

# ハイブリッド・データベースにおけるXMLデータモデリング

山川 多美

## XML Data Modeling for Hybrid Databases

Tami Yamakawa

増大するXMLデータをいかにビジネスに活用できるかが企業の課題となってきた昨今、XMLデータベース機能とリレーショナルデータベース機能を併せ持つハイブリッド・データベースが主流となり、急速に進化している。IBMも、ネイティブXMLデータベース機能を備えたハイブリッド・データベースであるDB2® 9を2006年夏に発表した。このような背景にあつて、本論文では、リレーショナルデータとXMLデータの両者を企業の資産として活用するためのデータモデリングについて述べる。XMLスキーマを中心に、XMLデータモデリングとしてのXMLスキーマ設計についての紹介を行い、リレーショナルデータモデリングとXMLスキーマ設計を併せた、ハイブリッド・データベースにおけるXMLデータモデリングを提案する。

Nowadays, with the question of how to utilize increasing XML data for business becoming an issue, hybrid databases that have both relational database functions and XML database functions have become mainstream and are rapidly evolving. In the summer of 2006, IBM also released DB2® 9 as a hybrid database. This paper discusses data modeling which utilizes both relational data and XML data as company assets. With a focus on the XML schema, the paper introduces the XML schema design as XML data modeling, and proposes XML data modeling in hybrid databases which unites XML schema design with relational data modeling is proposed.

Key Words & Phrases: XML, XMLスキーマ, データモデリング, DB2, ハイブリッド・データベース  
XML, XML schema, data modeling, DB2, hybrid database

### 1. はじめに

XML(eXtensible Markup Language)は、従来からのドキュメント管理やメッセージ交換だけでなく、昨今の情報処理のあらゆる場面で利用されている。SOA(Service Oriented Architecture)をはじめ、Web2.0によるビジネス・インテグレーションや、セマンティックWebでのメタデータ[1]など、新しいテクノロジーではXMLが不可欠になっている。Web上に存在するXMLデータ(コンテンツ)が膨大になるにつれ、このXMLデータをいかにビジネスに活用できるかが企業の課題となりつつある。

このようなXML普及の流れを受けて、XMLデータベース(以下、XML DB)機能とリレーショナルデータベース(以下、RDB)機能を併せ持つハイブリッド・データベース(以下、ハイブリッドDB)が主流となり、RDBベンダー各社は、続々とXML機能を強化してい

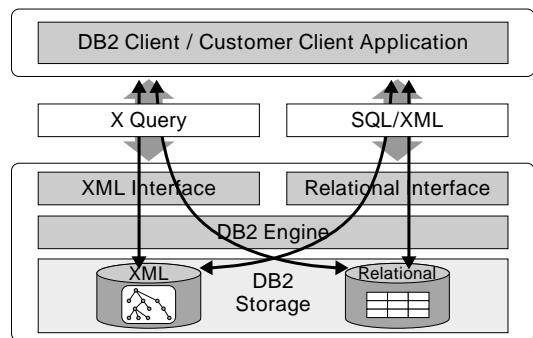


図1. ハイブリッド・データベース [DB2 9.2]

る。IBMも、ネイティブXML DB機能を備えたハイブリッドDBであるDB2 9を2006年夏に発表した(図1, [2])。データをDBに格納する際に、リレーショナルデータとXMLデータという二つの形態が選択でき、以下のような利点が享受できる。

- XMLデータもリレーショナルデータ同様に管理可能
- XQueryなどの標準アクセスが可能
- 両者を利用するアプリケーションが容易に開発でき

提出日: 2006年8月7日 再提出日: 2006年12月20日

る, など

このようなハイブリッドDBの急速な進化により, XMLデータを企業の資産として格納・分析・活用する基盤がそろってきた, 次はDB内にXMLデータをいかに整備していくかを考える必要がある.

XMLとして格納されるデータとしては, 各業界やアプリケーションが定めた標準XMLスキーマに基づくXMLデータやXMLメッセージ, XMLスキーマを持たない既存XMLデータをそのままDBに格納することがまず考えられる. XMLスキーマとはXML文書の構造や値の範囲などを, XMLスキーマ言語によりXML文書で記述したものである.

次の段階として, 情報の関係や構造をモデル化するデータモデリングにおいて, XMLデータもリレーショナルデータと同様にモデリング対象とし, 従来リレーショナルデータとして設計してきたデータをXMLデータとして設計・格納することも視野に入れていく必要がある. 企業のデータ資産をXML化することで, 次のような利点を楽しむことができる.

- ・システムの柔軟性
- ・XSLT( XML Stylesheet Language Transformations )を利用した容易なデータ加工
- ・システム連携の容易さ, など

本論文では, リレーショナルデータとXMLデータの両者を活用するシステムのためのデータモデリングについて述べる. 以下, 2章でXMLデータモデルのメタ定義を記述するXMLスキーマを紹介し, 3章でXMLスキーマ設計手法を中心にXMLデータモデリングを紹介しながら, リレーショナルデータモデリングとの共通点を明らかにする, 4章でXMLスキーマ設計にリレーショナルデータモデリング・フローを適用し, 二つのデータモデリングの成果物やフローの差異および考慮事項を明らかにした上で, ハイブリッドDBにおけるXMLデータモデリングについて提案する. 最後に5章で, 結論と今後の課題を述べる.

## 2. XMLデータモデルとしてのXMLスキーマ

### 2.1 XMLデータモデル

XMLデータもリレーショナルデータと同様に, データモデルを持つ.

例えば, 論文を例に考えてみる. 論文には執筆要領が定義されていて, タイトルや記述順序が決まっている. 論文をXMLで表現すると図2のようになる. XMLデータモデルを表現する一般的なDOM( Document Object Model )ツリーにすると図3のようになる. つまり, 論文XMLデータは, 執筆要領というデータモデルに基づくXMLデータといえる.

```

<Paper>
<Title>
<Ja_Title>ハイブリッド・データベースにおけるXMLデータモデリング
</Ja_Title>
<En_Title> XML data modeling... </En_Title></Title>
<Author><Auth1>山川多美</Auth1></Author>
.....
<Contents>
<Shou No=1>
  <Shou_Title>
    はじめに
    .....
</Shou_Title>
.....
</Paper>

```

図2. 論文のXML記述

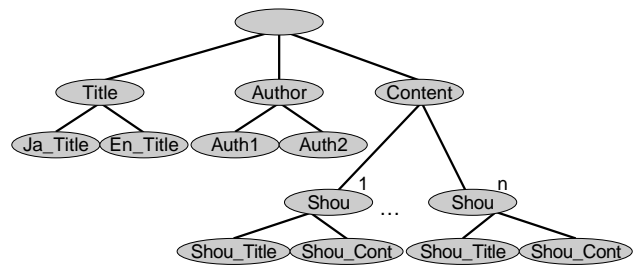


図3. 論文のXMLデータモデル

### 2.2 XMLスキーマの役割

実際の論文では, 執筆者が, 執筆要領に従って論文を記述する. 読者や審査員も執筆要領に従っていることを前提にして読み進めていく. 書き手と読み手が, 論文執筆要領というXMLデータモデルを共有しているからコミュニケーションが成り立っているのである. 論文執筆要領を人でもコンピュータでも理解できるように記述したものがXMLスキーマであり, コンピュータによる論文XMLデータ交換や加工を可能とする. つまり, XMLスキーマは, XMLデータモデルを表すメタデータ記述といえる.

XMLスキーマの主な役割は, 以下の二つである( 図4 [ 3 ] ).

- ① データの正当性( Validate )  
パーサーによりXMLデータがXMLスキーマに基づいているか妥当性を検証する
- ② 汎用性・拡張性  
変化する要件に対してはXMLスキーマを修正して対応していき, DB構造やプログラムへの影響を抑える.

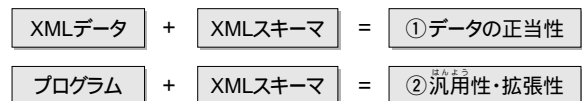


図4. XMLのスキーマの用途  
出典: XMLのスキーマ設計ハンドブック([ 3 p.21 ])

### 2.3 XMLスキーマ定義言語

ここでは, 代表的なXMLスキーマ定義言語について簡単に紹介する.

XMLスキーマとは, XMLデータモデルを記述する

体系の総称であり、実際の記述仕様として何種類かのスキーマ定義言語がある。以下に代表的なスキーマ定義言語を挙げる。

#### ① DTD( Document Type Definition )

XML1.0で策定された従来からのスキーマ定義言語。ドキュメント中心で、データ記述能力に欠けることがある反面、分かりやすく印刷出力には十分対応できる[ 3 ]。

#### ② XMLSchema

XMLSchemaは、XMLスキーマを記述するスキーマ定義言語の一つ( 固有名詞 )。

情報処理やスキーマ標準化に向いている。適応範囲が広い反面、複雑。ネームスペースと、XMLSchema自身がXMLデータであるという特徴がある[ 3 ]。

ドキュメント構造、要素や属性の内容について言及し、W3Cで勧告されている[ 4 ]。

#### ③ RELAX NG( REgular Language definition for XML New Generation )

RELAX Core( 2000年にJISで規格化 )とTREC( 2001年にJames Clark氏が公開 )を統合し、OASIS( 構造化情報標準促進協会 )により標準化。順序なしのコレクションのパターンもサポートしている、シンプルで強力なスキーマ定義言語[ 5 ]。

### 2.4 DB2におけるXMLスキーマ記述

前節で代表的なXMLスキーマ定義言語を挙げたので、ここではIBM DB2関連製品で扱われているXMLスキーマ定義言語を整理する。

#### ・ DB2 9

XMLSchemaを中心とした実装。妥当性検証やDecomposition( XMLデータのリレーショナルデータへの分解 )機能に使用している。DTDについてはシNTAXS・チェック・レベル( デフォルト値の有無など )での妥当性検証をサポート。

#### ・ RDA( Rational® Data Architect )

DB2ファミリー向けデータモデリングツール。XMLSchemaとデータモデルのマッピング、リバースエンジニアリング、XMLSchema出力などの機能が提供されている。

上記のようにIBM製品はXMLSchemaをXMLスキーマ定義言語の軸としている。その主な理由は、W3Cで勧告されているというオープン性と、きめ細かいデータ記述が可能であることである。

### 3 . XMLデータモデリング

前述の通り、XMLはシステム間や企業間で交換・共有するためのメッセージフォーマットとして広く浸透

してきている。その一方で、XMLはタグ名を自由に決められるなど柔軟性に富んでいる。柔軟性を維持しつつ、汎用的・標準的であるためには、XMLデータモデリングとしてのXMLスキーマ設計が非常に重要になってくる。

例えば、個人や特定部門でのみ通じるタグ名を使用していたのでは、プロセス間で利用できるXMLメッセージにはなりえない。

#### 3.1 XMLスキーマ設計手法

XMLデータとリレーショナルデータを同じ土俵でとらえて、一つのシステムまたは企業内に両者を適材適所に取り入れていくためには、XMLスキーマ設計( XMLデータモデリング )とリレーショナルデータモデリング、一つのデータモデリング・フローにのせることが必須である。

データモデリング・フローの一本化を考えるにあたり、まずは代表的なXMLスキーマ設計手法を以下に挙げる。

##### ( 1 ) オブジェクトモデリングの利用

オブジェクト指向のモデリング記法であるUML( Unified Modeling Language )はXMLと非常に親和性が高い。UMLオブジェクトクラス図のクラスをXML要素に変換する、シーケンス図からXMLメッセージフローを洗い出す、などが可能[ 6 ]。

##### ( 2 ) 正規化

RDB設計の代表的な手法。重複や依存関係を明らかにして論理データモデルを完成させていく[ 7 ]。

##### ( 3 ) ER( Entity Relationship )モデル

イベント系エンティティとリソース系エンティティからERモデルを完成させていく[ 7 ]。それぞれのエンティティをXMLスキーマで設計し、リソース系エンティティからマスター情報として共通にするネームスペース定義を設計することができる。

##### ( 4 ) ANSI/X3/SPARCの3層スキーマ・アーキテクチャ

1970年代にANSI( American National Standards Institute )により標準化された仕様。入出力と内部処理のスパゲティ関係をスター型にするために、中間に概念スキーマを採用している。入出力/内部処理/概念の3層をXMLで表現することで汎用的なシステムが構築できる[ 3 ]。

以上のように、XMLスキーマ設計には、既存の様々なモデリング手法が適用でき、なんら特別なスキルや手法は不要である。

#### 3.2 正規化によるXMLスキーマ設計

XMLスキーマ設計の一例として、前節で述べたXMLスキーマ設計手法の中からRDB設計時に必須

となる正規化を紹介する。

正規化の説明によく使われる注文伝票を用い、丸山 [ 3 ] のXMLスキーマの正規化を参考にし、XMLデータとリレーショナルデータそれぞれの正規化の例を示す。

( 1 ) 注文伝票からのデータ項目を拾い上げる

紙面の都合上、主要な項目のみを表1に表す。

表1. 注文伝票( 正規化前 )

伝票番号	注文年月日	商品コード	商品名	数量	...
1	2006.7.6	001	AA	100	
		002	BB	300	
2	2006.7.7	002	BB	20	
3	2006.7.7	003	CC	100	

( 2 ) 第1正規化

第1正規化とは、繰り返し項目をなくして単純化することで [ 8 ]、例えば、表1の商品コードや商品名などの繰り返し項目にあわせてフラットにする、もしくは、繰り返し項目の分離を行う。リレーショナルデータへの適用を表2に、XMLスキーマへの適用を図5に示す。

表2. 注文伝票( 第1正規形 )

伝票番号	注文年月日	商品コード	商品名	数量	...
1	2006.7.6	001	AA	100	
1	2006.7.6	002	BB	300	
2	2006.7.7	002	BB	20	
3	2006.7.7	003	CC	100	

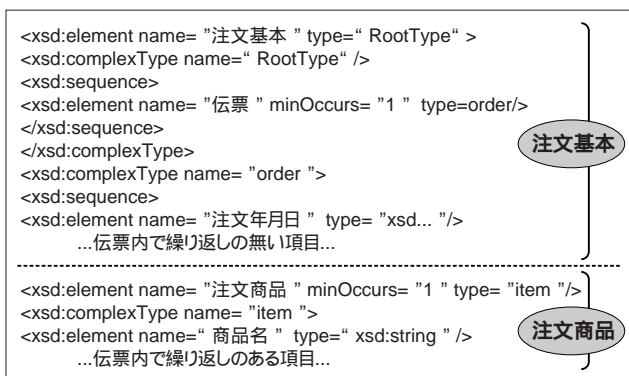


図5. 注文伝票XMLスキーマ( 第1正規形 )  
出典: XMLのスキーマ設計ハンドブック [ 3 p.70 ]

( 3 ) 第2正規化

第2正規化とは、第1正規形であることに加えて、すべての非キー属性がいかなる候補キーにも部分関数従属していないことである [ 8 ]。例えば、表2において、商品コードに関数従属している商品名を分離することにより、部分関数従属性が排除できる。リレーショナルデータへの適用例を図6にしめす。XMLスキーマの例では、共通情報 ( RDBにおけるマスターテーブル ) はネームスペースで共有化することができる。XMLスキーマそのものを記述すると長いので、XMLスキーマ構造だけを図7にしめす。

同様の方法が第3正規化にも適用でき、正規化によりリレーショナルデータとXMLスキーマ両方が設計できることがわかる。

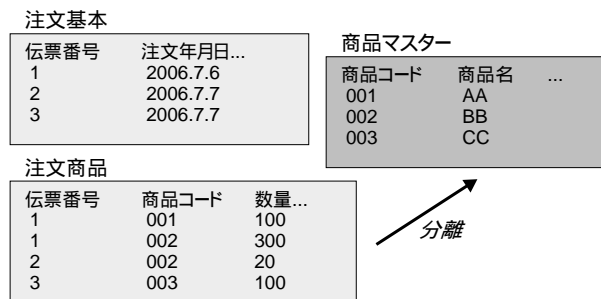


図6. 注文伝票表( 第2正規形 )

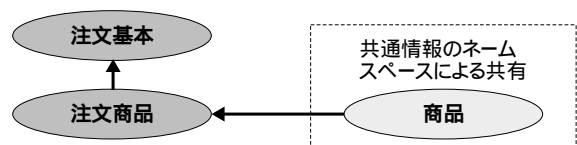


図7. 注文伝票XMLスキーマ構造( 第2正規形 )

#### 4. ハイブリッドDB設計のためのXMLデータモデリングのフロー

前章で、リレーショナルデータモデリングの手法も含めた従来からの手法が、XMLスキーマ設計に適用できることを述べた。ここでは、リレーショナルデータモデリングに使用される代表的なデータモデリング・フローをXMLスキーマ設計に適用し、二つのデータモデリングの各ステップにおける作業内容や成果物の差異を明らかにする。さらに、二つのデータモデリングを一つのフローにした、ハイブリッドDBにおけるデータモデリングを提案する。

図8にRDB設計で利用される代表的なデータモデリングのフローを挙げる [ 7 ]

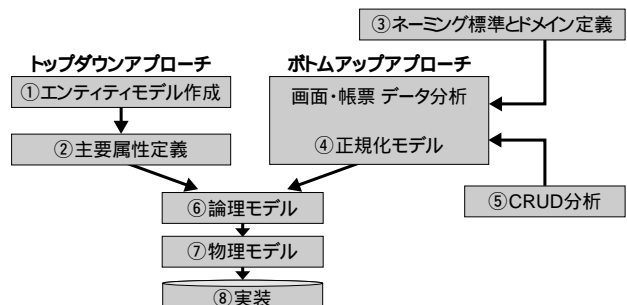


図8. RDB設計における代表的なデータモデリング・フロー  
出典: 実践的データモデリング入門 [ 7 p.vi ]

最後の実装をXMLデータとする場合に、図8の各モデリング手法がどのようにXMLスキーマ設計に置き換えられるのか、またはそのまま適用できるのかを

表3. リレーショナルデータモデリングによるXMLスキーマ設計

リレーショナルデータモデリング 作業内容 / 成果物	XMLスキーマ設計作業内容	XMLスキーマ設計成果物
① エンティティ切り出し 関連付け / ERモデル	リソースとイベントに分け、XMLデータエンティティ、つまり、XMLスキーマ設計対象候補の抽出。	ERモデル
② 主キー候補 主要属性付加 / 主要属性モデル	XMLスキーマ階層構造の組み立てを行う。エンティティからネームスペース候補を洗い出す。	リソースとイベントで区別された主要XMLスキーマ ネームスペース候補エンティティ
③ ネーミング標準定義 ドメイン定義	ネーミング標準に従ってタグを命名、ドメイン定義をXMLスキーマで記述。	各項目(要素)のXMLスキーマ。例えば、XML Schemaでは、コード値の型をtype、コード値の値リストをenumeration、で記述する。
④ 画面・帳票からのデータ分析 エンティティへの割り当て 正規化 / 正規化モデル	正規化手法を用いて、XMLスキーマを設計する。加えて、要素と属性の見直し要素数の確認、ネストの深さの確認など、を行う。例えば、繰り返し/ネストの可能性のある項目は属性ではなく要素にする[9]。	正規化XMLスキーマ ネームスペース構造
⑤ データフロー分析 / データフローダイアグラム、 Create Refer Update Delete分析表、	設計したXMLスキーマの重複や無駄がないかを、データのライフサイクル、利用、の観点から検証する。	リレーショナルデータモデリングの成果物と同じ
⑥ 主要属性モデルと正規化モデル の融合 / 論理モデル	トップダウンとボトムアップの成果物を融合する。	XMLスキーマ XMLネームスペース
⑦ 論理から物理への変換 / 物理モデル	リレーショナルデータモデリングと同様の手法でXMLインデックス設計を行う。性能を考慮し、要素数の確認や文書サイズ確認なども行う。	XMLスキーマ XMLインデックス定義

考察し、表3にまとめた。

リレーショナルデータモデリングと同様の手法やフローでXMLスキーマ設計を行い、必要な成果物が作成できることが表3よりわかる。同時に、ステップ④の作業内容に記述したような属性と要素の選択、要素数の確認など、XMLスキーマ設計特有の作業があることもわかった。

リレーショナルデータとXMLデータの両方を扱えるハイブリッドDBの特性を活かすには、データの特徴に応じた使い分けが必要である。一方で、ユーザーやアプリケーションからみたデータモデルはあくまで一つであって、単にモデルの表現方法が異なるだけである。そこで、リレーショナルデータモデリングとXMLデータモデリングを一つにしたフローとして図9を提案する。

図9にいたった考え方を以下に述べる。

- ・ リレーショナルデータとXMLデータという形式に依存しない、企業データのあり方を表すデータモデルの核として①のERモデルを位置付ける。
- ・ ① ②のエンティティモデルを作成する段階で、XMLエンティティ候補の洗い出しを行う。この理由は、この段階のイベント・リソース定義により、データがどこから来て、どのようにやり取りされるのかが明らかになり、入出力にXSLTを使用するか、外部連携にXMLを使用するかなど、XML化が望ましい要件がおおよそ決まるからである。
- ・ エンティティのリレーショナル/XML選択を受けて、ボトムアップアプローチに進むことで、リレーショナルデータとXMLデータそれぞれの正規化を実施できる。
- ・ 前述のように、ユーザーやアプリケーションにはXMLデータとリレーショナルデータを統一したデータモデルが必要なので、両方のエンティティを含み、かつ、区別がわかる論理モデルを⑥で作成する。具

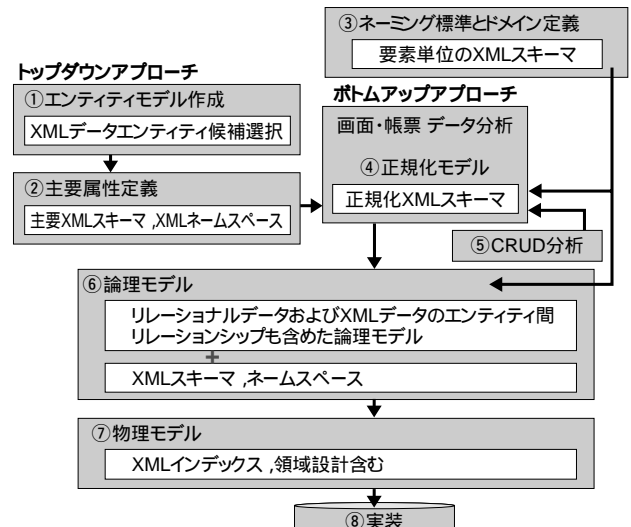


図9. ハイブリッドDB設計におけるデータモデリング・フロー

体的には、論理モデル上のXMLデータ部分のエンティティについては、エンティティ名と主要属性に加えて、XMLスキーマ名を記述しておく程度にしておく。これにより、リレーショナルデータとXMLデータの関係が把握できる論理モデルが実現できる。

- ・ 論理モデル作成後にXMLスキーマ変更が発生しても、①②の成果物となるERモデルや、⑥の論理モデル自体への変更が少ないフローである必要がある。
- ・ XMLのタグ名や値の型/範囲を定めた③のネーミング標準/ドメイン定義は、⑥で定めるXMLスキーマを構成するパーツであり、リレーショナルデータにおける標準化という役割以上に重要な役割を持つ。③のネーミング標準/ドメイン定義を、④の正規化モデルで表されるXML階層構造に従ってマージすることで、⑥のXMLスキーマを導き出すことができる。

## 5. おわりに

XML DB機能とRDB機能を併せ持つハイブリッドDBを、エンタープライズ規模のシステムで活用していくためのデータモデリングについて本論文で考察し、従来のリレーショナルデータモデリングとXMLデータモデリング(XMLスキーマ設計)をあわせたデータモデリング・フローを提案した。この提案では、ERモデルを共通の出発点とし、リレーショナルデータとXMLデータの両方を表現する論理モデルを作成することで、企業として統一されたデータモデルを実現している。さらに、XMLスキーマの柔軟性を損なわないように、XMLスキーマ変更時に論理モデルに変更が及ばないように配慮している。プロセスやユーザーにとっては両者ともデータであり、ビジネスの急速な進化に対応するためには統一され、かつ、柔軟なデータモデルが必要なのである。

業界や各種団体が規定しつつある標準XMLスキーマに基づくXMLメッセージやXMLコンテンツ以外に、企業内データをXMLデータとしてDBに格納することが定着するには時間がかかると思えるが、ハイブリッドDBが定着し、ハイブリッドDBを活用したソリューションが増えてくるにつれてXMLデータの割合が増えてくると筆者は考えている。そのためにも、本論文で述べたデータモデリング・アプローチは有用であると考えている。

本論文では、モデリングについての考察を述べたが、どのようなデータをXML化すべきかについてまで言及することができなかった。管理性や拡張性、XMLに適するデータ構造(階層構造や繰り返しが多いデータ)など、いくつかポイントが考えられる。また、実際のシステム開発の現場では、業務やシステム毎にデータをどちらの形態を主流とするか選択する必要がある。以下のような要素についても考慮する必要がある。

- ・ 技術者のスキルセット  
(SQLが主流かXQueryが主流かなど)
- ・ 既存データとの連携  
(既存データからのデータ取り込み度合いなど)
- ・ 既存アプリケーションの利用  
(SQL対応アプリケーションの存在)

筆者は、今後も継続的に企業データのXML化についての考察を深め、本論文で提言しているモデリング・アプローチを実際のビジネスケースで実践していく所存である。

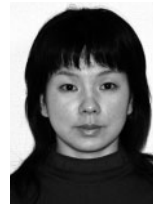
## 謝辞

本論文の執筆にあたって、日本アイ・ビー・エムの社内コミュニティ活動であるITSマスターコース'実

践!データベース論理設計'のメンバーの皆様との議論を参考にさせていただきました。また、XMLスキーマとXMLデータモデリングについてシリコン・バレー研究所XML開発チームに助言をいただきました。あらためて深謝いたします。

## 参考文献

- [1] 曾根原登, 岸上順一, 赤植淳一.: メタデータ技術とセマンティックウェブ, 電機大出版局, ISBN4-501-54060-5 (2006).
- [2] IBM Japan: <http://www.ibm.com/jp/software/data/db2/v9/> (2006.11).
- [3] 丸山則夫: XMLのスキーマ設計ハンドブック, ソフトリサーチ・センター, ISBN4-88373-170-7 (2002).
- [4] Uche Ogbuji: "XML標準の調査," IBM developerWorks, [http://www.ibm.com/jp/developerworks/xml/040227/j\\_x-stand1.html](http://www.ibm.com/jp/developerworks/xml/040227/j_x-stand1.html) (2004.1.20).
- [5] David Mertz: "RELAX NGによる逆襲," IBM developerWorks, [http://www.ibm.com/jp/developerworks/xml/030425/j\\_x-matters25.html](http://www.ibm.com/jp/developerworks/xml/030425/j_x-matters25.html) (2003.2).
- [6] Ayesha Malik: "Design XML Schema using UML," IBM developerWorks, <http://www.ibm.com/developerworks/library/x-umlschem/> (2003.2.1).
- [7] 真野正: 実践的データモデリング入門, 翔泳社, ISBN4-7981-0385-3 (2003).
- [8] 三好康之, 山下真吾, 松田幹子: テクニカルエンジニア「データベース」, 翔泳社, ISBN4-7981-1006-X (2005).
- [9] Matthias Nicola: "15 best practices for pureXML performance in DB2 9," IBM developerWorks, <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0610nicola/index.html> (2006.10).



日本アイ・ビー・エム株式会社  
アドバンスド・テクニカル・サポート  
ベータプログラム推進  
ICP-コンサルティングITスペシャリスト

**山川 多美** Tami Yamakawa

## [プロフィール]

オープン・プラットフォームにおけるInformation Management製品のベータプログラム推進を日本と韓国にて担当するチームリーダー。製品品質向上, 事例創出, スキル育成に力を注ぐと同時に、先進プロジェクトにおいて、DB2を中心とした技術支援活動を行っている。