and elements for the meaning and the pronunciation in addition to original descriptions.

In the following explanation, we presuppose one metadata model called ETO(entrance data units to objects). It is a data model for ensuring a relationship between related data units[21]. In this model, all the independent data units are supposed to be handled through ETO.(cf. Ohya and

[16]The system may only warn us of errors in uniqueness rules or do nothing, and sometimes stops.

[17]In RFC1737, "A URN identifies a resource or unit of information. It may identify, for example, intellectual content, a particular presentation of intellectual content, or whatever a name assignment authority determines is a distinctly namable entity." What is a namable entity is important in practical usage.

[18]It means that URN services should be provided with a function of showing metadata information. But we do not know whether it corresponds to URCs or not.

[19]RFC1737 introduces ISBNs as examples of URNs. And, RFC2288 explains that URNs can support ISBN naming systems. But, we think there should be more consideration.
ISBNs are assigned to material books as units of goods. For example, hard covered and paper books can take different ISBN numbers. It means that ISBNs are not purely assigned to content-based units. However, in RFC2288 ISBNs are regarded as identifiers of abstract meaning units. If so, we would handle all the information of different data types with one ISBN as the same object. It means that the different ISBNs can indicate different material books and concurrently the same information unit. Then, since URNs can take their aliases, we must establish a way of ensuring the sameness of the aliases in URNs.

[20]If a set of metadata elements functions as URNs, it need to be in an independent entrance data unit to target resources. Then, if we use URNs as a primitive way for data management, all the target resources must be handled through a hub-document with URNs and a common metadata element set(Ohya and Tutiya 1997, 1999).

[21]One of the shortcomings of present specifications of text-based data model like SGML/XML is that there is no common mechanism to ensure a relationship between related objects, e.g. SGML declarations, DTDs, and SGML/XML instances. RDFs also may not have such a function, although it is not fixed yet. Concerning application independency of text-based data models, main and meta contents in one data unit should be separated if at all possible. If so, we need new mechanism to ensure the relationship. That is an ETO.

Tutiya 1999)

5.1 A definition of contents

The following is a definition of metadata contents[22].

```
<<Entrance data unit To Objects.>>
ETO ::= IDs, MDs, LINKs
IDs ::= URNs, URLs, CMD
URNs ::= urn+
URLs ::= url+
CMD ::= TITLE?, CREATOR?, SUBJECT?, DESCRIPTION?,
        PUBLISHER?, CONTRIBUTOR?, date?, TYPE?,
        format?, identifier?, SOURCE?, language?,
        relation?, coverage?, RIGHTS?
TITLE ::= title_original+, title_meaning*,
          title_pronouncing*
CREATOR ::= creator_original+, creator_urn*,
            creator_pronouncing*, creator_alias*
SUBJECT ::= subject_original+, subject_meaning*,
            subject_pronouncing*
DESCRIPTION ::= description_original+,
      description_meaning*, description_pronouncing*
PUBLISHER ::= publisher_original+, publisher_urn*,
            publisher_pronouncing*, publisher_alias*
CONTRIBUTOR ::=
        contributor_original+, contributor_urn*,
        contributor_pronouncing*, contributor_alias*,
TYPE ::=
    type_original+, type_meaning*, type_pronouncing*
SOURCE ::= source_original+, source_meaning*,
           source_pronouncing*
RIGHTS ::= rights_original, rights_urn*,
           rights_pronouncing*, rights_alias*
MDs ::= md*
md ::= {TEI Independent Headerfile and other locally
        specified metadata format.}
LINKs ::= link*
link ::+ {declarations of referred actual data units.}
```

Members of CMD(core metadata) can be divided roughly into four groups[23]. One group consists of TITLE, SUBJECT, DESCRIPTION, and RIGHTS. Descriptions of the contents depend on characteristics of languages. Another group consists of CREATOR, PUBLISHER, and CONTRIBUTOR. This group represents unique objects in the real world. Then, there is no need to indicate semantics, but unique identifiers may be needed. The third group consists of TYPE and SOURCE. This group has some problems[24]. The last consists of the other elements; `date`, `format`, `identifier`, `relation`, and `coverage`. This group requires defining rigid data forms in specifications.

5.2 A definition in DTDs

The following DTD is for the example introduced later[25].

```
<?xml version="1.0" ?>
<!DOCTYPE ETO [
<!ELEMENT ETO (IDs, MDs, LINKs)>
<!ATTLIST ETO datatype CDATA #IMPLIED
              codetype CDATA #IMPLIED
              lang CDATA #IMPLIED
              version CDATA #IMPLIED >
<!ELEMENT IDs (urns, urls, cmd)>
<!ELEMENT urns (urn+)>
<!ELEMENT urn EMPTY>
<!ATTLIST urn id ID #IMPLIED
              server CDATA #IMPLIED
              type CDATA #IMPLIED
              content CDATA #REQUIRED >
<!ELEMENT urls (url+)>
<!ELEMENT url EMPTY>
<!ATTLIST url type (local | official ) #REQUIRED
              content CDATA #REQUIRED
              id ID #IMPLIED >
<!ELEMENT cmd (meta+) >
<!ATTLIST cmd type CDATA #REQUIRED
                   "extended dublin core"
              schemelocation CDATA #REQUIRED
              code CDATA #IMPLIED >
<!ELEMENT meta EMPTY>
<!ATTLIST meta name CDATA #REQUIRED
               type CDATA #IMPLIED
               lang CDATA #REQUIRED
               script CDATA #IMPLIED
               notation CDATA #IMPLIED
               olang CDATA #IMPLIED
               content CDATA #REQUIRED >
<!ELEMENT MDs (md*)>
<!ELEMENT md EMPTY>
<!ELEMENT LINKs (link+)>
<!ATTLIST LINKs relationtype CDATA #REQUIRED
                type (real | ETO) #REQUIRED "real" >
<!ELEMENT link EMPTY>
<!ATTLIST link datatype CDATA #IMPLIED
               role CDATA #REQUIRED
               locationtype CDATA #REQUIRED
               location CDATA #REQUIRED
               server CDATA #IMPLIED
               name CDATA #IMPLIED
               show CDATA #FIXED "embed"
               actuate CDATA #FIXED "auto" >
 ]>
```

In the DTD, the names of contents defined in the last section not appear for convenience. In actual, the members should be as alternatives in a position of attribute types at an attribute `name` of an element **meta**. And, a declaration of script code names is also ignored in this XML DTD.

We treat actual data units as referents of links. That is, an element **link** is automatically embedded into an ETO instance according to a value `embed` in an attribute `show`[26].

If a core metadata set is used as a part of URNs, since PUBLISHER and RIGHTS are changeable, they may had better not be in IDs but in MDs.

---

[22] Another of the shortcomings of SGML/XML is that there is no common way to define semantics of descriptions and contents. DTDs are definition formats only for syntactic features. It can not be used for defining styles of descriptions, semantics of the contents, or data forms. So, the following content definition is in our private way.

[23] It is different from the grouping in RFC2413, which takes three groups

[24] TYPE is ready for a category of resources, e.g. poem, novel, and so on. This kind of categories is easily affected by language cultures. For example, is Haiku a poem? What type is Kabuki? In addition, the present definition of TYPE is too ambiguous. For example, what type is E-journal by email? To speak frankly, we are not sure that TYPE works as identifiers.
SOURCE is for a unique identifier of the last original resource. It relates to what object URNs indicate. In this paper we propose the uniqueness of data units is ensured by using a set of descriptions. According to this position, SOURCE becomes too complex descriptions to be used as an element of core metadata.

[25] We ignore defining details of an element **md** in the DTD

[26] We well know that it includes possibility of leading to a problem of structure collision. What multiple instances are embedded into one ETO requires handling multiple DTDs within one instance. It means that we need a partial DTD for a portion of an instance. And, it leads to a problem what is an independent data unit.

5.3 Examples

According to the above DTD, we can get the following example[27].

```
<?xml version="1.0" ?>
<!DOCTYPE ETO SYSTEM "eto.dtd">
<ETO><IDs><urns>
 <urn id="urn1" server="url:http://grinfs.nd.chiba-u.ac.jp:8080"
       content="urn:local:ohya:cd1" /></urns>
 <urls><url type="official"
            content="http://cogsci.nd.chiba-u.ac.jp/ohya/cd1.xml" /></urls>
 <cmd type="extended dublin core" schemelocation="url:http://grinfs.nd.chiba-u.ac.jp/open/edc.xml">
   <meta name="DC.title" type="original" lang="ja" content="幸福の場所"/>
   <meta name="DC.title" type="meaning" lang="en" content=" A place for happiness"/>
   <meta name="DC.title" type="pronouncing" lang="ja" script="ascii" notation="kunrei"
         content="siawasenoarika"/>
   <meta name="DC.title" type="pronouncing" lang="ja" script="ascii" notation="iso3602"
         content="koufukunobasyo"/>
  <meta name="DC.creator" type="original" lang="ja" content="谷村有美"/>
  <meta name="DC.creator" type="pronouncing" lang="ja" script="ascii" notation="hepburn"
         content="Tanimura Yumi"/>
  <meta name="DC.description" type="original" lang="ja"
         content="シンガーソングライター谷村有美の 11 番目のアルバム"/>
  <meta name="DC.description" type="meaning" lang="en" script="ascii"
         content="Yumi Tanimura's the11th album"/>
  <meta name="DC.description" type="pronouncing" lang="ja" script="ascii" notation="hepburn"
         content="shingahsonguraitah tanimurayumi no juuichibanme no arubamu"/>
  <meta name="DC.publisher" type="original" lang="ja" content="SONY Records"/>
  <meta name="DC.publisher" type="original" lang="ja" script="ascii" content="SONY Records"/>
  <meta name="DC.publisher" type="pronouncing" lang="ja" script="ascii" notation="hepburn"
         content="sonih recohdo"/>
  <meta name="DC.contributor" type="original" lang="ja" content="編曲：清水信之"/>
  <meta name="DC.contributor" type="meaning" lang="en" content="co-produced by Nobuyuki Shimizu"/>
  <meta name="DC.contributor" type="pronouncing" lang="ja" script="ascii" notation="hepburn"
         content="henkyoku:shimizu nobuyuki"/>
  <meta name="DC.date" content="1994-12-01"/>
  <meta name="DC.type" type="original" lang="ja" content="CD"/>
  <meta name="DC.type" type="meaning" lang="en" content="CD"/>
  <meta name="DC.type" type="pronouncing" lang="ja" script="ascii" notation="hepburn" content="shiidyi"/>
  <meta name="DC.format" content="binary/cd-music"/>
  <meta name="DC.identifier" content="urn:local:sony_records:SRC3091"/>
  <meta name="DC.language" content="ja"/>
  <meta name="DC.rights" type="original" lang="ja" content="ソニーミュージックエンタテイメントジャパン株式会社"/>
  <meta name="DC.rights" type="alias" lang="ja" content="SONY Music Entertainment 株式会社"/>
  <meta name="DC.rights" type="alias" lang="en" content="SONY Music Entertainment(Japan) Inc."/>
  <meta name="DC.rights" type="pronouncing" lang="ja" script="ascii" notation="hepburn"
         content="sonih myuzikku entateimento kabusikikaisya"/>
 </cmd>
<MDs><md/></MDs><LINKs><link/></LINKs>  </ETO>
```

The example is a description about a music CD with a Japanese title. A problematic point in the example is whether a company name SONY is Japanese or not[28].

6 Another serious problem on metadata formats

In addition to language representations, metadata formats have problems of compound data units. When the target resource is a compound data unit comprising independent data units, there are following problems.

1. The sub-units of the target resource may have own metadata. If then, what relation can be there between metadata of the top-and the sub-units.
2. Since the sub-units are independent, they can be easily modified out of the control of the parent unit. How revised data can be reflected in the metadata of the top-units.

These topics are beyond this paper.

7 Conclusion

The new metadata format, differing to MARC format, should define not only what content to be described but also how to describe the content, and more prepare a field to reflect multiple language cultures not to invite persecution of minority languages in the global library. In order to do so, we need six parts of descriptions; language names, scripts, the codes, the notation, pronunciation, and the meanings. And more, a definite number of metadata elements should be used as components of URNs.

As you can see in the above example, practically the metadata description may become verbose. But, as a global standard, it is important that the possibility of describing the language features has been prepared. After providing this kind of distinctive feature of languages, metadata services can be realized as reflecting multiple language cultures.

---

[27] In the example, we use XML as meta languages. And the parts of MDs and LINKs elements are omitted for convenience.

[28] In our view, the question is meaningless. Essentially, proper names are independent of language names. It is sufficient for attributes of proper names to specify the name of the script. But it will cause difficulty in identifying language types of a title, for example, which includes only a proper name in other scripts than original.

References

1. R.Daniel, 1997-06, "A Trivial Convention for using HTTP in URN Resolution", RFC2169
2. L.Daigle, D.vanGulik, R.Iannella, and P.Falstrom, 1999-06, "URN Namespace Definition Mechanisms", RFC2611
3. ISO/IEC 2022, 1994, *International Standard, Information technology - Character code structure and extension techniques*, ISO
4. C.Lynch, C.Preston and R.Daniel Jr., 1998-02, "Using Existing Bibliographic Identifiers as Uniform Resource Names", RFC2288
5. M.Mealling and R.Daniel, 1999-01, "URI Resolution Services Necessary for URN Resolution", RFC2483
6. R.Moats, 1997-05, "URN Syntax", RFC2141
7. K.Ohya and S.Tutiya, 1997, "Hub-Metadata (in Japanese)", *IPSJ SIG Notes*, Vol.97 No.33, IPSJ
8. K.Ohya and S.Tutiya, 1999, "A metadata model ETO for image-based digital libraries (in Japanese)", *Proceedings of the 59th IPSJ National Conference*, Vol.3, IPSJ
9. K.Ohya and S.Tutiya, 1999x, "Independent metadata units for core systems of digital libraries", draft, (http://cogsci.nd.chiba-u.ac.jp/~ohya/1999draft1.ps)
10. K.R.Sollins, 1985, *Distributed Name Management*, MIT-LCS-TR-331, MIT
11. K.Sollins and L.Masinter, 1994-12, "Functional Requirements for Uniform Resource Names", RFC1737
12. K.Sollins, 1998-01, "Architectural Principles of Uniform Resource Name Resolution", RFC2276
13. N.Trubetzkoy, 1939, "Gedanken ueber das Indogermanenproblem", *Acta Linguistica*, Vol.1
14. The Unicode Consortium, 1996, *The Unicode Standard, Version 2.0*, Addison Wesley Developers Press,
15. S.Weibel, J.Kunze, C.Lagoze, and M.Wolf, 1998-09, "Dublin Core Metadata for Resource Discovery", RFC2413