オーディオビジュアル複合情報処理 40－11
（２００３．３．７）

IP ネットワークにおけるメタデータ交換サービスの実装と評価

鈴木　順一[†‡]　スィープラスートスック パオ[‡]　亀山　渉[‡]

[‡]早稲田大学大学院 国際情報通信研究科
〒169-0051　東京都新宿区西早稲田 1-3-10　早稲田大学 29-7 号館
[†]日本テレコム株式会社　情報通信研究所
〒104-0032　東京都中央区八丁堀 2-9-1 秀和東八丁堀ビル 5F

E-mail:junichi@fuji.waseda.jp

あらまし TV-Anytime Forum で標準化が進んでいる IP ネットワーク上でのメタデータ交換サービスのプロトタイプシステムを実装した。TV-Anytime メタデータサービスでは SOAP を使ってメタデータ交換を行っている。そのサービス概要やサービス API 及び実装したプロトタイプシステムの詳細について説明する。そしてシステムの評価実験の結果と考察についても報告を行う。また最後に今後のメタデータ流通について課題等を述べる。

An Implementation of Metadata Exchange Services Over IP Network

Junichi Suzuki[†‡]　Pao Sriprasertsuk[‡]　Wataru Kameyama[‡]

[‡]Graduate School of Global Information and Telecommunication Studies Waseda University
1-3-10 Nishiwaseda, Shinjuku, Tokyo, 169-0051 JAPAN
[†]Japan Telecom Co.,Ltd Information and Communication Laboratories
2-9-1 Hachoubori, Chuo-ku, Tokyo, 104-0032 JAPAN

E-mail:junichi@fuji.waseda.jp

Abstract Nowadays, the integration of computer world, broadcast world and communication world is progressing. Moreover, the development of multimedia technology standards and advanced technologies are making various new service models. For example, TV-Anytime Forum is defining service models in which users can explore and acquire TV Content distributed over a variety sources anytime and anywhere. In addition, the forum is also defining specification that allow consumer to personalize and control preferred content such as exchange, sharing and redistribution. To make this real, metadata exchange over IP network is a key technology.

Therefore, in this paper, implementation of metadata exchange service over IP networks based on TV-Anytime Forum's specification are described and some experimental results are also mentioned as well as its evaluation

1. はじめに

近年の CPU やハードディスクの低価格化、大容量化に伴い、数週間分の TV 番組を自分の家のハードディスクに全て録画しておき、いつでも好きな時に TV 番組を見ることが出来る時代も遠くなくなってきた。

また ADSL の普及によるインターネット回線の高速化、また携帯電話の進化無線技術の発展などによるアクセス手段の多様化により通信と放送とコンピューターを融合した新しいサービスが開始されようとしている。

このような時代背景のもと国際標準化団体である TV-Anytime Forum は 1999 年に発足し、現在も放送の即時性とインターネットの柔軟性を兼ね備えた新しいサービスとそれを実現するシステムの国際標準化を進めている。

その TV-Anytime システムを実現する上でのキーテクノロジーの１つにメタデータがある。

メタデータとはデータを記述する為のデータであり、コンテンツに関する情報（番組タイトル、概要、放送時間、権利情報等）をテキストデータとして記述しておく。必要な時にそのメタデータを参照して、そのコンテンツの情報を知ることが出来る。

2. TV-Anytime Forum メタデータサービス

TV-Anytime Forum では TV-Anytime システムで使用するメタデータのスキーマを標準化している[1]。

またユーザがどのようにして TV-Anytime メタデータを得ることが出来るのか、またメタデータサービスプロバイダーがどのようにして TV-Anytime メタデータを提供する（メタデータサービス）ことが出来るのかを記述し、その API を標準化している。

今回我々は TV-Anytime メタデータサービスの API が必要十分であるかどうかを検討することを目的に、その API に従ったメタデータサービスシステムのプロトタイプの実装を試みた。

本稿ではそのメタデータサービスの概要を説明すると共に、実装システムの詳細、システム評価について報告を行う。

3. メタデータサービス API

TV-Anytime メタデータサービス仕様書[2]には以下の４つのサービスが規定されている。

・Get_Data Service
・Describe_Get_Data Service
・Submit_Data Service
・Describe_Submit_Data Service

今回はその中の Get_Data Service と Describe_Get_Data Service の２つのサービスを実装した。

3.1. Get_Data Service

Get_Data Service とはクライアントからの要求に対して該当するメタデータを返信するサービスである。図 1 の Get_Data Service のメッセージ例を図 1 に示す。

図 1 の要求メッセージはタイトルが Titanic である番組や映画の ProgramReviewTable（番組批評情報メタデータ）を要求している例を表している。また図１の応答メッセージの TVAMain タグ以下は TV-Anytime で標準化されているスキーマのメタデータがそのまま入っている。

```
<get_Data xmlns='http://www.tv-anytime.org/2002/11/transport'
   xmlns:tvaf='http://www.tv-anytime.org/2002/11/transport/fieldIDs'>
   <QueryConstraints >
     <Predicate fieldID='tvaf:Title' fieldValue='Titanic'/>
   </QueryConstraints>
   <RequestedTables>
      <Table type='ProgramReview'/>
   </RequestedTables>
</get_Data>                                          要求メッセージ
```

```
<get_Data_Result>
  <TVAMain xmlns="http://www.tv-anytime.org/2002/08/metadata">
    <ProgramDescription>
      <ProgramReviewTable>
        <ProgramReviews version='1'>
          <Program crid='crid://km.giti.waseda.ac.jp/ Titanic'/>
          <Review>
            <FreeTextReview xml:lang='jp'> 面白い</FreeTextReview>
          </Review>
        </ProgramReviews>
      </ProgramReviewTable>
    </ProgramDescription>
  </TVAMain>
</get_Data_Result>                                   応答メッセージ
```

図 1 Get_Data Service メッセージ構造

3.2 Describe_Get_Data Service

Describe_Get_Data Service は Get_Data Service がどのようなサービスを提供しているのか（CapabilityDescription）を説明する為のサービスである。

クライアントは必ず初めに Describe_Get_Data Service 要求を行い、CapabilityDescription を

応答メッセージとして受信した後に、Get_Data Service に対して要求メッセージを作成する。また Describe_Get_Data Service と Get_Data Service は同じアドレス、同じポート番号で提供されなければいけないとされている。

Describe_Get_Data Service のメッセージ例を図 2 に示す。図 2 の応答メッセージの例では Describe_Get_Data Service 要求を送ったメタデータサービスは番組や映画の Title で検索が可能で、ProgramInformationTable(番組内容情報メタデータ)、ProgramLocationTable（番組配信情報メタデータ）、ProgramReviewTable（番組批評情報メタデータ）を提供可能であることが分かる。

```
<describe_get_Data xmlns='http://www.tv-anytime.org/2002/11/transport'/>
                                                        要求メッセージ
<describe_get_Data_Result service='1'
   xmlns='http://www.tv-anytime.org/2002/11/transport'
   xmlns:xsi='http://www.w3c.org/2001/XMLSchema-instance'>
   <Name>Waseda University's Test get_Data Service</Name>
   <Description>You can get a sample metadata.</Description>
   <AvaliableTables
   xmlns='http://www.tv-anytime.org/2002/11/transport/fieldIDs'>
      <Table canQuery='tvaf:Title' xsi:type='ProgramInformationTable'/>
      <Table canQuery='tvaf:Title' xsi:type='ProgramLocationTable'/>
      <Table canQuery='tvaf:Title' xsi:type='ProgramReviewTable'/>
   </AvaliableTables>
</describe_get_Data_Result>                             応答メッセージ
```

図 2　Describe_Get_Data Service メッセージ構造

4. メタデータサービスシステム概要

4.1 サービス手順

クライアントが TV-Anytime メタデータサービスを利用するまでには図 3 のステップを踏む必要がある。

①メタデータサービスを検索する。

②次に検索結果から得た URL に対して Describe_Get_Data Service の要求を行う

③Describe_Get_Data Service の応答メッセージよりメタデータサービスの Capability Description を得る。

④CapabilityDescription よりメタデータサービスに対して要求メッセージを送り、該当するメタデータを得る。

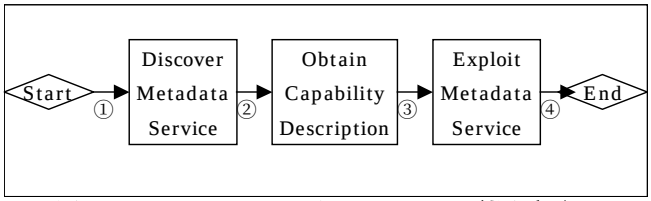

図 3 TV-Anytime メタデータサービス利用手順

4.2 Service Discovery

TV-Anytime メタデータサービスの検索（図 3 ①）には UDDI[3]と WS-Inspection[4]の一方又は両方を利用して行うことが可能である。サービス検索機能はオプション機能であったが、今回は UDDI を用いたシステムを実装した。

4.2 Transport Protocol

TV-Anytime メタデータの IP ネットワーク上での送信プロトコルには SOAP[5]を使うよう規定されている。ただし TV-Anytime メタデータサービスでは SOAP RPC(Remote Procedure Call)は使わないとされている。

次にプロトコルスタックを図 4 に示すが、オプションの Binary Encoding や Security Layer 部分はオプション機能であり今回は実装していない。

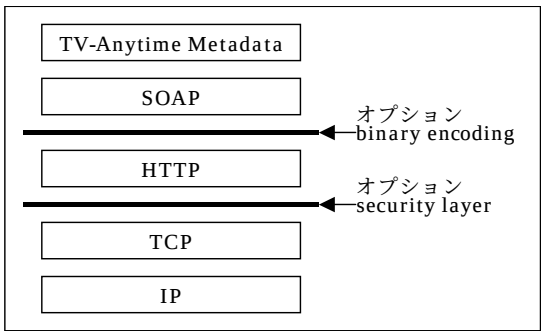

図 4プロトコルスタック

5. プロトタイプシステム実装

今回開発したプロトタイプシステムは UDDIRegistry と TV-Anytime メタデータサービスクライアントと TV-Anytime メタデータサービスサーバである。プロトタイプシステム概要図を図 5、使用したツールの一覧を表 1 に示す。

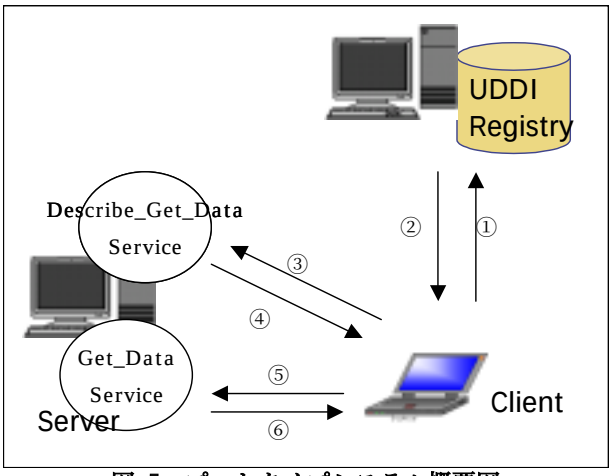

図 5　プロトタイプシステム概要図

表 1 実装システム使用ツール一覧

開発言語	Java1.4
XML パーサ	Xerces2.2
SOAP アプリケーション実行環境	Apache SOAP2.3.1+ Tomcat4.1
SOAP API	Apache SOAP API
UDDIRegistry	JWSDP1.0 付属 Registry サーバ（Tomcat+Xindice）
UDDI API	JAXR1.0
データベース	eXist0.9

（JWSDP1.0:SunMicroSystems 社 Java Web Services Developer Pack）

5.1 サービス実装

メタデータサービスの実装にはサーバ側の SOAP アプリケーションの実行環境として Apache XML プロジェクトが開発した Apache SOAP2.3.1 を使用した。また Web サーバには Tomcat4.1、データベースには XML ネイティブデータベースである eXist0.9 を利用した。

作成したサービスは Get_Data Service と Describe_Get_Data Service であり、図 1、図 2 に示すような要求メッセージに対して応答メッセージを送信する。

クライアントからの番組タイトル名での検索により番組内容情報メタデータ、番組配信情報メタデータ、番組批評情報メタデータを提供可能である。

メタデータは株式会社 WOWOW からご提供頂いた BS デジタル放送用の EPG(Electronic Program Guide、電子番組案内)データを TV-Anytime メタデータに変換したものをデータベースに格納した。

5.2 サービスの登録と検索

UDDIRegistry は JWSDP1.0 に付属している Registry サーバを利用した。

UDDIRegistry 内に登録してある様々なサービスを検索する方法の１つとして技術仕様（以下 tModel）による検索がある。サービスがどのようなインターフェースを持っているのかをキーとした検索方法である。

そこでまず初めに TV-Anytime が公開している WSDL[6]インターフェースファイル[7]に基づく tModel を UDDIRegistry に JAXR1.0(Java API for XML Registries)を使用して登録した。登録した際に tModelKey が UDDIRegistry より自動的に割当てられる。

またこの tModelKey を使用してメタデータサービスを UDDIRegistry に登録した。その登録 API にも JAXR1.0 を利用した。

クライアントはこの tModelKey を使用することにより、UDDI Registry 内で TV-Anytime メタデータサービス仕様準拠であるサービスを検索可能となる。

5.3 システムアーキテクチャ

次にプロトタイプシステムのアーキテクチャを図 6 に示す。UDDI Registry との API には JAXR1.0 を、サーバとの API には ApacheSOAP API 利用した。またユーザとのインターフェースには GUI（Graphical User Interface）を搭載した（図 7）。

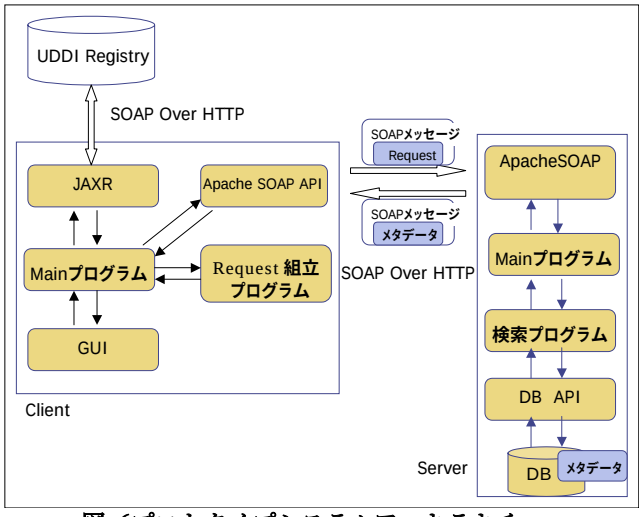

図 6プロトタイプシステムアーキテキチャ

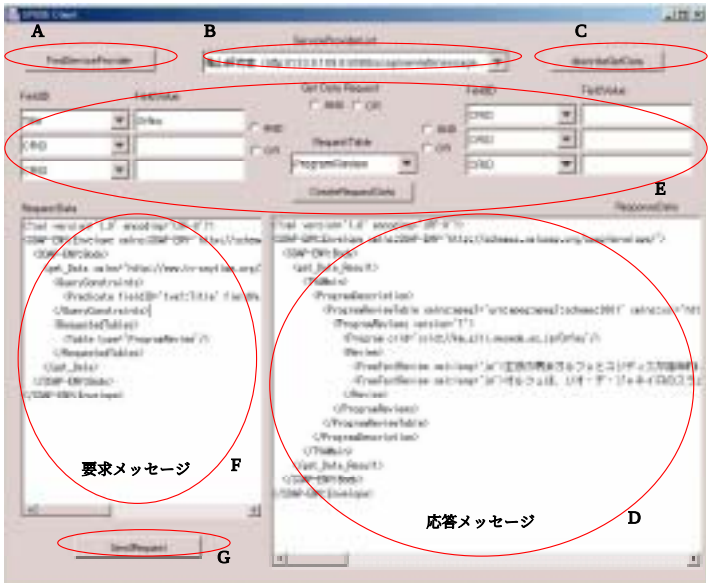

図 7クライアント GUI

5.4 システム詳細実行手順

本システムの詳細の実行手順を、以下の①～⑥（図 5①～⑥に対応）と図 5 と図 7 に基づき説明する。以下①～⑥の手順によりクライアントは GUI を使って必要なメタデータをサーバから得ることが出来る。

①図 7A ボタン押下。TV-Anytime メタデータサービス仕様に準拠しているサービスを tModelKey を利用して UDDIRegistry サーバから検索する。
②TV-Anytime メタデータサービスの URL 一覧が図 7B に表れる。
③図 7C ボタン押下し、URL 一覧から選択したメタデータサービスのアドレスに対して、Describe_Get_Data Service 要求メッセージを送る。
④メタデータサービスの Capability Description を得て、図 7D に表示する。
⑤CapabilityDescription を得ることにより、その Get_Data Service に対応した要求メッセージを図 7E で作成する。要求メッセージは図 7F に表示され、図 7G ボタンで送信する。
⑥サーバより該当するメタデータを受信したら図 7D に表示する。

6. 考察

6.1 評価

本システムの評価環境として表 2 に示す PC 上にシステムを実装し、研究室内の LAN に配置した。そして以下の評価方法・項目に従って実験をした。実験結果を表 3,4 に示す。

表 2　評価環境（PC スペック）

	クライアント	サーバ
OS	WindowsME	RedHatLinux7.3
CPU	Celeron600	PentiumIII800
Memory	192MB	256MB

評価方法・項目

・サーバから返って来るメタデータは１個に固定し、サイズを 10KB～1MB まで変化させた時の応答時間を測定（表 2）
・サーバから返って来るメタデータのサイズを 10KB 固定にし、メタデータの数を 5～100 個まで変化させた時の応答時間を測定（表 3）
（応答時間とはクライアントが Get_Data_Service に要求メッセージを送ってから応答メッセージを受け取るまでの時間とする。）

またメタデータを 10KB に固定したのは株式会社 WOWOW から頂いたメタデータの１番組あたりの平均メタデータサイズが 10KB 程度であった為である。

表 3　メタデータサイズ可変時の応答時間（メタデータ数１個）

メタデータサイズ(KB)	10	50	100	500	1000 (1MB)
応答時間(秒)	0.2	0.5	0.8	2.8	7.5

表 4　メタデータ数可変時の応答時間（メタデータサイズ１0KB）

メタデータ数(個)	5	10	20	50	100
応答時間(秒)	0.7	1.6	6.5	14.6	44.3

6.2 考察

表 2 を見ると１MB のメタデータを受け取っても 7 秒程度でメタデータサービスを受けられることが分かる。メタデータのサイズは今後ますます大きくなることが予想されるが、１番組あたりのメタデータサイズが１MB 程度を超えることは考えにくいので、返信するメタデータが１個であれば本システムでも十分サービスは提供出来る。

しかし、表 3 を見ると 10KB のメタデータが 100 個返って来るまでには 44 秒程度かかってしまう。これは実装したシステムのメタデータ（XML）の処理に DOM[8]を使い、１つのメタデータを読み出すたびにファイルに変換してから処理をしている為、サーバー側での処理時間が増えたことが予想される。実装方法を改善する必要がある。

クライアントからの要求メッセージの中にmaxPrograms 属性値（オプション機能）を入れることにより、返って来るメタデータの数の最大値を指定出来る仕様になっているので、応答時間が長くクライアント側でタイムアウトしてしまう場合などは、受け取るメタデータの最大値を指定する必要がある。

7. まとめ

本稿では TV-Anytime Forum にて標準化が進められている IP ネットワーク上でのメタデータサービスの実装を行った。

今回は最低限の機能のみでプロトタイプ実装を行ったが、本システムでもクライアントは欲しいメタデータをサーバから受け取れることが確認出来た。

またクライアントは AND や OR を使って複雑な要求メッセージを送ることが可能であるが、応答メッセージを受けた時に問題が起きることがあることが分かった。それは例えば図 8 のように 2 つのタイトル（Titanic と Startwars）の批評情報メタデータを要求した場合に２つのメタデータが返って来るが、その２つのメタデータがどちらのメタデータなのかが区別出来ない場合があることだ。

```
<get_Data xmlns='http://www.tv-anytime.org/2002/11/transport'
  xmlns:tvaf='http://www.tv-anytime.org/2002/11/transport/fieldIDs'>
  <QueryConstraints type='OR' >
    <Predicate fieldID='tvaf:Title' fieldValue='Titanic'/>
    <Predicate fieldID='tvaf:Title' fieldValue='Starwars'/>
  </QueryConstraints>
  <RequestedTables>
     <Table type='ProgramReview'/>
  </RequestedTables>
</get_Data>
```

図 8 Get_Data Service メッセージ

応答メッセージのスキーマ（TV-Anytime メタデータサービス仕様）に要求メッセージの属性値（fieldID 値と fieldValue 値）を入れるなどの若干の修正が必要である。

8. 今後の課題

今回はサーバ側の SOAP アプリケーションの実行環境として Apache SOAP2.3.1 を使用したが、その他にも SOAP 通信機能をもったソフトウェアや API は多数ある（ApacheAXIS、JAXR、GLUE 等）ので、様々なツールで実装した TV-Anytime メタデータサービスとクライアント間のインターオペラビリティを試験する必要がある。今後も TV-Anytime メタデータサービスの実装を続けていく予定である。

また今回実装した Get_Data Service と Describe_Get_Data Service は共に Request-Response 型であり、クライアントからサーバや別のクライアントに対して自分が持つメタデータを送ることが出来ない。そして TV-Anytime メタデータサービス仕様書には Get_Data Service と Describe_Get_Data Service 以外に Submit_Data Service があり、これはクライアントからサーバ等にメタデータを送ることが可能だが、送れるメタデータはユーザ・セントリックな情報（嗜好情報、視聴履歴等）に限られている。

今後のメタデータ流通の発展を考えると、ユーザ間での自由なメタデータ交換環境を実現する必要があると思う。それには Request-Response 型-の API だけではなくて、クライアントが他のクライアントに対してメタデータを送信可能な API も必要である。

ただしそのようなメタデータ流通においては不正なメタデータの送信をプロテクションすることや、送られてきたメタデータの信頼性、さらにメタデータの権利管理及び保護なども考慮しなければいけない。今後の課題として検討していく。

参考文献

[1]TV-Anytime Specification on Metadata:SP003v1.1,TV-Anytime Forum, http://www.tv-anytime.org

[2]TV-Anytime Specification on Metadata Services:SP006v1.0,TV-Anytime Forum, http://www.tv-anytime.org

[3]Universal Description Discovery & Integration,Version 3.0 T.Bellwood,et.al. http://uddi.org/pubs/uddi-v3.00-published-20020719.htm

[4]Web Services Inspection Language Version 1.0 K. Ballinger, P. Brittenham, A. Malhotra, W. A. Nagy, S. Pharies http://www.ibm.com/developerworks/webservices/library/ws-wsilspec.html

[5]Simple Object Access Protocol (SOAP) 1.1, W3C Note, 8 May 2002 D. Box, et. al. http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[6]Web Services Description Language, Version 1.1, W3C Note 15 March 2001 E. Christensen, F. Curbera, G. Meredith, S. Weerawarana http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[7]http://www.tv-anytime.org/wsdl/tva_transport_v10.wsdl#get_Data_SOAP, TV-Anytime Forum

[8]Document Object Model, http://www.w3.org/DOM/