XML出版の技術

森嶋　厚行

現在，複数の組織やソフトウェア間でデータ交換を行うために，XMLデータを利用する事が主流になりつつある．しかし，一方で世の中のデータの大部分はXMLデータではない．したがって，XMLデータでないデータをXMLデータに変換する「XML出版」技術が重要になっている．本稿では，特にデータベースを対象としたXML出版について，その技術説明およびいくつかの研究の紹介を行う．

キーワード：XML，データベース，データ交換，効率化

1. XML出版とは？

XMLは元々，情報の電子的な出版のために設計された規格であり，現在ではWWWなどを通じた電子的なデータ交換の手段として重要な役割を果たしている[14]．異なる組織間でデータを交換するためには，その組織間で，交換するデータの構造について合意されている必要がある．例えば，自動車部品業界や旅行業界などでも，業界標準のXMLデータの構造が，DTDなどのスキーマ言語[1]によって定められている[12, 13]．各組織はその構造に従ったXMLデータをお互いやりとりするのである．

一方，各組織は様々なデータを保持しているが，これらは実はほとんどXMLデータとしては保持されていない．例えば，関係データベースや，ファイルシステム中の各種データ（表形式のデータ，バイナリデータ，一般のテキストデータ，など）として管理されている．これらが，XML以外の形式のデータで保持されている理由には様々なものがある．第1に，単にアプリケーションプログラムがXML以外のデータを扱うように設計されているため，という場合がある．第2に性能や格納効率の問題がある．実は，XMLデータを読み込むためには，通常「パース（構文解析）」という処理が必要となるが，ここに大きなコストがかかる事が知られている．したがって，データの処理性能が重要視される場合，データはXMLでない形式で保持されていることが一般的である．また，テキスト形式のXMLデータは冗長性が高く，バイナリ形式よりデータ量が増える傾向がある．第3に，既存のデータ管理手法とのギャップがある．先に述べたように，XMLデータを用いてデータ交換を行う際には，交換するXMLデータの構造が公にされている必要がある(少なくとも，双方が合意している必要がある)．一方，各組織のデータベースに格納されているデータの構造は，データ管理に適した設計手法（正規化[11]など）に基づき設計されている．したがって，一般には各組織で管理されているデータの構造は，合意されたXMLデータの構造と一致しない．

簡単な例を使って説明する．ある自動車メーカが，「各部品メーカがどのような部品を扱っているか」に関する情報を，XMLの形式で関連会社に渡したいとする．このXMLデータの構造とそのデータの例を図1に示す．<name>は部品メーカの名前，<part>は部品メーカによって提供されている部品を表す．一方，自動車メーカでは，実際のデータは関係データベースに格納しており，そのデータベースの構造は図2で示されたものとする．このような時，XMLデータを通じてデータ交換をするためには，関係データベース中のデータをXMLデータに変換する必要がある．この処理を**XML出版**（XML Publishing）と呼ぶ．一般に，XML出版の結果は消費側に渡され，必要な部分がアプリケーションプログラムなどで利用される（図3）．

現実を考えると，XMLデータを格納するデータベースと同じぐらい，もしくはそれ以上に，XML出版は重要な技術である．なぜなら，上の例のように，我々が持つデータは必ずしもXMLデータとは限らないからである．特に，企業における重要なデータはXMLとしてではなく，関係データベースに格納されている事が一般的である．本稿では，XML出版の技

もりしま　あつゆき
筑波大学 大学院図書館情報メディア研究科/知的コミュニティ基盤研究センター
〒305-8550 つくば市春日1-2

```
<!ELEMENT suppliers (supplier*)>
<!ELEMENT supplier (name,  part*)>

<suppliers>
  <supplier>
      <name>A 工業</name>
      <part>タイヤ X</part>
      <part>ハンドル Z</part>
  </supplier>
  <supplier>
      <name>B 技研</name>
      <part> ... </part>
      ...
  </supplier>
  ...
</suppliers>
```

図1　出版するXMLデータの構造とデータ例

```
Supplier(suppkey, name, addr, nationkey)
PartSupp(partkey, suppkey, availqty)
Part(partkey, name, mfgr, brand, size, retail)
Nation(nationkey, name, regionkey)
```

図2　実際に格納しているデータベースの構造

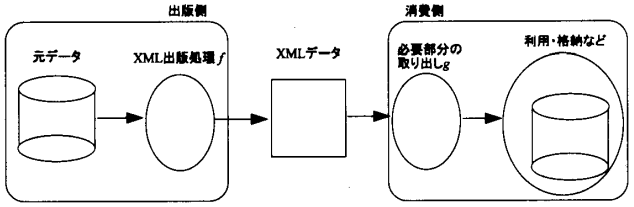


図3　XML出版とXMLデータの消費

術，特に関係データベースに対するXML出版の技術を紹介する．

2．XML出版技術の分類

ここでは，XML出版の技術を二つの視点から分類する．第一の視点は，元となるデータの種類は何か，第二の視点は，XML出版に必要な要素技術のうち，どの範囲をカバーしているか，である．

(A)　出版するデータの種類は何か

まず，企業の重要なデータはデータベースで管理されていることが多いため，データベースに格納されているデータのXML出版が，大きな問題として挙げられる．一口にデータベースといっても，関係データベースをはじめとしてオブジェクト指向データベースなどもあるが，最も需要が大きいと考えられるものは関係データベースのXML出版である．

次に，標準的に利用されているファイル形式のデータをXML化して出版する事が挙げられる．その代表は，表データを表すCSVファイルである．CSVファイルとは2次元の表データを表すもので，表の各列の区切りをカンマで表現し，各行の区切りを改行で表現している．Microsoft Excelをはじめとして，多くのアプリケーションプログラムがCSVファイルの形式でデータを出力する事ができる．他のファイル形式のデータとしては，ASN.1[3]といった特定の形式で構造が表現されたデータファイル，各種バイナリデータファイル，一般のテキストデータファイルなどが挙げられる．

以上のような様々なデータは次のように分類できる．下に行くほど，データのXML出版が困難になる．

(1) 規則的な構造を持ち，かつ，その構造が明示されているデータ：例えば，関係データベースは，データの構造が明示されている．また，バイナリファイルでも固定長フィールドの集まりであるようなものがあり，これらは規則的な構造を持つ．この場合，データ自身からは構造は分からないが，外部の情報を参照することによって取得できる．

(2) 論理的には規則的な構造を持つが，その構造をデータ自身や他のデータから発見するのが困難なもの：例えば，文献リストを表すテキストファイルがある．文献データは一般には特定の構造（著者，タイトルなど）を持つが，その抽出は簡単ではない．

(3) 論理的にも規則的な構造を持たないもの：特定のフォーマットに従わない一般のテキストデータファイルなどである．

(B)　どの要素技術を実現しているか

XML出版には次の要素技術が必要になる．

(a) 構造抽出処理：データの構造が明示的に表現されていない場合，何らかの手段を用いてデータの構造を抽出する．

(b) 構造変換処理：出版時に要求されているXMLデータの構造に合うようデータを変換する．例えば，元データには，部品の情報がメーカに関わらず一覧で保持されているが（図2），出版するXMLデータでは部品メーカ別に分類されている事が要求された場合には（図1），部品メーカ名での分類を行う．

(c) タグ付け処理：<supplier>等のタグを追加し，XML形式にする．

既存のXML出版に関連するシステムやソフトウェアツールは，以上の要素技術のうちのいずれかの機能を実現している．例えば，(a)(c)を実現したツールとして，テキストデータへのタグ付けツールが存在する．

また，(c)を実現したツールとしては，Excelデータの XML化ツールなどがある．また，節3で説明するデータベースに対するXML出版ツールは，(b)(c)を実現している．

2.1 XML出版処理の効率化

大規模なデータをXML出版するためには多大な実行コスト（処理時間）がかかるため，出版処理の効率化はXML出版における重要な問題の一つになっている．実は，XML出版技術を用いた単純なデータ交換には，効率化の余地がある．なぜなら，一般に，出版されたXMLデータは受け取り側で消費される（図3）が，そこでは，受け取ったXMLデータの一部のみだけが必要とされる場合がしばしばあるからである．説明のため，XML出版の処理を関数 $f: D \rightarrow X$ と表す（ここで，D は元データを表す型，X はXMLデータを表す型とする）．消費側で実際に必要なデータを取り出す処理を関数 $g: X \rightarrow X$ とする．このとき，通常は出版側で f を実行した後，結果のXMLデータが受け取り側に渡され，g が実行されて必要な部分が取り出される．もし，関数 $g \circ f: D \rightarrow X$ をXML出版側で実行し，必要な部分だけにデータの量を減らしてから受け取り側にその結果を渡す事ができれば，転送するデータ量を削減できる訳である．したがって，$g \circ f$ を求めることが重要になる．

3. 関係データベースのXML出版技術

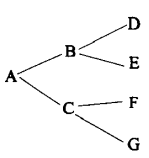
企業などでは多くのデータが関係データベースに格納されている事から，関係データベースに関するXML出版技術は重要な問題であると認識され，これまで活発に研究されてきた．ここではそれらについてより詳しく説明する．

3.1 関係データベースのXML出版における問題

関係データベースのデータは，その構造自体は明らかであるため，問題は，先に挙げた(b)構造変換処理と(c)タグ付け処理，およびXML出版の効率化，になる．これらをどう解決するかが焦点になるわけである．Shanmugasundaramらの文献[9]では，構造変換処理とタグ付け処理に関して，どのような手法が良いかを議論している．これらを次に紹介する．

3.2 構造変換処理

関係データベースでは，構造変換処理は，関係データベースの問合せ言語であるSQLによって記述され，処理される．しかし，SQLで記述された問合せ（以降SQL**問合せ**）が出力する関係表は2次元の表なので，木構造を持つXMLの構造を直接作るわけにはいかない．したがって，論理的に木構造と同等の情報を持つ表を作成する．文献[9]では，いくつかのSQL問合せの構築手法を実験によって比較しているが，Sorted Outer Unionプランと呼ばれる手法を用いて構築されたSQL問合せによる変換処理の効率が良い事が分かっている．Sorted Outer UnionプランによるSQL問合せが出力する関係表の各行（**タプル**と呼ばれる）は，XMLの木構造の根から各葉への経路一つに対応する（図4）．また，タプルが結果の表に現れる順序は，XMLデータを先頭から読んだときに出会う各経路の順序と一致するように出力される．したがって，関係表の各行を上から順に消費していけば，本質的にはXMLを順に読んでいる事と同じになる．先頭から順にタプルが消費されることを強調するために，この関係表は，しばしば**タプルストリーム**と呼ばれる．



Level 1	Level 2	Level 3
A	B	D
A	B	E
A	C	F
A	C	G

図4 Sorted Outer UnionプランによるSQL問合せが出力する関係表

3.3 タグ付け処理

タグ付けの処理については，文献[9]ではデータベースを管理するためのソフトウェア（RDBMSと呼ぶ）内部で処理を行う場合と外部のソフトウェアで別に行う場合が比較されており，内部で処理をする方が圧倒的に高速である事が示されている．その理由は，SQL問合せの実行によって生成されたタプルストリームを，タグ付けを行う外部ソフトウェアに受け渡すためのコストが高価なためである．

4. XML出版システムSilkRoute

SilkRoute[4]は，AT & Tで開発された，関係データベースのためのXML出版システムである．次のような特徴がある．(1) 関係データベースに対するXML出版処理（節2.1の関数 f）を記述するための汎用的な言語を提供する事，(2) XML出版の結果に対して，必要な部分だけを取り出すための処理記述（節2.1の関数 g に相当）が与えられたとき，出版側で $g \circ f$ を計算する仕組みが実現されている事，(3) 構造変換処理において，一つのタプルストリームを出

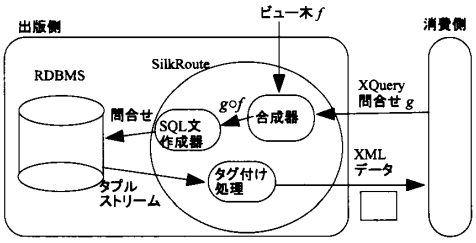


図5　SilkRoute のアーキテクチャ

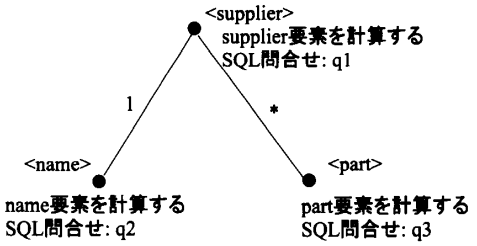


図6　XML 出版のためのビュー木

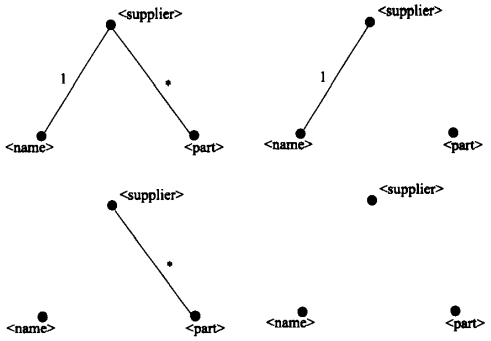


図7　ビュー木の分割

力する単純な Sorted Outer Union プランだけでなく，複数のタプルストリームを同時に用いる処理も選択肢として考慮し，それらの選択肢の中から，最も実行コストの低い処理を自動的に選択・実行する事．

図5は SilkRoute のアーキテクチャである．XML への変換 f はビュー木（Viewtree）という言語で記述され[1]，入力される．利用者はまずビュー木 f を与える事により，データベースをどのように XML として出版するかを SilkRoute に指示する．具体的には，データベース中からどのデータを抽出し，どのようにタグ付けするかを指定する．例えば，図2のデータを基に，図1の構造を持つ XML 出版を行うためのビュー木は図6のようになる．これは，図1で表される XML の構造を木構造で表し，それぞれのノードに対応する XML 要素を計算するための SQL 問合せ（図6では詳細は省略）を記述したものである．

SilkRoute は**仮想出版モード**と**実体出版モード**のどちらかで実行される．仮想出版モードでは，XML 出版処理 f はそのまま実行されず，$g \circ f$ が実行される．したがって，節2.1に書いたように，出版するデータの量を削減することができる．このモードで実行する場合は，SilkRoute の入力として，ビュー木 f に加え，部分データを抽出するための記述を g を与える．g は，XML 問合せ言語である XQuery で記述する．その後，SilkRoute は合成器（図5）によって $g \circ f$ を求め，次にそれを処理するため SQL 問合せを生成する．したがって，消費側で必要とされるデータのみを計算する SQL 問合せが，データベースに投入される．その結果のタプルストリームは先頭のタプルから順にタグ付け処理器に次々と渡され，タグが追加されながら，XML データとして消費側に渡されていく．

一方，実体出版モードで実行する場合に必要な入力は f だけである．仮想モードと違い，大規模な XML データを作成する事が考えられるため，SilkRoute はまず XML 出版処理の**最適化**を行う．一般に，データベース処理における最適化とは，（必ずしも最適でなくても）効率のよい実行方法を発見する作業の事である．実は，ビュー木をうまく分割して各部分木ごとにタプルストリームを生成すると，効率よい処理が可能な事が分かっているため[5]，SilkRoute は，「うまい」分割方法を探す処理を行う．一般には，木の辺が n 個ある場合，木の分割方法はその辺の組み合わせの数と同じ 2^n 通りになる．例えば，図6のビュー木の場合，分割方法は4通りある（図7）．SilkRoute では，分割された部分木それぞれに対し SQL 問合せを一つずつ作成し，これらを並列に実行して，その結果であるタプルストリームをタグ付け処理器で結合する．ある同一のデータベースに対して，10ノードを持つビュー木を用いて実験したところ，次のような結果が得られた[5]．

タプルストリーム数	処理時間
10	1837秒
5	592秒
1	2729秒

この場合は，5タプルストリームの場合が他と比べてかなり高速である事が分かる．SilkRoute は，このような複数の選択肢の中から，高速に動くと考えられるものを自動的に選択し，実行する．

[1] 実際には，人間にとってより書きやすい言語が用意されており，それを用いて記述したものを，内部でビュー木に変換し利用する．

5. XML出版技術に関する各種研究

XML出版技術に関しては，SilkRouteの他にも様々な研究が行われてきた．ここでは，データベースに関するものを中心に，日本での研究も含めてそのいくつかを紹介する．

5.1 XPERANTO

IBMのAlmaden研究所によって開発されてきたXPERANTOは，関係データベースに対するXML出版機能を持つ．SilkRouteとならび代表的なXML出版システムである．XML出版機能はSilkRouteの仮想出版モードとほぼ同じであるが，処理と最適化の仕組みがSilkRouteと異なる．具体的には，SilkRouteではビュー木で表現されていたXML出版の処理が，XPERANTOではXQGM（XQuery Graph Model）と呼ばれるグラフ構造で表現され，このグラフを用いて合成処理などが行われる．詳細は文献[7]等に説明されている．

5.2 Sihem Amer-Yahiaらのシステム

AT & TのSihem Amer-YahiaとYannis Kotidisの論文[2]では，節2.1で説明したXML出版処理の効率化をより一般化したアプローチを提案している．具体的には，XMLの出版側だけではなく，消費側の処理までも考慮して最適化を行う．この研究では，図3の右端に示されるように，XMLデータの消費側では，XML出版と逆の操作が行われる事に着目している．すなわち，XMLデータを他の形式のデータに変換する必要があるという事である．

ここで理想的な場合を考える．つまり，XMLを受け取った消費側に出版側と全く同じ構造のデータベースがあり，受け取ったXMLデータをそのデータベースに格納するような場合である．このようなときには，XMLデータへの変換を完全に省略することができるので，変換処理によるオーバヘッドをなくすことができる．一般的には，必ずしもこのようにはうまくいかないが，このように消費者側での変換も考慮する事により，XMLデータへの変換コストをさらに削減する事ができる．

この仕組みは次のように実現されている．まず，出版側と消費側で，それぞれXMLビューに対するフラグメントの指定を行っておく．ここでいう**フラグメント**とは，出版側と消費側それぞれにおいて都合良く扱える（すなわち，分割しない方が低コストで処理できる）データの単位である．図8はフラグメントの例である．例えば，それぞれの側のデータベースに格納されているデータの単位に一致させておくと，分割しない方が低コストで処理できる．

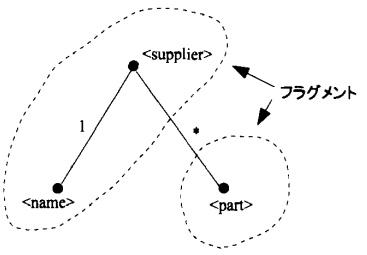


図8 フラグメント指定の例

Sihem Amer-Yahiaらのシステムでは，指定されたフラグメントを単位として，処理の最適化を行う．その際には，データベースによる問合せ処理コスト，データのネットワーク転送コスト，消費側でのデータ書き出しコストなどを総合的に考慮する．

フラグメントの考え方は汎用性が高いため，出版側・消費側のデータが関係データベース以外の場合にも本手法は適用可能である．

5.3 SuperSQL

SuperSQL[10]は，慶應義塾大学の遠山が提案しているSQL拡張であり，関係データベースに対して出版機能を追加するものである．様々な形式でデータを出版可能であるが，その一つとしてXML出版を行う事ができる．SuperSQLでは，特にXML出版に特化した処理の効率化の機能はないが，出版方法指定のためにTFE（Target Form Expression）と呼ばれる簡潔な記法を用意していることが特徴である．

5.4 BiXA

BiXA[6]は，筑波大学の品川，北川が開発している，バイナリデータのXML出版システムである．バイナリデータの特徴を生かしたXML出版処理の効率化が特徴である．このシステムでは，固定長のフィールドの組合せであるようなバイナリデータを前提としている．これにより，データへのアクセス要求gが与えられると，それを処理するために必要なフィールドのアドレスをオフセット計算によって計算することができる．また，バイナリデータでは，フィールドの繰り返し回数やフィールドのデータ型など，構造を決定するための値がデータヘッダなどで与えられることがあるが，オフセット計算時にはこれらも自動的に読まれる．同様なバイナリデータのXML出版技術として，BinX[15]などもある．

6. おわりに

XMLはWWWなどを通じた電子的なデータ交換の手段として重要な役割を果たしているため，各種データをXMLに変換するXML出版は重要な問題の一つである．SQLの最新標準規格であるSQL 2003[8]でも，XML出版のための関数が用意されており，データベースにおいてXML出版を行うための機能が身近なものになりつつある．

データベースに対するXML出版技術に関する研究では，これまで処理の効率化が一つのポイントとなり，様々な研究が行われてきた．特に，不必要なXMLへの変換は行わなくても良いような仕組みが開発されてきている．そもそも，ソフトウェア間のデータのやりとりでは，XMLデータは「論理的に」やりとりされれば良く，実行時に実際に「XMLデータ」が転送される必要はない．このことをうまく利用すると，処理の大幅な効率化が可能になる．データを扱うための共通のプロトコルとしてXMLは多大な貢献を果たしているが，処理のオーバヘッドが大きいことも知られている．したがって，データベースのXML出版における処理の効率化の研究成果を，他のXML関連技術と組み合わせることも，これからの重要な課題の一つと言える．

謝辞 有益なコメントをいただきました科学技術振興機構CREST研究員の品川徳秀氏に感謝いたします．

参考文献

[1] 天笠俊之，吉川正俊，「XMLデータベース技術概説」オペレーションズ・リサーチ．50 (6)：365-372 (2005)

[2] Sihem Amer-Yahia, Yannis Kotidis：Web-Services Architecture for Efficient XML Data Exchange. ICDE：523-534 (2004)

[3] The ASN. 1 consortium. http://www.asn1.org/.

[4] Mary F. Fernandez, Yana Kadiyska, Atsuyuki Morishima, Dan Suciu, Wang Chiew Tan：SilkRoute：A framework for publishing relational data in XML. ACM Trans. Database Syst. 27 (4)：438-493 (2002)

[5] Mary F. Fernandez, Atsuyuki Morishima, Dan Suciu：Efficient Evaluation of XML Middle-ware Queries. SIGMOD Conference 2001

[6] 品川徳秀，北川博之，「バイナリデータに対するXMLビューの実現」電子情報通信学会論文誌．J 88-D-I (3)：604-616 (2005)

[7] Jayavel Shanmugasundaram, Jerry Kiernan, Eugene J. Shekita, Catalina Fan, John Funderburk：Querying XML Views of Relational Data. VLDB：261-270 (2001)

[8] ISO/IEC 9075-14：2003. Database language. SQL Part 14：XML-Related Specifications (SQL/XML).

[9] Jayavel Shanmugasundaram, Eugene J. Shekita, Rimon Barr, Michael J. Carey, Bruce G. Lindsay, Hamid Pirahesh, Berthold Reinwald：Efficiently publishing relational data as XML documents. VLDB J. 10 (2-3)：133-154 (2001)

[10] Motomichi Toyama：SuperSQL：An Extended SQL for Database Publishing and Presentation.SIGMOD Conference：584-586 (1998)

[11] Jeffrey D. Ullman, Jennifer Widom：A First Course in Database Systems. Prentice-Hall 1997

[12] SAE J 2008. http://www.xmlxperts.com/sae.htm.

[13] OpenTravel Alliance. http://www.opentravel.org/.

[14] W 3 C. Extensible Markup Language (XML). http://www.w3.org/XML.

[15] edikt :: BinX, http://www.edikt.org/binx/.