Define-XML 2.0.0　解説

注意：本文書はとあるサークルが作成した私家的なメモです。内容には意図しない誤りが存在する可能性があります。本文書の使用は自己責任にて行って下さい。本文書の使用にて発生したあらゆる損害に対して、当サークルは一切の責任を負いません。また、本文書に関する質問に対して当サークルは基本的に回答いたしません。しかしながら、コメントや提案を歓迎しております。

現バージョンの公開は以下のサイトに限定しています。
http://doubledealer989.blog74.fc2.com/blog-category-14.html

1 はじめに

このテキストは Define-XML 2.0.0 のドラフトバージョンを元に作成した解説です。このテキストでは、読者に以下の知識がある前提で話を進めます。

- SDTM を理解している。または SEND・ADaM を理解している
- XML の基礎知識を有している

上記二つの条件を満たさない場合、これから先を読むことに意味はないでしょう。もし、あなたが CDISC 標準の知識を得たいなら、ためらわずに CDISC が発行している文書を精読して下さい。もし、XML の勉強をしたいなら、オライリー社の「XML 入門」を推薦します。

1.1 Define.XML 2.0 の配布パッケージに関する注意

読者の皆さんは Define.xml 2.0 のパッケージを CDISC の公式サイト(http://www.cdisc.org/)から入手することができます。このパッケージには、Define.xml の仕様解説書と実サンプルが梱包されています。以下にパッケージの概要を図示します。例えば、SDTM 向けの Define.xml のサンプルは SDTM フォルダの中にあります。Define.xml を学習する最もよい方法の一つは、サンプルから自分が学びたい箇所を見つけ、それを読み解くことです。サンプルを有効に使って効率的に学習してください。



† : SDTM フォルダの中には Data Reviewers Guide のサンプルが含まれています。FDA は Reviewers Guide の提出を求めており、サンプルとしての実用度を挙げる意味で、このファイルは重要です。しかしながら、この Reviewers Guide の内容には慎重な態度を取るべきです。なぜならば、Data Reviewers Guide のフォーマットは未確定だからで

す。より現実的で最新のフォーマットは、PhUSE の公式サイトにあります。

2 イントロダクション

Define-XML は、SDTM・SEND・ADaM を FDA に提出する際に求められる付属文書です。Define-XML の実体は XML ファイルです。このファイルには、提出データセットの内容に関する説明を記述します。例えば、データセットの一覧・データセット内の変数の一覧・コードリストの一覧などです[†1]。また、ハイパーリンクを利用して、必要な情報へ適宜アクセスできる工夫を盛り込みます。

define.xml は ODM（CDISC Operational Data Model）の拡張形式に当たります。ODM はデータの交換・保存を目的として作成された形式です。これを申請用にカスタマイズしたものが define.xml に当たります。ODM の基本構造を知りたい場合、ODM の規格を参照して下さい。

†1：このような、データに関する説明のことをメタデータと呼びます。

2.1 XML ファイルか PDF ファイルか

FDA へ電子的に臨床試験データを提出する場合、メタデータの添付が必須です。1999 年の時点では、メタデータの形式として PDF ファイルが指定されていました。2005 年になると、Study Data Specification という文書にて、define.xml の使用が推奨されます。2011 年に発行された CDER Common Data Standards Issue では、define.xml が重要な意味を持つことが記述されています。

歴史的には PDF ファイルから始まり、続いて XML ファイルが推奨されています。XML ファイルには以下のような利点があります。

- ファイル内の情報がツリー構造を取っており、プログラムで処理できる
- 適切なスキーマを提供することで、バリデーションが取れる
- スタイルシートを適用することで、人間が読みやすい形式に変換できる

2.2 ODM について

ODM(Operational Data Model)とは、システム間でデータを交換するための標準です。ODM が取り扱うデータは広大です。臨床データ・メタデータ・管理データ・参照データなどを対象としています。ODM は拡張性を備えており、その規則に従えば自由に追加データを取り扱えるようなります。

Define.xml は、この拡張性を利用して開発されたものです。この文書では ODM の詳細に触れません。詳細を知りたい場合は、別文書を参照してください。

3　XML ファイルの基本要件

3.1 基本要件

まず、define.xml の基本的な要件を確認しましょう。

- Define.xml は ODM v1.3.2 をベースに作成されています。したがって、ODM v1.3.2 の規格をすべて満たす必要があります。
- 名前空間は　http://www.cdisc.org/ns/def/v2.0　です。
- ODM v1.3.2 の定義により、ユーザーは任意に Define.xml を拡張できます。しかし、独自の拡張に起因する問題の責任は、拡張を行った側にあります。また Define.xml に拡張を行った場合、該当するタグは一切解釈されません。
- 拡張を行った場合、次の要件が求められます。
  - 拡張部分の構造を説明すること。拡張の説明は他のすべてのユーザーに対して公開しなければなりません。
  - 拡張時に既存の Element や Attribute の定義をオーバーライドしてはいけません。
  - 必要のない拡張は禁止されています。既存の定義で表現できる情報を、拡張した規格で表現するべきではありません。例えば、独自に拡張した Element を用いて、データセット名を記述することは違反です。
  - 拡張部分を取り去った Define.xml は、Well Formed な XML であるべきです。

3.2 Define.xml の構造

Define.xml の基本構造は次の通りです。灰色のブロックには、提出データセットのメタデータが記述されます。その前後には、XML ファイルとして求められる約束事や試験全体の情報が記述されます。特に灰色のブロックに注目しましょう。次のことが分かります。

- annotation CRF のエリア・データセット定義のエリア・変数定義のエリア…と、情報の種類毎にセクションが設けられています。すなわち、メタデータはカテゴリ毎にまとめて記述されます。
- この構造は論理的ですが、各要素のリンクを張る必要が生じます。このため、define.xml の内部では、各要素に名前が与えられます。名前は OID attribute で定義さ

れます。そして、名前を指定することでリンクが構成されます。当然のことですが、OID で示される「名前」はユニークでなければなりません。

```
<?xml version="1.0" encoding="UTF-8"?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:def="http://www.cdisc.org/ns/def/v2.0"
  FileType="Snapshot"
  FileOID="Studycdisc01-Define-XML_2.0.0"
  CreationDateTime="2012-06-21T11:07:23-05:00"
  Originator="CDISC XML Technologies Team">
  <Study OID="cdisc01">
    <GlobalVariables>
      <StudyName>CDISC01</StudyName>
      <StudyDescription>CDISC Test Study</StudyDescription>
      <ProtocolName>CDISC 01</ProtocolName>
    </GlobalVariables>
    <MetaDataVersion OID="MDV.CDISC01.SDTMIG.3.1.2.SDTM.1.2"
      Name="Study CDISC01, Data Definitions"
      Description="Study CDISC01, Data Definitions"
      def:DefineVersion="2.0.0"
      def:StandardName="CDISC SDTM"
      def:StandardVersion="3.1.2">
        < Annotated Case Report Forms (def:AnnotatedCRF) >
        < Supplemental Data Definitions (def:SupplementalDoc) >
        < Value Level Metadata (def:ValueListDef) >
        < Where Clause Definitions (def:WhereClauseDef) >
        < Domain Level Metadata (ItemGroupDef) >
        < Variable Level Metadata (ItemDef) >
        < Controlled Terminology Metadata (CodeList) >
        < Computational Algorithms (MethodDef) >
        < Comments (def:CommentDef) >
        < Referenced Documents (def:leaf) >
    </MetaDataVersion>
  </Study>
</ODM>
```

3.3 注意するべき全体的な規則

3.3.1 /@OrderNumber

Define.xml では、OrderNumber という Attribute があります。この Attribute は、要素の順番を明確

に記述します。この Attribute を用いる場合、同レベルの全要素に対して適用します。一部の要素に部分的に利用するとエラーになります。

正しい例：

```
<ItemRef ItemOID="IT.STUDYID" OrderNumber="1".../>
<ItemRef ItemOID="IT.AE.DOMAIN" OrderNumber="2".../>
<ItemRef ItemOID="IT.USUBJID" OrderNumber="3".../>
```

誤った例：

```
<ItemRef ItemOID="IT.STUDYID" OrderNumber="1".../>
<ItemRef ItemOID="IT.AE.DOMAIN".../>
<ItemRef ItemOID="IT.USUBJID" OrderNumber="3".../>
```

3.3.2 Def と Ref

「Def」と「Ref」の文字列に注目することで、Define.xml の学習を効率的に進められます。前述の通り、Define.xml ではメタデータの種類ごとに情報が記載されます。一般的にメタデータの実体を記述するタグには"Def”という文字列が含まれます。例えばデータセット定義は次のようになります。

```
<ItemGroupDef OID="IG.AE" Domain="AE" Name="AE" Repeating="Yes"...>
 …
 ＃ここにデータセットの定義が記述されます。
 …
</ItemGroupDef>
```

何かを参照する場合、タグの名前に "Ref”という文字列が含まれます。例えば、変数の定義を参照する場合、次のようになります。

```
<ItemRef ItemOID="IT.STUDYID" OrderNumber="1".../>
<ItemRef ItemOID="IT.AE.DOMAIN" OrderNumber="2".../>
<ItemRef ItemOID="IT.USUBJID" OrderNumber="3".../>
```

3.3.3 /@OID

OID とは Object ID の略です。OID により個々の定義に名前を与えます。この名前を参照することで、Define.xml 内部でデータを相互に参照したり、再利用できます。OID はユニークな値でなければなりません。

具体的な例を挙げましょう。以下の XML では、変数が定義されています。"IT.DM.BRTHDTC”は誕生日という変数に与えられた **Define.xml 上の名前**です。同様に”IT.DM.AGE"は年齢の変数に与えられた名前です。

```
<ItemDef  OID="IT.DM.BRTHDTC"  Name="BRTHDTC" ...> … </ItemDef>
<ItemDef  OID="IT.DM.AGE"  Name="AGE" ...> … </ItemDef>
```

この変数の情報は、データセットの定義から参照されます。その際に OID が利用されます。

```
<ItemGroupDef  OID="IG.DM" Domain="DM"…>
 ...
 <ItemRef ItemOID="IT.DM.BRTHDTC" OrderNumber="8" Mandatory="No"/>
 <ItemRef ItemOID="IT.DM.AGE" OrderNumber="9" Mandatory="Yes"/>
 ...
</ItemGroupDef>
```

3.4 Define.xml の Validation

Define.xml が正規であるかを判断するためには、2 つの原則が必要です。1 つはスキーマに準拠すること、もう 1 つはビジネス上の要件を満たすことです。前者は XML ファイルが Well-formed であることです。後者は FDA の要求や、Define.xml の目的（すなわち提出データセットのメタデータを完全に記述する）を達成することです。

適切な XML エディタとスキーマを用いれば、前者のチェックを簡単に実施できます。また後者の要件のチェックを行うには、OpenCDISC や WebSDM といったバリデータを利用すると効率的です。

Define.xml のバリデーションを実施する際に『大文字・小文字が厳密に使い分けられている』ことに注意してください。例えば、データセットのクラスを表現するリテラルは大文字です。"Findings”は違反です。"FINDINGS”が有効な値とみなされます。