社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2007－ＤＤ－62（１）
2007／7／26

博物館情報を用いたメタデータスキーマ統合機構の実装と評価

秋元　良仁　　亀山　　渉

早稲田大学大学院国際情報通信研究科　〒367-0035 埼玉県本庄市西富田 1101
E-mail: ryoji@fuji.waseda.jp, wataru@waseda.jp

あらまし　膨大な情報から利用者にとって有益な情報を探し出すにはメタデータの利用が必要不可欠である．しかし，様々なメタデータスキーマが既に定義されているため，異なるスキーマで定義されたメタデータ間の情報交換は困難な状況にある．この解決策として，筆者らは異なるスキーマ間の関係性を記述できる言語 Fuzzy Schema を提案してきた．Fuzzy Schema はスキーマで定義される項目間の対応パターンとその類似性を記述できる言語である．本稿では，博物館情報を対象に，Fuzzy Schema を用いたメタデータスキーマ統合機構の実現方法について論じる．
キーワード　メタデータ，メタデータスキーマ，Fuzzy Schema，博物館情報

Implementation and Evaluation of Metadata Schema Integration Mechanism Based on Museum Information

Ryoji AKIMOTO and Wataru KAMEYAMA

Graduate School of Global Information and Telecommunication Studies, Waseda University　1011 Nishitomida, Honjo-shi, Saitama 367-0035 Japan
E-mail: ryoji@fuji.waseda.jp, wataru@waseda.jp

Abstract　A technology is necessary to extract useful information according to requirements from various information by using metadata. However, because various metadata schema have been defined, it is difficult to exchange different metadata. For this reason, we propose the concept that is called Fuzzy schema. Fuzzy Schema is a language based on XML that can describe the correspondence pattern and the ambiguity between items defined by metadata schema. In this paper, we describe a method of implementing metadata schema integration system based on museum information.
Key words　Metadata, Metadata Schema, Fuzzy Schema, Museum Information

1. はじめに

情報技術の急速な普及に伴い，人類によって創出される情報量は爆発的に増加している．また，これらの情報にアクセスできるユビキタスな情報環境基盤も整いつつあり，博物館においては，資料の収集，保存，整理，公開，展示等を情報技術を用いて積極的に行う取り組みが行われつつある．

このような状況において，膨大な情報の中からユーザにとって有益な情報を探し出すには，メタデータ [1] を用いて情報を整理する技術が必要となる．従来よりメタデータはその重要性が認知されており，様々なメタデータの構築，蓄積，利用が行われている．しかしながら，メタデータは予め定義されたメタデータスキーマに基づいて作成されるため，異なるメタデータスキーマに基づくメタデータ間の情報交換は困難な状況にある．

これまでにも，その解決方法の１つとして，語彙の意味を正確に記述できるスキーマと豊富な語彙を用意し，国際標準として確立する方式が提案されている [2]．しかしながら，巨大なスキーマと膨大な語彙は記述者による記述の差異を許容するため，情報検索等相互利用した際に記述の異なりが原因で精度が落ちる可能性がある．また，そもそも人間の知識表現は環境や状況，経験によって作られるものであり，固定的なスキーマで許容できるものであるとは考えづらい．

そこで，筆者らは個別にスキーマが存在することを認めた上で，そのスキーマ間の関係性が記述できる言語 Fuzzy Schema を提案している [3]．Fuzzy Schema は各スキーマで定義される項目間の関係性とそのあいまいさ (各項目にどの程度類似性があるか) が記述できる XML 形式の言語である．

本稿では，Fuzzy Schema を用いたメタデータ統合機構の試作について報告する．以下，２節でシステムの要件，３節でプロトタイプシステムの実現方法について述べる．４節で関連研

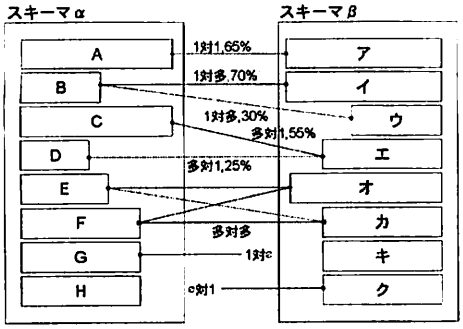

図 1　マッピング・パターンとあいまい度の例
Fig. 1　An example of Mapping Pattern/The degree of ambiguity

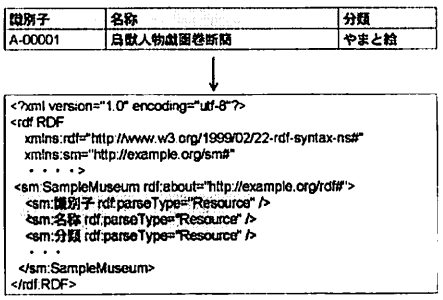

図 2　関係データベースからの変換
Fig. 2　Conversion from Relational database

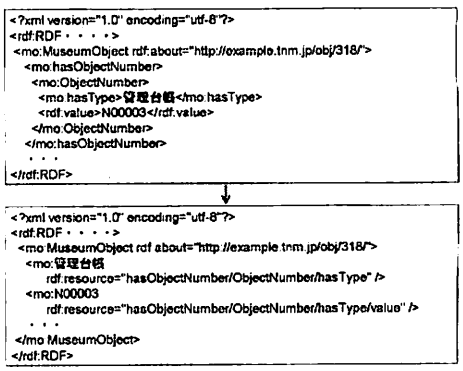

図 3　フラットな XML/RDF への変換
Fig. 3　Conversion into flat XML/RDF

究について述べ，最後に 5 節でまとめと今後の課題について述べる．

2.　システムの要件

2.1　Fuzzy Schema の概要

人間は経験や環境，状況を踏まえてコミュニケーションを図る．そこでは，コミュニケーションの対象となる相手が持つ知識表現を緩やかに理解し，都度自分の知識表現を作り変えることで円滑な知識の交換や蓄積が行われているものと考えられる．

著者らは，この発想を基に個別に存在するスキーマ間の関係性を記述できる言語 Fuzzy Schema を提案している．Fuzzy Schema はマッピング・パターンとあいまい度の 2 つの機構を有する．

マッピング・パターンとは，写像のパターンであり，各スキーマで定義された項目を比較する際の対応パターンを指す．具体的には，項目間の対応が 1 対 1 で対応付けられるもの，1 対多の関係で対応付けられるもの，多対多の関係で対応付けられるもの，1 対空の関係で対応付けられるもののに分類できる．

あいまい度とは，マッピング・パターンによって分類された各項目間の対応関係の類似性を定量的に示す尺度である．あいまい度を用いることで対応項目間の親密度を示すことができる．図 1 にマッピング・パターンとあいまい度の例を示す．

2.2　Fuzzy Schema 言語の要件と設計

前項で示した Fuzzy Schema を実現するにあたり，その概念を記述できる言語が必要となる．本研究では，マッピング・パターン及びあいまい度は XML/RDF 形式で記述できることを要件とする．XML はデータ及び文書の表現形式として浸透してきており，柔軟なデータ構造を表現できる．また，RDF は Web 上で識別できるリソースを論理的に記述表現するためのフレームワークであり，トリプル (主語，述語，目的語) を用いてシンプルに記述することができる．XML によるデータ交換や相互運用に高い親和性を持つため，本研究のようにデータ構造間の関係性を記述するのに適している．

Fuzzy Schema 言語は RDF 間の関係性を表現する言語となる．そのため，対象となるデータが関係データベースで管理されている場合，一旦 XML/RDF 形式に変換する必要がある．この場合，項目名 (関係データベースの属性名) を要素とし，各データ (各属性値) をリテラル値として RDF に置き換えることができる．この際，ノード要素は直接リテラル値を持つことができないため，項目名がプロパティ要素となるよう，rdf:parseType="Resource"を与える (図 2)．また，対象となるデータが XML に見られる階層構造を持つ場合，構造の深さを rdf:resource 属性を用いて Path 表記することでフラットな XML/RDF 形式に書き換え，対象 RDF 間の関係をより明確に記述できるようにする (図 3)．

リスト 1 に Fuzzy Schema 言語の概要を示す．

まず，2 行目で比較対象となっている双方の名前空間の宣言を行う．

次に，5-8 行目で対象となるリソースをそれぞれ rdf:about 属性を用いて指し示す．

10 行目以降は，項目間の関係が 1 対 1，1 対多，多対 1，多対多，1 対空，空対 1 であった場合の対応関係を示す．ここでは比較元/比較先の要素位置の記述と，比較元-比較先間で各要素がどの程度類似しているのかを示すあいまい度の記述を行う．

リスト 1: Fuzzy Schema 言語

```
 1  <?xml version="1.0" encoding="utf-8"?>
 2  <名前空間 />
 3  <ルート>
 4  <!-- 対象とするリソースの記述 -->
 5    <対象リソース>
 6      <比較元，リソース位置 />
 7      <比較先，リソース位置 />
 8    </対象リソース>
 9  <!-- 1 対 1 対応 -->
10    <1 対 1>
11      <比較元，要素位置 />
12      <比較先，要素位置 />
13      <あいまい度，比較先-比較元の割合 />
14    </1 対 1>
15  <!-- 1 対多対応 -->
16    <1 対多>
17      <比較元，要素位置 />
18      <比較先，要素位置，割合 (比較先)/>
19      <あいまい度，比較先-比較元の割合 />
20    </1 対多>
21  <!-- 多対 1 対応 -->
22    <多対 1>
23      <比較先，要素位置，割合 (比較先)/>
24      <比較元，要素位置 />
25      <あいまい度，比較先-比較元の割合 />
26    </多対 1>
27  <!-- 多対多対応 -->
28    <多対多>
29      <比較元，要素位置，割合 (比較元) />
30      <比較先，要素位置，割合 (比較先) />
31      <あいまい度，比較先-比較元の割合 />
32    </多対多>
33  <!-- 1 対空対応 -->
34    <1 対空>
35      <比較元，要素位置，割合 (比較元) />
36    </1 対空>
37  <!-- 空対 1 対応 -->
38    <空対 1>
39      <比較先，要素位置，割合 (比較先) />
40    </空対 1>
41  </ルート>
```

対応関係が 1 対多，多対 1，多対多の場合，比較元/比較先には要素が複数個記述される．これらの要素はシーケンシャルなリストとして記述するため，rdf:Seq タイプを用いた RDF コンテナモデルとして記述する．

また，1 対空，空対 1 の場合，対応関係が取れないため，比較元・比較先それぞれの要素位置のみを示す．

なお，Fuzzy Schema 言語を設計するにあたり，RDFS で定義されるボキャブラリ以外に，上記を実現するために 15 要素，3 属性を新しくボキャブラリとして定義した．

2.3 対象とする領域とデータ

本研究では，博物館におけるメタデータ間の情報交換を対象に Fuzzy Schema を適用する．

博物館は古くから紙の台帳あるいは目録という形で文化財に関するデータを収集管理してきた．近年では情報技術を用いた台帳の電子化，データベース化が進みつつある．また，それに伴い様々な博物館用メタデータスキーマが提案されている (表 1)．

この背景には，博物館が管理対象とする文化財が歴史・芸術・民族・産業・自然科学等多岐に渡ることが挙げられる．近年ではあるテーマに沿って複数の分野にまたがる文化財を一堂に会して展覧会が開催されたり，ある歴史研究を行う上で複数の分野にまたがる博物館情報を横断的に収集する必要があるといった，博物館情報の横断利用の要求も大きい．

表 1 様々な博物館メタデータスキーマ

Table 1 Metadata schema of museum information

名称	内容
ミュージアム資料情報構造化モデル [4]	東京国立博物館が国内向けに提案．国際標準と互換あり．博物館業務支援及び情報共有を目的とする．
CIDOC/IC	国際博物館会議 (ICOM) のドキュメンテーション委員会 (CIDOC) が提案する国際標準ガイドライン．
CIDOC/CRM	CIDOC IC の情報共有を目的とした概念参照モデル．XML/RDF 形式で記述可能．
SPECTRUM	英国の博物館ドキュメンテーション協会 (MDA) が提案する手続き型モデル．
CDWA	米国の Getty Research Institute が中心となり策定された美術情報のメタデータスキーマ．XML Schema をベースとする．
遺物分類標準	韓国国内で利用されるメタデータスキーマ．Dublin Core をベースとした基本 16 項目，オプション 116 項目を有する．

そこで，本研究では，既に公開されている仕様を用いて疑似的に作成したスキーマと実際に博物館で運用されているスキーマを用意し，これらの異なるスキーマに対して Fuzzy Schema を適用することで高精度な情報検索を可能とするシステムを試作する．

具体的に用いるデータは，前者は東京国立博物館を中心に策定されたミュージアム資料情報構造化モデル [4] をベースに疑似的に作成したスキーマ，後者は印刷技術や印刷物を中心に資料収集を行っている印刷博物館 [5] のデータベーススキーマを用いる．

3. プロトタイプシステムの実現方法

3.1 システム構成

図 4 にプロトタイプシステムの構成を示す．システムは Web アプリケーションであり，仮想的に 2 つの異なるデータベースを配置し，ユーザは Web ブラウザを介して各データベースからスキーマの XML/RDF 形式への変換，Fuzzy Schema の生成，各データベースの検索，Fuzzy Schema を介した検索を実行できる．実装は Java1.4.2 を用い，環境として OS は Debian GNU/Linux を用いた．また，データベース管理ソフトはネイティブ XML データベースである eXist 及びリレーショナルデータベースである MySQL を用いた．アプリケーションサーバには Apache HTTP サーバ，Apache Tomcat を用いている．

3.2 データベースの設計

本システムでは異なる 2 つのデータベースを用いる．以下，それぞれのデータベースについて述べる．

a) ネイティブ XML データベース (NXD)

eXist は Java で記述されたオープンソースのネイティブ XML データベースシステムであり，XQuery，XPath，XUpdate をサポートしている．また，Java API によるネストしたクエリの発行や結果ソースの取得等も行える．実行アーキテクチャとして，war ファイルをデプロイすることでサーバ上の 1 アプリケーションとして動作させることができる．

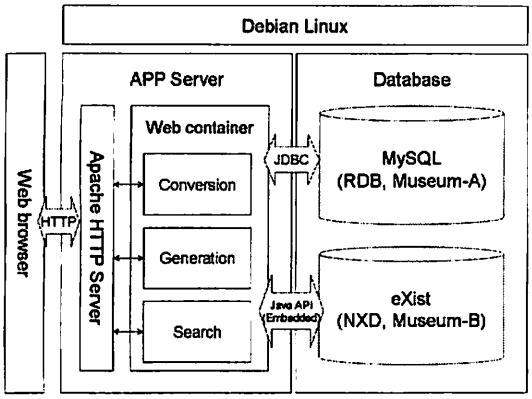

図 4　メタデータスキーマ統合システム

Fig. 4　Metadata schema integration system

リスト 2: XML Schema 例

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  ... >
<!-- ルートの定義 -->
 <xs:element name="sampleMuseum"
             type="sampleMuseum"/>
  <xs:complexType name="sampleMuseum">
   <xs:sequence>
    <xs:element name="museumObject"
                type="museumObject"
                maxOccurs="unbounded"/>
   </xs:sequence>
  </xs:complexType>
<!-- 各要素の定義 -->
  <xs:complexType name="museumObject">
   <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="identifier"
                type="xs:string"/>
     <xs:element name="objectNumber"
                 maxOccurs="1">
     </xs:element>
    ...
<!-- 各要素で使用する要素定義 -->
<!-- 期間，人間情報，場所情報等 -->
</xs:schema>
```

リスト 3: XML インスタンス例

```
<!-- リスト 2 から生成したインスタンス例 -->
<?xml version="1.0" encoding="UTF-8"?>
<sampleMuseum>
 <museumObject>
  <identifier>http:// ... </identifier>
   <objectNumber>
    <objectNumberString>0001</objectNumberString>
    <objectNumberType>資料 A</objectNumberType>
   </objectNumber>
   <title>
    <titleString>タイトル 1</titleString>
   </title>
   <classification>
    <classificationString>
      分類 A
    </classificationString>
   </classification>
   <originalUse>用途 A</originalUse>
    ...
 </museumObject>
</sampleMuseum>
```

表 2　印刷博物館のスキーマ例

Table 2　An example of Printing museum metadata schema

番号	項目名	データ型
1	旧整理番号 (old_arrangement_number)	int
2	親番号 (parents_number)	int
3	枝番号 (branch_number)	int
4	簡易名称 (name_simple)	char(100)
5	資料よみ (name_note)	char(100)
6	正式名称・日本語 (name_jp)	char(100)
7	正式名称・英語 (name_en)	char(100)
8	収蔵分類 (collection_classification)	char(100)
9	分類 (classification)	char(100)
10	分類番号 1(classification1)	char(10)
・	‥	・
16	キーワード (kerword)	char(50)
17	モチーフ (motif)	char(50)
18	発行・製作年月日 (西暦) (creating_date)	char(50)
・	‥	・

本システムでは，eXist にミュージアム資料情報構造化モデルをベースに疑似的に作成したスキーマ及びインスタンスを格納した．ミュージアム資料情報構造化モデルは博物館間での情報共有を意識しており，RDFS により語彙が定められている等，XML 処理を見据えたモデルとなっている．スキーマは XML Schema 形式で，要素 95 項目，属性 1 項目が定義されている．また，このスキーマを元に疑似的に 1000 件の XML インスタンスを作成した．リスト 2 にスキーマ例，リスト 3 にインスタンス例を示す．

b)　リレーショナルデータベース (RDB)

MySQL はオープンソースの RDBMS の実装の 1 つで世界的に広く指示されている．著者らの調査によると [6]，国内の中小規模の博物館の多くは利用の都合上，学芸員に負担のかからない簡易な RDBMS を用いて文化財の管理がなされている．そこで，本システムでは，MySQL に印刷博物館で実際に用いられているスキーマ及びテーブルを格納した．なお，データ項目数は 116 項目，データ格納件数は 997 件である．表 2 にスキーマ例を示す．

3.3　変換機能

それぞれのデータベースに格納されたデータは，アプリケーションサーバの変換機能でフラットな XML/RDF 形式のデータに変換される．以下の手順では，ユーザがブラウザで確認することを前提に処理内容を記してあるが，変換後の XML/RDF 形式のデータは別途 NXD に格納され，Fuzzy Schema 生成機能で利用される．

c)　NXD からの変換

作成した疑似スキーマとインスタンスを予め NXD に格納しておく (図 5(1))．ユーザは Web ブラウザを介してサーバにリクエストを発行し (図 5(2))，変換機能を介して NXD にアクセスする (図 5(3))．変換機能は NXD からスキーマを取得し (図 5(4))，XML/RDF 形式への変換を行う．その後，変換結果を Web ブラウザに返す (図 5(5))．

リスト 4: Fuzzy Schema 例

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:fs="http://example.org/fs#">
<fs:FuzzySchema>
<fs:TargetSchema>
 <fs:Relations>
  <fs:Location>
   <fs:OriginLocation rdf:about="URI" />
   <fs:EndLocation rdf:about="URI" />
  </fs:Location>
 </fs:Relations>
</fs:TargetSchema>
<fs:OnetoOneMatching>
 <fs:Relations>
  <fs:Location>
   <fs:OriginLocation rdf:resource="Path" />
   <fs:EndLocation rdf:resource="Path" />
  </fs:Location>
 </fs:Relations>
</fs:OnetoOneMatching>
<fs:OneToManyMatching>
 <fs:Relations>
  <fs:Location>
   <fs:OriginLocation rdf:resource="Path" />
   <fs:EndLocation>
    <rdf:Seq>
     <rdf:_1
      rdf:resource="Path"
      fs:EndLocationRate="%"/>
    </rdf:Seq>
   </fs:EndLocation>
  </fs:Location>
  <fs:Rates>
   <fs:ItemList>
    <fs:Item fs:Rate="%">
     <fs:OriginLocation rdf:resource="Path" />
     <fs:EndLocation rdf:resource="Path" />
    </fs:Item>
   </fs:ItemList>
  </fs:Rates>
 </fs:Relations>
</fs:OneToManyMatching>
<fs:ManyToOneMatching>
 <fs:Relations>
  <fs:Location>
   <fs:OriginLocation>
    <rdf:Seq>
     <rdf:_1
      rdf:resource="Path"
      fs:OriginLocationRate="%"/>
    </rdf:Seq>
   </fs:OriginLocation>
   <fs:EndLocation rdf:resource="Path" />
  </fs:Location>
  <fs:Rates />
 </fs:Relations>
</fs:ManyToOneMatching>
<fs:ManyToManyMatching>
 <fs:Relations>
  <fs:Location>
   <fs:OriginLocation />
   <fs:EndLocation />
  </fs:Location>
  <fs:Rates />
 </fs:Relations>
</fs:ManyToManyMatching>
<fs:OneToNothingMatching>
 <fs:Relations>
  <fs:Location>
   <fs:OriginLocation rdf:resource="Path" />
  </fs:Location>
 </fs:Relations>
</fs:OneToNothingMatching>
<fs:NothingToOneMatching>
 <fs:Relations>
  <fs:Location>
   <fs:EndLocation rdf:resource="Path" />
  </fs:Location>
 </fs:Relations>
</fs:NothingToOneMatching>
</fs:FuzzySchema>
</rdf:RDF>
```

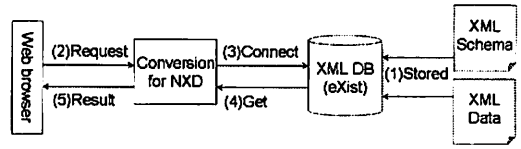

図 5 NXD を用いた XML/RDF 変換機能
Fig. 5 XML/RDF Conversion function using NXD

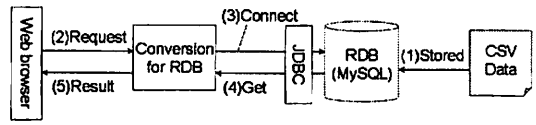

図 6 RDB を用いた XML/RDF 変換機能
Fig. 6 XML/RDF Conversion function using RDB

d) RDB からの変換

RDB に予め印刷博物館用のスキーマとデータを格納しておく (図 6(1)). ユーザは Web ブラウザを介してサーバにリクエストを発行し (図 6(2)), 変換機能は JDBC を介して RDB にアクセスし (図 6(3)), RDB から取得したスキーマ (図 6(4)) を XML/RDF 形式に変換する. その後, 変換結果を Web ブラウザに返す (図 6(5)). 今回は複雑な関係表は用いず, 1 テーブルから変換を行っている.

3.4 Fuzzy Schema 生成機能

Fuzzy Schema 生成機能は変換機能を介して生成された XML/RDF 形式のスキーマからマッピング・パターンとあいまい度を計算して Fuzzy Schema を生成する. 現時点では, 計算アルゴリズムに未着手であるため, 今回は手作業で Fuzzy Schema を作成した. リスト 4 に作成した Fuzzy Schema 例を示す.

3.5 検索機能

検索機能の概要を図 7, 図 8 に示す.

本システムでは対象データベース毎に異なる検索機能を配置した. ユーザは Web ブラウザを介して, 検索機能にクエリを発行する (図 7(1), 図 8(3)). 検索機能はそれぞれのデータベースに接続し (図 7(2), 図 8(4)), 取得した検索結果 (図 7(3), 図 8(5)) をブラウザに返す (図 7(4), 図 8(6)). 検索言語は NXD では XQuery を, RDB では SQL を用いた. なお, 今回は手作業で作成した Fuzzy Schema(図 8(2)) は NXD に格納し, XQuery による検索機能を実装した.

4. 関連研究

4.1 データ構造とその記述

情報の共有を目的とした異種情報の統合に関する研究は古くから行われている [7]. データ構造の異種性に着目した研究には, Kim のデータの異種性の分類 [8] がある. 関係データベー

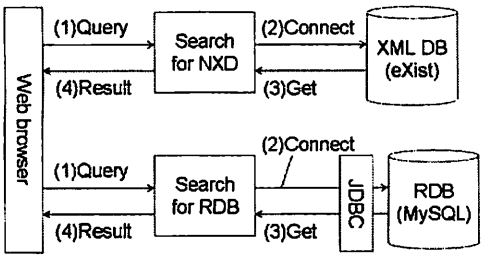

図 7 NXD/RDB を用いた検索機能
Fig. 7 Search function using NXD/RDB

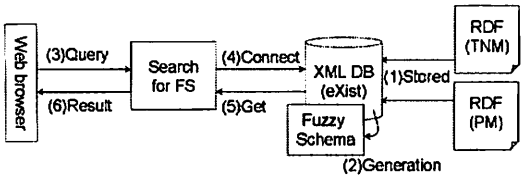

図 8 Fuzzy Schema を用いた検索機能
Fig. 8 Search function using Fuzzy Schema

スにおけるスキーマ定義言語，質問言語に要求される記述能力を明らかにすることを狙いとした分類で，スキーマの異種性をレコード間，データ項目間，レコードとデータ項目間に大分類し，各々異名同義や同名異義の名称の差異，属性数の多寡等で小分類がされている．

また，半構造データモデル [9] を記述する言語として，Object Exchange Model(OEM) がある [10]．様々な資源からのデータを，ある統一的なモデルにマップすることによって統一的なインターフェースを提供するシステムにおける統一モデルとして提案された．

1990 年代半ば頃からは，WWW 上のデータ (主に HTML, XML) に対するスキーマ定義言語，質問言語の記述に関する研究が取り上げられている [11]．

また，意味情報の異種性の観点から見ると，オントロジを用いて異種性の解消を試みる取り組みがある．大規模なオントロジ活用プロジェクトの Cyc [12]，概念辞書の WordNet [13]，エージェントの通信・知識記述言語の KQML [14] や KIF [15] 等が挙げられる．Web 上のデータに対しては，W3C のウェブオントロジ言語に OWL [16] がある．OWL は半構造データを対象としたオントロジの構築言語と言える．

4.2 異種システムの統合

異種システムを統合する取り組みとして，メディエータを用いたメディエータシステムがある [17]．メディエータシステムは対象とする情報源 (データベース，知識ベース，Web 等) の変化に対してサービスの安定的供給，異種性解消，情報統合，要求応答，といった一連の処理を目指したシステムである．通常各情報源にラッパをかけ，共通モデルに変換した情報を利用するが，一意的なラッパの作成で拡張や変更に柔軟に対応することは難しい．

また，近年の取り組みとして，大規模システムをサービスの集合として構築する Service Oriented Architechture(SOA) がある．システム間に共通のインターフェースを配置しシステム間の連携を図るという考え方であり．技術的基盤としては，XML Web サービスがその中心になる．

5. まとめと今後の課題

本稿では異なるスキーマを持つメタデータ間の情報交換を目的としたメタデータスキーマ統合機構の試作について論じた．試作ではスキーマ間の関係性記述言語 Fuzzy Schema を基に，異なる博物館メタデータスキーマの XML/RDF 形式への変換機能，Fuzzy Schema 生成機能，各データベースへの検索機能を実装した．

今後の課題として，本システムを評価するために，予め用意した正解に対するクエリの差異，あるいは異なるデータベースへの同一クエリの結果等を用いて適合率/再現率を測り，システムの検証を行う予定である．

文　献

[1] Dempsy *et al.*: "Metadata: A Current View of Practice and Issues", J. of Documentation, Vol. 54, No. 2, pp. 145-172(Mar. 1998).
[2] Multimedia Content Description Interface(参照 2007-06-28) http://www.itscj.ipsj.or.jp/mpeg7/
[3] 秋元良仁, 亀山渉: "博物館情報に基づくメタデータスキーマ統合機構の構築", 情処研報, Vol. 2007, No. 34, pp. 9-16, 2007.
[4] ミュージアム資料情報構造化モデル (参照 2007-06-28) http://webarchives.tnm.jp/docs/informatics/smmoi/
[5] 印刷博物館 (参照 2007-06-29) http://www.printing-museum.org/
[6] 秋元良仁, 鈴木理洋: "博物館情報の相互利用を目的とした文化財情報システムの提案", 情処シンポジウム「じんもんこん 2004」, 2004.
[7] C. Batini *et al.*: "A Comparative analysis of methodologies for database schema integration", ACM Computing Surveys, No. 10, Vol. 4, pp. 323-364, 1986.
[8] W. Kim *et al.*: "Classifying Schematic and Data Heterogeneity in Multidatabase Systems", Computer, Vol. 24, No. 12, pp. 12-18, 1993.
[9] 田島敬史: "半構造データのためのデータモデルと操作言語", 情処論, Vol. 40, No. SIG3, pp. 157-170, 1999.
[10] Y. Papakonstantinou *et al.*: "Object Exchange across Heterogeneous Information Sources, Proc. of IEEE ICDE, pp. 251-260, 1995.
[11] 吉川正俊: "データベースの観点から見た XML の研究", 情報学シンポジウム, 2002.
[12] Cycorp(参照 2007-02-27) http://www.cyc.com/
[13] WordNet(参照 2007-06-29) http://wordnet.princeton.edu/
[14] Y. Labrou *et al.*: "A Proposal for a new KQML Specification", TR CS-97-03, 1997.
[15] M. R. Genesereth *et al.*: "Knowledge Interchange Format draft proposed American National Standard (dpANS) NCITS. T2/98-004"
[16] OWL Web Ontology Language(参照 2007-06-29) http://www.w3.org/TR/owl-features/
[17] G. Wiederhold: "Glossary: Intelligent Integration of Information", Jounal of Intelligent Information System, Vol. 5, pp. 101-112, 1996.