

O-030

RELAX NGによるUBLスキーマライブラリの開発 Development of UBL Schema Library in RELAX NG

衣川 勇樹†
Yuuki Kinugawa

内藤 広志‡
Hiroshi Naito

1 はじめに

B2Bにおいて業界を問わず利用可能なビジネス文書のライブラリである Universal Business Language (UBL) が開発されている[1]。UBLはXMLのビジネス文書を作成する際に使用する共通の部品をW3C XML Schema (XSD) で定義している。今回XSDで記述されたUBLスキーマを、シンプルでツール開発が容易なスキーマ言語であるRELAX NG (RNG) [2]で記述するため、UBLのXSDスキーマをRNGスキーマに変換するツールを開発した。

本発表は[3]の研究を元に、最新版のUBLへ対応させ、さらにXSDの派生型の変換とスキーマ情報の追加を行った結果を述べたものである。

2 UBLの構成

UBLは以下の3つを提供している。

- (1) スキーマライブラリ：ビジネス文書の部品（コアコンポーネント）を定義するためのXSDで記述されたスキーマ群
- (2) ビジネス文書：(1)のスキーマを組み合わせて定義された受発注用のXML文書
- (3) 拡張用のガイドライン：(1)のスキーマをカスタマイズし、業界特有の文書を作成するための規定

(1)は次の8種類のスキーマから構成されており、図1のような参照関係にある。⑦と⑧のスキーマには複数のスキーマが存在する。

- ① CoreComponentParameters：コンポーネントの注釈を記述
- ② CoreComponentTypes：コアコンポーネントの型を定義
- ③ UnspecializedDatatypes：②を派生させ、ビジネス文書で使用するデータ型を定義
- ④ SpecializedDatatypes：ビジネス文書で使用する付加的なデータ型を定義
- ⑤ CommonBasicComponents：ビジネス文書に使用する基礎的な共通コンポーネントを定義
- ⑥ CommonAggregateComponents：③④⑤を集約してビジネス文書に使用する共通コンポーネントを定義
- ⑦ codelist：通貨コードなど、様々なコードを選択するリストを定義（13個）
- ⑧ maindoc：各コンポーネントを組み合わせ、標準的なビジネス文書を定義（8個）

図1のとおり、下位のスキーマはより上位のスキーマによって参照され、最上位の⑧maindocスキーマは(2)のXML文書を定義する。なお、①のスキーマは全てのスキーマが

参照しているため図1では省略した。

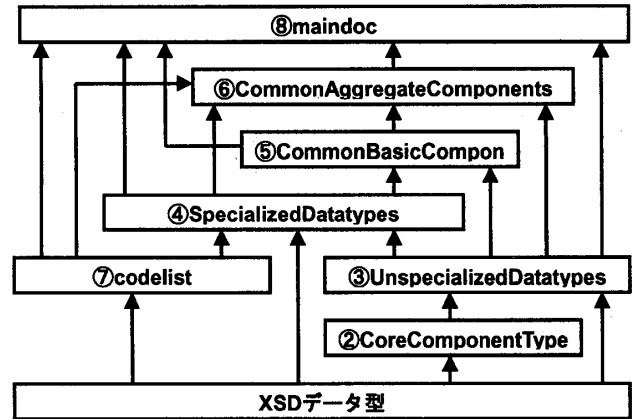


図1 スキーマの参照関係

3 RNGスキーマライブラリの開発

3.1 開発の目的

現在UBLが提供しているスキーマライブラリはXSDで記述されている。XSDはW3Cが勧告したスキーマ言語で、多くの要素や属性を持ち、高い表現力を持っている。しかし、それゆえ構造が複雑で、理解しにくいという欠点もある。そこでスキーマライブラリをRNGに変換するツールを開発した。RNGはOASIS[3]によって標準化されたスキーマ言語で、シンプルで柔軟な構造を持つ。RNGでスキーマライブラリを記述することで、RNGを使用したツールの利用を可能にし、UBLの開発の幅を広げることができる。

3.2 開発の方法

図2に示すとおり、RNGスキーマライブラリはXSDスキーマライブラリをツールで変換したものである。ツールはJavaで開発されており、8種類のXSDスキーマと変換用の設定ファイル（plefix-config.xml）を入力とし、それぞれに対応するRNGスキーマを出力する。

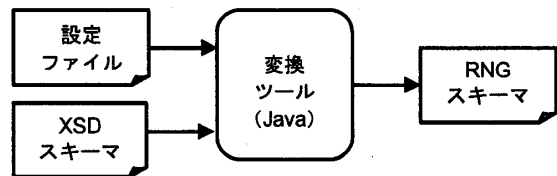


図2 スキーマの変換

† 大阪工業大学 情報科学研究科

‡ 大阪工業大学 情報科学部 情報メディア学科

4 スキーマの変換

4.1 型定義と要素定義の変換

CoreComponentTypes の Measure と MeasureType の定義を例として図3に示す。XSDの要素と型は RNG ではパターンとして定義する。RNGのパターン名には名前空間を設定できないため、"ubl-バージョン-スキーマの接頭辞-要素名(または型名)"として重複を避けた。スキーマの接頭辞は変換用の設定ファイルでスキーマごとに設定しておく。

XSD記述例	RNG記述例
<pre><xsd:element name="Measure" type="MeasureType" /> <xsd:complexType name="MeasureType"> <xsd:simpleContent> <xsd:extension base="xsd:decimal"> <xsd:attribute name="unitCode" type="xsd:token" use="optional"/> </xsd:extension> </xsd:simpleContent> </xsd:complexType></pre>	<pre><define name= "ubl-1.0-cct-Measure"> <element name="Measure"> <ref name= "ubl-1.0-cct-MeasureType"/> </element> </define> <define name= "ubl-1.0-cct-MeasureType"> <data type="decimal"/> <optional> <attribute name="unitCode"> <data type="token"/> </attribute> </optional> </define></pre>

図3 型定義と要素定義の変換

4.2 派生型定義の変換

XSDスキーマは型を派生させて新たな型を定義できる。最下位の派生元はXSDデータ型(例: decimal、string)である。派生には元の型の値や属性値を限定する制限と元の型に内容を追加する拡張がある。どちらも型を参照して新たな型を定義することになる。

RNGは拡張はできるが制限はできない。よって派生は派生元を辿り、最下位までの型情報を全て含んだ新しいパターンを定義する必要がある。

5 スキーマ情報の追加

5.1 バージョニングとカスタマイゼーション

UBLのバージョンニングとカスタマイゼーションはスキーマを派生させて行う。例えば ver.1.1 のスキーマライブラリは ver.1.0 のスキーマライブラリを派生させて作られる。したがって新バージョンのスキーマライブラリは、その前のバージョンのスキーマライブラリを包括していることになる。

5.2 スキーマ情報

異なるバージョンかカスタマイズされたスキーマで取引を行う場合、取引当事者間で共通する最新のスキーマを用いる。つまり ver.1.2 のスキーマと ver.1.1 カスタムスキーマで取引を行う場合、共通する最新バージョンである ver.1.1 のスキーマを採用するということである。これは5.1で述

べたように、バージョンニングとカスタマイゼーションが派生元のスキーマを包括しているため可能となる。

このとき、そのスキーマがどのスキーマを派生させて作られたものかという情報が必要になる。この情報は Post Schema Validation Infoset (PSVI) に含まれ、XSDの特徴となっている。しかし RNG では PSVI の取得はできず、スキーマの派生元を辿ることはできない。そこで RNG スキーマに必要なスキーマ情報を追加した。

5.3 スキーマ情報の追加

追加したスキーマ情報は以下の4つである。

- ① 変換元になった XSD スキーマの型名 AType
- ② AType の派生元となった XSD スキーマの型名 BType
- ③ AType の派生の種類 (拡張か制限)
- ④ BType の名前空間 (スキーマの個有値)

スキーマ情報は RNG スキーマの注釈として追加した。これにより RNG スキーマにおいても派生元スキーマ情報を取得でき、異なるバージョン間で取引を行うことができる。

スキーマ情報の例として UnspecializedDatatypes の MeasureType の RNG スキーマの一部を図4に示す。この例では変換元の MeasureType が CoreComponentTypes (ver.1.0) の MeasureType を制限して定義されたという情報が含まれている。

```
<define name="ubl-1.0-udt-MeasureType">
<a:documentation>
<xsdTypeInfo>
<xsdType>MeasureType</xsdType>
<xsdBaseType>MeasureType</xsdBaseType>
<xsdDerivedType>restriction</xsdDerivedType>
<xsdBaseTypeNs>urn:oasis:names:tc:ubl:
CoreComponentTypes:1.0</xsdBaseTypeNs>
</xsdTypeInfo>
</a:documentation>
```

図4 スキーマ情報

6 まとめ

開発した RNG スキーマライブラリは XSD スキーマライブラリと等価な XML インスタンスを定義できるため、通常取引においては相互運用性があると言える。しかし異なるバージョン間の取引においては、今回追加したスキーマ情報を取得するためのツールが必要となる。このツールを開発し、RNG スキーマを用いて異なるバージョン間で取引を行うことが今後の課題である。

参考文献

- [1] OASIS Universal Business Language TC
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl
- [2] RELAX NG Specification
<http://www.oasis-open.org/committees/relax-ng/spec-20011203.html>
- [3] 内藤広志、衣川勇樹、井戸翔子、田口華香、二宮智映、細島賢：UBL用の Relax NG スキーマの開発
第66回情報処理学会全国大会公演論文集(2004)
- [4] OASIS
<http://www.oasis-open.org/home/index.php>