

国地方連携システムのためのスキーマ設計

村田 真[†]

[†] 国際大学研究所

E-mail: †eb2m-mrt@asahi-net.or.jp

あらまし 「電子自治体のシステム構築のあり方に関する検討会」に関連する総務省の調査研究の1つとして、「国・地方を通じた業務の手順をパターン化して自動化するシステム(連携ワークフローシステム)及び国・地方で共有すべき文書・データの連携を可能とするシステム(連携文書管理システム)に関する調査研究」が平成16年度から2カ年にわたって行われた。連携文書管理システムでは、XMLを用いて文書・データを表現している。本論文では、本事業におけるスキーマ設計、とくに都道府県用スキーマと市町村用スキーマとの共通化について解説する。スキーマのスタイルとして Garden of Eden を採用し、スキーマ変換ツールを作成・使用することによって、スキーマの共通化を行うことができた。

キーワード 電子自治体, 国地方連携, XML, スキーマ, スタイル, RELAX NG, 共通化

Schema Design for Information Liaison between Local Governments and the Central Government

Makoto MURATA[†]

[†] Research Institute, International University of Japan

E-mail: †eb2m-mrt@asahi-net.or.jp

Abstract Under the auspices of “E-Local Governments System Development Panel” (founded by Ministry of Internal Affairs and Communications), a workflow system and document management system for information liaison between the central government and local governments have been studied from 2004 to 2005. The document management system adopts XML for representing documents and data. This paper focuses on schema design in this project, especially factorization among the schemas for prefectures and those for local governments. The adoption of the “Garden of Eden” style and schema conversion tools allows such schema factorization.

Key words e-local government, liaison between the central government and local governments, XML, schema, style, RELAX NG, factorization

1. はじめに

効果的・効率的な電子自治体のシステム構築, 国・地方を通じた業務の手順の見直しや国・地方で共有すべき文書・データの連携, 電子申請等のシステムと地方公共団体等における既存システムとの円滑な連携等の課題を解決するため, 電子自治体のシステム構築のあり方について検討することを目的として, 総務省は, 平成16年4月27日に「電子自治体のシステム構築のあり方に関する検討会」を発足させた。

この検討会に関連する総務省の調査研究の1つとして、「国・地方を通じた業務の手順をパターン化して自動化するシステム(連携ワークフローシステム)及び国・地方で共有すべき文書・データの連携を可能とするシステム(連携文書管理システム)に関する調査研究」がある。これらのシステムは、霞が関 WAN

及び LGWAN 上で動作することを想定している。この調査研究は、平成16年度から2カ年の事業として実施された。平成16年度には方式の検討を行い、平成17年度には設計・開発と実証実験を行った。

本論文では、この調査研究に携わった成果について解説する。連携文書管理システムをおもに解説し、連携ワークフローシステムについては扱わない(連携ワークフローシステムについては報告書[1]を参照)。連携文書管理システムでは、XML[8]を用いてデータを表現している。

1.1 照会業務

国・地方を通じた業務のうち、総務省において定期的に実施される調査・照会業務(これ以降、照会業務と表記する。)を本事業では扱った。照会業務とは、国又は地方局が地方公共団体などの行政機関に対して何らかの報告を求める業務である。行

政機関の報告は、国が集計・分析・加工し、政策立案に利用する。地方公共団体も他の地方公共団体の報告データを行政の参考とする。一部の報告データは一般にも公開され、公的なデータとして各方面で広く活用されている。

照会業務の数は、総務省だけでも 250 を越え、全省庁を合わせると 700 を越えることが確認されている。1 つの照会業務で使用される調査票は、1 ページのものから 400 ページ近いものまで存在することを確かめている。

照会業務は、地方公共団体と総務省(省庁)の連携を要する代表的な業務であり、国と地方公共団体の双方にとって多大の負担となっている。

1.2 XML 化のねらい

照会業務でやりとりされる調査票の表現として、XML(Extensible Markup Language) を本事業では採用している。XML を採用する理由には、消極的なものと積極的なものに分けられる。

消極的な理由は、XML が国際的な標準であって、特定の企業に支配されないことである。調査票を XML で表現しておけば 50 年後にも読めるものと期待できる。一方、特定の企業に支配された形式で表現した場合は、その企業の方針によって読めなくなることがあり得る。

積極的な理由は、回答負担の軽減および利活用の促進である。回答負担を軽減するため、照会業務を見なおし、重複を排除する。重複を排除するには自動転記機構や文書自動生成が必要であるが、これらは XML のタグを利用して実現できる。利活用を促進するため、重要な調査票を XML で統一的に表現・管理し、検索や再構成を実現する。

XML 化にあたっては、データ標準化推進地方公共団体協議会(第 1 回)[2] 資料 5 「要素属性などの命名規則」に従い、日本語タグを積極的に採用する。また、データ標準化推進地方公共団体協議会(第 1 回) 資料 8 「スキーマ言語プロファイル」の選択肢 1 を選択し、RELAX NG [9] でスキーマを作成し、必要に応じて W3C XML Schema [10] と DTD に自動変換する。

1.3 スキーマの作成手順

スキーマを作成するにはいろいろな方法がある。UML やスプレッドシートで最初に設計してからスキーマを自動生成する方法、スキーマを直接書き下す方法、XML 文書からスキーマを自動生成する方法などである。本事業では、まず調査票から XML 文書を手作業で作成し、この XML 文書からスキーマを自動生成し、修正を加えてスキーマを完成させるという方法をとった^(注1)。

1.4 課題：都道府県用スキーマと市町村用スキーマの共通化

本事業における数多くの課題のうち、本論文ではスキーマ設計に関する課題のみを考慮する。とくに、都道府県用スキーマ

と市町村用スキーマの共通化を扱う。

都道府県用スキーマとは、都道府県が国に提出する調査票を表現するためのスキーマである。同様に、市町村用スキーマとは、市町村が国に提出する調査票を表現するためのスキーマである。都道府県と市町村がほぼ同一様式の調査票を提出する場合は、この二種類のスキーマは、ほぼ同一であるが、完全に同一なわけではない。このような調査票の例として、地方財政状況調査がある。都道府県用の地方財政状況調査様式と市町村用の地方財政状況調査様式とでは、同一なのがほぼ半分であり、相違があるのが残り半分である。例えば、地方財政状況調査のうち一般行政経費について、都道府県の調査票は地域安全対策等に要する経費(警察費)を含むが、市町村の調査票は含まない。

都道府県用スキーマと市町村用スキーマを無関係に作成することには、さまざまな問題がある。まず、同一部分を二回作成することは工数の無駄である。より、重要なのは、必要以上の相違が発生する危険があること、将来スキーマを変更するとき両方を同じように変更しなければならないことである。

これらの問題を解決するため、都道府県用スキーマと市町村用スキーマを共通化する必要がある。以降では、共通化とは、都道府県用スキーマと市町村用スキーマの共通化を指すものとする。

1.5 構成

本論文の残りは次のように構成される。2. では、スキーマのスタイルについて説明する。3. では、共通化の方法を具体的に述べる。4. では、関連する仕事について概観する。

2. スキーマのスタイル

スキーマ書法にはいくつかのスタイルがある。一つのスキーマはいくつかの定義文から構成されるのであるが、どんな単位で定義文を書くか(たとえば定義文をできるだけ細かく分割するか、それともできるだけ分割しないか)が各スタイルによって異なる。

どのスタイルを採用するかは、共通化に大きく影響する。また、スキーマのスタイルによっては、RELAX NG スキーマからの DTD の自動生成が困難になる。

2.1 有力なスキーマスタイル

よく知られているスキーマスタイルとして、Russian Doll, Salami Slice, Venetian Blind, Garden of Eden の四つがある。最初の三つは Roger L. Costello^(注2)が W3C XML Schema のためのスタイルとして提案したもので、Garden of Eden は Eve Maler が追加提案 [7] したものである。

- Russian Doll(定義文をできるだけ分割しない)
 - Salami Slice(要素・属性が現れると定義文を分割する)
 - Venetian Blind(要素・属性の内容が現れると定義文を分割する)
 - Garden of Eden(Salami Slice と Venetian Blind の混合)
- ここでは、簡単のためスキーマ言語 RELAX NG を用いて、

(注1): 回答負担の軽減および利活用の促進には、行政の業務の流れを見直すこと、道路台帳などの情報資源の利用方法を見直すことなどが重要である。調査票の XML 化に関する限り、UML やスプレッドシートを用いたスキーマ設計に工数を費やすより、このような見直しに工数を掛けるべきであると著者は考えている。

(注2): <http://www.xfront.com/BestPracticesHomepage.html>

四つのスタイルで書いたスキーマを示す。しかし、これらのスタイルはもともと W3C XML Schema のために提案されたものであり、W3C XML Schema でも問題なく利用できる。また、Salami Slice と Garden of Eden については DTD にも利用できる。

以下の XML 文書を考える。この文書には、foo, bar, sat, baz1, baz2, baz3 の 6 つのタグ名が現れていることに注意。

```
<foo>
  <bar>
    <baz1>1</baz1>
    <baz2>2</baz2>
    <baz3>3</baz3>
  </bar>
  <sat>
    <baz1>1</baz1>
    <baz2>2</baz2>
    <baz3>3</baz3>
  </sat>
</foo>
```

2.2 Russian Doll

Russian Doll スタイルで書いたスキーマを次に示す。全体が一つの定義文 (start = ...) で書かれており、XML 文書と同じ構造をもつ。

```
default namespace = ""
start =
  element foo {
    element bar {
      element baz1 { xsd:integer },
      element baz2 { xsd:integer },
      element baz3 { xsd:integer }
    },
    element sat {
      element baz1 { xsd:integer },
      element baz2 { xsd:integer },
      element baz3 { xsd:integer }
    }
  }
```

Russian Doll は簡明で読みやすいという特徴がある。

Russian Doll スタイルのスキーマを与えられたとき、それをカスタマイズして用いる方法はない。別ファイルにスキーマをコピーして修正するしかない。

また、同じ形をした行が何度も現れたとき、それらを何度も

記述しなければならない。たとえば、bar と sat の内容はまったく同じだが、繰り返して記述されている。保守のときも、これらの記述をまったく同様に変更しなければならない。

2.3 Salami Slice

Salami Slice スタイルで書いたスキーマを次に示す。定義文は七つあり、そのどの右辺も element キーワードで始まっている。逆に、element キーワードが現れるのは、右辺の先頭に限る。

```
default namespace = ""
start = element foo { bar, sat }
bar = element bar { baz1, baz2, baz3 }
sat = element sat { baz1, baz2, baz3 }
baz1 = element baz1 { xsd:integer }
baz2 = element baz2 { xsd:integer }
baz3 = element baz3 { xsd:integer }
```

Salami Slice スタイルは、Russian Doll ほど簡明ではないが、十分に読みやすい。同じ形をした行を何度も記述しなければならないという欠点は、Russian Doll よりは軽減されているが、Salami Slice でも存在する。例えば、baz1 というタグ名、baz2 というタグ名を繰り返して記述する必要はないが、baz1, baz2, baz3 の三つの並びを二度繰り返して記述する必要がある。

カスタマイズという点で、Salami Slice は Russian Doll より優れている。すなわち、Salami Slice スタイルのスキーマを与えられたとき、それをカスタマイズして用いることができる。これは、与えられたスキーマを参照するスキーマを作成し、一部の定義文を上書きする^(注3) ことによって行う。定義文は要素・属性に対応するので、上書きできるのは要素・属性の定義に限られる。

SGML 時代からの伝統に基づいたスキーマ (Docbook, TEI, XHTML2 など) では、Salami Slice を原則として採用している。

2.4 Venetian Blind

Venetian Blind スタイルで書いたものを次に示す。定義文は三つあり、どの定義文も、要素の内容を規定している。実際、foo.content は foo 要素の内容を、bsm.content は bar, sat 要素の内容を規定している。element キーワードは、定義文の右辺になんども出現する。

```
default namespace = ""
start = element foo {foo.content}
foo.content =
  element bar { bsm.content },
  element sat { bsm.content }
bsm.content =
  element baz1 { xsd:integer },
  element baz2 { xsd:integer },
  element baz3 { xsd:integer }
```

Salami Slice のときと同様に、Venetian Blind スタイルのスキーマを与えられたとき、それをカスタマイズして用いること

表 1 スキーマスタイルの比較

Table 1 Comparison of schema styles

	カスタマイズ	可読性	行に関する重複
Russian Doll	×		あり
Salami Slice			あり
Venetian Blind			なし
Garden of Eden			なし

(注3): DTD, W3C XML Schema, RELAX NG において、この上書き方法は異なる。RELAX NG では、上書きではなく、組み合わせ (combine) という方法もあるが、本論文では考慮しない。

ができる。Venetian Blind では定義文は要素・属性の内容に対応するので、上書きできるのは要素・属性の内容の定義に限られる。

Venetian Blind は、同じ形をした行は一度だけ記述すればよいという利点をもつ。bar と sat の内容が bsm.content に一本化されていることに注意されたい。表の形をしたものを XML 化すると、この一本化が行なえることがきわめて多い。

一方、Venetian Blind は読みやすさという点ではやや劣る。また、bsm.content のように、もともとの XML 文書にない名前を導入することは、スキーマの可読性を低めているということが出来る。

また、Venetian Blind は、RELAX NG から DTD に変換しにくいという問題点がある。これは、同一のタグ名が複数箇所に書かれることがあるのが原因である。

最近のスキーマでは Venetian Blind を採用しているものがある。W3C XML Schema で名前を持つ複合型を多用した場合には、Venetian Blind スタイルのスキーマになる。

2.5 Garden of Eden

Garden of Eden スタイルで書いたものを次に示す。Salami Slice スタイルと Venetian Blind スタイルにおける定義文分割の両方がこのスタイルでは用いられている。

```
default namespace = ""
start = element foo {foo.content}
foo.content = bar, sat
bar = element bar { bsm.content }
sat = element sat { bsm.content }
bsm.content = baz1, baz2, baz3
baz1 = element baz1 { xsd:integer }
baz2 = element baz2 { xsd:integer }
baz3 = element baz3 { xsd:integer }
```

Garden of Eden では、要素・属性の定義を上書きすることも、要素・属性の内容の定義を上書きすることもできるので、カスタマイズにとくに優れている。

同じ形をした行は一度だけ記述すればよいという点では、Garden of Eden は Venetian Blind と同等である。

一方、Garden of Eden の欠点は、Venetian Blind と比べてもやや劣る可読性である。これは、Venetian Blind と同様にもともとの XML 文書にない名前が現れること、Venetian Blind よりさらに定義文が増えることが原因である。

最近のスキーマでは Garden of Eden を採用しているものがある (Universal Business Language など)。

3. 都道府県用スキーマと市町村用スキーマの共通化

本章では、どのスタイルを採用するかを 3.1 でまず説明したのち、共通化前のスキーマと共通化後のスキーマを 3.2 で示す。続いて、共通化の手順と支援ツールを 3.3 で説明する。最後に、共通化できた割合を 3.4 で示す。

3.1 共通化のためのスキーマスタイル

本事業では Garden of Eden スタイルを採用した。このスタ

イルはカスタマイズにおいて優れているため、都道府県用スキーマと市町村用スキーマの共通化を容易に実現することができる。一方、定義文の並べ方を工夫すること、スキーマにしか現れない名前には体系的な命名をすることによって、ある程度は可読性を改善している。

Venetian Blind は、本事業で扱ったスキーマに関する限り、Garden of Eden と同様の共通化を実現することができる。しかし、RELAX NG から DTD への変換を困難にするという理由から、本事業では採用しなかった。

Salami Slice は、共通化という点で、Garden of Eden 及び Venetian Blind よりやや劣っている。これは、ある要素の構造が都道府県用調査票と市町村用調査票とで異なるとき、この要素のタグ名も二回記述しなければならないことによる。また、同じ形をした行をなんども繰り返して記述すると、調査票スキーマが長大になる傾向がある。

3.2 共通化前のスキーマと共通化後のスキーマ

本事業では、都道府県用スキーマと市町村用スキーマの共通化を、RELAX NG スキーマの include 機構を用いて行った。具体的には、同一部分の一つのスキーマにまとめて記述し、都道府県固有部スキーマと市町村固有部スキーマから include によって取り込んだ。

スキーマの共通化を説明するため、簡単な例題を用いて説明する。まず、共通化を行っていない以下の二つのスキーマ test1.rnc と test2.rnc を考える。

スキーマ 1(test1.rnc)

```
start = 学生
学生 = element 学生 { 学生.内容 }
学生.内容 = 小学生, 中学生, 高校生, 大学生
小学生 = element 小学生 { 各学校の生徒.内容 }
中学生 = element 中学生 { 各学校の生徒.内容 }
高校生 = element 高校生 { 各学校の生徒.内容 }
大学生 = element 大学生 { 各学校の生徒.内容 }
男 = element 男 { xsd:long }
女 = element 女 { xsd:long }
各学校の生徒.内容 = 男, 女
```

スキーマ 2(test2.rnc)

```
start = 学生
学生 = element 学生 { 学生.内容 }
学生.内容 = 小学生, 中学生, 高校生, 短大生, 大学生
小学生 = element 小学生 { 各学校の生徒.内容 }
中学生 = element 中学生 { 各学校の生徒.内容 }
高校生 = element 高校生 { 各学校の生徒.内容 }
短大生 = element 短大生 { 各学校の生徒.内容 }
大学生 = element 大学生 { 各学校の生徒.内容 }
男 = element 男 { xsd:long }
女 = element 女 { xsd:long }
各学校の生徒.内容 = 男, 女
```

これらのスキーマには、まったく同じいくつかの定義文 (小学生など) が現れている。同一の定義文を複数のスキーマに置くことは、1. で論じたいくつかの問題がある。

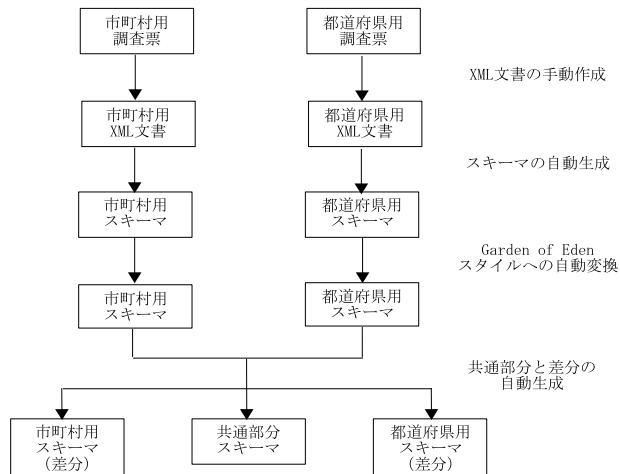


図 1 スキーマ作成ステップ
Fig. 1 Steps for creating schemas

次に共通化を行った結果を示す。まず、両者に共通する定義文をまとめた共通部分スキーマ testCommon.rnc を示す。

共通部分スキーマ (testCommon.rnc)

```
start = 学生
学生 = element 学生 { 学生. 内容 }
小学生 = element 小学生 { 各学校の生徒. 内容 }
中学生 = element 中学生 { 各学校の生徒. 内容 }
高校生 = element 高校生 { 各学校の生徒. 内容 }
大学生 = element 大学生 { 各学校の生徒. 内容 }
男 = element 男 { xsd:long }
女 = element 女 { xsd:long }
各学校の生徒. 内容 = 男, 女
```

testCommon.rnc を include によって参照し、固有の定義文だけをもつように、test1.rnc と test2.rnc を書き改めたスキーマ test1rev.rnc と test2rev.rnc を次に示す。

スキーマ 1 の改訂版 (test1rev.rnc)

```
include "testCommon.rnc"
学生. 内容 = 小学生, 中学生, 高校生, 大学生
```

スキーマ 2 の改訂版 (test2rev.rnc)

```
include "testCommon.rnc"
学生. 内容 = 小学生, 中学生, 高校生, 短大生, 大学生
短大生 = element 短大生 { 各学校の生徒. 内容 }
```

3.3 スキーマ作成手順と支援ツール

3.3.1 調査票から XML 文書の手動作成

調査票をもとに業務見直し（業務フローの見直しや情報資源の見直し）をした上で、XML 文書を手動で作成する。都道府県用の調査票からは、都道府県用 XML 文書を作成し、市町村用の調査票からは市町村用 XML 文書を作成する。

3.3.2 XML からスキーマの自動生成

1. で示したように、都道府県用 XML 文書と市町村用 XML 文書から都道府県用 RELAXNG スキーマと市町村用 RELAX

NG スキーマを自動生成する。自動生成は、trang^(注4) というツールによって行う。

2. で示した XML 文書から trang によって自動生成したスキーマを次に示す。スタイルは Salami Slice に近いが、一回だけしか参照されない要素については定義文を設けることなく直接展開している点が異なる。

```
default namespace = ""
start =
  element foo {
    element bar { baz1, baz2, baz3 },
    element sat { baz1, baz2, baz3 }
  }
  baz1 = element baz1 { xsd:integer }
  baz2 = element baz2 { xsd:integer }
  baz3 = element baz3 { xsd:integer }
```

3.3.3 Garden of Eden スタイルへの自動変換

Garden of Eden への自動変換を行なうためのツールを本事業で開発した。このツールは、Windows のパッチファイルであるが、内部的にいくつかのプログラムを用いている。とくに重要なのは、Salami Slice スタイルの RELAX NG スキーマ (XML 構文) を受け取って Garden of Eden スタイルの RELAX NG スキーマ (XML 構文) に変換するプログラムである。これは XSLT 2.0 のスタイルシートとして実現されている。二つの要素が子孫まで含めて同一であるかの判定を行なっているため XSLT 1.0 では実現できず、XSLT 2.0 を必要とした。

3.3.2 に示したスキーマを自動変換した結果を次に示す（ただしわずかに手を加えている）。

```
default namespace = ""
## bar-sat. 内容
start = foo
foo = element foo { foo. 内容 }
foo. 内容 = bar, sat
bar = element bar { bar-sat. 内容 }
sat = element sat { bar-sat. 内容 }
baz1 = element baz1 { xsd:integer }
baz2 = element baz2 { xsd:integer }
baz3 = element baz3 { xsd:integer }
bar-sat. 内容 = baz1, baz2, baz3
```

なお、このツールは一般の RELAX NG スキーマを対象としたものではなく、本事業で扱う範囲のスキーマに特化したものである。例えば、属性に対処していないし、同一のタグ名が文脈によって異なる内容を持つことにも対処していない。

3.3.4 共通部分と差分の自動作成

Garden of Eden スタイルの都道府県用スキーマと市町村用スキーマから、本事業で開発したツールによって、共通部分スキーマと差分スキーマを作成する。

共通化ツールは、2つのスキーマ i_1 と i_2 を入力として動作し、共通部分スキーマ o_c と差分スキーマ o_1, o_2 を出力する。

(注4): <http://www.thaiopensource.com/relaxng/trang.html>

共通部分スキーマ o_c は、二つの入力スキーマに共通する定義文だけを集めたものである。差分スキーマ o_1 と o_2 は、それぞれ入力スキーマ i_1 と i_2 に対応し、共通しない定義文だけを集めたものである。

3.2 の test1.rnc と test2.rnc を入力として、共通化ツールを実行したとする。 o_c としては、3.2 で示した共通部分スキーマ testCommon.rnc と同じものが出力される。 o_1 と o_2 としては、以下に示す差分 test1diff.rnc と test2diff.rnc がそれぞれ出力される。

差分スキーマ 1(test1diff.rnc)

学生.内容 = 小学生, 中学生, 高校生, 大学生

差分スキーマ 2(test2diff.rnc)

学生.内容 = 小学生, 中学生, 高校生, 短大生, 大学生
短大生 = element 短大生 { 各学校の生徒.内容 }

test1diff.rnc と test2diff.rnc を修正し、testCommon.rnc を参照する include 文を追加すれば共通化は終了する。

共通化ツールは、一般の RELAX NG スキーマを対象として、XSLT 2.0 を用いて開発されている。二つの定義文が完全に同じかどうかを判定するため、XSLT 1.0 ではなく、XSLT 2.0 の機能が必要となっている。

3.3.5 W3C XML Schema と DTD への変換

先に言及した trang を用いて、RELAX NG から DTD 及び W3C XML Schema スキーマへの変換を行う。testCommon.rnc と test1rev.rnc から自動生成された DTD を次に示す。Garden of Eden スタイルが踏襲されていることに注意。

共通部分 DTD(testCommon.dtd)

```
<?xml encoding="UTF-8"?>
<!ELEMENT 学生 (%学生.内容;)>
<!ENTITY % 各学校の生徒.内容 "男,女">
<!ELEMENT 小学生 (%各学校の生徒.内容;)>
<!ELEMENT 中学生 (%各学校の生徒.内容;)>
<!ELEMENT 高校生 (%各学校の生徒.内容;)>
<!ELEMENT 大学生 (%各学校の生徒.内容;)>
<!ELEMENT 男 (#PCDATA)>
<!ELEMENT 女 (#PCDATA)>
```

差分 DTD1(test1diff.dtd)

```
<?xml encoding="UTF-8"?>
<!ENTITY % 学生.内容 "小学生,中学生,高校生,大学生">
<!ENTITY % testCommon SYSTEM "testCommon.dtd">
%testCommon;
```

続いて、testCommon.rnc と test1rev.rnc から自動生成された自動生成された W3C XML Schema スキーマを示す。ここでも Garden of Eden スタイルが踏襲されていることに注意。共通部分スキーマ (testCommon.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema
```

```
xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```
elementFormDefault="qualified">
<xs:element name="学生" type="学生.内容"/>
<xs:element name="小学生"
type="各学校の生徒.内容"/>
<xs:element name="中学生"
type="各学校の生徒.内容"/>
<xs:element name="高校生"
type="各学校の生徒.内容"/>
<xs:element name="大学生"
type="各学校の生徒.内容"/>
<xs:element name="男" type="xs:long"/>
<xs:element name="女" type="xs:long"/>
<xs:complexType name="各学校の生徒.内容">
<xs:sequence>
<xs:element ref="男"/>
<xs:element ref="女"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

差分スキーマ 1(test1rev.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
<xs:include schemaLocation="testCommon.xsd"/>
<xs:complexType name="学生.内容">
<xs:sequence>
<xs:element ref="小学生"/>
<xs:element ref="中学生"/>
<xs:element ref="高校生"/>
<xs:element ref="大学生"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

3.4 結果

表 3.4 に、本事業における共通化の結果を示す。都道府県と市町村が類似した調査票を記入している照会業務は、公共施設状況調査、地域活性化事業債最終充当額一覧、地方公務員給与実態調査、地方財政状況調査の 4 つである。最初の二つについては共通化の効果は低いが、残りの 2 つについては効果が高いことが分かる。

表 2 共通定義文と固有定義文の数の一覧

Table 2 Numbers of common definitions and specific definitions

公共施設状況調査	0	110	186
地域活性化事業債最終充当額一覧	5	34	37
地方公務員給与実態調査	718	205	67
地方財政状況調査	2362	456	549

4. 関連する仕事

XMLのためのスキーマ書法については、数多くの試みがなされつつある。最近では、電子政府のためのスキーマ書法もいくつか現れている。これらは、Cover Pages: Naming and Design Rules^(注5)にまとめられている。

SGMLのDTDの書法として、書籍[6]が広く知られている。この書籍は、スキーマ作成の上流工程から下流工程まで広い範囲を扱う。下流工程であるスキーマのコーディング作業についても、文法的に正しいDTDの書き方にとどまることなく、拡張・保守・カスタマイズに優れた書法を追求している。

現在、W3C XML Schemaの書法に関する試みが数多くなされている(XML Schemas: Best Practices^(注6)など)。それらの試みから、Russian Doll, Salami Slice, Venetian Bind, Garden of Edenの四つのスタイルが生み出された。しかし、この四つのスタイルを生み出した点を除いて、書籍[6]以降に大きな進展はないと著者は考えている。現在は、文法的に正しいW3C XML Schemaスキーマをどう作成するか、W3C XML Schemaの陥穽をどう避けるか、W3C XML Schemaをどう技巧的に使うかというレベルに留まっており、拡張・保守・カスタマイズについての考慮は乏しい。電子政府のためのスキーマ書法についても同様である。

本事業では、RELAX NGのinclude機構のみを用いて、都道府県用スキーマと市町村用スキーマの共通化を行った。W3C XML Schemaを用いた場合にも、今回の共通化に関する限り、同等のことは可能である。実際、本事業のRELAX NGスキーマはすべてW3C XML Schemaに変換されている。変換後のスキーマは、W3C XML Schemaのinclude機構を利用している。

最初からW3C XML Schemaを用いる場合は、複合型のderivation by extensionによる共通化も考えられる。すなわち、共通部分のみを記述した複合型から、都道府県用複合型と市町村用複合型をderivation by extensionによって作成する方法である。しかし、derivation by extensionには、末尾に要素を追加することしかできず、先頭や途中で要素を追加することはできないという制限がある。都道府県スキーマと市町村スキーマの差異は末尾だけとは限らないので、derivation by extensionによる共通化には強い制限がある。

Coates[5]は、相互に関連しあうスキーマ群の開発・保守についての興味深い報告をしている。彼は、スキーマ言語W3C XML Schemaの機能(importやinclude)を用いてスキーマ群を開発・保守することを試みたが、困難であるとして断念している。その代わりに、スキーマを構成する要素・型の宣言のみを専用レポジトリに格納し、そこからスキーマを自動生成している。要素・型の依存関係をレポジトリの機能によって管理し、特定のスキーマに必要な宣言だけを取り出している。

5. まとめと今後の課題

本論文では、都道府県スキーマと市町村スキーマの共通化について示した。Garden of EdenスタイルのRELAX NGスキーマをまず作成したのち、共通部分スキーマと差分スキーマを自動作成することによって、共通化を達成することができた。

本論文で示した共通化は、必ずしも満足のいくものではない。これは、都道府県スキーマの定義文と市町村スキーマの定義文が完全に同じであるかどうか調べるとき、厳しすぎる判定をするからである。例えば、タグ名の僅かな違い(送り仮名の違いなど)やデータ型の違いがあれば、二つの定義文は異なるとみなされてしまう。また、要素の出現順が違えば、まったく異なる定義文とみなされてしまう。

共通化に関する今後の課題の一つとして、業務そのものの見直しによる共通化がある。例えば、地方債に関する多くの調査票は、それぞれ似た業務フローと構造をもつ。これらをまとめて扱うことにより、照会業務全体の効率化に繋がることが期待される。

文 献

- [1] 総務省, 国・地方公共団体を通じた業務改革に対応したシステムの調査研究報告書, http://www.soumu.go.jp/denshijiti/pdf/060525_1.pdf, 2006
- [2] 総務省, 「データ標準化推進地方公共団体協議会」第1回配布資料, 2006
- [3] 「国・地方公共団体連携システム」事業に基づく「調査・照会業務システム」開発に対する提言, 連携文書管理 照会業務編, 2006
- [4] 「国・地方公共団体連携システム」事業に基づく「調査・照会業務システム」開発に対する提言, 連携文書管理 XML技術編, 2006
- [5] Anthony B. Coates, Creating and Maintaining Large Families of Related Schemas, XML Conference 2005
- [6] Eve Maler and Jeanne El Andaloussi, Developing SGML DTDs: from text to model to markup, Prentice-Hall, 1996
- [7] Eve Maler, Schema Design Rules for UBL...and Maybe for You, XML Conference 2002
- [8] T. Bray, J. Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau(Eds). "Extensible Markup Language (XML) 1.0 (Third Edition)". W3C Recommendation, Feb. 2004. <http://www.w3.org/TR/2004/REC-xml-20040204/>.
- [9] J. Clark and M. Murata (Eds). "RELAX NG Specification", Dec. 2001. <http://www.oasis-open.org/committees/relaxng/spec-20011203.html>.
- [10] H. S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn (Eds). "XML Schema Part 1: Structures". W3C Recommendation, May 2001. <http://www.w3.org/TR/xmlschema-1/>.

(注5): <http://xml.coverpages.org/ndr.html>

(注6): <http://www.xfront.com/BestPracticesHomepage.html>