ＲＤＢを用いたＸＭＬ差分管理と部分木取得

応用技術部会　XML-DB　WG
NTTソフトウェア株式会社
山本浩一

はじめに

- 私は応用技術部会の活動の中で、XMLの1ノードを1行としてRDBに格納する手法（私はこの手法を1行1ノード格納法と呼んでいます。）を考案しました。今回の発表では、この1行1ノード格納法を用いて、「XML差分管理」および「XML部分木取得」を行う方法についてご説明とデモを行います。

1行1ノード格納法

- 1行1ノード格納法とは、XMLを複数のノードの集まりとしてとらえ、以下に示す5つの「ノード情報」を1つの行に格納する手法です。

ノード情報

- ノードのパス情報
  - ①親ノードのXPATH
  - ②順番（親ノード内の子供ノードの通番）
- ノードのプロパティ情報（アトリビュート）
  - ③ノード名
  - ④ノード値
  - ⑤ノードタイプ

1行1ノード格納法（具体例）

- まず、あるノード（students）の持つ「ノード情報」がどのようなものかを以下に示します。

```
<?xml version="1.0" ?>
<classes>
    <class name="2年1組">
        <teacher>青木</teacher>
        <students>
            <student>山下</student>
            <student>平井</student>
        </students>
    </class>
</classes>
```

- ノードのパス情報
  ①親ノードのXPATH
  /classes[1]/class[1]
  ②順番（親ノード内の子供ノードの通番）
  1番目（0相対）
- ノードのプロパティ情報（アトリビュート）
  ③ノード名　：students
  ④ノード値　：なし
  ⑤ノードタイプ：要素

1行1ノード格納法（具体例）

- XMLの全ノードから「ノード情報」を作成し、RDBの表に格納した例を以下に示します。

親ノードのXPATH	順番	ノード名	ノード値	ノードタイプ
/	0	classes		要素
/classes[1]	0	class		要素
/classes[1]/class[1]	-1	name	2年1組	属性
/classes[1]/class[1]	0	teacher		要素
/classes[1]/class[1]/teacher[1]	0		青木	テキスト
/classes[1]/class[1]	1	students		要素
/classes[1]/class[1]/students[1]	0	student		要素
/classes[1]/class[1]/students[1]/student[1]	0		山下	テキスト
/classes[1]/class[1]/students[1]	1	student		要素
/classes[1]/class[1]/students[1]/student[2]	0		平井	テキスト

1行1ノード格納法（具体例）

- RDBの表から元のXMLを取得する方法を以下に示します。

SELECT * FROM 表 ORDER BY 親ノードのXPATH , 順番

親ノードのXPATH	順番	ノード名	ノード値	ノードタイプ
/	0	classes		要素
/classes[1]	0	class		要素
/classes[1]/class[1]	-1	name	2年1組	属性
/classes[1]/class[1]	0	teacher		要素
/classes[1]/class[1]	1	students		要素
/classes[1]/class[1]/students[1]	0	student		要素
/classes[1]/class[1]/students[1]	1	student		要素
/classes[1]/class[1]/students[1]/student[1]	0		山下	テキスト
/classes[1]/class[1]/students[1]/student[2]	0		平井	テキスト
/classes[1]/class[1]/teacher[1]	0		青木	テキスト

```
<?xml version="1.0" ?>
<classes>
    <class name="2年1組">
        <teacher>青木</teacher>
        <students>
            <student>山下</student>
            <student>平井</student>
        </students>
    </class>
</classes>
```

■この表からXMLを取り出す方法■

①空のDocumentを作成します。

②XMLを格納した表を『親ノードのXPATH』と『順番』でソートし、取得した順番に対応するノードを作成し、Documentに対してXPATHで特定したノードに追加します。

XML部分木取得

- ここまで1行1ノード格納法によるXMLのRDBへの登録とRDBからXML全体を取得する方法について説明いたしました。そこで次にXMLから必要な部分だけを取り出す方法（XML部分木取得）についてご説明します。

XML部分木取得（目標と現状）

＜目標＞

- XPathによる検索を可能とします。XPathをSQLに変換し、RDBから必要最小限のデータのみを取り出して、 NodeList（XML部分木）を再構築することを目標とします。

＜現状＞

- 現状のデモプログラムでは、条件のないXPathによる検索が可能です。

XML部分木取得（具体例）

- XMLの登録時に、XpathでXMLを探索するためのインデックス用の表を作成します。

インデックス表

XPATHのID	XPATH
0	/classes
1	/classes/class
2	/classes/class@name
3	/classes/class/teacher
4	/classes/class/students
5	/classes/class/students/student

XML部分木取得（具体例）

- XMLを格納した表に、インデックスに対応する「XPathのID」の列と、自分の親ノード内の自分と同名の要素があった場合、その中の何番目なのかをあらわす列（「同名通番」）を作成します。

XML格納表

XPathのID	同名通番	親ノードのXPATH	順番	ノード名	ノード値	ノードタイプ
0	1	/	0	classes		要素
1	1	/classes[1]	0	class		要素
2		/classes[1]/class[1]	-1	name	2年1組	属性
3	1	/classes[1]/class[1]	0	teacher		要素
		/classes[1]/class[1]/teacher[1]	0		青木	テキスト
4	1	/classes[1]/class[1]	1	students		要素
5	1	/classes[1]/class[1]/students[1]	0	student		要素
		/classes[1]/class[1]/students[1]/student[1]	0		山下	テキスト
5	2	/classes[1]/class[1]/students[1]	1	student		要素
		/classes[1]/class[1]/students[1]/student[2]	0		平井	テキスト

XML部分木取得（具体例）

- 指定したXPathを「インデックス表」から探し、対応する「XpathのID」を持つ行を、「XML格納表」から取り出します。

```
SELECT * FROM XML格納表  WHERE XPATHのID IN
 (SELECT XPATHのID from インデックス表
   WHERE XPATH='/classes/class/students/student')
```

XPathに対応するノード（親ノード表）

XPathのID	同名通番	親ノードのXPATH	順番	ノード名	ノード値	ノードタイプ
5	1	/classes[1]/class[1]/students[1]	0	student		要素
5	2	/classes[1]/class[1]/students[1]	1	student		要素

この例では、
指定したXPath：/classes/class/students/student
対応するノード：上記の2ノード
となります。この対応するノードから、その子孫ノードを全て取り出す必要があります。

XML部分木取得（具体例）

- Xpathに対応するノードから、子孫ノードを全て取り出すSQLを作成します。

XPathに対応するノード（親ノード表）

XPathのID	同名通番	親ノードのXPATH	順番	ノード名	ノード値	ノードタイプ
5	1	/classes[1]/class[1]/students[1]	0	student		要素

```
SELECT * FROM XML格納表
WHERE XPATH LIKE '/classes[1]/class[1]/students[1]/student[1]%'
ORDER BY 親ノードのXPATH,順番
```

この LIKE の条件値は、以下のように生成します。

「親ノードのXpath」＋”/”＋「ノード名」＋”[“＋「同名通番」＋”]%”

XML部分木取得（具体例）

- XPathに対応するノードから、子孫ノードを全て取り出します。

XPathに対応するノード（親ノード表）

XPathのID	同名通番	親ノードのXPATH	順番	ノード名	ノード値	ノードタイプ
5	1	/classes[1]/class[1]/students[1]	0	student		要素

XPathに対応するノードの子孫ノード全て（子孫ノード表）

XPathのID	同名通番	親ノードのXPATH	順番	ノード名	ノード値	ノードタイプ
	1	/classes[1]/class[1]/students[1]/student[1]	0		山下	テキスト

相対パス：．（親ノードそのものなので）

■XPathに対応するノードからのXML部分木を生成する方法■
①空のDocumentを作成します。（このDocumentは全ての親ノードで共通して使用します。）
②XPathに対応するノード（これを、TopNodeとします。）を作成します。
③子孫ノード表の全てのノードに対して、「親ノードのＸＰath」をTopNodeからの相対パスに変換します。
④子孫ノード表を『親ノードのXPATH』と『順番』でソートし、取得した順番に対応するノードを作成し、TopNodeに対して相対パスで特定したノードに追加します。

XML部分木取得（具体例）

- 全ての親ノードに対してXML部分木を作成します。すると、NodeListができます。

<student>山下</student>
<student>平井</student>

XML部分木取得（今後の課題）

＜今後の課題＞

- 条件判定を含むXPathによる検索を可能とします。
- XPathを使用した検索の速度（性能）を実測し、その特性を調査します。
- XPathによって特定したノードの更新を可能とします。

XML差分管理

- XMLの差分データを、2つのXMLを「1行1ノード格納法」で別々の表に格納し、2つの表をOuter Join することによって抽出する方法について、説明します。

この方法を用いれば、XMLの差分をノード単位で管理できます。

XMLの差分データ抽出の具体例

現在XML

```
<?xml version="1.0" ?>
<classes>
    <class name="2年1組">
        <teacher>青木</teacher>
        <students>
            <student>山下</student>
            <student>平井</student>
        </students>
    </class>
</classes>
```

『山下』君が転校したらどうなるか？

親ノードのXPATH	順番	ノード名	ノード値	ノードタイプ
/	0	classes		要素
/classes[1]	0	class		要素
/classes[1]/class[1]	-1	name	2年1組	属性
/classes[1]/class[1]	0	teacher		要素
/classes[1]/class[1]/teacher[1]	0		青木	テキスト
/classes[1]/class[1]	1	students		要素
/classes[1]/class[1]/students[1]	0	student		要素
/classes[1]/class[1]/students[1]/student[1]	0		山下	テキスト
/classes[1]/class[1]/students[1]	1	student		要素
/classes[1]/class[1]/students[1]/student[2]	0		平井	テキスト

XMLの差分データ抽出の具体例

『山下』君転校後XML

```
<?xml version="1.0" ?>
<classes>
    <class name="2年1組">
        <teacher>青木</teacher>
        <students>
            <student>平井</student>
        </students>
    </class>
</classes>
```

『山下』君が転校したので、
<student>山下</student>
がXMLから削除されました。

親ノードのXPATH	順番	ノード名	ノード値	ノードタイプ
/	0	classes		1
/classes[1]	0	class		1
/classes[1]/class[1]	-1	name	2年1組	2
/classes[1]/class[1]	0	teacher		1
/classes[1]/class[1]/teacher[1]	0		青木	3
/classes[1]/class[1]	1	students		1
/classes[1]/class[1]/students[1]	0	student		1
/classes[1]/class[1]/students[1]/student[1]	0		平井	3

XMLの差分データ抽出の具体例

現在XML

親ノードのXPATH	順番	ノード名	ノード値	ノードタイプ
/	0	classes		要素
/classes[1]	0	class		要素
/classes[1]/class[1]	-1	name	2年1組	属性
/classes[1]/class[1]	0	teacher		要素
/classes[1]/class[1]/teacher[1]	0		青木	テキスト
/classes[1]/class[1]	1	students		要素
/classes[1]/class[1]/students[1]	0	student		要素
/classes[1]/class[1]/students[1]/student[1]	0		山下	テキスト
/classes[1]/class[1]/students[1]	1	student		要素
/classes[1]/class[1]/students[1]/student[2]	0		平井	テキスト

オレンジ色の行は、上と下の両方にある行を除いた残り (OUTER JOIN を使用)を求めたものです。この残りが差分となります。

『山下』君転校後XML

親ノードのXPATH	順番	ノード名	ノード値	ノードタイプ
/	0	classes		要素
/classes[1]	0	class		要素
/classes[1]/class[1]	-1	name	2年1組	属性
/classes[1]/class[1]	0	teacher		要素
/classes[1]/class[1]/teacher[1]	0		青木	テキスト
/classes[1]/class[1]	1	students		要素
/classes[1]/class[1]/students[1]	0	student		要素
/classes[1]/class[1]/students[1]/student[1]	0		平井	テキスト

XMLの差分データ抽出の具体例

追加XML差分

親ノードのXPATH	順番	ノード名	ノード値	ノードタイプ
/classes[1]/class[1]/students[1]/student[1]	0		平井	テキスト

削除XML差分

親ノードのXPATH	順番	ノード名	ノード値	ノードタイプ
/classes[1]/class[1]/students[1]/student[1]	0		山下	テキスト
/classes[1]/class[1]/students[1]	1	student		要素
/classes[1]/class[1]/students[1]/student[2]	0		平井	テキスト

現在のXMLから削除XML差分の行を削除し、追加XML差分の行を追加すると、『山下』君転校後XMLとなります。

この2つの表を複数世代管理することで、XMLの差分管理を行うことが出来ます。

お礼

XMLの差分データ管理ツールを作成するにあたり、日本IBM社様のご厚意により、『DB2 ユニバーサル・データベース エンタープライズ・エディション バージョン7．2』をお貸しいただきました。

本当にありがとうございました。