

UBL用のRelax NGスキーマの開発

内藤広志 衣川勇樹 井戸翔子 田口華香 細島 賢

大阪工業大学 情報科学部

1 はじめに

Universal Business Language (UBL)^[1] は、XML を用いた電子商取引の相互運用性の問題を解決するために、(a) 基本的なビジネスプロセスの受発注業務で業界によらず共通に利用できるビジネス文書を定義したスキーマと、(b) ビジネス文書のスキーマを記述するために使用する再利用可能なデータコンポーネントのスキーマを含むライブラリを提供することを目的としている。現在のUBLバージョンでは、(a) と (b) のスキーマはW3C XML Schema (XSD) で記述されているが、UBLのスキーマ設計が特定のスキーマ言語に依存していないことは、より多くの開発ツールを提供するために重要である。

本研究では、XSD言語で書かれたUBLスキーマをRelax NG言語^[3]に変換することによって、UBLスキーマ設計における言語依存性の問題を検証した。

2 UBLスキーマライブラリの構成

UBLスキーマライブラリ (UBLライブラリ) は次の7種類のスキーマから構成される。これらのスキーマ間には、図1のような参照関係がある。なお、CoreComponentParameters と DataTypes は省略した。

- (1) CoreComponentParameters: コンポーネントのドキュメンテーションを記述するためのスキーマ。
- (2) CoreComponentTypes: XSDデータ型をもとに、コアコンポーネント技術仕様書^[2] (CCTS) で定義されているCore Component Typeを定義する。
- (3) RepresentationTerms: (2)を派生させて、ビジネス文書で使用するデータ型を定義する。
- (4) DataTypes: ビジネス文書で使われる付加的なデータ型を定義する。現在は使用されていない。
- (5) Reusable: ビジネス文書を記述するための標準化され、再利用可能なデータコンポーネントのスキーマを定義する。
- (6) Codelist: ビジネス文書で利用される様々なコードを定義するためのフレームワークを提供する。
- (7) maindoc: 一般的な受発注業務で使われる8種類のビジネス文書を定義する。

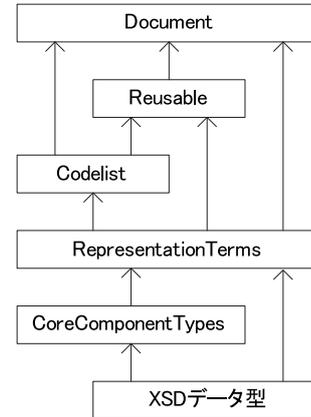


図1: コンポーネント間の参照関係

これらのXSDスキーマは、図2のようにディレクトリに分けて格納されている。「自動生成スキーマ」は、図4のようなスプレッドシートで記述された定義からプログラム (UBLish^[1]) によって生成されたスキーマである。一方、「人手編集スキーマ」は、人間が作成したスキーマで、上記の(1), (2), (3), (4)が該当する。

| ディレクトリ | 人手編集スキーマ | 自動生成スキーマ | 総数 |
|-----------------------|----------|----------|----|
| xsd/codelist/placebo/ | - | 56 | 56 |
| xsd/common/ | 4 | 1 | 5 |
| xsd/maindoc/ | - | 8 | 8 |

図2: UBLライブラリのファイル構成

3 Relax NGスキーマの生成

Relax NGスキーマへの変換は、図3のように、「人手編集スキーマ」はXSDスキーマよりXSLTを用いて行い、「自動生成スキーマ」はスプレッドシートよりPerlを用いて行った。(2)と(5)のスキーマを例にして、XSDスキーマからどのようなRelax NGスキーマを生成したかを説明する。

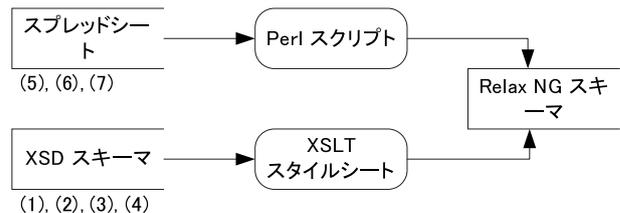


図3: Relax NGスキーマの生成フロー

3.1 CoreComponentTypes

UBLライブラリでは、再利用性を高めるため、グローバル要素として要素とデータ型を定義するスタイ

Translating from W3C XML Schema to Relax NG for UBL Schema Library
Naito Hiroshi, Kinugawa Yuki, Ido Shoko, Taguchi Haruka, Hosohata Ken
Faculty of Information Science and Technology, Osaka Institute of Technology

ル (Garden of Eden^[4]) を用いている。たとえば、次のようにデータ型 MeasureType と、それを参照する要素 Measure のペアをグローバル要素として定義している。データ型は XSD データ型の拡張として定義している。

```
<xsd:element name="Measure" type="cct:MeasureType" />
<xsd:complexType name="MeasureType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:decimal">
      <xsd:attribute name="unitCode"
        type="xsd:token" use="optional"/>
      <xsd:attribute name="unitCodeListVersionID"
        type="xsd:token" use="optional"/>
    </xsd:extension>
  </xsd:simpleContent></xsd:complexType>
```

これを Relax NG では次のスキーマに変換した。XSD スキーマの要素とデータ型のそれぞれを define パターンとして表現する。パターン名に名前空間接頭辞を使うことができないため、cct-Measure のように名前空間接頭辞をパターン名に追加している。XSD データ型は、Relax NG でも利用でき、data 要素の type 属性で指定している。

```
<define name="cct-Measure">
  <element name="Measure">
    <ref name="cct-MeasureType"/></element></define>
<define name="cct-MeasureType">
  <data type="decimal"/>
  <optional>
    <attribute name="unitCode">
      <data type="token"/></attribute></optional>
  <optional>
    <attribute name="unitCodeListVersionID">
      <data type="token"/></attribute></optional></define>
```

3.2 Reusable

CCTS では、ビジネス文書を構成するコンポーネントを Business Information Entity (BIE) と呼ぶ。BIE は単一のデータからなる BBIE、複数の BBIE からなる ABIE などに分かれる。Reusable では、図 4 のようにスプレッドシートをもちいて BIE を記述する。この BIE 情報は、UBLish によって次の XSD スキーマに変換される。BBIE と ABIE とともにグローバル要素として定義される。

| UBL Name | Object Class | Representation Term | Occurrence | BIE Type |
|----------------|--------------|---------------------|------------|----------|
| Dimension | Dimension | | | ABIE |
| AttributeID | Dimension | Identifier | 1..1 | BBIE |
| Measure | Dimension | Measure | 0..1 | BBIE |
| Description | Dimension | Text | 1..1 | BBIE |
| MinimumMeasure | Dimension | Measure | 0..1 | BBIE |
| MaximumMeasur | Dimension | Measure | 0..1 | BBIE |

図 4: Reusable を定義するスプレッドシート

```
<xsd:element name="Dimension" type="DimensionType" />
<xsd:complexType name="DimensionType">
  <xsd:sequence>
```

```
<xsd:element ref="AttributeID"
  minOccurs="1" maxOccurs="1"/></xsd:element>
<xsd:element ref="Measure"
  minOccurs="0" maxOccurs="1"/></xsd:element>
<xsd:element ref="Description"
  minOccurs="1" maxOccurs="1"/></xsd:element>
<xsd:element ref="MinimumMeasure"
  minOccurs="0" maxOccurs="1"/></xsd:element>
<xsd:element ref="MaximumMeasure"
  minOccurs="0" maxOccurs="1"/></xsd:element>
</xsd:sequence></xsd:complexType>
<xsd:element name="MinimumMeasure"
  type="rt:MeasureType"/> ...
```

XSD スキーマの要素とデータ型に対応した define パターンを生成する。

```
<define name="cat-Dimension">
  <element name="Dimension">
    <ref name="cat-DimensionType"/>
  </element></define>
<define name="cat-DimensionType">
  <ref name="cat-AttributeID"/>
  <optional>
    <ref name="cat-Measure"/></optional>
  <ref name="cat-Description"/>
  <optional>
    <ref name="cat-MinimumMeasure"/></optional>
  <optional>
    <ref name="cat-MaximumMeasure"/></optional></define>
<define name="cat-MinimumMeasure">
  <element name="MinimumMeasure">
    <ref name="rt-MeasureType"/></element></define> ...
```

4 UBL スキーマ設計の問題点

UBL の XSD スキーマを Relax NG に変換することができたが、UBL スキーマ設計に次の言語依存の問題点があることがわかった。特に (3) は大規模なスキーマを開発する上で Relax NG に追加すべき機能である。

- (1) 同名の要素とデータ型の宣言があるが、Relax NG では両者を分けて定義できない。
- (2) Relax NG では要素の出現回数の上限を宣言できない。
- (3) 要素とデータ型を含むパッケージを名前空間で識別しているが、Relax NG では define で名前空間を指定できないし、スキーマの import 機能もサポートされていない。
- (4) 派生を多用し、参照した宣言を拡張したり制限しているが、Relax NG では拡張は容易だが制限はサポートされていない。

参考文献

- [1] Meadows, B. et al., Universal Business Language 1.0 Beta Committee Draft, OASIS, 2003.
- [2] UN/CEFACT, ebXML Core Components Technical Specification, 2003.
- [3] OASIS, RELAX NG Specification Committee Specification 3, 2001.
- [4] Maler, E., Schema Design Rules for UBL...and Maybe for You, XML2003, 2003.