

フラグメントメタデータ流通における フラグメント発見とフラグメント更新手法に関する研究

渡邊雅義 亀山渉

早稲田大学大学院国際情報通信研究科

1. はじめに

コンテンツ流通を促進させるためには消費者にそのありかを効果的に知らせる必要があり、その手段としてメタデータが注目されている。すなわち、有効なメタデータを多数発見し、それらを取得する環境を整備することが必要不可欠である。そこで我々は同一コンテンツを対象にしたメタデータをフラグメント化し、それを多様なノード間で交換できる方式を提案している[1]。

本報告では[1]において課題であったフラグメント発見と更新手法に主眼を置いて検討した結果を報告する。

2. フラグメントメタデータ交換手順

フラグメントメタデータ交換手順の概要を図1に示す。まず Step1 でフラグメントメタデータ（以下 FMD）を交換できる端末を発見する。次に Step2 でそのサービスの詳細を確認する。最後に Step3 において確認した内容をもとに要求を作成し、FMD の交換を行う。

[1]において Step3 を FMD 交換サービスとして提案しているが、本報告では Step1 で FMD の Discovery サービスを、Step2 で Describe サービスを、Step3 で更新を可能にするため FMD 交換サービスに変更を加えたものを新たに追加した。

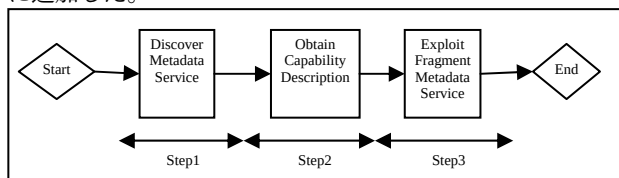


図1 FMD 交換手順概要

3. 提案サービスについて

3.1 Discovery サービス

Discovery サービスとは、ある端末が、既知の端末を通して新たな端末を発見するサービスである。端末Aが、既知である端末Cを通してCが知っている端末であるD・Eを発見する手順を例にとり、図2を用いて説明する。ここでは、A・B・Cがお互いに既知で、C・D・Eがお互いに既知と仮定する。AがCに対してDiscoveryサービスを行う（図2①）とすると、その結果、Cは自分の情報と合わせてD・Eの情報もAに返す（図2②）。これによって、Aは、既知の端末Cを通して新しい端末D・Eを発見できる。

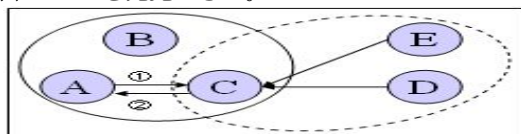


図2 未知の端末の発見例

Discovering and Updating Fragment-Metadata on Fragment-Metadata Distribution Environment
GITS, Waseda University

3.2 Describe サービス

Describe サービスとは端末の状態の詳細を確認するためのサービスである。クライアントが FMD を取得する場合、サーバの対応していないパラメータで要求を出すと FMD が取得できない場合がある。無駄な要求を省くため、相手のサービスの名前、内容、どのような要求に対応しているかなど、サービスの詳細を知るために describe_get_Fragment サービスを用いる。

一方、クライアントが相手に FMD を送りつける場合、FMD を受け取る側は余計な FMD を受け取らないために特定のクライアントのみから FMD を受け取る設定にしている場合がある。このような場合、無駄な FMD の送信を防ぐため、相手が FMD を受け取るかどうかを調べるためのサービスとして describe_send_Fragment を用いる。

3.3 get_Fragment サービス

更新された FMD の取得を可能とするため、[1]で定義した仕様を変更した。[1]において FMD を特定するために ID 情報や Version 情報が規定されているが、その Version 情報と、図3の下線部の fragmentVersion の値とを下線部 test の値に従って比較することによって、より新しく更新された FMD の取得や、その FMD より古い Version の FMD の取得も可能となる。

<pre> <get_Fragment> <Fragment ParentID="http://A.jp/Original.xml#xpointer()"/> </get_Fragment> </pre>	文献[1]の仕様
<pre> <get_Fragment> <Fragment ParentID="http://A.jp/Original.xml#xpointer()"/> <VersionInfo fragmentVersion="20031217180000" test="greater"/> </Fragment> </get_Fragment> </pre>	本報告での仕様

図3 get_Fragment 例

4. フラグメントメタデータ交換手順詳細

FMD サービスとして FMD 取得や FMD 送信などがあるが、本節では更新された FMD 取得を例にとって説明する。また以下、FMD の交換メッセージはすべて SOAP エンベロープに包んで SOAP/HTTP で受け渡しが行われるものとする。

4.1 Discovery サービス例

クライアントがある端末におけるサービスを発見したい場合は、既知の端末に対して要求を送信する Discovery サービスのレスポンスによって、サービスを行える新たな端末が発見できる。図4はこのやり取りの SOAP エンベロープを省略した具体的な記述例である。

4.2 Describe, get_Fragment サービス例

クライアントが FMD を更新する場合、まず相手のサービス詳細を得るため要求（図5①）を送信する。Describe サービスによって返されたレスポンス（図5

②)を参照し、FMDを更新するための要求(図5③)を出す。その結果フラグメントサービスにより更新されたFMD(図5④)が返ってくる。図6は図5の①②、図7は図5の③④のやり取りのSOAPエンベロープを省略した具体的な記述例である。

以上をTV-Anytime Forum 準拠のメタデータ[3]を使い実装し、動作を確認した。

<discovery/>	Request Message
<pre><discovery_Result> <Person persontype="friend"> <PersonInfo> <name>A</name> <sex>female</sex> <interest>fashion</interest> </PersonInfo> <ServiceInfo serviceName="send_Fragment"> <serviceURL>http://www.A.jp/FragmentService.wsdl</serviceURL> </ServiceInfo> </Person></pre>	
	Response Message

図4 Discovery サービスメッセージ例

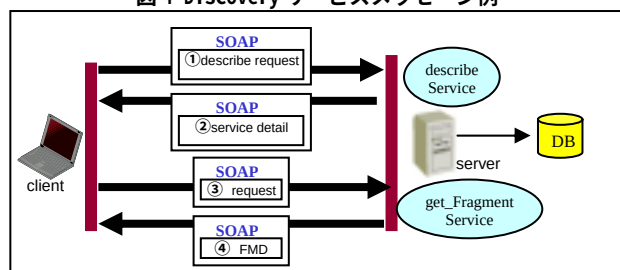


図5 Describe,get_Fragment サービス手順例

<describe_get_Fragment/>	Request Message
<pre><describe_get_Fragment_Result> <Name>Fragment Metadata Service</Name> <Description>you can get metadata</Description> <AvailableValue ParentID="true" filename="true" access="true" table="true" version="true"/> </describe_get_Fragment_Result></pre>	
	Response Message

図6 describe_get_Fragment メッセージ例

<get_Fragment/>	Request Message
<pre><get_Fragment> <Fragment> <VersionInfo fragmentVersion="20031217180000" test="greater"/> </Fragment> </get_Fragment></pre>	
	Response Message
<pre><get_Fragment_Result> <Fragment ParentID=" " http://www.A.jp/OriginalA.xml#xpointer("/") fragmentVersion="20031231000000" table="ProgramLocationTable" access="many" interval="short"> <ProgramLocationTable> <Schedule serviceIDRef="1"> <ScheduleEvent> <Program crid="crid://www.waseda.ac.jp/WCSoccer"/> </ScheduleEvent> </Schedule> </ProgramLocationTable> </Fragment> </get_Fragment_Result></pre>	

図7 FMD 更新メッセージ例

5. FMD の更新に関する考察

5.1 問題点

以上よりサーバ側で更新された FMD をクライアントが取得できるようになったが、クライアントはサーバでいつ FMD が更新されたか分からない。そのためクライアントは FMD の更新を調べるためサーバへ定期的なアクセスが必要となる。この問題の解決手段として、該当の FMD

へ、以下の表1に示すアクセス頻度と更新頻度のヒント情報を FMD に加え、クライアントがサーバの更新を推定できるようにした。また FMD の種類として番組内容情報、配信情報、批評情報などがあるが、これらも情報の性質により更新されるタイミングが違うので、クライアントが判断する材料と考えられる。このことにより、更新を調べるための無駄なアクセスを減らすことができると考えられる。以上のことを確かめるため、シミュレーションを行った。

表1 FMD に加えたヒント情報

ヒント情報	Access(アクセス頻度)	Interval(更新頻度)
とりうる値	small,middle,many	short,normal,long

5.2 シミュレーション実験

サーバは1タイムスロットごと、該当 FMD の更新確率関数(1)に従って更新を実行する。関数(1)は文献[3]を参考に設定した。定数 a,b は文献[3]の最大値とその位置を参考に a=2 b=1, a=0.5 b=0.3 の2通りにおいてシミュレーションを行った。

$$Y = aX \times e^{bX} \dots (1)$$

(Y:更新確率 X:タイムスロット a,b:定数)

クライアントは取得した FMD に書きこまれているヒント情報を使い、サーバのメタデータ更新を予測する関数を作成する。本シミュレーションでは10タイムスロット分のデータを用いて更新予測関数を作成し、サーバの更新を予測し適合率を調べた。

クライアントが推定で使えるヒント情報の組み合わせは27通りとする。クライアント、サーバは共にこの27通りのMetadataを所有する。クライアントはデータ解析手法として数量化II類[4]を使い更新の評価関数を作成する。

5.3 シミュレーション結果

サーバにおける FMD の更新をクライアントで推定した場合の適合率を表2に示す。その結果より、FMD にヒント情報を加えることによって、ある程度の確率でクライアントがサーバの更新を判断できることが分かった。

表2 シミュレーション結果

定数	a=2 b=1	a=0.5 b=0.3
適合率	85.2%	77.8%

6. まとめ

本稿ではメタデータをユーザ間で自由かつ柔軟に流通可能とするため、課題であったサービス発見と更新手法に関する提案をし、それを実現するための実装をした。

今後はメタデータにアプリケーション等をリンク付けし、自動的に流通させる方式を検討する予定である。

参考文献

- [1]鈴木順一, 亀山渉, “メタデータ流通を促進するフラグメントメタデータ交換システム”, 情報科学技術フォーラム 2003,D-029(2003年8月)
- [2]TV-Anytime Forum, <http://www.tv-anytime.org>
- [3]山田宰, “放送から見たユビキタスネットワーク”, ユビキタスネットワーク技術の将来展望に関する調査研究会, 資料1-7,p14(2001年11月) <http://www.soumu.go.jp/joho-tsusin/policyreports/chousa/yubikitasu/>
- [4]菅民朗, “多変量解析の実践”, ISBN:4768702120(1993年3月)