

RDBを用いたXML差分管理

XMLをRDBに格納する新手法の提案

XMLの差分データの抽出手法

応用技術部会 XML-DB WG プロトタイプ サブWG
NTTソフトウェア株式会社 山本 浩一

XMLをRDBに格納する今までの手法

- XMLを1つのテキストデータのかたまりとして、1つの列に保存します。(1列格納法)
- XMLから要素や属性を取り出し、表とマッピングします。(表分解法)

があります。

次にそれぞれの例と特徴を見てみましょう。

と は、以降の説明の為に付けた方式の名前です。

1 列格納法 (今までの手法その1)

```
<?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>
  <class name="2年2組">
    <teacher>坂井</teacher>
    <students>
      <student>三木</student>
      <student>佐藤</student>
    </students>
  </class>
</classes>
```



id	XML_DATA
1	<pre><?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> <class name="2年2組"> <teacher>坂井</teacher> <students> <student>三木</student> <student>佐藤</student> </students> </class> </classes></pre>

1 列格納法 (特徴)

- 格納したXMLを完全にもとの形に復元して取り出せます。(メリット)
- XMLを更新する時は、XML全体を単位として更新しなければいけません。(一部分の変更を行うことが出来ません。)(デメリット)

XMLの部分更新が出来ないため、あとで説明するXML差分管理ではこの方式を採用できませんでした。

表分解法(今までの手法その2)

```
<?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>
  <class name="2年2組">
    <teacher>坂井</teacher>
    <students>
      <student>三木</student>
      <student>佐藤</student>
    </students>
  </class>
</classes>
```



class

class_id	name	teacher
1	2年1組	青木
2	2年2組	坂井

student

student_id	class_id	student
1	1	山下
2	1	平井
3	2	三木
4	2	佐藤

要素の1:N関係は外部キーで保持します。

表分解法(特徴)

- 既存のRDBを用いたアプリケーションとの連携が容易です。(メリット)
- XMLの一部分だけを更新することが出来ます。(メリット)
- 格納したXMLを完全にもとの形に復元出来ません。(要素と属性以外のノードはなくなり、また、XML内での要素の出現順序も変わります。)(デメリット)

XMLを元の形に復元出来ないため、あとで説明するXML差分管理ではこの方式を採用できませんでした。

表分解法(特徴)

- XMLに**要素の繰り返し構造**があったとき、対応できません。(デメリット)

```
<?xml version="1.0"?>
<説明>XMLの紹介
  <本文>XMLはW3Cによって制定され、...
    <説明>W3Cの紹介
      <本文>W3C(World Wide Web Consortium)とは、...
        <説明>World Wide Web の紹介
          <本文>World Wide Webとは、...</本文>
        </説明>
      </本文>
    </説明>
  </本文>
</説明>
```

左の例のように『説明』と『本文』が繰り返していて、いくらでも階層が深くなるXMLのことです。

XMLをRDBに格納する新手法

- XMLを複数のノードの集まりとしてとらえ、ノードのパス情報とノードのプロパティ情報を1つの行に保存します。(**1行1ノード格納法**)

次に**1行1ノード格納法**の例と特徴を見てみます。

1行1ノード格納法(新格納法)

```
<?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>
  <class name="2年2組">
    <teacher>坂井</teacher>
    <students>
      <student>三木</student>
      <student>佐藤</student>
    </students>
  </class>
</classes>
```

- ノードのパス情報
親ノードのXPATH
`/classes[1]/class[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)
ノード名 : `students`
ノード値 : なし
ノードタイプ : `要素`

1行1ノード格納法(新格納法)

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
/classes[1]	1	class		要素
/classes[1]/class[2]	-1	name	2年2組	属性
/classes[1]/class[2]	0	teacher		要素
/classes[1]/class[2]/teacher[1]	0		坂井	テキスト
/classes[1]/class[2]	1	students		要素
/classes[1]/class[2]/students[1]	0	student		要素
/classes[1]/class[2]/students[1]/student[1]	0		三木	テキスト
/classes[1]/class[2]/students[1]	1	student		要素
/classes[1]/class[2]/students[1]/student[2]	0		佐藤	テキスト

1行1ノード格納法(新格納法)

SELECT * FROM NEW_XML_TEMP ORDER BY xpath , seq

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]	1	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
/classes[1]/class[2]	-1	name	2年2組	属性
/classes[1]/class[2]	0	teacher		要素
/classes[1]/class[2]	1	students		要素
/classes[1]/class[2]/students[1]	0	student		要素
/classes[1]/class[2]/students[1]	1	student		要素
/classes[1]/class[2]/students[1]/student[1]	0		三木	テキスト
/classes[1]/class[2]/students[1]/student[2]	0		佐藤	テキスト
/classes[1]/class[2]/teacher[1]	0		坂井	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>
  <class name="2年2組">
    <teacher>坂井</teacher>
    <students>
      <student>三木</student>
      <student>佐藤</student>
    </students>
  </class>
</classes>
```

この表からXMLを取り出す方法
空のDocumentを作成します。

XML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。

ソートは必ず親を子より先に作る為
に行っています。

1行1ノード格納法(特徴)

- 格納したXMLインスタンスを完全にもとの形に復元して取り出せます。(メリット)
- XMLの一部だけを更新することができます。(メリット)
- XMLを更新する為の特別な更新言語は不要です。標準SQLだけで更新できます。(メリット)
- XMLに要素の繰り返し構造があっても、対応できます。(メリット)

1行1ノード格納法(特徴)

- 他の格納法に比べ、格納に必要な資源を多く必要とします。(デメリット)
- XMLの階層が深くなると XPATH の文字列が長くなり、格納できなくなります。(デメリット)

プロトタイプを作成ではDB2を使用しました。DB2では、VARCHAR(文字列を格納する列を定義するデータ型)を最大32Kバイトまで確保できるため、まずXPATHの文字列長がオーバーすることはないと思います。

XMLの差分データ

XMLの差分データはどんな時に必要とされるのか？

データ交換にXMLを使用する時、データ通信量を最小限にするために、XMLの差分だけを送受信したい。

経済指標など、時間と共に変わる情報の変更履歴を管理したい。

などが考えられます。

XMLの差分データの抽出手法

- XMLの格納方法は、先程説明した**1行1ノード格納法**を使用します。
- XMLの差分データは、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、抽出します。

次に具体例を示します。

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できます。

今後の課題

- 1行1ノード格納法の新たな適用分野を模索する。(差分管理以外にも使えるはず。)
- XMLの差分を通信する場合の最適な方法を模索する。
- XMLの差分を元に、経済指標などの時間と共に変わる情報を、時系列にグラフとして表示する方法を模索する。

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。