

## 組込み用 XML データベースの試作 An Experimental Implementation of XML Database for Embedded Systems

峯村 治実<sup>†</sup>      水口 武尚<sup>†</sup>      大塚 義浩<sup>†</sup>  
Harumi Minemura   Takehisa Mizuguchi   Yoshihiro Otsuka

### 1. はじめに

現在、最も一般的に用いられているデータベースシステムはリレーショナルデータベース (RDB) であり、さまざまな商用製品やオープンソースのRDBが存在する。RDBは2次元の表でデータを管理するため分かりやすく、安定性や運用ノウハウの面でも優れている。しかし、RDBは厳密なスキーマを前提としたシステムであり、スキーマ変更時にテーブルやインデックスの構成変更が必要になるという課題がある。このため、後から追加するデータに新しい項目が含まれていた場合等に対応するのが難しい。これを解決できるデータベースシステムとして、近年、XMLデータベース (XML-DB) が注目されており、いくつかの製品が出ている。しかし、既存のXML-DBの製品は大規模システム用であり、組込み向けXML-DBは現時点では存在しない。この理由の一つとして、メモリやディスク等のリソース消費量が、RDBに比べて大きい点が挙げられる。この課題の解決を目的として、組込み用の簡易的なXML-DBの試作と評価を行った結果について報告する。

### 2. 基本方針

XMLデータは、主に要素 (element)、属性 (attribute)、値 (value) から構成されており、1つの要素には複数 (0個以上) の要素・属性・値が付随し、1つの属性には1つの値が付随する。したがって、1つのXMLデータは図1に示すような木構造で表現することができる。

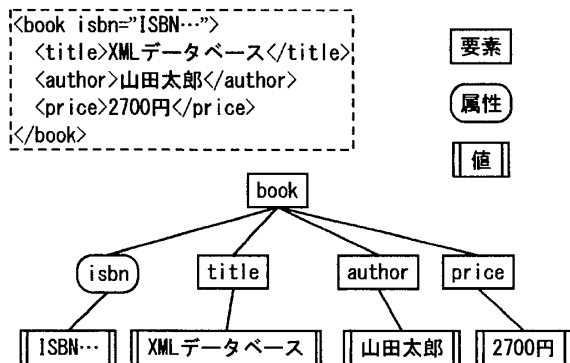


図1 XMLデータの構造

XMLデータをデータベースに格納する方式は、この木構造のデータをそのまま表現できるデータ構造で管理する方式と、XMLデータの構成要素および構成要素間の関係をRDBに写像して管理する方式の2つに大別される。前者はネイティブ方式、後者はRDBマッピング方式と呼ばれる。RDBマッピング方式としては、構造写像アプローチ<sup>[1]</sup>およびモデル

写像アプローチ<sup>[2,3]</sup>が知られている。構造写像アプローチは、個々のXMLデータの論理構造をそのままRDBのスキーマにマッピングする方式、モデル写像アプローチは、XMLデータの構成要素 (木構造のノード、枝等) をRDBのテーブルで管理する方式である。例を図2に示す。一般的に、構造写像アプローチはXML文書ごとに新たなテーブルを構築する必要があるのに対し、モデル写像アプローチは1つのスキーマですべてのXMLデータを管理できるため、汎用性・拡張性の面で優れている。

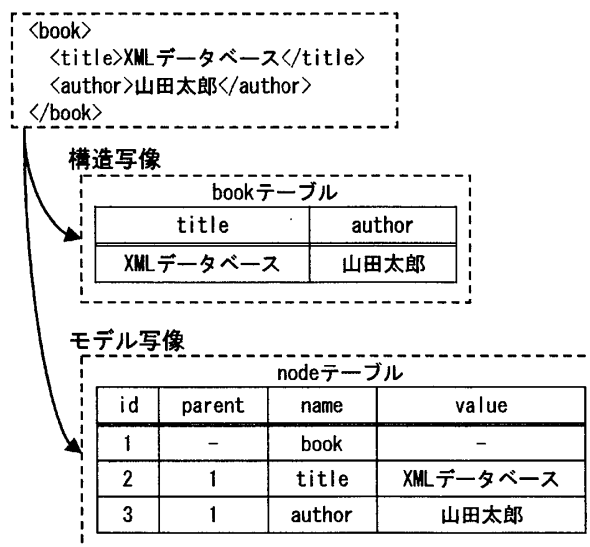


図2 構造写像とモデル写像

ネイティブ方式はXMLデータをそのまま扱うため、子ノードの情報を木構造をたどって取得する等の処理を高速・柔軟に実現可能である。一方、RDBマッピング方式は既存のRDBシステムを利用できるため、信頼性や開発工数の面で優れている。今回は、以下の理由から、RDBマッピング方式 (モデル写像アプローチ) を採用することとした。

- ① 多くの組込み機器では、大部分のデータはスキーマ不変の固定データで、RDBによる管理に適しており、一部の可変データのみをXML-DBで管理する形となる。このため、RDBとXML-DBが共存できる必要があり、XML-DBをRDBマッピングで実装した方が共通部分が多くなるため、使用リソース量の面で有利である。
- ② RDBで管理するデータとXML-DBで管理するデータをアトミックに処理する機能を実現しやすい。例えば、書籍情報のうち固定部分をRDBで、可変部分をXML-DBで管理する場合、一つの書籍の情報を入力する処理でRDBの処理完了後、XML-DBでの処理を開始した時点で電源が落ちた場合、再起動後に一つの書籍情報のうち一部分のみが入力された状態になるのを避けなければならない。これを

実現するには、RDBでの処理とXML-DBでの処理を一つのトランザクションとして実行し、電源断等によって完了しなかったトランザクションを再起動後にロールバックできる必要がある。XML-DBをRDBマッピングで実装すると、同一のトランザクション管理機能を利用できるため、この処理の実現が容易になる。

### 3. テーブル構成

RDBとしては、オープンソースのRDBとして実績があるSQLite<sup>[4]</sup>を使用し、pathテーブルとnodeテーブルによってXMLデータを管理する。データの格納は、入力元のXMLファイルをXMLパーサによって解析し、pathテーブルおよびnodeテーブルにマッピングすることによって行う。

〈pathテーブル〉

カラム名	型	内容
id	整数	path名のID
path	文字列	path名
name	文字列	path名最後の要素名または属性名

〈nodeテーブル〉

カラム名	型	内容
NodeID	整数	ノードのID
PathID	整数	path名のID
ParentID	整数	親ノードのID
Type	整数	要素:0、属性:1
Value	文字列	ノードの値

pathテーブルはXMLデータの各ノード(要素、属性)についてrootノードからのpath名を管理するテーブルである。path名はrootノードからの要素名および属性名を「/」(要素名の場合)または「@」(属性名の場合)で連結した文字列で表現する。pathテーブルにより、pathを指定したXMLデータの検索を高速に行える。nodeテーブルは各ノードのデータ本体を格納したテーブルである。また、nodeテーブルに、path名そのものではなくpath名のidを格納することによって、データ量の削減を図っている。なお、SQLiteでは文字列はすべて可変長で管理される。

### 4. 検索機能

XML-DBの問い合わせ言語の標準仕様としては、XPathとXQueryが知られている。組み込み機器では処理性能と用途の面からあまり複雑な検索は要求されないと思われることから、一般のアプリケーションがXQueryを使いこなすのは難しいことから、今回の試作では、XPath 1.0のサブセットによって簡易的に検索条件を指定することとした。

XPathによる検索式はパーサによって解析され、SQL文が生成される。例えば「//book[author="山田太郎"]」という検索式の場合、「SELECT id FROM path WHERE path LIKE '%book/author%'」で得られたidの値Xについて、「SELECT ParentID FROM node WHERE PathID=X AND Value='山田太郎'」を実行することになる。

### 5. 試作と性能評価

実際の組み込み機器上でどの程度の性能が得られるかを検証するため、以下の組み込み機器評価プラットフォームで試作と性能評価を行った。

項目	仕様
CPU	組み込み用プロセッサ(400MHz)
メモリ	SQLite使用メモリ最大約6MB
ハードディスク	4200rpm、シーク16msec
測定対象データ	元XMLファイル…………… 約100Mバイト pathテーブル…………… 120件 nodeテーブル…………… 260万件

測定内容と測定結果を以下の表に示す。

No.	検索式	件数	処理時間[秒]
1	//e0	1	0.2
2	//e1[.="s1"]	18	0.2
3	//e2[e1="s1"]	18	0.8
4	//e3[e2/e1="s1"]	18	923.6
5	//e4[./e1="s1"]	18	5,479.6
6	//e2[e1="s1" and e5="s2"]	10	17.9

単純な検索(No. 1、No. 2)では実用的な処理時間(0.2秒)が得られているが、孫ノードの条件による検索(No. 4)や、兄弟ノードの条件による検索(No. 5)では、数百秒以上の時間がかかっている。また、AND条件を用いた検索(No. 6)でも、十数秒の時間がかかっている。これは、XPath検索式をSQLに変換するときに、複雑な検索式ではコストの高いJoin演算が多用されるためと考えられる。

### 6. おわりに

組み込み用の簡易的なXML-DBをRDBマッピング方式により試作し、単純な検索式では実用的な処理速度が得られることを確認できた。今後の課題は複雑な検索式の場合の性能改善であり、Join演算をできるだけ使用せずに複数のselect演算に分解し、段階的に検索を行う等の工夫が必要と考えられる。

### 〈参考文献〉

- [1] J. Shanmugasundaram, K. Tufte, C. Zhang, G. He, D. J. DeWitt, and J. F. Naughton, "Relational databases for querying XML documents: Limitations and opportunities," In VLDB, pages 302-314, September 1999.
- [2] D. Florescu and D. Kossmann, "Storing and querying XML data using an RDBMS," IEEE Data Engineering Bulletin, vol. 22, no. 3, pages 27-34, 1999.
- [3] M. Yoshikawa, T. Amagasa, T. Shimura, and S. Uemura, "XRel: A pathbased approach to storage and retrieval of XML documents using relational databases," ACM Transactions on Internet Technology, vol. 1, no. 1, pages 110-141, 2001.
- [4] <http://www.sqlite.org/index.html>