ISSN 2320 -2602

International Journal of Advances in Computer Science and Technology (IJACST), Vol.2 , No.8, Pages : 28-34 (2013)
Special Issue of ICCECT 2013 - Held during 16-17 August, 2013, Thailand

Language Supports for Multinational Clinical Trials in CDISC Platform

Jihyeon Yeom, Sooa Jung, Hyeokman Kim
School of Computer Science, Kookmin University, Seoul, Korea
{jhyum, jsa0820, hmkim}@kookmin.ac.kr

Abstract : The CDISC language supports are the ones for identifying languages for CDISC data described in multiple languages. This study addresses issues relevant to the language supports in the CDISC framework, to be encountered during the conduction of multinational clinical trials. In this paper, we propose the extensions of SDTM and ODM for the language supports applied not only to tabulated data in SDTM but also to XML representation of the tabulated data in ODM instances. Specifically, a new special-purpose domain called Language Support (LS) for SDTM, and an ODM extension schema using subtyping or type inheritance mechanism are implemented. According to the extensions, any granule of SDTM data entities such as a single attribute value, a record, a set of records, entire records of a subject or a study and ODM data entities such as an attribute value and a whole content of an element including its children can be described in different languages. The language supports will be essential to increase international adoption and support on the CDISC standard.

Key words : CDISC, clinical trials, language identification

INTRODUCTION

Clinical trials are the most expensive and time consuming stages in the process of drug development. While conducting the clinical trials, tremendous effort and time are needed for design and maintenance of study data, data extraction from operational database, data transformation for submitting to regulatory authorities, and data exchange.

The goal of Clinical Data Interchange Standards Consortium (CDISC) is established to develop and support global, platform-independent data standards to improve ineffective processes of clinical trial studies, and also to generate, exchange, and acquire electronic clinical trial documents [1]. The standards define structure and data format to facilitate access and orientation to the data and analysis tools, which, in turn, requires less training, improves communication and results in faster review cycles [2]. CDISC is developing several standards [3]. Among them, Submission Data Tabulation Model (SDTM) defines a standard format of tabulated datasets for clinical trials, and Operation Data Model (ODM) defines a standard XML representation of the tabulated datasets and its metadata to submit to a regulatory authority.

Special concerns are needed to support multilanguage environment in multinational clinical trials [4], which can be grouped into three categories. First, Case Report Form (CRF) in a multinational trial needs to be translated into the respective languages so that patients, clinical monitors, and investigators have the same knowledge regarding the concepts that the trial is intended to capture. A poorly translated CRF may mislead patients and provide inaccurate or even wrong information to the intended trial. Second, when a clinical trial is conducted in muliti-racial environment, multi-lingual CRF is needed to acquire valid data according to each language. For example, free-text comments in the CRF will be described in the participants and researchers native language. Finally, English terms, such as Electrocardiogram (ECG), can be used for multinational trials conducted in non-English speaking countries, without translation.

From the perspective of Clinical Trial Management System (CTMS), there are many benefits if data collected from the CRF can be described in multiple languages for the same clinical trial as depicted in Figure 1. First, no matter what language is used as a primary language in a CRF, the CTMS can be programmed to automatically generate the submission data package in any target language that the regulatory authority wants. The CTMS can also be programmed to automatically generate eCRF described in multiple desired languages. Relevant datasets including variable names, code lists, questionnaires, and inclusion/exclusion criteria, should be translated into the target languages in advance to enable automatic generation of multi-language eCRF.

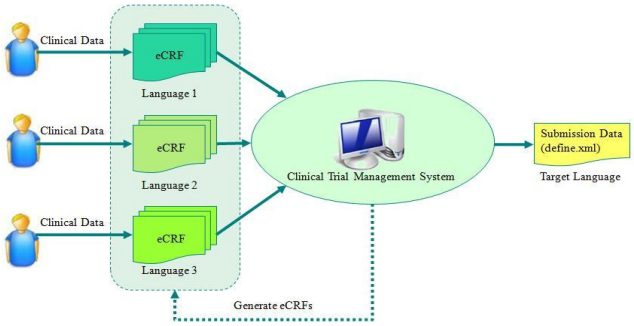


Fig 1: Multi-language identification in the CDISC

This paper addresses the issues relevant to language identification in the CDISC framework, which could be encountered during the conduction of multinational studies. In the documents conformant to CDISC standard specifications, it might be useful or required to identify the natural or formal language in which the content of the document is written. The identification of languages should be equally applied to both tabulated domain data in SDTM

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2012133).

ISSN 2320 -2602

International Journal of Advances in Computer Science and Technology (IJACST), Vol.2 , No.8, Pages : 28-34 (2013)
Special Issue of ICCECT 2013 - Held during 16-17 August, 2013, Thailand

documents and XML representation of the tabulated data in ODM documents. Moreover, the granularity of the content identified might range from a single attribute value of a record in a given domain or of an element in an ODM document, to a set of all related domains or the entire ODM document. A Korean CRF to evaluate the efficacy of red ginseng for the glucose control in type 2 diabetic patients was used to develop multi-language support environment throughout this paper.

This paper is comprised as follows: Chapter 2 addresses issues regarding multilanguage identification of SDTM and suggests how to extend this standard. Chapter 3 addresses obstacles to express multilanguage in ODM and proposes an extension of ODM schema. Finally, Chapter 4 describes the conclusion and future research directions.

LANGUAGE IDENTIFICATION IN SDTM

Introduction to SDTM

The SDTM, which is based on a number of domains, provides the guidance of structure, and format of standard clinical trial tabulation datasets submitted to a regulatory authority (i.e. FDA) [5]. The SDTM defines variables of collected data, which are categorized into the following classes of domains: 1) general-purpose domains representing interventions, events, and observation findings such as vital signs (VS), concomitant medications (CM), and adverse events (AE); 2) special-purpose domains for demographics (DM) and comments (CO), which cannot be extended with any additional variables other than those specified; 3) trial design domains regarding study design such as Trial Summary (TS); and 4) special-purpose relationship domains like supplemental qualifiers (SUPPQUAL). 24 domains are currently specified, and more domains will be continuously added.

In SDTM, observations are listed as the values of each variables defined at specific domains. That means, a domain corresponds to a table and the variables of the domain correspond to columns of the table, respectively. Table 1 shows an example dataset of VS domain in the study on the efficacy of the Ginseng Steamed-Red (GSR). In the table, the column headings are composed of variable names of the VS domain. The table illustrates the use of vital sign variables for three subjects, GSR-005, GSR-006, and GSR-007 (USUBJID). The subject GSR-005 has three observations including weight, systolic blood pressure and diastolic blood pressure, while GSR-006 and GSR-007 have only one observation of weight. The three observations of GSR-005 are identified by the sequence variable (VSSEQ). Each record includes the observation code (VSTESTCD), the actual name of the observation (VSTEST), the category of the observation, and the original result of the observation.

Table 1: Example of VS domain

VS							
STUDYID	DOMAIN	USUBJID	VSSEQ	VSTESTCD	VSTEST	VSCAT	VSORRES
GSR	VS	GSR-005	1	WEIGHT	Weight	Somatometry	60
GSR	VS	GSR-005	2	SYSBP	Systolic Blood Pressure	Cardinal Sign	120
GSR	VS	GSR-005	3	DIABP	Diastolic Blood Pressure	Cardinal Sign	80
GSR	VS	GSR-006	1	WEIGHT	체중	신체계측	65
GSR	VS	GSR-007	1	WEIGHT	체중	Somatometry	50

Language Identifiation in SDTM

The Supplemental Qualifiers (SUPPQUAL) is one of the special-purpose relationship domains. The SUPPQUAL domain is used to capture non-standard variables not presently included in general-purpose domains, and their association to parent records in the general-purpose domains, and allows capturing values for the variables. Because the SDTM does not allow the addition of new variables directly into the domains, it is necessary for sponsors to represent the metadata and data for each non-standard variable/value combination in the SUPPQUAL dataset.

By utilizing the SUPPQUAL domain, SDTM can describe data in multiple languages [5]. That is, adding a new variable for identifying languages into the domain allows the description of data in different languages. Table 2 demonstrates how the SUPPQUAL domain can express data in different languages with the VS dataset of Table 1. In Table 2, the variable RDOMAIN specifies a target general-purpose domain, and the variables STUDYID and USUBJID are identifiers of a study and a subject in the domain, respectively. IDVAR has a variable name in the target domain, and IDVARVAL has a value of the variable described in the IDVAR. By utilizing the combination of the variables IDVAR and IDVARVAL, a record or set of records in the target domain can be specified. So, the pair of two variables is used as a selection condition. Finally, QNAM has a new variable name, VSLANG, for identifying languages for the specified record or set of records, and QVAL has a language code which is a value of the VSLANG. The language code is one of two-letter (ISO 639-1) or three-letter (ISO 639-2) abbreviations for names of languages in the world [6].

For example, the first record in Table 2 shows that the language for the first record in Table 1 (VSSEQ=”1”) of the subject GSR-005 is English (VSLANG=”en”). Similarly, the second record in Table 2 expresses that the language for the second and third records (VSCAT=”Cardinal Sign”) of the subject GSR-005 is also English. Finally, the third record in Table 2 means that the language for the fourth record in Table 1 (VSSEQ=”1”) of the subject GSR-006 is Korean (VSLANG=”ko”).

Table 2: Language identification using SUPPQUAL domain

Supplemental Qualifiers							
STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	QNAM	QLABEL	QVAL
GSR	VS	GSR-005	VSSEQ	1	VSLANG	Vital Sign Language	en
GSR	VS	GSR-005	VSCAT	Cardinal Sign	VSLANG	Vital Sign Language	en
GSR	VS	GSR-006	VSSEQ	1	VSLANG	Vital Sign Language	ko

As shown in Table 2, using the SUPPQUAL domain, SDTM can describe data in different languages for a single record or a set of records in any general-purpose domain. However, SDTM cannot describe data in different languages for a single variable of a record in the domain. For example, the variable VSTEST of the fifth record in Table 1 has a value described in Korean, and the variable VSCAT of the same record has another value in English.

Furthermore, the current version of SDTM cannot describe a default language for a study or a subject explicitly. By describing the default language, the major language for the study or preferred language for the subject will be clearly

ISSN 2320 -2602

International Journal of Advances in Computer Science and Technology (IJACST), Vol.2 , No.8, Pages : 28-34 (2013)
Special Issue of ICCECT 2013 - Held during 16-17 August, 2013, Thailand

identified. Also, this will help to curb the increase in number of describing language identification, thus keeping the number of records in the SUPPQUAL dataset as small as possible.

Extension of SDTM

This subsection proposes three extensions of SDTM to support the language identification problems in SDTM. First, a new parameter "DLANG" for defining a default language of a trial is added into the parameter list for a trial. The variable TSPARMCD in the trial summary (TS) domain will then be able to have "DLANG" to define the default language. Table 3 shows the example of defining a default language of the GSR study as English by inserting the record describing DLANG="en".

Table 3: Defining a default language of a study in TS domain

TS					
STUDYID	DOMAIN	TSSEQ	TSPARMCD	TSPARM	TSVAL
GSR	TS	1	DLANG	Default Language	en

Secondly, the new parameter "DLANG" for defining a default language of a subject is added into the test code list for a subject. The variable SCTESTCD in the subject characteristics (SC) domain will then be able to have DLANG to define the default language. Table 4 shows that the default language for the subjects GSR-005 and GSR-006 is English while that of GSR-007 is Korean.

Table 4: Defining default languages of subjects in SC domain

SC							
STUDYID	DOMAIN	USUBJID	SCSEQ	SCTESTCD	SCTEST	SCORRES	SCSTRESC
GSR	SC	GSR-005	1	DLANG	Default Language	ENGLISH	en
GSR	SC	GSR-006	1	DLANG	Default Language	ENGLISH	en
GSR	SC	GSR-007	1	DLANG	Default Language	KOREAN	ko

Finally, Language Support (LS) domain, a new special-purpose domain for language identification, should be created. Note that current SDTM suggests the creation of new records in SUPPQUAL domain for language identification. A record in the newly suggested LS domain includes a condition for selecting a data entity in a target domain and a language code of a target language for the selected data entity. The selection condition is described by one of the pairs of variables, IDVAR and IDVARVAL, or SEQVARVAL and IDVAR. Note that the new variable SEQVARVAL has the value of the sequence variable XXSEQ defined in all general-purpose domains where the two characters XX stands for a domain prefix defined in SDTM. The selection condition expressed by IDVAR and IDVARVAL specifies a record or a set of records in a target domain, as utilized in the domain SUPPQUAL. Similarly, the other kind of a condition expressed by SEQVARVAL and IDVAR specifies a specific value of a record. With these conditions of the LS domain, users can specify a record, a set of records, or even a single value of a record in a target domain, for identifying the language for the specified data entity. In addition, the language code for the specified data entity is recorded in the variable LANGCD.

Table 5 illustrates how the new LS domain can identify languages for broad range of data entity granules, utilizing the same VS dataset of Table 1. In the first three records of the table, the pairs of IDVAR and IDVARVAL express the selection conditions VSSEQ="1", VSCAT="Cardinal Sign", and VSSEQ=1", respectively, which are the same as Table 2. However, the pair of SEQVARVAL and IDVAR in the fourth record expresses the selection criteria specifying the value of the variable VSTEST of the record selected by the condition VSSEQ="1". So, the fourth record in Table 5 shows that the language for the value of VSTEST of the fifth record in Table 1 (VSSEQ="1") of the subject GSR-007 is Korean (LANGCD="ko").

As shown in Table 5, the proposed LS domain enables users to identify languages for data entities in target domains, ranging from a value of a variable to a record or a set of records. Also, when specifying languages for data entities, the users don't need to create new variables such as the variable VSLANG in Table 2. Usually, the users might feel it is difficult to create new variables as a value of the variable QNAM in the SUPPQUAL domain, rather than just filling data in a predefined format such as the LS domain.

Table 5: Language identification using LS domain

LS						
STUDYID	RDOMAIN	USUBJID	SEQVARVAL	IDVAR	IDVARVAL	LANGCD
GSR	VS	GSR-005		VSSEQ	1	en
GSR	VS	GSR-005		VSCAT	Carinal Sign	en
GSR	VS	GSR-006		VSSEQ	1	ko
GSR	VS	GSR-007	1	VSTEST		ko

LANGUAGE IDENTIFICATION IN ODM

Introduction to ODM

The ODM is a vendor neutral and platform independent data model to generate easily understandable electronic document for acquisition, exchange and submission of clinical study data. ODM adopts XML Schema [7] to define a standard format or schema for representing study data, clinical data, administrative data, and reference data associated with a clinical trial study [8].

Especially, the study data of ODM can define a logical structure of clinical datasets using the element MetaDataVersion in which the datasets are grouped into items, item groups, forms, study events, and protocols in sequence. This logical structure will correspond to a structure of CRF. Also, clinical data can be transformed into a XML instance document according to the logical structure. Therefore, CDISC enables the clinical trial data collected with CRF to be populated into standard SDTM tables. Then, the data of each table can be transformed into a XML instance document conformant to the ODM schema.

Figure 2 is an example of an ODM instance document conformant to the ODM schema. This document has study data for the clinical study GSR. It defines basic information and a logical structure for submission data. The element GlobalVariable includes elements StudyName and StudyDescription, which are the basic information of the study. The element MetaDataVersion shows a single ItemDef, a component of the logical structure. Figure 2 thus shows that this study is designed to test efficacy of Red Ginseng, and submits data for Systolic Blood Pressure.

ISSN 2320 -2602

International Journal of Advances in Computer Science and Technology (IJACST), Vol.2 , No.8, Pages : 28-34 (2013)
Special Issue of ICCECT 2013 - Held during 16-17 August, 2013, Thailand

```
<?xml version="1.0" encoding="UTF-8"?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3">
  <Study OID="GSR">
    <GlobalVariables>
      <StudyName>A study on the efficacy of the ginseng steamed red</StudyName>
      <StudyDescription>Examine the efficacy of GSR on controling blood sugar</StudyDescription>
      <ProtocolName>CDISC-Protocol-007</ProtocolName>
    </GlobalVariables>
    <MetaDataVersion OID="GSR-MDV-001" Name="Version 0.1">
      <ItemDef OID="VS.SYSBP" Name="Systolic Blood Pressure" DataType="integer">
        <Question>
          <TranslatedText xml:lang="en">Systolic Blood Pressure (mmHg)</TranslatedText>
          <TranslatedText xml:lang="ko">수축기 혈압 (mmHg)</TranslatedText>
        </Question>
      </ItemDef>
    </MetaDataVersion>
  </Study>
</ODM>
```

Fig 2: Example of ODM instance

Language Identification in ODM

In ODM, a special element TranslatedText is utilized for identifying languages [8]. That is, if some contents can be described in more than two languages, the contents are tagged with the TranslatedText. The element has a special attribute named xml:lang, which may be inserted in documents to specify the natural or formal language used in the contents and attribute values of any element in an XML document [9]. The values of the attribute are language identifiers defined by IETF RFC 3066 [10]. The language specified by the xml:lang applies to the element where it is specified, the values of its attributes, and to all child elements in its content unless overridden with another instance of xml:lang [9].

In ODM, there are five elements having TranslatedText as a child element; Decode, Description, ErrorMessage, Question, and Symbol. Those elements don't have an attribute, and have only the single element TranslatedText. Figure 3 illustrates, using the element TranslatedText, how to identify two languages, English and Korean, for decoded texts in "Vital Signs Test Code". The figure shows that the elements TranslatedText under a single Decode element represent multiple translated texts having the same meaning. Thus, the element Decode is only used to make a group for the translated texts.

```
<CodeList OID="C66714" DataType="string" Name="Vital Signs Test Code">
  <CodeListItem CodedValue="SYSBP">
    <Decode>
      <TranslatedText xml:lang="en">Systolic Blood Pressure</TranslatedText>
      <TranslatedText xml:lang="ko">수축기 혈압</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue="WEIGHT">
    <Decode>
      <TranslatedText xml:lang="en">Weight</TranslatedText>
      <TranslatedText xml:lang="ko">체중</TranslatedText>
    </Decode>
  </CodeListItem>
</CodeList>
```

Fig 3: Language identification using TranslatedText element

Previously explained language identification in ODM has the following problems. First, according to the definition of a xml:lang [7], it is suggested to insert the attribute directly to an element required to translate, for example, the element Decode in Figure 3. However, as depicted in Figure 3, the language identification for the element Decode using the additional child elements TranslatedText only increases depth of the document, and doesn't have any distinct benefit. Figure 4 shows another example of language identification for the element Decode by inserting the attribute xml:lang directly to the element as suggested in XML, and not utilizing the special element TranslatedText. In the figure, the depth of a document is not increased, and the same meaning can be delivered. Also, the same method can be applied to the other four elements Description, ErrorMessage, Question, and Symbol, as well as to the element Decode.

```
<CodeList OID="C66714" DataType="string" Name="Vital Signs Test Code">
  <CodeListItem CodedValue="SYSBP">
    <Decode xml:lang="en">Systolic Blood Pressure</Decode>
    <Decode xml:lang="ko">수축기 혈압</Decode>
  </CodeListItem>
  <CodeListItem CodedValue="WEIGHT">
    <Decode xml:lang="en">Weight</Decode>
    <Decode xml:lang="ko">체중</Decode>
  </CodeListItem>
</CodeList>
```

Fig 4: Language identification without TranslatedText

Second, elements associated with a string datatype or datatypes derived from the string datatype can have data described in different languages. For example, the elements StudyName and StudyDescription of the clinical study in Figure 2 may be described in more than one language, if the study is conducted in multiple countries. However, the current ODM schema defines both minimum and maximum occurrences of the elements as 1. Therefore, they can appear only once, that is, described in only one language.

Third, attributes having a string value can also have data described in different languages. For example, the value of the attribute Name in the ItemDef, "Systolic Blood Pressure", in Figure 2 may be described in multiple languages. For this, other name attributes for each language would be defined, which might be impossible.

Fourth, the ODM schema doesn't provide the capability for defining a default language for a whole study as well as for a specific subject in a study.

Figure 5 shows an ideal ODM instance which remedies all the problems explained above. The document is derived by modifying the instance in Figure 2, in order to describe data in Korean as well as in English. An attribute xml:lang is inserted to the root element ODM, to represent the default language of this document as English. Also, the elements StudyName and StudyDescription are described in Korean well as in English. The two elements having an attribute xml:lang describe Korean study name and description, and the other two elements having no attribute describe study name and description in the default language, English. The attribute Name of the element ItemDef in Figure 2 is converted to the new element Name in Figure 5, to support multilanguage expression. The converted element Name without the attribute xml:lang describes the item name in the default language, in English, and another element Name having the attribute xml:lang describes the item name in Korean. Furthermore, a language for the element Question is described by the attribute xml:lang instead of its child element TranslatedText. The language for the element Question having no xml:lang is also defined by the default language of the element ODM, that is, English.

ISSN 2320 -2602

International Journal of Advances in Computer Science and Technology (IJACST), Vol.2 , No.8, Pages : 28-34 (2013)
Special Issue of ICCECT 2013 - Held during 16-17 August, 2013, Thailand

```
<?xml version="1.0" encoding="UTF-8"?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3/ideal" xml:lang="en">
  <Study OID="GSR">
    <GlobalVariables>
      <StudyName>A study on the efficacy of the ginseng steamed red</StudyName>
      <StudyName xml:lang="ko">홍삼의 효능 연구</StudyName>
      <StudyDescription>Examine the efficacy of GSR on controling blood sugar</StudyDescription>
      <StudyDescription xml:lang="ko">홍삼의 혈당 조절 효능 연구</StudyDescription>
      <ProtocolName>CDISC-Protocol-007</ProtocolName>
    </GlobalVariables>
    <MetaDataVersion OID="GSR-MDV-001" Name="Version 0.1">
      <ItemDef OID="VS.SYSBP" DataType="integer">
        <Name>Systolic Blood Pressure</Name>
        <Name xml:lang="ko">수축기 혈압</Name>
        <Question>Systolic Blood Pressure (mmHg)</Question>
        <Question xml:lang="ko">수축기 혈압 (mmHg)</Question>
      </ItemDef>
    </MetaDataVersion>
  </Study>
</ODM>
```

Fig 5: Example of an ideal ODM instance for language identification

To generate ideal ODM instances like the one in Figure 5, the ODM schema should be revised to remedy the problems relevant to the language identification. Like many other standard bodies, however, CDISC doesn't allow any revision of the ODM schema without their approval. To revise the ODM schema for private purposes (vendor extensions in ODM terminology), additional extension schemas of the ODM schema should be created separately, and used together with the ODM schema. The following subsections address limitations when extending the ODM schema according to the ways recommended by CDISC, and then propose a new extension schema based on subtyping.

Limitation of ODM Extension Using redefine

The extensions of ODM schema must satisfy the conformance rules or requirements for vendor extensions [8]. The most important requirements for the vendor extension are as follows. First, the new XML elements and attributes can be added, but may not render any standard ODM elements or attributes obsolete. Secondly, all new element and attribute names must use distinct XML namespace to ensure that there are no naming conflicts with the ODM schema and other vendor extension schemas.

To enforce above requirements, every complex datatypes in the ODM schema has two dummy groups, group and attributeGroup for element and attribute extensions, respectively. The dummy groups provide a more constrained "do-it-here" environment for adding content to the existing datatype. They also enable XML Schema redefine to be used as a mechanism for defining extension schema [11].

The XML Schema redefine is a mechanism that redefines a group or attributeGroup of a complex datatype, or redefines an existing simpleType or complexType by extending or restricting the datatype [12]. Especially, ODM recommends redefining an element group and/or an attribute group. This is the reason why the dummy groups are defined in all complex datatypes in ODM.

Redefining an element group of a complex datatype can add new elements or narrow down occurrence ranges of existing elements in the datatype. For example, an occurrence range of an element from 1 to 10 (minOccurs="1" and maxOccurs="10") can be redefined from 2 to 7 (minOccurs="2" and maxOccurs="7"). However, the range cannot be changed from 0 to 7, because it is out of the original range. Similarly, redefining an attribute group can add new attributes or change properties of existing attributes. For example, use property of an attribute can be switched from "optional" into "required".

In order to remedy the problems related with the language identification explained in the subsection 3.2, extending the ODM schema utilizing the redefine mechanism has problems. First, to describe content of a complex datatype in multiple languages without the special element TranslatedText as depicted in Figure 4, the TranslatedText should be deleted from the datatype or can be omitted in its instantiation. The datatype should then be allowed to have a string content with an additional attribute xml:lang which identifies the language of the string content on behalf of the element TranslatedText. By redefining the datatype, TranslatedText can be defined as being able to be omitted in its instantiation and xml:lang added into the datatype, but the datatype cannot have a string content. In order to allow the existing complex datatype to have a string content, the property mixed of the datatype should be set to "true" (mixed="true"). However, the redefine mechanism cannot change property values of a complex datatype though it can change those of an attribute.

Second, in order to describe content of an element having string content like the elements StudyName and StudyDescription in Figure 2 in multiple languages, the upper bound of an occurrence range for the element should be changed into "unbounded", if it is set to "1". The change will allow multiple element contents rather than a single one. However, the redefine mechanism cannot extend the occurrence range though being able to narrow down the range.

Third, there is no way to describe a string value of an attribute of a datatype in multiple languages. In order to identify language for the attribute value, the attribute should not be used or activated, and then a new element having the same name of the attribute be added into the datatype. The new element will have an attribute xml:lang and an occurrence range whose upper bound is set to "unbounded". However, the redefine mechanism doesn't allow deactivation of an attribute in a datatype as well as adding a new element into the datatype at the same time. The deactivation is achieved by restricting the datatype, and the addition by extending the datatype. However, both the restriction and extension to the same datatype cannot be applied in a single redefine element.

Fourth, to define a default language for an entire ODM instance document, an attribute xml:lang needs to be added into the associated datatype of the root element ODM. However, the datatype is defined as anonymous, thus no way to redefine the datatype.

Fifth, according to semantical restrictions of XML Schema, the extension schema having the redefine element should have the same XML namespace of its original schema [12], in this case, that of the ODM schema. Also, all new element and attribute names must use distinct XML namespace, which is the second conformance rule for the vendor extension of ODM [8]. Therefore, if new elements or attributes are required for extending the original schema,

ISSN 2320 -2602

International Journal of Advances in Computer Science and Technology (IJACST), Vol.2 , No.8, Pages : 28-34 (2013)
Special Issue of ICCECT 2013 - Held during 16-17 August, 2013, Thailand

they should be defined in another separate extension schema whose namespace is different from that of the ODM schema. The second extension schema is named a namespace extension schema [11]. In other words, the redefine mechanism in ODM always results in the following two extension schemas; an extension schema having the redefine element, and another namespace extension schema defining new elements and attributes.

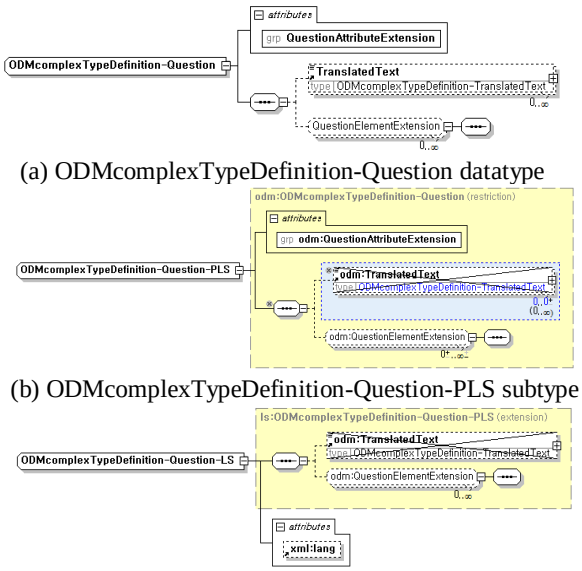
Because of the problems of extending the ODM schema for the language identification, utilizing the redefine mechanism as recommended in ODM is not appropriate for the extension.

ODM Extension Using Subtyping

This subsection proposes to extend the ODM schema for the language identification, based on XML Schema subtyping. The XML Schema subtyping is a mechanism that derives a new `simpleType` or `complexType` by extending or restricting an existing datatype called a base datatype [12]. The newly derived datatype becomes a subtype of the existing base datatype, since it inherits declarations of the base datatype and then adds new attributes/elements to the declarations or constrains the declarations.

In the proposed extension, some datatypes in the ODM schema are extended using the subtyping according to the four cases [13]. In this paper, we describe one representative case. The case is to identify language for element content without utilizing the child element `TranslatedText`. For this purpose, we will derive a new subtype from each datatype having `TranslatedText` in the ODM schema. While deriving the subtype, the `TranslatedText` is set not to appear by restricting the maximum occurrence of the element be "0" (`maxOccurs="0"`). Then, the subtype is allowed to have a string content which was described in the `TranslatedText`, by setting the property `mixed` of the subtype as "`true`" (`mixed="true"`). At the same time, the subtype is allowed to identify a language for the string content by adding an attribute `xml:lang` to the subtype.

For example, Figure 6 shows how the element `Question` having a child element `TranslatedText` in Figure 2 is changed into the one having an attribute `xml:lang` without `TranslatedText` in Figure 5. In Figure 6, each definition of a datatype is graphically presented with a schema diagram generated by XMLSpy [14]. Figure 6.a is a schema diagram of an original datatype `ODMcomplexTypeDefinition-Question`. Figure 6.b shows a subtype `ODMcomplexTypeDefinition-Question-PLS`, which is derived by restricting the maximum occurrence of the element `TranslatedText` be "0". Figure 6.c shows another subtype `ODMcomplexTypeDefinition-Question-LS`, which is derived by extending the PLS subtype. The extension includes changing the property `mixed="true"`, which is not drawn in the schema diagram, and adding a new attribute `xml:lang` to the subtype.



(a) ODMcomplexTypeDefinition-Question datatype

(b) ODMcomplexTypeDefinition-Question-PLS subtype

(c) ODMcomplexTypeDefinition-Question-LS

Fig 6: Language identification without using TranslatedText

It should be noted that the subtype LS is eventually derived from the original datatype, but has the interim subtype PLS since both the restriction and extension of XML Schema are not allowed to be applied at the same time. That is, the subtype PLS is derived as an intermediate step. So, the datatype PLS can be prohibited not to be utilized in instance documents by setting the attribute `abstract="true"`. Also noted that the original datatype was defined in the ODM schema, and the two derived subtypes PLS and LS are defined in a separate extension schema.

Also noted that, in Figure 6.a and 6.b, the attribute group `QuestionAttributeExtension` is a dummy group without any attribute. If more than one attribute is actually defined in a subtype such as in Figure 6.c, the dummy attribute group is not depicted in the schema diagram of XMLSpy.

CONCLUSION

Current CDISC has many restrictions for describing data in multiple languages, which might prevent sponsors from conducting multinational studies or national studies involving multinational patients in CDISC platform. Among various CDISC standards, SDTM and ODM defining contents and format of clinical trial data are lacking the language supports.

In this paper, we propose the extensions of SDTM and ODM to overcome the current problems relevant to the language supports, that is, identifying languages for the data described in multiple languages. Specifically, a new special-purpose domain called Language Support (LS), which has the structure and access method similar to the existing domains, and two global parameters are proposed for identification of languages in SDTM. The proposed domain and parameters allow users to identify languages of the data at any data granule they want, for example, from a single attribute value of a record to a record, a set of records,

or all records of a subject or a study. Also, an ODM extension schema is proposed to support the multilanguage expressive power of the new LS domain and parameters. With the schema, any granules of ODM instance, from an attribute value to a whole content of an element including its child elements, can be described in different languages.

The extended CDISC model adopting the language identification will also be a basis for automatically translating an electronic CRF (eCRF) described in a language into other language versions. We will investigate techniques that do the automatic translation in future work.

REFERENCES

[1] Wayne R. Kubick, Stephen Ruberg, Edward D. Helton, "Toward a Comprehensive CDISC Submission Data Standard," Drug Inf Journal, Vol. 41, No. 3, pp.373-82, 2007.
[2] W. Kuchinke, S. Wiegelmann, P. Verplancke, C. Ohmann, "Extended cooperation in clinical studies through exchange of CDISC metadata between different study software solutions," Methods Inf Med, Vol. 45, No, 4, pp.441-6, 2006.
[3] Tammy Souza, Rebecca Kush, Jullie P. Evans, "Global clinical data interchange standards are here!," Drug Discovery Today, Vol. 12, No. 3/4, pp.174-81, 2007.
[4] Hsin-Tsung Ho, Shein-Chung Chow, "Design and analysis of multinational clinical trials," Drug Inf J., Vol. 32, No. 4, 1998.
[5] Study Data Tabulation Model Implementation Guide: Human Clinical Trials, V3.1.1, CDISC Standard, 2005.
[6] Codes for the representation of names of languages, ISO Standard
[7] XML Schema Part 0: Primer, Second Edition, W3C Recommendation, 2004.
[8] Specification for the Operational Data Model (ODM), Version 1.3, CDISC Standard, 2006.
[9] Extensible Markup Language (XML) 1.0, Third Edition, W3C Recommendation, 2004.
[10] RFC 3066: Tags for the Identification of Languages, IETF (Internet Engineering Task Force), 2001.
[11] Specification for the Operational Data Model (ODM), Version 1.2.1, CDISC Standard, 2005.
[12] Jon Duckett, Nik Ozu, Kevin Williams, Stephen Mohr, Professional XML Schemas, Chicago: Wrox Press Ltd., 2001.
[13] Jihyeon Yeom, Hyeokman Kim, "ODM extension for language supports," Technical Report, Kookmin University.
[14] Altova GmbH, XMLSpy Enterprise Edition User Manual. 2005.