

IMS における XDMS を利用した文書管理システムの検討

須賀院 吉伸[†] 乃村 能成[‡] 谷口 秀夫[†][†]岡山大学工学部情報工学科 [‡]岡山大学大学院自然科学研究科

1. はじめに

次世代ネットワーク (NGN) における IP Multimedia Subsystem (IMS) は, XML Document Management Server (XDMS) を提供している. XDMS は, Instant Messaging (IM) のコンタクトリストや携帯電話の通信記録など, 比較的小さな XML 文書を保存, 管理する目的で使用される. このため, XDMS は, 一般文書を扱うための検討が十分ではない. そこで, 本稿では, オフィス文書をはじめとする一般文書を XDMS で管理する際の利点と問題点を明らかにし, XDMS を利用した文書管理システムを検討する.

2. データベースとしての XDMS

2.1. XDMS 概要

IMS では, IM や Push-to-Talk over Cellular (PoC) など様々なサービスがアプリケーションサーバ (AS) によって提供される. IMS で提供されるサービスには, 複数の AS が連携するものがある. このようなサービスでは, 各 AS が個別にユーザ情報や設定情報を管理すると, 複数 AS 間での情報共有が困難となる. IMS では, このような共有情報を格納するため, XDMS を提供している. XDMS は “OMA XML Document Management”^[1] により定義されている. また, NGN では, IMS の外にも AS を設置できる. このため, AS による様々な付加サービスが期待できる. AS は, XML Configuration Access Protocol (XCAP)^[2] を用いて XDMS 上の XML 文書を操作する. XDMS と AS の関係を図 1 に示す.

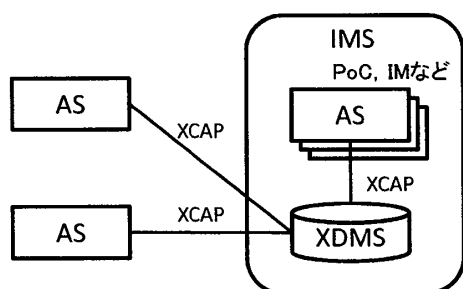


図 1 XDMS と AS の関係

Design of the document management system using XDMS in IMS
Yoshinobu SUGAIN[†], Yoshinari NOMURA[‡], and
Hideo TANIGUCHI[†]

[†]Faculty of Engineering, Okayama University

[‡]Graduate School of Natural Science and Technology,
Okayama University

2.2. XML Configuration Access Protocol

XCAP とは, XDMS 上の XML 文書を操作するプロトコルであり, HTTP をベースとしている. AS は, XDMS 上の URI に GET, PUT, DELETE 要求を送信することで, XML 文書の取得, 更新, 削除を行う. また, XCAP では, Application usage という定義書を AS に要求している. これは, AS が操作する XML 文書の XML スキーマや XDMS へのアクセス制限, URI の解釈について定義した文書である. XCAP を利用した XML 文書取得例を図 2 に示す.

```
GET https://xdms.example.com/my-service/users/
                                     (1)      (2)
nomura/friend-list/~~/header
(3)      (4)      (5)
```

図 2 XML 文書取得例

XDMS では, 操作対象の XML 文書をリソースと呼ぶ. 図 2 の例では, xdms.example.com サーバの保持する (1), Application usage “my-service” に基づいた (2), ユーザ nomura の (3), friend-list リソース中の (4), header 要素のタグ情報 (5) を取得する.

2.3. 文書管理における XDMS の利点と問題点

XDMS を用いて一般文書を管理する利点を以下に述べる.

- (1) データストレージを AS 自身で保持しなくてよい. このため, 低コストで AS を構築および運用できる.
- (2) XCAP は, HTTP ベースのオープンなプロトコルであり, 動作も単純である. このため, 特殊なソフトウェアに依存することなくクライアント機能を実装できる.
- (3) XDMS では, 認証機能と各リソースへのアクセス権管理機能を標準で提供している. また, これらの機能は, Application usage を記述することにより動作を制御できる. この機能を利用することにより, AS にデータ共有インタフェースを実装することなく, データを共有できる. 通常, AS のデータを他の AS と共有する場合, データ共有インタフェースを実装する必要がある. このインタフェースは, 複雑な機能を持ち, 実装が難しい.

次に, 問題点を以下に述べる.

- (1) XDMS では, ディレクトリ操作にあたる操作がない. このため, 大量の XML 文書を階層構造

のあるファイルシステム内の要素とみなして管理しにくい。

- (2) XCAP には、リソースの一覧を取得する機能がない。このため、AS は URI を知らないリソースへアクセスできない。
- (3) XCAP には、リソースの排他制御機構がない。XDMS 上のリソースは複数の AS から参照可能だが、リソースを更新する AS は慣例的に 1 つである。このため、複数の AS から同時に更新されることが想定されていない。同一リソースを更新する場合は、AS 間の排他制御機構が必要となる。

3. システム設計

3.1. 設計方針

2.3 節で述べた問題を解決するため、XDMS 上にある各リソースの一覧を保持するリソースを作成する。本文書管理システムでは、このリソースをインデックスと呼ぶ。また、本文書管理システムでは、XDMS 上の各リソースをユニークな ID で管理する。インデックスは、この ID と文書名の対応情報を格納する。また、インデックス内において文書間の階層構造と文書のロック状態を表現する。インデックスの導入によって、2.3 節で述べた問題点は以下のように解決できる。

- (1) インデックス内の階層構造を編集することで、ディレクトリの作成、変更および削除が行える。また、文書そのものに相当するリソースを編集せずに文書のディレクトリ間移動操作ができる。
- (2) AS は、インデックスを参照することで文書名一覧および各文書名に対応するリソースの ID を取得できる。
- (3) インデックス内に排他制御情報を表す要素を用意して排他制御を行う。また、HTTP のステータスコードを利用し、テストアンドセット操作を実現する。

3.2. インデックス

インデックスは、各リソースの情報を保持する。インデックスの例を図 3 に示す。directory 要素が仮想的なディレクトリを表す。また、document 要素は、name 属性と id 属性を持つ。これは、それぞれ文書名と対応するリソースの ID を表す。また、lock 要素は、その親要素に対応するリソースがロックされていることを表す。リソースをロックするためには、テストアンドセット操作を XCAP で実現する必要がある。この実現には、PUT 要求に対するレスポンスのステータスコードを用いる。AS は、ロック対象の document 要素に対して lock 要素の作成要求を送信する。lock 要素が存在せず、ロックに成功した場合、ステータス

コードは 201 Created である。すでに lock 要素が存在し、ロックに失敗した場合、ステータスコードは 200 OK である。このようにして、テストアンドセット操作を実現する。

```
<?xml version="1.0" encoding="UTF-8" ?>
<index xmlns="http://okayama-u.ac.jp/xdms">
  <head> ... </head>
  <home-directory>
    ⋮
    <directory name="dir1">
      <document name="sample.xml" id="412f85">
        <lock/>
        <created> 2008-09-10 </created>
        <author> nomura </author>
      </document>
    </directory>
  </home-directory>
</index>
```

図 3 インデックス例

4. おわりに

IMS における XDMS の概要と XDMS において操作インタフェースとして使用される XCAP の概要について述べた。また、XDMS を用いて一般文書を管理する利点および問題点を明らかにした。

インデックスの導入により XDMS 上で仮想的なディレクトリ操作およびリソースのディレクトリ間移動操作が実現できることを示した。また、インデックスを参照することで、リソース一覧の取得操作が実現できることを示した。PUT 要求に対するレスポンスのステータスコードを用いることで、テストアンドセット操作を実現し、この操作を使用したリソースの排他制御機構が実現できることを示した。

残された課題として、文書管理システムの実装と性能評価がある。

謝辞 本研究の一部は、科学研究費補助金・若手研究 (B) (課題番号: 21700139) による研究費、及び日本電信電話株式会社 NTT サービスインテグレーション基盤研究所と国立情報学研究所の提供する研究設備、回線を活用した。ここに記して謝意を示す。

参考文献

- [1] “XML Document Management Specification V1.1”, Open Mobile Alliance, 27 June 2008.
- [2] J. Rosenberg, “The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)”, IETF RFC4825, May 2007.