構造化文書情報源に対するラッパーの構築手法

2Q－2

古舘丈裕　石川佳治　植村俊亮
奈良先端科学技術大学院大学 情報科学研究科

1. はじめに

電子化された文書をネットワークを通して入手できる機会が多くなるにつれて，それらを意味的に統合し共有したいという要求が高まってきている．外部の情報源から得られる様々な形式の情報を統合的に扱えるシステムの研究が盛んに行われており，その中のアプローチの一つとしてメディエータ (mediator) とラッパー (wrapper) による環境が提案されている[1)2)]．

本稿では，構造化文書を取り扱う情報検索システムを対象として，メディエータへのインタフェースを与えるラッパーを構築する手法，特に問合せ変換について述べる．

2. 情報統合のためのアーキテクチャ

メディエータは複数の情報源から得られた情報を意味的に統合する役割を果たす．情報を統合するためには共通のデータモデルが不可欠であり，ここではオブジェクトデータベース標準 ODMG-93[3)]のオブジェクトモデルが採用されていると想定する．

ユーザから発行された問合せはメディエータに送られ，メディエータは適切な情報源を選択し，各情報源に問合せを送る．ここでの問合せは OQL で表現される．ラッパーは各情報源ごとに存在し，OQL で表された問合せを情報源固有の検索コマンドに変換する．また検索結果をオブジェクトモデルに適合するように変型してメディエータに返す．

3. ラッパー内部における問合せ変換

ここでは，情報源のサンプルとメディエータ内に構築されるオブジェクトの例を挙げ，両者の間での問合せの変換について述べる．

3.1 情報源とオブジェクトの例

ここでは例として音楽用 CD の紹介記事が SGML 文書として格納されている情報源を考える．紹介記事の DTD の例を図 1に示す．これに基づいて文書の特定の領域を指し示すのに経路式を用いる．例えば CD のタイトルの部分ならば report.disc.title となる．

情報源はこの DTD に基づく文書インスタンスを多数格納し，title と musician についての索引を構築しているものとする．検索コマンドとしては，図2のような仕様が与えられているとする．例えば search 'queens' and 'ogre' というコマンドでは二つの語を文書中のどこかに含むような文書インスタンスが返される．search -t 'ogre' なら，文書中の CD のタイトルの領域に 'ogre' が含まれるような文書インスタンスを返す．

一方，メディエータ側では，図3のような型が定義されているとする．ユーザ／アプリケーションはこれらの型の情報を元にして問合せを記述する．

```
<!ELEMENT report    - - (disc+)>
<!ELEMENT disc      - - (title,musician,body)>
<!ELEMENT title     - O (#PCDATA)>
<!ELEMENT musician  - O (#PCDATA)>
<!ELEMENT body      - O (tune+, dscr)>
<!ELEMENT tune      - - (title,words,music)>
<!ELEMENT words     - O (#PCDATA)>
<!ELEMENT music     - O (#PCDATA)>
<!ELEMENT dscr      - O (#PCDATA)>
```

図 1 サンプル DTD

```
%search [opt] string [and [opt] string..]
  option: t -- report.disc.title が対象
          m -- report.disc.musician が対象
  既定値では，文書全体が検索の対象
```

図 2 検索コマンドの構文

```
interface MusicCD
{
 attribute     String title;
 attribute     String dscr;
 relationship Musician musician
   inverse Musician::works;
 relationship List<Tune> tunes
   inverse Number::included_in;
}

interface Musician
{
 attribute     String name;
 relationship List<MusicCD> works
   inverse MusicCD::musician;
}

interface Tune
{
 attribute     String title;
 attribute     String words_by;
 attribute     String music_by;
 relationship List<MusicCD> included_in
   inverse MusicCD::tunes;
}
```

図 3 インタフェース定義

3.2 問合せ変換の方針

情報源の検索機能は様々であり，表現力の高い OQL で記述された問合せ要求を満たすことができるとは限らない．例えば，問合せ中に「12 曲以上収録されている CD」という条件が含まれている場合，例の情報源

A Wrapper Generating Method for Structured Document Sources
Takehiro Furudate, Yoshiharu Ishikawa, Shunsuke Uemura
Graduate School of Information Science,
Nara Institute of Science and Technology (NAIST)

ではそのような機能はサポートされていないので検索できない.

この場合は，与えられた問合せに含まれる検索条件のうち，対象の情報源でサポートされている条件式だけを情報源の検索システムに実行させ，その結果に対して残っている条件を適用してフィルタリングを行う．また，情報検索システムで直接はサポートされていないが，元の問合せの条件を包摂する検索がサポートされている場合もある．その場合は元の条件を，実行可能な条件に置き換えて検索する．その結果にはミスヒットも含まれているが，元の条件をフィルタとして適用して最終的な結果を得る[4]．

与えられた問合せ中に情報源でサポートできる検索条件が無い場合は，その問合せは拒否される．

3.3 問合せ変換の例

図 3で定義されたクラス群に対して，「'BEATLES' の CD で，'Yesterday' を収録している CD のタイトルを求めよ」という問合せは OQL では次のように表される．

```
select d.title
form   MusicCD d
where  d.musician.name = 'BEATLES'
and    d.tunes.title = 'Yesterday'
```

ラッパーはこの OQL 文を図 2の構文を持つ検索コマンドに変換する．そのためには，オブジェクトおよびその属性と，文書中の項目とのマッピング情報が必要となるが，それはラッパー生成時にあらかじめ記述しておく[5]．マッピング情報は例えば，次の様な対応関係を保存する．

オブジェクトの属性	文書中の領域
MusicCD::title	report.disc.title
Musician::name	report.disc.musician
Tune::title	report.disc.tune.title

ラッパーは与えられた OQL 文を解析し，where 句から and や or で結ばれた条件節を順に取り出し，それぞれ情報源で検索可能かどうか調べていく．上の問合せ例では次の二つの条件が含まれている．

1. d.musician name = 'BEATLES'
2. d.tunes.title = 'Yesterday'

条件 1 の経路式からは検索の対象が Musician オブジェクトの name 属性であることが分かる．それに対応する構造化文書中の領域をマッピング情報から求める．ここでは report.disc.musician で表される領域が検索の対象となり，この中に 'BEATLES' という語を含むような文書インスタンスの集合が検索結果となる．例の情報検索システムではこの領域に対する検索機能がサポートされている．よって検索式 -m 'BEATLES' をそのまま与えれば良い．

条件 2 からも同様に，対応する構造化文書中の領域 report.disc.tune.title が得られるが，例の検索システムは検索範囲をこの領域に特定した機能は直接サポートしていない．しかし，文書の全領域を対象として検索する機能はサポートしているので，それを実行した結果に対して，report.disc.tune.title の領域に 'Yesterday' が含まれている文書だけを通すフィルタリング処理を行うことによって元の問合せの要求に答えることができる．

以上から，ラッパーは検索式 **search** -a 'BEATLES' **and** 'Yesterday' を生成し情報源に与える．またその検索結果に対してフィルタリングを行うスクリプトを生成する．

また OQL 問合せ中の select 句では，title だけを要求している．そのため，検索結果の文書集合から title に対応する領域のみを切り出す必要がある．ラッパーは，その結果から ODMG オブジェクト (ここではリテラル) の集合を生成し，メディエータに返す．

4. まとめ

構造化文書情報源向けのラッパーを構築するための手法の一部として，ラッパー内部における問合せ変換機構について簡単な例を用いて説明した．今後は，複雑な問合せに対応した変換アルゴリズム，およびマッピング情報からラッパー自体を生成するアルゴリズムについて検討する．

謝辞

日頃より有益な御指導・御討論を頂く，植村研究室の皆様に感謝致します．

参 考 文 献

1) Y. Papakonstantinou, H. Garcia-Molina and J. Ullman. MedMaker: A Mediation System Based on Declarative Specifications. *Proc. of 12th Intl. Conf. on Data Engineering*, pp.132–141, New Orleans, Louisiana, Feb.–Mar.1996.

2) Y. Arens, R. Hull, and R. King. (eds.), Reference Architecture for the Intelligent Integration of Information, Version 2.0 (Draft). August 1995.

3) R. G. G. Cattell (ed.), *The Object Database Standard: ODMG-93, Release 1.2.* Morgan Kaufmann, 1996.

4) K. Chen-Chuan, H. Garcia-Molina, and A. Paepcke. Boolean Query Mapping Across Heterogeneous Information Sources. *IEEE Trans. on Knowledge and Data Engineering*, Vol.8, pp. 515–521, August 1996.

5) Y. Ishikawa, T. Furudate, S. Uemura. A Wrapping Architecture for IR Systems to Mediate External Structured Document Sources. *DASFAA'97*, Australia, April 1997. (to appear)