

IMSにおけるXDMS高度利用のためのソフトウェアフレームワーク

乃村 能成[†] 長尾 武憲^{††} 谷口 秀夫[†] 南 裕也^{†††} 並河 大地^{†††}[†] 岡山大学大学院自然科学研究科 ^{††} 岡山大学工学部^{†††} 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

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, 排他制御をはじめとした複雑な操作を備えていない^[1]. このため, XDMS を利用する Application Server (AS) は, 必要に応じて独自にこれらの機能を実装しなければならない. 将来の AS 展開を考えた場合, これらの独自実装は, AS 開発の工数増大ばかりか, AS 間連携の標準不在による相互運用性の低下を招く恐れがある. そこで, 本稿では, XDMS をより高水準なインタフェースで扱えるソフトウェアフレームワークについて検討する. これにより AS 開発の工数削減や相互運用性の向上が期待できる.

2 XDMS と AS の関係

IMS では, Instant Messaging (IM) や Push-to-Talk over Cellular (PoC) など様々なサービスが AS によって提供される. AS は, 自身のデータを保存するために XDMS を利用できる. XDMS は, “OMA XML Document Management”^[2] 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. XDMS と AS の関係を 図 1 に示す.

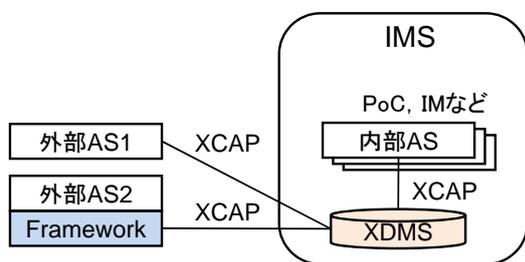


図 1 XDMS と AS の関係

AS は, XML Configuration Access Protocol (XCAP)^[3] を用いて XDMS 上の XML 文書 (リソースと呼ぶ) を

Software framework for effective utilization of the XDMS in IMS

Yoshinari Nomura[†], Takenori Nagao^{††}, Hideo Taniguchi[†], Hiroya Minami^{†††} and Daichi Namikawa^{†††}

[†] Graduate School of Natural Science and Technology, Okayama University ^{††} Faculty of Engineering, Okayama University ^{†††} NTT Service Integration Laboratories, NTT Corporation

操作する. XCAP は, HTTP をベースとしており, AS は, XDMS 上の URI に GET, PUT, DELETE 要求を送信することで, リソースの取得, 更新, 削除を行う.

図 1 において, 内部 AS は, XDMS を IM のコンタクトリスト管理や携帯電話の通信記録管理に用いている. 内部 AS が扱うリソースは, 比較的サイズが小さく, 単一の AS のみから操作される. 一方で, 今後発展が期待される外部 AS では, AS 間リソース共有/排他制御や全文検索など, AS をまたがった高度な操作の実現が求められる. RDB を用いた開発では, これらの操作は, RDB 側に具備されているため, AS 開発の負担は小さいが, XDMS を用いた場合, すべてを AS 側で実装せざるを得ない. これら XDMS を操作する機能は, AS 独自ではなく, 何らかの共通フレームワーク上で構築されることが望ましい.

3 XDMS 高度利用のためのソフトウェアフレームワーク

3.1 要求

XDMS 対応の AS を通常の Web アプリケーションに近い開発効率で実現するために, フレームワークに対し以下が求められる.

- (1) 既存フレームワークとの高い親和性
通常の Web アプリケーション (RDB+DBI+O/R マップ) を少ない工数で XDMS に対応させたい.
- (2) オブジェクトとリソースのマッピング機能
XDMS (とそのデータ) とプログラミング言語におけるオブジェクトのマッピングを抽象化する機能が望まれる. 特に, XML とオブジェクトのマッピング機能が必要である.
- (3) XCAP プロトコルの隠蔽機能
XDMS へのアクセスプロトコルである XCAP は, REST インタフェースを意識した構造を持つ. リソース名と URI の相互変換, HTTP メソッドと DB 操作とのマッピング, 認証方式を隠蔽することが必要である.
- (4) インデックス機能
XDMS には, ある AS に属するリソースの一覧 (インデックス) を取得する操作がない. フレームワーク独自にインデックスを管理し, AS から隠蔽する必要がある.

(5) 排他制御機能

XDMS は、排他制御機構を持たない。そのため、排他制御機構をフレームワークに内蔵し、AS 間でプロトコルを統一する。

3.2 対処と実現

まず、要求(1),(2),(3)を考慮し、既存 Web アプリケーションフレームワークである Ruby On Rails [4] (以下 Rails) とその一部である ActiveRecord を採用し、それらを改変することで実現する。ActiveResource は、XML 文書を HTTP で取得して Ruby のオブジェクトにマッピングする操作を抽象化するライブラリで、RDB の O/R マップ (ActiveRecord) と類似した機能を持つ。従って、ActiveResource により、XCAP (HTTP) を隠蔽しつつ、オブジェクトとリソース (XML 文書) のマッピングを実現できる。また、ActiveResource と ActiveRecord のインタフェースには互換性があるため、既存の RDB + ActiveRecord に基づくアプリケーションを少ない工数で XDMS + ActiveResource 対応に改変できる。ただし、XCAP と ActiveResource が想定する HTTP の間には若干の違いがある (例えば、XCAP は POST を規定していない) ため、ActiveResource の改変を必要とする。

次に、要求(4),(5)について、著者らは、XDMS における両機能の欠如が実際のアプリケーションに与える問題を指摘し、その解決方法としてインデックスリソースの導入を提案している [1]。具体的には、インデックスリソースと呼ばれる XML 文書を新たに規定し、その中に AS が保持するリソースの一覧を列挙し、メタ情報や排他制御の情報までを含める方式である。このインデックスリソースによる管理方式を ActiveResource 上に実現する。

3.3 利用例

提案した方針に基づいてプロトタイプを作成した。ここでは、プロトタイプにおけるリソースの新規作成、読み出し、及び更新の例を示す。図 2 は、XDMS アクセスに必要な URI やパスワードを設定する例である。Document というクラスを作成し、XDMS のリソースと結び付けている。このようにわずかな設定で XDMS を操作できる。図 3 は、title と comment を属性として持つリソースを新規作成し、XDMS へ保存する例で、図 4 は、同リソースを XDMS から取得し、title を変更し、再び XDMS に書き戻す例である。最後に削除 (delete) している。# で始まる行は、対応する XCAP プロトコルの流れと、doc 変数の内部状態を示している。例にはないが、Document.find(:all) とすること

で、インデックスの取得ができる。

```
class Document < ActiveRecord::Base
  self.site = "https://xdms.example.com/au/nom/"
  self.user = "nom"
  self.password = "*****"
  self.auth_type = :digest
end
```

図 2 設定の例

```
doc = Document.new(:id => 1, :title => "タイトル",
                  :comment => "コメント")
doc.save

# PUT http://xdms.example.com/au/nom/1.xml
# --> 201 Created
```

図 3 新規作成の例

```
doc = Document.find(1)
doc.title = "タイトル改"
doc.save
doc.delete

# GET http://xdms.example.com/au/nom/1.xml
# --> 200 OK
# doc: {:comment => "コメント", :title => "タイトル"}
#
# PUT http://xdms.example.com/au/nom/1.xml
# --> 200 OK
#
# doc: {:comment => "コメント", :title => "タイトル改"}
#
# DELETE http://xdms.example.com/au/nom/1.xml
# --> 200 OK
```

図 4 取得、更新、削除の例

4 おわりに

本稿では、NGN における IMS と XDMS の概要、それらと Application Server (AS) との関係について説明し、今後の AS 発展において必要とされるソフトウェアフレームワークに対して、要求される機能を示した。また、要求に対する解決案を提示し、それらの提案に基づいたプロトタイプとその動作例を示した。残された課題として、未実装部分 (排他制御) の実装及び性能評価がある。

参考文献

- [1] 須賀院 吉伸, 乃村 能成, 谷口 秀夫, “IMS における XDMS を利用した文書管理システムの検討,” 第 72 回 情報処理学会全国大会講演論文集, DVD-ROM, March, 2010.
- [2] Open Mobile Alliance, “XML Document Management Specification V1.1,” 27 June 2008.
- [3] J. Rosenberg, “The Extensible Markup Language (XML) Configuration Access Protocol (XCAP),” IETF RFC4825, May 2007.
- [4] “Ruby On Rails,” <http://rubyonrails.org/>