

コンテンツ統合言語としての XQuery の再考

湯本 高行[†] 田中 克己[†]

Web コンテンツや蓄積型テレビに蓄積されるビデオコンテンツなど、膨大かつ多様な XML コンテンツが利用できるようになってきている。閲覧効率の向上やコンテンツの理解支援などを目的として、これらの多様なメディアのコンテンツを統合して利活用することが考えられる。従来、コンテンツ統合は、SMIL などを用いて手動でコンテンツの同期化記述を行うことで統合したり、特定の用途に応じたアプリケーションとして実現されていた。これに対して、XML 質問言語を用いて統合に必要なコンテンツを検索しこれを再構成することでコンテンツ統合を行うことが考えられる。本稿では、コンテンツ統合言語としての XQuery の再考を行い、コンテンツ統合にどのような機能が必要となるかについて考察する。具体的には、XQuery における文書解析を行うための関数の不備、ポリモルフィズムへの非対応、XML データ操作機能の不備などの問題が考えられるが、これらはそれぞれ XQuery の標準関数の拡張、ポリモルフィズムの導入の対応、他の言語との併用によって克服できると考えられる。

XQuery Reconsidered As a Content Integration Language

TAKAYUKI YUMOTO,[†] and KATSUMI TANAKA[†]

Recently, a vast volume of XML contents of several media types are available such as Web contents and video contents both of them can be stored into personal video recorder. We consider to integrate these different-media contents into a compound content to access effectively and to make these contents more comprehensive. Conventionally, the content integration have been achieved by hand-writing SMIL data or by adhoc application systems. We propose a way to achieve the content integration by using XML query language, which enables us to retrieve necessary assets for integration and to compose them. In this paper, we investigate the functions of XQuery as a content integration language. We consider that XQuery needs functions for document analysis, polymorphism, and XML data manipulation as a content integration language.

1. はじめに

家庭へのブロードバンドネットワークやハードディスクレコーダーの普及により、Web コンテンツ、ビデオコンテンツ、音楽コンテンツなど多種多様なコンテンツが簡単に入手できるようになっている。しかし、入手できるコンテンツの数が膨大になっているにもかかわらず、コンテンツは1つ1つクリックし、閲覧するというスタイルが主流であり、自由な見方ができない。例えば、自分がいつも閲覧するニュースサイト、定期的に新着記事を取得している RSS サイト、ハードディスクレコーダーに記録されている1週間分のニュース番組があったとき、これらの内容を見ようとすると、記事を1つ1つクリックし、ビデオを(ときには早送りをしながら)1つ1つ見ていかなくてはならな

い。各コンテンツには重複する内容もあり、自分が興味があることだとしてもかなり面倒である。また、さまざまなメディアに関連した情報が分散しており、わかりづらい。このような状況は、「内容的に重複するコンテンツをひとつのコンテンツに統合する」だけでもかなり改善する。また、逆にひとつのニュースをさまざまな見方をしたい場合、同じニュースに関する記事を集めてきて、それらを並べて表示すればよい¹⁾。これらはどちらもコンテンツ統合と呼ばれるものであるが、これらを実現するためには、SMIL²⁾ などを使って、手動でコンテンツの同期化記述を行うか、特別なソフトウェアとして実現するしかないが、コンテンツをまとめて見たい(コンテンツを統合したい)という欲求は一般ユーザが普通に持っている欲求であり、もっと一般ユーザでも簡単にコンテンツ統合を実現できるべきである。

本稿では、コンテンツ統合言語としての XQuery³⁾ の再考を行い、コンテンツ統合にどのような機能が必要

[†] 京都大学大学院 情報学専攻 社会情報学専攻
Department of Social Informatics,
Graduate School of Informatics, Kyoto University

要となるかについて考察する。

以下に論文の構成を示す。2節ではコンテンツ統合の概要について述べ、どのようなコンテンツ統合手段が必要とされているかに述べる。3節ではコンテンツ統合を XQuery を用いて実現する方法について述べる。4節では XQuery を用いてコンテンツ統合を実現した場合の問題点と解決策について考察する。5節ではまとめと今後の課題について述べる。

2. コンテンツ統合

2.1 動 機

コンテンツ統合は、複数のコンテンツから新たなコンテンツを生成することである。コンテンツの統合の目的はさまざまであるが、例えば以下の3つが考えられる。

(1) コンテンツの閲覧効率の向上

複数のコンテンツに分散していた内容をひとつの統合コンテンツにまとめることによって、ユーザはより簡単に情報にアクセスできる。さらに、重複した内容を集約することによって、情報の冗長性を下げることが可能である。

例:複数 RSS⁵⁾ の集約

複数のニュースサイトから配信される RSS の中から同じ出来事を伝えるものはひとつにまとめ、ひとつの RSS に統合し、見せる。

(2) コンテンツのわかりやすさの向上

テキストだけのコンテンツよりもビデオのコンテンツも合わせて閲覧した方がわかりやすい内容もある。このように、異種のメディアの表現能力を利用することによって、コンテンツのわかりやすさを向上させることに可能となる。

例:画像のビデオによる置きかえ

ニュースサイトのスポーツの試合結果を伝える記事中の画像を試合の映像で置きかえる。

(3) コンテンツ理解の促進

ニュースの場合においては、Web 上では同じ出来事に関してさまざまな伝え方や意見がある。これらを統合して見せることによって、ユーザは単に列挙された情報をのみを受け取るのではなく、さまざまな見方があることを理解し、各情報間のつながりを意識しながら、情報を受け取る。これによって、ユーザのコンテンツに対する理解を促進される。

例:関連情報の同時呈示

複数のサイトから同じ出来事に対するニュース記事、ニュース映像、blog などによる意見を集

めて、多角的に出来事をとらえる。

このように、コンテンツ統合は、異なるメディアの表現能力の違いやコンテンツごとの多様な視点を利用しながら、メディアリッチなコンテンツを生成し、コンテンツの閲覧効率およびわかりやすさ、理解度を向上させるものである。

これらを多くのメディアにおいて実現するならば、XML ベースのコンテンツおよび XML ベースのメタデータが付加されたコンテンツを対象にすることが得策であると考えられる。XML は XHTML⁴⁾ や SMIL²⁾ などのコンテンツ記述に使われると共に、RDF⁶⁾ や MPEG-7⁷⁾、MPEG-21⁸⁾ などメタデータ記述にも使われる。例えば、ニュース記事なら NewsML、スポーツニュース記述 SportsML、ラジオ・TV 番組案内 ProgramGuideML、イベント記述 EventsML などが IPTC (国際新聞電気通信評議会) によって規定され利用されている^{9),10)}。MPEG-7 は XML スキーマに準拠している XML ドキュメントで、マルチメディアのメタデータの記述方法を定める規格であり、MPEG-21 はさらにデジタル権利処理にも対応している。TV Anytime は放送、インターネットなどさまざまな方法で配信される番組 (コンテンツ) を大容量のサーバに蓄積し、利用者がいつでも視聴できる放送システムである。TV Anytime Forum で検討されている放送用のメタデータでは、XML スキーマを使っている¹¹⁾。

このように XML は非常に広汎な用途で利用されているため、以下では XML コンテンツおよびメタデータを対象としたコンテンツ統合について述べる。

2.2 コンテンツ統合の実現手法

統合に利用可能な既存の技術および統合を実現する研究について考察する。統合には、静的な統合と動的な統合がある。前者は統合に用いる素材コンテンツをどのように配置するか空間的レイアウト、時間的レイアウトをひとつひとつ記述していく方式である。このような方式の言語としては、HTML⁴⁾、SMIL²⁾ がある。これらの言語はコンテンツ記述言語としてとらえることが一般的であるが、現在完全に単一のメディアからなるコンテンツしか記述できないということはほとんどなく、コンテンツ記述言語のほとんどは静的なコンテンツ統合が可能であると言うこともできる。この静的なコンテンツ統合の問題点としては、他のコンテンツを使って、新しい統合コンテンツをつくる場合であっても、普通のコンテンツをつくる場合と同じ労力がかかる。

この問題を解決するのが動的なコンテンツ統合であ

る。動的なコンテンツ統合は、与えられたコンテンツ集合(ときにはクエリに基づいてコンテンツ集合の取得も行う場合もある)に対して、一定のルールに従って、それらから統合コンテンツを生成するというものである。これらの多くは目的毎のアプリケーションとして実現されており、それぞれの目的において優れた成果をあげている^{12)~14)}。しかし、コンテンツを統合して見るという行為は人それぞれであり、さまざまな統合方式が存在してよいはずである。これを実現するためには、統合のテンプレートが必要である。OssenbrugenらはXMLコンテンツからマルチメディアコンテンツを生成するスタイルシートを提案している¹⁵⁾。これによって、従来のスタイルシートのようにコンテンツの空間的なレイアウトを変化させるだけではなく、時間的なレイアウトを変化させることも可能である。筆者らが提案しているMetaSMILはコンテンツ記述言語に検索機能を付加することによって、コンテンツ統合のテンプレートを実現するものである¹⁶⁾。また、一般的な言語によって、統合のテンプレートが実現できる可能性もある。XSLT¹⁷⁾はXML文書のフォーマットの変換を行う簡易なスクリプト言語である。XSLTを用いて、さまざまなフォーマットに変換できるが、XQueryで利用できるような入れ子状の質問などには対応しておらず、単純な統合の要求にしか答えられない。それに対して、XQuery³⁾はコンテンツ統合に関して高い記述能力を持っている。XQueryはXML文書のための検索言語であり、入れ子質問や変数の利用などが可能である。さらに注目すべき特徴として、解の出力の形式の高い記述能力があげられる。XQueryでは、解を出力するreturn節において、出力する解の形式を柔軟に変更することが可能である。

2.3 アプローチ

前述したようにコンテンツ統合の目的や用途はさまざまである。そのため、その目的や用途に合わせたコンテンツ統合を実現するテンプレートが必要である。コンテンツ統合は以下のような動作の組み合わせによって実現されることが多い。

- (1) 統合に用いるコンテンツの選定
- (2) メディア形式の変換
- (3) 再構成

XQueryでは上記3つの操作を表現することが可能である。そのため、XQueryはコンテンツ統合のテンプレートを実現する有用な手段であると考えられるため、XQueryを用いてコンテンツ統合のテンプレートを実現する方法について考察する。

3. XQueryによるコンテンツ統合

XQueryでは、前節で述べた3つの操作を基本的には表現することができる。また、XQueryのFLWOR表現式による問い合わせはFLWORの5つの節からなっているが、各節の中でFLWOR表現式を用いることができるので、3つの操作を組み合わせ、統合を実現することができる。

3.1 統合に用いるコンテンツの選定

与えられたコンテンツ集合からある条件を満たすコンテンツを選定する。

例:キーワード”松井”を持つコンテンツ集合の抽出

```
for $x in (doc(file1),...,doc(filen))
for $y in $x/*/text()
where contains($y,"松井")
return $x
```

図1 コンテンツ選択の例

3.2 メディア形式の変換

コンテンツ統合では異なる形式のメディアを統合する場合がある。この場合、内容の統合だけではさまざまな言語のタグが混在し、一般的なブラウザやプレイヤーで解釈できない。そのため、タグを一般的なブラウザおよびプレイヤーで解釈できるように1種類の言語のタグに変換する必要がある。これは一般的なXML文書の統合と同種の問題である。さらにそれだけではなく、コンテンツ統合においては、テキストなどの非時系列コンテンツとビデオなどの時系列コンテンツを統合する場合がある。このような場合は、非時系列コンテンツを時系列コンテンツに変換したり、時系列コンテンツを非時系列コンテンツに変換した上で、統合を行う必要がある。

例:MPEG-7メタデータからSMILへの変換

MPEG-7のメタデータからシーン情報を得て、それをSMILに変換する。図2は変換を行うXQuery質問で、図3,4はそれぞれ変換前のMPEG-7のデータと変換後のSMILのデータである。

3.3 再構成

問い合わせの出力結果をタグによって構造化して出力する。

3.4 質問のネストの例

XHTMLで書かれたニュース記事とMPEG-7のメタデータが付加されたビデオを統合し、HTML+SMILコンテンツに変換する場合を考える。外側の質問で、

```

let $x := document(uri)
return
<smil>
<head>[略]</head>
<body>
<video>
{
element img {
attribute
src {$x//MediaLocator/text()},
attribute region {rg1},
attribute
begin {$x//MediaRelTimePoint/text()},
attribute
dur {$x//MediaDuration/text()}
}
}
</body>
</smil>

```

図2 MPEG-7 から SMIL への変換を行う XQuery 質問

```

<Mpeg7 type = "complete" ... >
<ContentDescription
xsi:type="ContentEntityType">
<MultimediaContent
xsi:type="AudioVisualType">
<AudioVisual id="id1">
<MediaLocator>uri</MediaLocator>
<MediaTime>
<MediaRelTimePoint>
t1</MediaRelTimePoint>
<MediaDuration>t2</MediaDuration>
</MediaTime>
[中略]
</AudioVisual>
</MultimediaContent>
</ContentDescription>
</Mpeg7>

```

図3 変換前の MPEG-7 データ

XHTML のコンテンツを取得し、<html>,</html> を出力し、内側の質問の1つ目では、XHTML コンテンツの外側のタグを外したものを出力、2つ目はキーワードに”松井”を持つシーンを MPEG-7 のメタデータから発見し、形式を SMIL に変換して出力している。図5にその XQuery での表記を示す。

```

<?xml version="1.0"?>
<smil>
<head>[略]</head>
<body>
<video src="uri" region="rg1"
begin="t1" dur="t2"/>
</body>
</smil>

```

図4 変換後の SMIL データ

```

let $x := doc(uri1)
return
<html>
{for $y in $x/* return $y}
{for $s
in doc(uri2)//TemporalDecomposition
for $t in $s//text()
where contains($t,"松井")
return
{
element smil:video {
attribute
src {$s//MediaLocator/text()},
attribute
begin {$s//MediaRelTimePoint/text()},
attribute
dur {$s//MediaDuration/text()}
}
}
}
</html>

```

図5 質問のネストの例

4. コンテンツ統合における XQuery の問題点

4.1 文書解析用の関数の不備

XHTML や SMIL など表示を重視して記述されるコンテンツでは、単に文字列マッチングなどでは統合に適しているコンテンツかどうかを判別できない。このため、文書解析にも利用可能な関数が必要である。しかし、XQuery の仕様¹⁸⁾では、そのような関数は用意されていない。特に日本語においては対応が遅れている。XQuery では、標準関数などを用いて、独自の関数をユーザが定義できるため、基本的な関数が用意されれば、この状況は大幅に改善すると思われる。

4.2 ポリモルフィズムへの非対応

異種のコンテンツおよびメタデータに対して統合を行う場合、コンテンツごとに処理を変える必要がある。例えば、“松井”に関するコンテンツを発見する場合、MPEG-7のメタデータであれば、MPEG-7のツリーをたどり、キーワードとして“松井”が付加されているかどうかを判定すればよいが、XHTMLのコンテンツの場合は多くの場合は、文書解析的な手法を使うしかない。このように処理が異なるので、統合のための問い合わせを書くときは、どのタイプのコンテンツが処理の対象になるかをいちいち判定して記述しなくてはならず、非常に面倒である。

別の例として、ビデオユニットの長さを返す問い合わせを以下に示す。

```
let $x:=doc(uri)
return
<duration>
  if (string(name($x/*))="smil")
  then string($x//video/@dur)
  else
    if (string(name($x/*))="Mpeg7")
    then $x//MediaDuration/text()
    else "unknown"
</duration>
```

図6 ポリモルフィズムを導入しない例

4.3 XMLデータの操作機能の不備

XQueryは問い合わせの結果を独自に構造化することはできるが、元のXML文書の構造を利用して構築することはできない。よりルートノードに近い部分に対しては、XQueryの問い合わせを記述するときあらかじめ記述しておかなくてはならない。図7はノードがタグを表している。この図のノードAを別のノードEに置き換える場合、サブツリー1、ノードBについては動的に取得して出力できるが、ノードC,Dに関してはすべてXQueryの質問のreturn節に記述しておかなくてはならない。

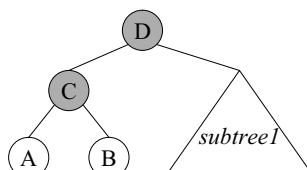


図7 データ操作の例

例えば、HTMLの画像の部分に映像を埋め込み、HTMLのレイアウトはそのまま使うというようにことがうまく表現できない。

SQLが問い合わせ言語でありながら、データ操作言語としての側面も持っていたのに対して、XQueryはデータ操作機能を持っていない点がネックになっている^{19),20)}。しかも現在、XMLのデータ操作言語には標準が存在していない。(XUpdate²¹⁾が有力であるという意見もあるが、W3Cなどによる標準化が行われているということは確認できなかった。)このような場合はXSLTなどの変換用の言語の方がむしろ処理に適している場合もある。図8,9はそれぞれ、置き替えるビデオがわかっている場合にHTMLコンテンツ中の画像をビデオに置き替えるための記述例を示す。

*

```
<xsl:stylesheet [略]>
  <xsl:template match="/*">
    <xsl:apply-templates/>
  </xsl:template>
  <xsl:templates match="//img">
    <smil:video
      src="uri" begin="t1" dur="t2"/>
    </xsl:template>
  <xsl:template match="*">
    <xsl:element name="name()">
      <xsl:apply-templates/>
    </xsl:element>
  </xsl:template>
</xsl:stylesheet>
```

図8 XSLT(一部拡張あり)による例

5. おわりに

本稿では、XQueryをコンテンツ統合言語として再考した。XQueryをコンテンツ統合言語として使う場合は、統合コンテンツのレイアウトの方針がある程度、決まってい、各部分にどの素材を当てはめるかを決めたい場合には有効であると考えられる。ただし、複数のメディアが混在している場合は、前処理として、メディアを分類してから、XQueryを用いるなどしないと記述が煩雑になる。

これに対して、欠点としては以下のようなものがある

* ただし、以下の例はXSLTの処理系のベンダが独自の仕様拡張を行ったことにより、動的にXML要素を生成が可能になっているために実現した。

```

<xupdate:modifications>
  <xupdate:insert-before select="//img">
    <xupdate:element name="smil:video">
      <xupdate:attribute name="src">
        uri</xupdate:attribute>
      <xupdate:attribute name="begin">
        t1</xupdate:attribute>
      <xupdate:attribute name="dur">
        t2</xupdate:attribute>
    </xupdate:element>
  </xupdate:insert-before>
  <xupdate:remove select="//img"/>
</xupdate:modifications>

```

図 9 XUpdate による例

げられる。

- (1) 文書解析を行うための関数の不備
- (2) ポリモルフィズムへの非対応
- (3) XML データ操作機能の不備

これらはそれぞれ XQuery の標準関数の拡張，ポリモルフィズムの導入，他の言語との併用によって克服できると考えられる。

また，Web ページの受動的視聴²²⁾，TV2Web²³⁾ などのコンテンツ変換も同種の技術で対応可能であり，さまざまな新たなコンテンツの利用法につながる技術であると言える。これによって，XQuery は問い合わせ言語として有用であるだけでなく，コンテンツ統合/変換言語としても利用できる可能性がある。

本稿では，コンテンツ統合言語というものを考察したが，コンテンツ統合は XML データの統合の特別なケースとして位置付けられる。このため，本稿で見つかったコンテンツ統合言語としての XQuery の問題は，XQuery を XML データの統合に用いるときの問題点とも言える。

今後の課題としては以下のようなものが考えられる。

- ポリモルフィズムに対応した XQuery の処理系の開発とその統合言語としての能力の検証
- 変換言語 (XSLT)，データ操作言語 (XUpdate) など他の言語を併用したコンテンツ統合手法の開発

謝辞 本研究の一部は，平成 16 年度科研費基盤研究 (A)(2)「モバイル環境におけるコンテンツのマルチモーダル検索・呈示と放送コンテンツ生成」(課題番号:14208036，代表:田中克己)及び 21 世紀 COE プログラム「知識社会基盤構築のための情報学拠点形成」によるものです。ここに記して謝意を表すものとします。

参 考 文 献

- 1) Akiyo Nadamoto, Katsumi Tanaka, “A Comparative Web Browser (CWB) for Browsing and Comparing Web Pages”, Proceeding of the 12th International World Wide Web Conference (WWW2003), Budapest, Hungary (2003).
- 2) W3C Synchronized Multimedia Home page, <http://www.w3.org/AudioVideo/> (2004).
- 3) W3C XML Query (XQuery), <http://www.w3.org/XML/Query/> (2004).
- 4) W3C HTML Home Page, <http://www.w3.org/Markup/> (2004).
- 5) The Web KANZAKI, “RSS – サイト情報の要約と公開”, <http://www.kanzaki.com/docs/sw/rss.html> (2004).
- 6) W3C, Resource Description Framework (RDF), <http://www.w3.org/RDF/> (2004).
- 7) MPEG-7 Overview (version 9), <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm> (2003).
- 8) MPEG-21 Overview v.5, <http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm> (2002).
- 9) IPTC/NewsML Web, <http://www.newsml.org/pages/index.php> (2004).
- 10) 社団法人 日本新聞協会, newsml.jp, <http://www.newsml.jp/> (2004).
- 11) TV-Anytime Forum, <http://www.tv-anytime.org/> (2004).
- 12) 寺田 努, 塚本昌彦, 西尾章治郎, “アクティブデータベースを用いたカラオケの背景作成システム”, 情報処理学会論文誌, Vol. 44, No. 2, pp. 235–244 (2003).
- 13) Qiang Ma and Katsumi Tanaka, “WebTelop: Dynamic tv-content augmentation by using web pages”, Proceedings of IEEE International Conference on Multimedia and Expo (ICME2003) (II), pp.173–176 (2003).
- 14) Columbia Newsblaster, <http://www1.cs.columbia.edu/nlp/newsblaster/> (2004).
- 15) Jacco van Ossenbruggen, Joost Geurts, Lynda Hardman, Lloyd Rutledge, “Towards a Multimedia Formatting Vocabulary for Time-Based Hypermedia”, Proceedings of The Twelfth International World Wide Web Conference (WWW2003), Budapest, Hungary, pp. 384 – 393 (2003).
- 16) Takayuki Yumoto, Qiang Ma, Kazutoshi Sumiya and Katsumi Tanaka, “A Dynamic Content Integration Language for Video Data

- and Web Content”, Proceedings of the 4th International Conference on Web Information Systems Engineering (WISE2003), pp.83–92, Roma, Italy (2003).
- 17) XSL Transformations (XSLT), <http://www.w3.org/TR/xslt> (1999).
 - 18) W3C, XQuery 1.0 and XPath 2.0 Functions and Operators, <http://www.w3.org/TR/xpath-functions/> (2003).
 - 19) 芝野 耕司, “SQL の 20 年と現状および今後の展開 (前編)”, 情報処理, Vol.45, No.5, pp.518–521 (2004).
 - 20) 土田 正士, 小寺 孝, 芝野 耕司, “SQL の 20 年と現状および今後の展開 (後編)”, 情報処理, Vol.45, No.6, pp.624–630 (2004).
 - 21) XML:DB Initiative, “XUpdate - XML Update Language”, <http://www.xmldb.org/xupdate/index.html> (2003).
 - 22) 灘本明代, 服部多栄子, 近藤宏行, 沢中郁夫, 田中克己, “Web コンテンツの受動的視聴のための自動変換とスクリプト作成マークアップ言語”, 情報処理学会論文誌: データベース, Vol.42, No.SIG1 (TOD8), pp.103–116 (2001).
 - 23) Kazutoshi Sumiya, Mahendren Munisamy, Katsumi Tanaka, “TV2Web: Generating and Browsing Web with Multiple LOD from Video Streams and their Metadata”, Proceedings of The 13th International World Wide Web Conference (WWW2004), New York City, U.S. (2004).