

# データベースのXML化による効果

柴田 和紀

富士通ネットワークテクノロジーズ株式会社

## 1. はじめに

近年 Extensible Markup Language (以下、XML) がインターネット標準メタ言語として注目され、論理的な文書構造の記述やデータ交換等幅広い分野での活用が報告されている。現在、C言語で開発された既存ソフトウェアと Web 系 (Java, HTML) ソフトウェアとのインタフェース技術として XML を導入し、開発に効果を上げている。

本論文では、XML が既存ソフトウェアと Web 系ソフトウェアとを組み合わせた開発にもたらす効果について効率、品質の2点から報告する。

## 2. XML 選択の経緯

既存ソフトウェア製品に Web 系言語で開発したアプリケーション機能を追加するソフトウェア開発における効率面の課題として、以下の3点が挙げられる。

新規および既存ソフトウェアがアクセスする DB の構成

Web 系言語との共存を前提として作られていない既存ソフトウェアの改造量の縮小  
試験用の DB データ作成方法

これらの課題を解決するために言語間 I/F と DB アクセス処理が小規模で開発できる技術を中心に調査した。その結果、次の理由から DB を XML 化することにした。

XML はマークアップ言語であり、文書はテキストファイルになるため、データ作成が容易である

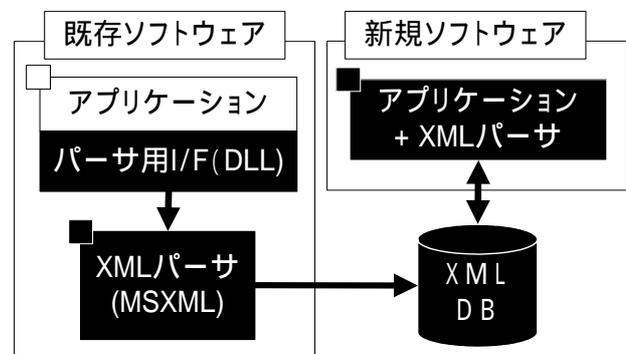
ソフトウェアの動作環境に合わせたデータの読み替え処理が不要

DB の I/O として XML パーサが公開されており、標準化 API (DOM) を使用することでデータアクセスが可能

XML パーサは様々な言語で作成されたものが公開されているため、既存ソフトウェアとも親和性が良い

## 3. XML を組込んだ開発

XML を C 言語開発された既存ソフトウェアに組込むため、以下の構成とした (図1)。



矢印はデータの流を表す

図1. ソフトウェア構成

既存ソフトウェアの改造は、次の作業を実施することで開発量を抑えた。

### XML パーサの移植

XML パーサ (MSXML) は、C++用ソースコードが公開されているため、既存ソフトウェアと同一環境で動作するようにした。さらに、XML パーサをアプリケーションと分離し、アプリケーションの動作負荷を軽減させた。

### パーサ用 I/F DLL の作成

XML-DB へのアクセスが必要なアプリケーション用に、XML パーサとの I/F DLL を作成した。そのため、アプリケーションの改造は DLL 呼び出し処理の追加だけとした。

新規ソフトウェアには、開発言語ごとに対応する XML パーサ (C++:MSXML Java:XML Parser for Java) を組込んだ。

## 4. 導入効果

XML を利用した効果を、従来型 DB を使用した作業と比較し評価した。

### 4-1. 開発効率

効率面での効果を設計/製造/試験の工程ごとに検証する。

#### 設計工程

バイナリ形式の従来 DB は、バイトバウンダリや DB サイズを考慮した設計が必要である。また、リレーション用のオフセットや仕様追加のための領域確保も必須である。そのため、DB 構成検討や設計レビューに多くの工数を必要とする。

DB を XML 化し、XML の技術を利用することでリレーションが簡易化し、必要以上の DB 領域が不要となった。そのため、DB サイズが半減し、さらにレビュー工数削減に繋がった。

#### 製造工程

従来型 DB での開発は、アプリケーション用に DB アクセス用のモジュール開発が必須である。また、データ展開や排他制御のための処理が複雑化する傾向にある。

DB アクセスに XML パーサを利用することでモジュール作成、データ展開処理、排他制御の開発量がゼロとなった。また、XML で用意されている DTD を活用することで DB に登録するデータの正当性検証も不要となった。

#### 試験工程

XML パーサでデータアクセス部が隠蔽されているため、通常、DB 新規作成時に必要であるアクセス処理、アプリケーションの動作検証が不要となった。また、テキスト形式であること、XML データ作成用のソフトが多く公開されていることから、これらを活用することでさらに試験用 DB 構築作業が効率化された。

しかし、DB を XML 化することで効率を悪化させる要因もあった。設計時は、XML による DB 設計ノウハウ吸収時間。製造工程では XML パーサの動作理解とその I/F 確立処理作成。試験工程では、XML パーサとの I/F 検証と問題切分けである。

次に、開発全体で XML 適用効果を検証する。従来手法で予想した開発工数を 100%とすると、XML 適用時では 84%となり、16%の工数削減となった(表1)。



表1. 開発工数の比較

### 4-2. 品質

XML パーサにより DB へのアクセス処理が隠蔽されたため、DB への I/O 処理に問題が発生しなかった。また、アプリケーションと DB との I/F を DLL 化して開発量を抑えたこと、DTD を利用してデータの正当性を確保したことから DB 処理問題も予想件数を下回る結果となった(表2)。

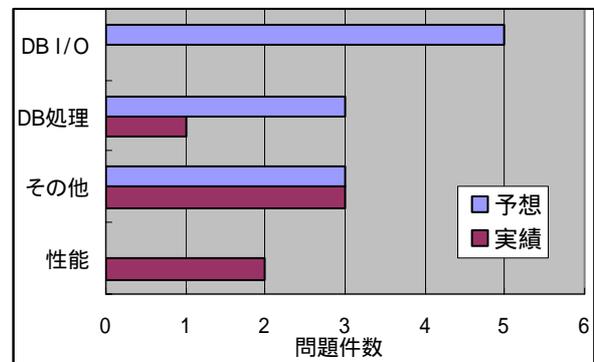


表2. DBに関する問題内訳

しかし、XML パーサとの I/F 用 API として DOM を使用したため、DB 内の保存データ量が多くなるとメモリ資源が枯渇してしまう性能問題が発生してしまった。これは、DOM が XML データの構造解析結果をメモリ上に構築するためである。

現在では、DOM に較べて使用メモリ量の少ない SAX(The Simple API for Xml)が公開されているため、次版以降で、XML パーサを変更する予定である。

## 6. まとめ

今回の開発では、XML の使用を前提としていない既存ソフトウェアに対して技術を導入した。その結果、効率、品質面ともに良好の結果が得られた。

今後は、XML の適用範囲を拡大し、その効果を検証する予定である。