

SuperSQL を用いた XML 生成系の機能拡張

青木 隆龍[†] 遠山 元道[†]

[†] 慶應義塾大学理工学部情報工学科 〒 223-8522 神奈川県横浜市港北区日吉 3-14-1

E-mail: †{ryuryu@db.ics.keio.ac.jp, toyama@ics.keio.ac.jp}

あらまし 関係データベースの既存のデータから XML を生成する従来手法に SQL/XML がある。しかし、グルーピング階層が多段階の XML を生成する場合、SQL/XML では複数のクエリーを入れ子にする必要があるため、質問文が論理的に複雑になるという問題点が存在した。そこで本論文ではこの問題を解決するために、SuperSQL を利用して、質問文の論理的な複雑さが生成する XML のグルーピング階層レベルに影響されない XML 生成システムの実現とその機能拡張について提案する。そして、以前の実現におけるグルーピング要素に属性を付加できないという制約を言語仕様の変更によって排除する。提案システムでは、単一のクエリーのみによって、グルーピング階層が多段階の XML を容易に生成可能であり、従来手法と比較して記述力がより一層向上している。

キーワード SuperSQL, XML, 関係データベース

Extensions of XML Generation System Using SuperSQL

Takatatsu AOKI[†] and Motomichi TOYAMA[†]

[†] Department of Information and Computer Science,

Keio University

Hiyoshi3-14-1, Kouhoku-ku, Yokohama-shi, Kanagawa, 223-8522 Japan

E-mail: †{ryuryu@db.ics.keio.ac.jp, toyama@ics.keio.ac.jp}

Abstract SQL/XML is in the conventional technique which generates XML from an existing the relational database. However, when we generate the XML of a multi-level grouping hierarchy, in SQL/XML, there are problems to construct logically complicated query that it requires multiple nested queries. In this paper, in order to solve this problem, we propose the XML generation system the logical complexity of the query is not influenced by the grouping hierarchy level of the XML to generate and the using SuperSQL. We also remove limitation, not be able to add an attribute to a grouping element in the former implementations, by extending the language. In the proposal system, we can easily generate the XML multi-level a grouping hierarchy by a single query.

Key words SuperSQL, XML, Relational database

1. はじめに

SuperSQL とは、関係データベースへの問い合わせと同時にその検索結果の構造化を行い、出力メディアや多様なレイアウト表現の指定を可能とした出力を行う処理系である。SuperSQL 質問文の Generate 句にある TFE が複雑な木構造を表現可能であることを利用して、以前にも慶應義塾大学遠山研究室では、XML データドキュメントの生成 [?] をはじめ、関係データベースと XML データの統合利用 [?] [?] に関する研究など、SuperSQL における XML 生成系の開発が行われていた。しかし、以前の実現においては、グルーピング要素に属性を付加できないという制約が存在していたため、様々な構造をとりうる XML を柔軟に表現するには機能が不十分であった。また、

Java のバージョンアップにともなって、SuperSQL がバージョンアップされたため、SuperSQL の以前のバージョン上に開発された XML 生成部の実装方式 [?] の設計変更および改良を行う必要があった。

一方、ISO の SQL:2003 標準で定義されている XML 関連機能である SQL/XML [?] (ISO/IEC 9075-14 XML-Related Specifications) では、グルーピング階層が複数レベルにわたる XML を生成する場合、複数のクエリーを入れ子にして記述する必要がある。そのため、質問文が論理的に複雑になってしまい、思い通りの出力を得るには習熟が必要であるという問題点が存在していた。

本稿では上記の問題を解決するために、SuperSQL の現在のバージョンにおいて、新たに言語仕様を変更した XML 生成系

を実現した上で、SQL/XML に対応した機能拡張を報告する。

以下、本稿の構成を示す。2 章では SuperSQL の無次元化について述べる。3 章では提案システムである SuperSQL における XML 生成系について述べ、その機能拡張について 4 章で述べる。5 章では評価について述べ、最後に 6 章でまとめを述べる。

2. SuperSQL の無次元化

この章では本論文で改善を試みる SuperSQL について簡単に述べる。SuperSQL は関係データベースの出力結果を構造化し、多様なレイアウト表現を可能とする SQL の拡張言語であり、慶應義塾大学遠山研究室で開発されている [?] [?]。そのクエリは SQL の SELECT 句を GENERATE< *media* >< *TFE* > の構文を持つ GENERATE 句で置き換えたものである。ここで < *media* > は出力媒体を示し、HTML、PDF などの指定ができる。また < *TFE* > はターゲットリストの拡張である Target Form Expression を表し、結合子、反復子などのレイアウト指定演算子を持つ一種の式である。XML など 0 次元のデータを生成する場合は、グルーピングの指定のみが必要であり、縦や横などの各次元のレイアウトの指定は必要ない。そこで、次元を考慮しなくていいよう従来の SuperSQL の文法を簡素化することにより、SuperSQL を無次元化する。ここでは、従来の SuperSQL と無次元の SuperSQL を比較した結合子および反復子の相違点について述べる。

2.1 結合子

従来の SuperSQL では、結合子はデータベースから得られたデータをどの方向（次元）に結合するかを指定する演算子である。それに対して、無次元の SuperSQL では、結合子はデータベースから得られたデータを無次元に結合する演算子であり、結合演算子（,）を指定する。

2.2 反復子

従来の SuperSQL では、反復子は指定する方向にデータベースの値があるだけ繰り返して表示する演算子である。それに対して、無次元の SuperSQL では、反復子はデータベースの値があるだけ繰り返して生成する演算子であり、反復演算子（[]）を指定する。なお、反復子はそのネストによって属性間のグルーピングを指定できる。

3. SuperSQL による XML の生成

SuperSQL を用いて、データベース中の既存のデータから様々な構造の XML を生成するシステムについて提案する。SuperSQL 質問文の GENERATE 句中の < *media* > に、出力媒体として XML を指定した場合は、SuperSQL の無次元化が自動的に適用される。そのため、結合子は（,）によって指定し、反復子は（[]）によって指定する。

3.1 装飾子

各データベース属性や '{ }' で括った部分式、反復子、文字列に対するオプションを指定する。オプションは '@' の後に続く '{ }' の間に記述し、オプション名と引数の対を '=' で結合して指定する。複数のオプションを指定する場合は ',' を挟

んで並べて記述する。

<属性名>@[<装飾指定>]

ここでは、SuperSQL における XML 生成に特化した各装飾子の機能を紹介する。

3.1.1 tag (XML 要素名付加)

対象（データベース属性、'{ }'、反復子、文字列）の XML 要素名を名付ける。もし、tag オプション、att オプション、もしくは notag オプションのいずれも指定しない場合には、デフォルトとして tag オプションが適用され、データベース属性名が要素名となる。

3.1.2 att / attstart / attend (XML 属性付加)

att, attstart, および attend は、属性付加における一連の装飾子オプションである。att オプションの対象（データベース属性）の属性名を名付ける。もし、引数に何も指定しない場合には、データベース属性名が XML 属性名となる。

なお、属性を付加する要素指定、および属性付加の終了指定は必須である。属性を付加する要素を決定するには、対象（データベース属性、'{ }'）に対して、attstart オプションを指定する必要がある。また、要素に対する属性の付加を終了するには、対象（データベース属性）に対して、attend オプションを指定する必要がある。

要素に対して属性を付加するには、例えば、

```
{e.name@{tag=氏名, attstart},
 e.id@{att=従業員 ID},
 e.sex@{att=性別, attend}
}@{tag=従業員}
```

と記述すれば、

```
<従業員>
  <氏名 従業員 ID="1" 性別="男">青木 隆龍</氏名>
</従業員>
```

といった形式で XML データが出力される。

また、グルーピング要素に対して属性を付加するには、木構造の深さ優先探索に従って、attstart オプション、att オプション、および attend オプションを指定する。例えば、

```
{e.id@{att=従業員 ID},
 e.sex@{att=性別, attend},
 e.name@{tag=氏名}
}@{tag=従業員, attstart}
```

と記述すれば、

```
<従業員 従業員 ID="1" 性別="男">
  <氏名>青木 隆龍</氏名>
</従業員>
```

といった形式で XML データが出力される。

3.1.3 notag (要素型の非表示指定)

対象 (データベース属性, 文字列) の要素型 (タグの対) を表示しない場合に指定する。つまり, タグを付加せず値のみを出力したい場合に使用するオプションであり, 混在内容の出力を得たい場合などに使用する。

タグを付加せず値のみを出力するには, 例えば,

```
{e.name@{notag},
 e.email@{tag=Eメール}
}@{tag=従業員}
```

と記述すれば,

```
<従業員>
 青木 隆龍
  <Eメール>takatatsu@xyz.ne.jp</Eメール>
</従業員>
```

といった形式で XML データが出力される。

3.1.4 EMPTY_ON_NULL / ABSENT_ON_NULL / NULL_ON_NULL (Null 値の扱い指定)

Null 値の扱いは, tag オプション指定, att オプション指定, および notag オプション指定に対する副次的な指定である。対象 (データベース属性) の値が空値であるときの扱いとして, SQL/XML の XML 内容オプションである「EMPTY ON NULL」「ABSENT ON NULL」, および「NULL ON NULL」に対応して, 提案システムでは以下の 3 通りの指定ができる。

- EMPTY_ON_NULL (デフォルト)
 - 空要素または空属性 (タグのみ) を生成する。
- ABSENT_ON_NULL
 - 空要素または空属性を生成しない (タグも出力しない)。
- NULL_ON_NULL
 - 空値に対しては, 要素値または属性値を 'null' として, 要素または属性を生成する。

これらいずれのオプションも指定しない場合には, デフォルトとして「EMPTY_ON_NULL」が適用される。

3.1.5 comment (コメント付加)

comment オプションは, tag オプション指定に対する副次的な指定である。'{}', 反復子を対象とし, comment オプションの引数に文字列を指定することにより, XML コメントを含んだ XML 値を作成する。

3.2 関数

SuperSQL における XML 生成系では, 装飾子を用いることで, 生成する XML の構造のすべてを表現可能である。一方, 典型的な例においてセマンティクスを集約する目的で, 関数を補助的に用いる。TFE 内には関数の記述が可能であり, TFE 中のデータベース属性, もしくは '{} ' で括った部分式や反復子等で構造化されたデータベース属性等を引数とする。ここでは, SuperSQL における XML 生成に特化した各関数の機能を紹介する。

3.2.1 fatelement 関数 (要素名付加・属性付加)

対象 (データベース属性, 文字列) の要素名を名付ける。必要

に応じて, その要素に対する属性の付加を同一関数内において, 同時に行うことができる。要素名は装飾子オプション element の引数に指定する。もし, element オプションの引数に何も指定しない場合には, データベース属性名が要素名となる。

生成する要素に対して属性を付加する場合は, 付加する属性数に応じて, 以下のように指定する。引数に指定したデータベース属性名を属性名として名付け, データベース属性値を属性値とする。

- 付加する属性数が 1 つのとき
 - 対象 (データベース属性) を 'att=' の引数に指定する。
- 付加する属性数が複数のとき
 - 対象 (データベース属性) を 'att1=, att2=, att3=, ..., attN=' の各引数にひとつずつ指定する。

要素に対して属性を 2 つ付加するには, 例えば,

```
{fatelement(e.name@{element=氏名}, att1=e.id,
             att2=e.sex)
}@{tag=従業員}
```

と記述すれば,

```
<従業員>
  <氏名 id="1" sex="男">青木 隆龍</氏名>
</従業員>
```

といった形式で XML データが出力される。

fatelement 関数は, SQL/XML 公開関数の「XMLELEMENT」および「XMLATTRIBUTES」に対応した仕様となっている。属性付加における一連の装飾子オプション (att, attstart, および attend) と比較して, 典型的な例においてセマンティクスを集約できるため感覚的に記述しやすい。しかしながら, fatelement 関数には, 以下に挙げる制約がある。

- グループ要素に対して属性の付加はできない。
- 要素に対する属性名は常にデータベース属性名となるため, 属性名の名前付けを自由に指定できない。
- 対象 (データベース属性) の値が空値であるときの扱いとして, Null の要素値に対しては常に空要素を生成する (EMPTY_ON_NULL)。一方, Null の属性値に対しては常に空属性を生成しない (ABSENT_ON_NULL)。そのため, EMPTY_ON_NULL, ABSENT_ON_NULL, および NULL_ON_NULL のオプションを自由に指定できない。

したがって, 要素に対する属性付加を行う場合, 上記の制約が存在する fatelement 関数ではなく, 柔軟なオプション指定ができる一連の装飾子オプション (att, attstart, および attend) を用いて, 通常は質問文を記述することを想定している。

3.2.2 comment 関数 (コメント付加)

文字列およびデータベース属性を対象として, XML コメントを含んだ XML 値を作成する。XML コメントを任意の位置に指定することができる。comment 関数は, SQL/XML 公開関数の「XMLCOMMENT」に対応した仕様となっている。そのため, tag オプション指定に対する副次的な指定である装飾子 comment と比較して, セマンティクスを集約できるため感

覚的に記述しやすい。

3.2.3 noelement 関数 (要素型の非表示指定)

対象 (データベース属性, 文字列) の要素型 (タグの対) を表示しない場合に指定する。装飾子オプション notag と同様の機能をもつ。

3.2.4 null 関数 (出力値の非表示指定)

データベース属性, 関数, '{ }', 反復子を対象とする。構造化の際に引数を利用するが, その引数を出力したくない場合に指定する。

3.3 GENERATE 句におけるアスタリスク記述

GENERATE 句に '*' を記述した場合, その関係のすべての属性を求める。そして, 各データベース属性ごとにデータベース属性名を要素名として名付けて, 要素を生成する。例えば,

```
GENERATE XML
[
  {e.*}@{tag=従業員}
]@{tag=従業員リスト}
FROM employee e
```

と記述すれば,

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<従業員リスト>
  <従業員>
    <id>1</id>
    <name>青木 隆龍</name>
    <sex>男</sex>
    <year>2010</year>
    <salary>280</salary>
    <email>takatatsu@xyz.ne.jp</email>
  </従業員>
  .....
</従業員リスト>
```

といった形式で XML データが出力される。

3.4 実体参照

XML 文書で特殊な記号 (<, >, &, ', ") を表すには, 実体参照を使う必要がある。そこで既存のデータベース中に特殊な記号が含まれている場合, XML を生成する際に提案システムの処理系内で, 表 1 に示す実体参照による置換が行われる。

表 1 実体参照

実体参照	表す記号
<	<
>	>
&	&
'	'
"	"

4. XML 生成系の機能拡張

SuperSQL の以前のバージョン上を実現した XML 生成系と, 本研究において XML 生成部の実装方式を新たに開発しなおし, SuperSQL の現在のバージョン上を実現させた XML 生成系 (提案システム) を比較する。提案システムにおいて拡張された機能の一覧を表 2 に示す。SuperSQL の現在のバージョンには, 以前のバージョンにはなかった機能である「属性値や集約値に対する局所的ソーティング機能」が実装されている。そのため, 提案システムでは, 属性値や集約値に対して局所的ソーティングを行うことができる。なお, DTD の生成機能については, 提案システムには現時点で実装を行っていない。

表 2 XML 生成系の機能拡張一覧

機能	以前の実現	提案システム
SuperSQL の無次元化の適用	×	
グルーピング要素に対する属性付加	×	
GENERATE 句中のアスタリスク記述	×	
属性値や集約値に対する局所的整列	×	
装飾子の種類	7 種類	10 種類
関数の種類	2 種類	4 種類
Null 値の扱い指定	2 通り	3 通り
DTD の生成		×

5. 評価

提案システムを評価するために, SuperSQL と SQL/XML を用いて, 既存の同一データベースから同一の XML を生成することを考える。このとき必要となる SuperSQL 質問文と SQL/XML 質問文を比較し, 論理的な複雑さに基づく, 質問文の記述のしやすさの視点から両者の記述力の差異を評価する。評価として生成する XML の構造には,

- (1) グルーピング階層が一段階の XML
- (2) グルーピング階層が多段階の XML

の 2 つのパターンを対象とする。なお, 今回は, 生成する XML のグルーピング階層が多段階のときの評価は, グルーピング階層を三段階に設定して行った。

5.1 テストデータ

テストデータとして, 表??, 表??, および表?? に示すリレーションを使用する。これらのリレーションに格納された既存のデータベースから, 例えば, 各支店各売場ごとに従業員の氏名や給料などの情報を一覧するといった XML を生成する。

表 3 リレーション store (支店)

属性名	内容
id	支店コード
city	支店名
address	所在地

表 4 リレーション dept (売場)

属性名	内容
id	売場コード
name	売場名
store	売場のある支店コード
manager	売場のマネージャーの従業員 ID

表 5 リレーション employee (従業員)

属性名	内容
id	従業員 ID
name	従業員の氏名
sex	従業員の性別
year	従業員の入社年度
salary	従業員の給料
email	従業員の E メールアドレス
manager	従業員のマネージャーの従業員 ID

5.2 結果

図 1 に示すようなグルーピング階層が一段階の XML を生成するときに必要な SuperSQL 質問文と SQL/XML 質問文は、それぞれ図 2、図 3 のとおりである。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<従業員リスト>
  <従業員 従業員 ID="1">
    <氏名 性別="男">鈴木 太郎</氏名>
    <salary>300</salary>
    <入社年度>2008</入社年度>
    <email>taro@xyz.ne.jp</email>
  </従業員>
  .....
</従業員リスト>
```

図 1 グルーピング階層が一段階の XML 例

```
GENERATE XML
[
  {e.id@{att=従業員 ID, attend},
    e.name@{tag=氏名, attstart},
    e.sex@{att=性別, attend},
    e.salary,
    e.year@{tag=入社年度},
    e.email
  ]@{tag=従業員, attstart}
]@{tag=従業員リスト}
FROM employee e
```

図 2 SuperSQL 質問文：グルーピング階層が一段階の XML 生成

```
SELECT
XMLELEMENT(NAME "従業員リスト",
  XMLAGG(
    XMLELEMENT(NAME "従業員",
      XMLATTRIBUTES(e.id AS "従業員 ID"),
      XMLELEMENT(NAME "氏名",
        XMLATTRIBUTES(e.sex AS "性別"), e.name),
      XMLFOREST(
        e.salary,
        e.year AS "入社年度",
        e.email
      )
    )
  )
)
FROM employee e
```

図 3 SQL/XML 質問文：グルーピング階層が一段階の XML 生成

また、図 4 に示すようなグルーピング階層が多段階の XML を生成するときに必要な SuperSQL 質問文と SQL/XML 質問文は、それぞれ図 5、図 6 のとおりである。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<支店リスト>
  <支店 支店コード="1">
    <支店名>東京</支店名>
    <address>渋谷</address>
  <売場リスト>
    <売場 売場コード="3">
      <売場名>婦人衣料品</売場名>
      <従業員リスト>
        <従業員 従業員 ID="2">
          <氏名 性別="女">佐藤 花子</氏名>
          <salary>500</salary>
          <入社年度>2005</入社年度>
          <email>hanako@xyz.ne.jp</email>
        </従業員>
        .....
      </従業員リスト>
    </売場>
    .....
  </売場リスト>
</支店>
.....
</支店リスト>
```

図 4 グルーピング階層が多段階の XML 例

6. ま と め

本研究では、SuperSQL の現在のバージョンにおける XML 生成系の実現とその機能拡張を行った。XML 生成部の実装方式を新たに設計・改良をし、要素に対して属性を付加するための言語仕様を変更することによって、以前の実現における問題点であったグルーピング要素に対して属性を付加できないという制約を排除した。そして、論理的に単純な SuperSQL 質問文によって、複雑でかつ柔軟な構造をとりうる XML を生成可能とする処理構造を実現した。

既存の同一データベースから同一の XML を生成する際に、提案システムおよび SQL/XML においてそれぞれ必要となる質問文の比較を行った。その結果、特にグルーピング階層が多段階の XML を生成する場合において、SQL/XML では複数のクエリーを入れ子にして記述する必要があるのに対して、提案システムではひとつのクエリーのみ記述すれば済む。これにより、提案システムは SQL/XML より記述力の面で優れていることが確認できた。

文 献

- [1] 赤堀 正剛, 田中 隆一, 遠山 元道: 『SuperSQL による XML データドキュメントの自動生成』, 信学技報, pp.79-86, 電子情報通信学会, 2000
- [2] 赤堀 正剛, 有澤 達也, 遠山 元道: 『SuperSQL による関係データベースと XML データの統合利用』, 情報処理学会論文誌, データベース, Vol.42, No.SIG08(TOD10), pp.66-95, 2001 年 7 月
- [3] 赤堀 正剛: 『SuperSQL による関係データベースと XML データの統合利用』, 修士論文, 慶應義塾大学大学院理工学研究科開放環境科学専攻, 2001
- [4] 慎 祥探, 有澤 達也, 遠山 元道: 『SuperSQL の XML 生成部の実装方式』, 信学技報, pp.79-84, 電子情報通信学会, 2002
- [5] Jim Melton, Stephen Buxton: 『Querying XML: XQuery, XPath, and SQL/XML in context』, Morgan Kaufmann, 2006
- [6] SuperSQL: <http://SuperSQL.db.ics.keio.ac.jp/>
- [7] M. Toyama, “SuperSQL: An Extended SQL for Database Publishing and Presentation”, *Proceedings of ACM SIGMOD '98 International Conference on Management of Data*, pp.584-586, 1998
- [8] 遠山 元道: 『ターゲットリストの拡張によるデータベース出版と概視の実現』, 信学技報, Vol.93, No.65, pp.243-252, 電子情報通信学会, 1993