異種情報源統合のためのXMLメディエータ

出口　彰，小西　修
高知大学理学部数理情報科学科
780-8520 高知県高知市曙町 2-5-1
{98ss054, konishi}@is.kochi-u.ac.jp

概要

Web 環境に分散して存在する異種情報源間のセマンティクスレベルの異種性を解消し，XML の柔軟なデータ表現能力を活かし XML レベルでの情報統合を目的とする．エージェント型メディエーションシステムである HI-AMS と CORBA 環境を組み合わせたアーキテクチャを基本モデルとし，HI-AMS 上に，スキーマ管理，トレーダサービス，ブローカー機能，オントロジから構成されるメタデータ機構を構築することでセマンティクスレベルの異種性を解消した．また，HI-AMS における情報収集エージェントであるアクセスエージェントが収集した情報を XML 出版し，XML レベルで統合した．そしてそれを蓄積していくことで XML データベースを HI-AMS 上に構築した．これとエージェントの学習機能を組み合わせることで，エージェントの自律性，成長性を最大限活かした HI-AMS を構築した．この HI-AMS によって，利用者は異種分散情報源によって構築されたネットワーク環境においても HI-AMS に問い合わせをするだけで自分の目的とする情報を的確に獲得することが可能となった．

キーワード：メディエータ，XML 情報統合，KQML-CORBA，メタデータ機構，RDF，DublinCore，Datalog

XML-Mediator for Integration of Heterogeneous Information Sources

Akira DEGUCHI, Osamu KONISHI
Dept.of Information Science,Faculty of Science,Kochi University
2-5-1 Akebono-cho，Kochi City，Kochi，780-8520 Japan
{98ss054, konishi}@is.kochi-u.ac.jp

Abstract

Unific access of the information sources by the mediation system in distributed heterogeneous information sources is described. HI-AMS which is a mediation system consists of two main mechanisms. They are a metadata mechanism for canceling the different-species nature of the sematics level which exists between the heterogeneous information sources, and an information integration mechanism for integrating the information collected from heterogeneous information sources. A metadata mechanism consists of schema management, a trader service, a broker function, and an ontology, and an information integration mechanism consists of an integration algorithm, a schema control, an agent learning function, and XQuery. HI-AMS consists of a user interface agent, an access agent, an integrated agent, and facilitator. And an agent's communication by KQML. Moreover, KQML-CORBA environment was built by exchanging the message of KQML form with the wrapper installed in heterogeneous information sources through ORB. In this paper, the architecture, schema definition by RDF in schema management, and deductive axiom and the rule processing by Datalog in ontology is described in full detail.

Key words: Mediator，XML Integration，KQML-CORBA，MetaData Mechanism，RDF，DublinCore，Datalog

1 はじめに

近年のネットワークはインターネットの急速な発展によって，利用者の目的とする情報を蓄えた情報源は世界中に分散して存在している．利用者は世界中のサイトから自分の目的とする情報を獲得することができるようになってきた．しかし，これが逆にダイナミックに変化する異種情報源の構築を促してしまうという問題も引き起こしている．この環境には，HTML ファイルのような半構造データも含まれており明確にデータ定義はされていない．また，異なるプラットフォーム上で異なる形式で存在している．さらに動的に情報源が出現・消滅・移動するため利用者がその情報源の位置情報を的確に把握することは難しくなっている．このような問題の解決策の一つとしてメディエーションシステムがある．これは，利用者と情報提供者との間に位置する仲介役であり，このシステムによって利用者は，情報源の位置・検索方法などを知らなくてもメディエータに問い合わせるだけで目的とする情報が獲得できる．しかし，情報源のメタデータやデータ表現方法などの異種性は存在する．このために，利用者が異種性を考慮して検索する必要性が生じている．本研究では，メディエーションシステムである HI-AMS(High Intelligence Active Mediation System)[3] にメタデータ機構を構築することでメタデータやデータ表現方法の異種性を解消した．さらに，近年インターネットにおいて，データベースやデータ交換フォーマットなどとして注目を集めている XML に注目して，異種情報源から検索した情報を XML で統合し，結果の XML 文書をシステム上に構築することでデータベースとして利用した．

2 関連研究

本研究の関連研究としては，利用者からの問い合わせに対しラッパーを用いて情報源固有の検索式へ変換して，その結果を共通モデルである OEM(Object Exchange Model) へ変換する研究として，Stanford 大学の TSIMMIS プロジェクト [5] があげられる．OEM とは，半構造データであり，ラベルのついた有向辺で頂点間をつないだグラフ構造によってデータベースを表現する．これは，敢えて利用者の目的とする情報を，リレーショナルデータベースやオブジェクト指向データベース，XML に格納せず半構造データにすることによって，柔軟な問い合わせをすることを可能としている．OEM に対する問合わせ言語として Lorel 言語などがある．また，メディエータの研究として，Microelectronics and Computer Technology Corporation(MCC) の InfoSleuth プロジェクト [6] があり，これはオントロジの概念を用いて Web 環境における情報源を検索し，検索結果は Web ドキュメントとして返される．さらに，近年注目を集めている XML とデータベースに関する研究について説明する．XML が注目を集めているが，企業の基幹系の業務などでは，依然として情報源にリレーショナルデータベースが使用されているという背景から，XML 文書をリレーショナルデータベースにマッピングし SQL ベースの問合わせによって利用者が情報を獲得するという研究が多くなされている．また，XML 出版とアウタージョインの概念を利用し情報統合を扱った研究や XML データベースを利用した研究などもなされている [7][8].

そこで，本研究とこれらの研究との違いを述べておく，まずメディエータ，ラッパーの研究においては，本研究ではすべての情報源を XML 出版し結果を統合するという点や，メタデータ機構を有するメディエーションシステムであるという点において異なる．後半の XML とデータベースの研究については，基本的に本研究はメディエーションシステムをメインに取り扱っている点で異なるのであるが，システムをすべて XML ベースで構築しているという点でも他のものとは異なるといえる．

3 アーキテクチャ

3.1 HI-AMS

本研究の基本モデルである HI-AMS は，ユーザインタフェースエージェント，アクセスエージェント，統合エージェント，ファシリテータから構成され，それぞれのエージェントが自律的な振る舞いをする．ユーザインタフェースエージェントは，利用者の検索要求を WWW ブラウザで受け付ける GUI 機能と，その検索要求をメッセージとしてファシリテータに送信して検索結果を，利用者に提示する機能を提供する．アクセスエージェントは，検索要求メッセージを受信し異種情報源にアクセスし利用者の要求を満たす情報を収集する機能を提供する．統合エージェントは，アクセスエージェントによって収集された情報を統合する機能を提供する．ファシリテータは，これらのエージェントを協調促進させる役目をもつ．HI-AMS は，これらのエージェントに基づいて異種情報源を統合し，利用者の質問に対して，最適な結果を返す自律的なシステムである．今回の研究では，HI-AMS にスキーマ管理，トレーダサービス，ブローカー機能，オントロジから構成されるメタデータ機構を構築することで，情報源間におけるメタデータやデータの表現方法などの異種性を解消した．これによって，HI-AMS を構成するエージェント群は異種性を考慮した検索が可能となる．そして HI-AMS のアクセスエージェントが異種情報源から収集した情報を XML 出版し XML データベースに格納していく．また，その XML データベースに対する検索は XQuery によって表現され処理される．HI-AMS 全体としては，RDF[9]，RDF-Schema[10]，オントロジ記述の XML による表現，XML 出版，XML 情報統合，XQuery

などの技術を利用することで XML ベースの HI-AMS を構築した.

4 KQML-CORBA 環境

本研究では HI-AMS 内におけるエージェント通信言語として KQML(Knowledge Query and Manipulation Language) を採用している．KQML は，知識システム間で様々な知識を交換することを目的として設計された発話行為論に基づいたエージェント間知識操作言語である．これは，エージェントの会話内容に関しては特別な規定がなく一貫性制約なども表現することができるため，柔軟なコミュニケーションが構築できる.

4.1 KQML-CORBA

HI-AMS における情報収集エージェントであるアクセスエージェントは，ORB を介して異種情報源を検索する．このために，HI-AMS は分散オブジェクト技術 CORBA を利用する．このとき，情報源と HI-AMS とのインタフェースを以下のように IDL によって KQML オブジェクトの構造体を埋め込み定義する.

```
module HIAMS{

  typedef sequence<octet> ByteString;

  struct KQMLmsgObj{
    string performative;
    string sender;
    string receiver;
    long id;
    string ontology;
    ByteString msg;
  };
```

これによって，HI-AMS を構成するエージェント群は外部の情報源に設置されたラッパーと通信する際も KQML 形式のメッセージでやり取りを行う．つまり内部だけでなく外部との通信にも KQML の利点を活かすことが可能となる．以下は，KQML-CORBA 環境下でアクセスエージェントが検索要求を受け付け，適切なメッセージであったなら，ラッピングクラスで包括して ORB を介して情報源に設置されたラッパーに検索要求をするプログラムコードの抜粋である.

```
if(sender.equals("Facilitator")){
  if(performative.indexOf("query") >= 0){
//KQMLmessagem（ラッピングクラス）化する.
          KQMLmessage message = new KQMLmessage();
          message.setperformative(performative);
          message.setsender(sender);
          message.setreceiver(receiver);
          message.setid(id);
          message.setontology(ontology);
          message.setmsg(content);
          KQMLmessage res_message = getInfo(ontology,message,name);
            …
public static KQMLmessage getInfo(String ServerName
      ,KQMLmessage KQML,String name){
    Properties props = new Properties();
    props.put("org.omg.CORBA.ORBInitialHost",args[0]);
    props.put("org.omg.CORBA.ORBInitialPort","1050");
// create and initialize the ORB
    ORB orb = ORB.init(args,props);
// get the root naming context
    org.omg.CORBA.Object objRef =
        orb.resolve_initial_references("NameService");
    NamingContext ncRef = NamingContextHelper.narrow(objRef);
// resolve the Object Reference in Naming
    NameComponent nc = new NameComponent(ServerName,"");
    NameComponent path[] = {nc};
    Wrapper WrapperAgentRef =
        WrapperHelper.narrow(ncRef.resolve(path));
// call the Wrapper server object and print results
    WrapperAgentRef.init();
    KQMLmessage msg = new
        KQMLmessage(WrapperAgentRef.receiveMsg(KQML.getData()));
```

5 メタデータ機構

本研究では，メタデータの異種性を解消するための一つの手段として，メタデータ機構を HI-AMS 上に構築することを提案する．メタデータ機構は，HI-AMS スキーマ管理，CORBA トレーダサービス，ブローカー機能，オントロジから構成される.

5.1 スキーマ管理

データベースを構築する際には，データに対してメタデータが定まっている必要がある．メタデータによってそのデータが何に関するものであるかを知ることができる．ここに HI-AMS を構成するエージェント群が使用するメタデータを RDF-Schema によって定義する.

今回の実験では，書誌情報にシステムを適用したので，書誌情報に関するスキーマ定義の一部を図 1 に示す．これとあわせて DublinCore メタデータ [11] を使うことでシステムのメタデータとする．この定義によって，例えば Price が Book の値段を表しているスキーマだということを定義できた．これによって，スキーマのセマンティクスを考慮することが可能である.

■ HI-AMS のスキーマ定義

```
<?xml version=1.0>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax#"\
     xmlns:rdfs="http://w3.org/2000/01/rdf-schema#">

    <rdfs:Class rdf:ID="BOOK">
        <rdfs:label>book</rdfs:label>
        <rdfs:comment>BOOK CLASS</rdfs:comment>
        <rdfs:isDefinedBy resource =
            "http://peeko.is.kochi-u.ac.jp/~98ss54/schema/"/>
        <rdfs:subClassOf rdf:resource =
            "http://www.w3.org/1999/02/22-rdf-syntax-ns#Resource">
    </rdfs:Class>
         …
    <rdf:Property rdf:ID="Price">
        <rdfs:label>Price</rdfs:label>
        <rdfs:comment>Price of a book</rdfs:comment>
        <rdfs:domain rdf:resource="BOOK">
        <rdfs:isDefinedBy resource =
            "http://peeko.is.kochi-u.ac.jp/~98ss54/schema/"/>
        <rdfs:range rdf:resource=
            "http://www.w3.org/1999/02/22-rdf-syntax-ns#Literal">
    </rdf:Property>
         …
```

図 1: RDF によるスキーマ定義

そして，RDF による書誌情報のオントロジ記述と情報源のオントロジ記述を図 2 に与える．このオントロジ記述によって情報源と HI-AMS のオントロジの対応付けを行う．また，このように RDF によって HI-AMS のオントロジ記述を行うことによって，書誌情報以外の情報を検索対象として取り扱う場合でもその情報に関するオントロジ記述を定義するのみで HI-AMS に機能を追加することが可能となる．このことから情報源として様々なものを取り入れることが可能となり，汎用的なシステムとなる．

■情報源のオントロジ記述

```
<?xml version="1.0" encoding="Shift_JIS"?>
<!DOCTYPE Ontology SYSTEM
          "http://peeko.is.kochi-u.ac.jp/~98ss54/MetaData.dtd">
<Ontology name="XMLOntology">
    <DataSource name="SoftBank">
        <Title name="タイトル" value="false"/>
        <Creator name="著者" value="false"/>
        <category name="分野" value="true">
            <value name = "文学"/>
            <value name = "自然科学"/>
                ....
            <value name = "情報工学"/>
            <value name = "数学"/>
        </category>
        <ISBN name = "ISBN" value="false"/>
                ....
    </DataSource>
</Ontology>
```

■ HI-AMS の書誌情報に関するオントロジ

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdf:Description
            about="http://peeko.is.kochi-u.ac.jp/~98ss54/MetaData">
        <rdf:Seq>
            <rdf:li>
                <Attribute name="Title" value="false"/>
            </rdf:li>
            <rdf:li>
                <Attribute name="Creator" value="false"/>
            </rdf:li>
            <rdf:li>
                <Attribute name="category" value="true">
                    <value name="文学">
                    <value name="自然科学">
                        ...
                    <value name="電気・電子">
                    <value name="数学">
                </Attribute>
            </rdf:li>
            <rdf:li>
                <Attribute name="ISBN" value="false"
                                    PrimaryKey="true"/>
            </rdf:li>
                ...
        </rdf:Seq>
</rdf:RDF>
```

図 2: オントロジ記述

5.2 トレーダサービス

CORBA トレーダサービスを実装することで，ネーミングサービスに登録された情報源がどのような情報を持っていて，どのようなメタデータであるか，という情報を取得することが可能となる．

ネーミングサービスに登録された情報源に設置されたラッパーが図 2 に示した情報源のオントロジ記述より必要な情報を抽出しファシリテータに通知する．ファシリテータは，この情報より知識ベースを構築していく．この知識ベースより，以下に説明するブローカー機能とオントロジ変換ルールを生成することが可能となる．

5.3 ブローカー機能

情報源側に設置されたラッパーから HI-AMS への宣伝情報の通知により図 3 のような知識ベースをファシリテータは構築していく．つまり，情報源が何に関する情報を取り扱っていて，どのようなスキーマを持っているかである．この知識ベースは，情報源のラッパーが起動された時点で自動的にファシリテータに通知され，知識ベースも自動的に構築されていく．

Schema	category	ServerName
Title	book	XMLWrapper_aquila
Creator	book	XMLWrepper_aquila
	...	
Title	book	RDBWrapper_peeko
Price	book	RDBWrapper_peeko
	...	

図 3: ブローカー機能

これによって，HI-AMS のエージェント群が利用者から検索要求を受け付けたときに，この知識ベースより検索対象となる情報源を特定することが可能となる．図 3 に示す属性名 Schema の値は情報源固有のスキーマ名ではなく RDF-Schema によって定義された HI-AMS のスキーマである．

5.4 オントロジ

異種情報源間に存在するセマンティクスレベルの異種性を解消するために，本研究では，演繹データベースの演繹公理の概念を利用して HI-AMS にオントロジ変換ルールを定義した．ルール部は Datalog を利用して表現される．

5.4.1 オントロジ変換ルール

まず，前述のトレーダサービスによって図 4 に示す二つのファクト集合が得られる．それぞれ，equals が概念的に同じものであることを表し，usedby は情報源で使用されるスキーマ名を表す．

source	destination
Title	タイトル
Creator	著者
Price	値段
⋮	⋮

(a) テーブル equals

agent	destination
XMLWrapper_aquila	タイトル
XMLWrapper_aquila	著者
XMLWrapper_aquila	値段
⋮	⋮

(b) テーブル usedby

図 4: ファクト集合

ここで，schema(X,Y,Z) を「属性名 X は 情報源 Y において Z と表される」と定義する．そして，演繹公理として，

「X と Z は同じものであり，Z は Y において使用されているならば，属性名 X は情報源 Y において Z で表される」が導かれる．これを，Datalog によって表現すると以下のようになる．

```
schema(X,Y,Z) <- equals(X,Z),usedby(Y,Z)
```

そして，これに対する問い合わせとして考えられる問い合わせが

```
? - schema('Title','XMLWrapper_aquila',X)
```

であり，これは「属性名 Title は情報源 XMLWrapper_aquila において何と表されるか」を示している．このとき演繹公理を

```
create view MetaData as
select equals.source,usedby.agent,usedby.destination
where equals.destination = usedby.destination
```

という SQL 文に変換する．これによって二つのファクト集合から図 5 のようなビュー表を生成する．

agent	source	destination
XMLWrapper_aquila	タイトル	Title
XMLWrapper_aquila	著者	Creator
XMLWrapper_aquila	Creator	著者
XMLWrapper_aquila	Price	値段
⋮	⋮	⋮

図 5: オントロジ変換ルール

そして，先ほどの Datalog 表現による問い合わせは，

```
select  destination
from  MetaData
where source = 'Title'
      and agent = 'XMLWrapper_aquila'
```

というビュー表に対する問い合わせに変換されて実行される．これによって，検索メッセージを検索対象の情報源のオントロジへと変換することが可能となる．また，このように演繹公理を作成しオントロジ変換ルールを作成することで，単純な変換だけでなく細かく制約条件を指定することが可能となる．

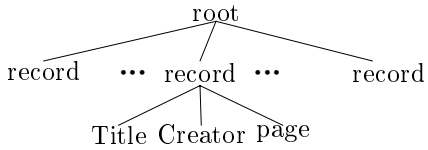
6 XML 情報統合

異種情報源からの検索結果を利用者の目的とするものに近づけるためには，それぞれの情報源からの結果を統合する必要がある．また，XML の普遍性，計算機での可読性，汎用性等から考えて結果を XML として利用者に提示することが最適と考えられる．

6.1 XML 出版

異種分散情報源の共通モデルとして XML を使用する．これまでは，ODB や RDB を共通モデルとして使用されてきたようであるが，たとえば RDB では，データ構造が硬いために異種分散情報源のデータ構造が複雑になるにつれて情報源間のデータ構造の違いを吸収することが困難になる．しかし，XML は木構造によって構造化され，柔軟なデータ表現能力をもつ．このために，異種分散情報源の共通モデルとして最適であると考えられる．

本研究では，XML 出版として異種分散情報源のデータベースを



のように root，record，そしてそれ以下に情報源の有するスキーマという形で行う．それによって，以下で示す統合アルゴリズムによって様々な情報源を統合することが可能となる．

6.2 統合アルゴリズム

異種情報源からの結果を XML レベルで統合するアルゴリズムについて述べる．まず，情報源から収集した結果を，DOM(Document Object Model) を利用して木構造でモデル化する．図 6 に木構造にモデル化された XML-A と XML-B を統合する例を示す．そして，図 7 にアルゴリズムを示す．図 7 の 5 行目の「$A.Child$ と $B.Child$ が同じデータである」ことを確認するには，ノード集合 A, B(record ノードの子ノード) の共通部分で表されるノード集合の内容をすべて比較することで実現できる．

ここで，図 2 の HI-AMS のオントロジ記述の属性 PrimaryKey に注目する．これは，その要素における name の属性値で表されるスキーマがすべての情報源を通してデータを一意的に識別できるスキーマということを表しているとする．つまり，PrimaryKey を指定することで上の説明のように共通するノード集合の内容すべてを比較することなく同じデータであるかどうかを判定することが可能となる．さらに，PrimaryKey を比較し同じデータと判定されると，ほかのスキーマの内容を比較することでデータの内容の表現の違いによる異種性を解消することが可能である．例えば，

```
Publisher       岩波書店
ISBN            ISBN1-2345-6789-0

Publisher       iwanami
ISBN            ISBN1-2345-6789-0
```

というデータであっても，同じデータと判断でき，さらに「iwanami」と「岩波書店」が同じものであるという知識を獲得できる．統合例を図 8 に 示す．

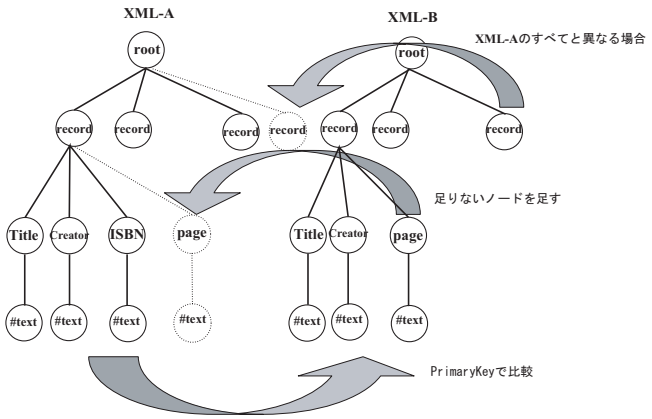


図 6: 木構造にモデル化された XML の統合

```
 1  while(XML-B の root の子が存在する){
 2      B.Child = B.root.getChild();
 3      while(XML-A の root の子が存在する){
 4          A.Child = A.root.getChild();
 5          if(A.Child と B.Child が同じデータであるなら){
 6              A := A.Child の子のノード集合
 7              B := B.Child の子のノード集合
 8              case A =  B
 9                  continue;
10              case A >  B
11                  continue;
12              case A <  B
13                  A-B のノードを A.Child に append
14              case A <> B
15                  A-B のノードを A.Child に append
16
17              Break;      //B.Node を次へ進める
18          }
19          else{
20              continue;   //A.Node を次へ進める
21          }
22      }
23      A.root へ B.Node を append
24  }
```

図 7: XML 情報統合アルゴリズム

6.3 スキーマ

HI-AMS のアクセスエージェントによって異種情報源から収集された情報は，DublinCore の名前空間に沿ったメタデータと RDF-Schema によって定義した HI-AMS のメタデータによって表現され，その XML データが統合される．このために，XML 出版された XML データは 6.1 節で述べたような構造をしている必要がある．よって XML 出版した XML データは，RELAX NG によって制御される．では，RELAX NG によって表現したスキーマの一部を以下に示す.

```
<?xml version="1.0"?>
<grammar ns = "" xmlns="http://relaxng.org/ns/structure/0.9"
    datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
  <start>
    <choice>
      <ref name="category"/>
            …
      <ref name="Title" ns="http://purl.org/dc/elements/1.1/"/>
            …                          // 名前空間の使用
      <element name="root">
        <zeroOrMore>
```

■ XML-A

```
<?xml version="1.0" encoding="Shift_JIS"?>
<root xmlns:dc="http://purl.org/dc/elements/1.1/">
    <record>
        <dc:Title>XML ハンドブック</dc:Title>
        <dc:Creator>渡辺竜生</dc:Creator>
        <Price>1400</Price>
        <PublicationDate>2001-01-25</PublicationDate>
        <catogory>コンピュータ</catogory>
        <ISBN>ISBN4-7973-1471-0</ISBN>
        <dc:Publisher>SOFTBANK</dc:Publisher>
        <Type>A4<Type>
    </record>
</root>
```

■ XML-B

```
<?xml version="1.0" encoding="Shift_JIS"?>
<root xmlns:dc="http://purl.org/dc/elements/1.1/">
    <record>
        <dc:Title>XML ハンドブック</dc:Title>
        <dc:Creator>渡辺竜生</dc:Creator>
        <Price>1400</Price>
        <PublicationDate>2001-01-25</PublicationDate>
        <catogory>コンピュータ</catogory>
        <ISBN>ISBN4-7973-1471-0</ISBN>
        <dc:Publisher>SOFTBANK</dc:Publisher>
        <page>250</page>
    </record>
    <record>
        <dc:Title>安楽に死にたい</dc:Title>
        <dc:Creator>松田道雄</Creator>
        <Price>1400</Price>
        <PublicationDate>1997-04-24</PublicationDate>
        <catogory>社会科学</catogory>
        <ISBN>ISBN4-00-002474-4</ISBN>
        <dc:Publisher>岩波書店</dc:Publisher>
        <page>120</page>
    </record>
</root>
```

■上の XML-A,XML-B の統合結果

```
<?xml version="1.0" encoding="Shift_JIS"?>
<root xmlns:dc="http://purl.org/dc/elements/1.1/">
    <record>
        <dc:Title>XML ハンドブック</dc:Title>
        <dc:Creator>渡辺竜生</dc:Creator>
        <Price>1400</Price>
        <PublicationDate>2001-01-25</PublicationDate>
        <catogory>コンピュータ</catogory>
        <ISBN>ISBN4-7973-1471-0</ISBN>
        <dc:Publisher>SOFTBANK</dc:Publisher>
        <Type>A4</Type>
        <page>250</page>
    </record>
    <record>
        <dc:Title>安楽に死にたい</dc:Title>
        <dc:Creator>松田道雄</dc:Creator>
        <Price>1400</Price>
        <PublicationDate>1997-04-24</PublicationDate>
        <catogory>社会科学</catogory>
        <ISBN>ISBN4-00-002474-4</ISBN>
        <dc:Publisher>岩波書店</dc:Publisher>
        <page>120</page>
    </record>
</root>
```

図 8: 統合例

```
          <ref name="record"/>
        </zeroOrMore>
      </element>
      <ref name="Creator"/>
    </choice>
  </start>
       …
  <define name="record">
    <element name="record">
      <group>
        <oneOrMore ns="http://purl.org/dc/elements/1.1/">
                                          // 名前空間の使用
          <ref name="Title"/>
        </oneOrMore>
        <optional>
          <ref name="category"/>
        </optional>
        <optional>
             …
      </group>
    </element>
  </define>
  <define name="Title" ns="http://purl.org/dc/elements/1.1/">
    <element name="Title">
      <data type="string"/>
    </element>
  </define>
       …
  <define name="Price">
    <element name="Price">
      <data type="integer"/>              // 整数型の定義
    </element>
  </define>
</grammar>
```

6.4 XQuery

本研究における HI-AMS は，XML データベースを内部に有している．この XML データベースに格納されている XML データは，6.1 節で述べた構造をしている．そして，このデータベースへのアクセスは XQuery にて行われる．タイトルが「アルゴリズム理論」であるものを検索する具体例を以下に示す．

```
For $x IN /root/record
Where $x/Title=“ アルゴリズム理論 ”
return $x
```

6.5 エージェント学習機能

HI-AMS を構成するエージェントであるファシリテータの学習機能について述べる．これまでのエージェントのもつ実行部に学習のための学習部を設計・追加する．この学習部は環境に関する知識とエージェントが行っている行為に関するフィードバックを基にして，将来実行部がよりよい振る舞いをするためにはそれをどのように修正すればよいかを決定する．

エージェントの経験として，利用者の検索要求に対して異種分散情報源へアクセスし利用者へ検索結果を返す．この経験より，エージェントは利用者のどのような検索に対してどのような情報が異種分散情報源から返され，どの情報を利用者に提示したか，またどのような検索が多くなされるかなどの知識を獲得できる．つまり，エージェント群は利用者の検索要求に対して，過去に経験した異種分散情報源検索より得た知識によって情報源までアクセスせずとも自らの経験と知識で利用者の要求に対応できるかどうかを判定する．このとき自ら結果を利用者に返せると判断するとエージェントは情報源検索を避けエージェントの知識ベースより結果を返すことになる．図 9 に書誌情報検索によるエージェントの新たに獲得する知識を示す．

id	schema	comp	value	address	category
1	Title	=	情報科学	2002_01_21/1.xml	book
2	Title	=	アルゴリズム	2002_01_21/2.xml	book
2	Price	<=	3000	2002_01_21/2.xml	book
3	Title	~	哲学	2002_01_21/3.xml	book
4	Price	<=	7000	2002_01_21/4.xml	book
⋮	⋮	⋮	⋮	⋮	⋮

図 9: エージェントの知識ベース

この例は，利用者からの検索要求における検索条件とその結果の XML データを対応付けたものである．そして，利用者から検索要求を受けたとき，エージェントがこの知識より利用者に回答することができるかを判定する．HI-AMS を構成するエージェント群は，この自らの経験によって獲得した知識より自律的に判断し行動することになる．

```
1 condition = 検索条件
2 if(完全一致が存在する)
3   return;
4 else
5   for(sub = condition.getFirstCondition; sub != null
            ;sub=sub.nextCondition){
6     if(sub と完全一致するものが存在する){
7       XML = getXML;
8       temp_sub = sub;
9       for(sub = condition.getFirstCondition; sub != null
              ;sub = sub.nextCondition){
10          if(sub != temp_sub)
11            XML.search();
12        }
13        return;
14      }
15    }
16    for(sub = condition.getFirstCondition; sub != null
            ;sub=sub.nextCondition){
17      if(利用可能なものが存在する){
18        XML = getXML;
19        for(sub = condition.getFirstCondition; sub
              != null;sub = sub.nextCondition){
20          XML.search();
21        }
22      }
23    }
```

図 10: 学習アルゴリズム

6.5.1 アルゴリズム

では，図 10 にエージェントの学習アルゴリズムを示す．このアルゴリズムによって，エージェント群は自己の知識より利用者の要求に応えることができるかを判断する．できると判断した場合は，エージェントは知識より必要なものを取り出し利用者に提示したり，知識より必要なもの全てを取り出し利用者の要求するものを作り提示したりする必要がある．この学習機能によって，ORB を介して情報源まで検索にいかず利用者に結果を提示できるということで，利用者からみればレスポンスの向上が期待できる．さらに，HI-AMS の XML 出版，XML 情報統合などの負荷を軽減できる．

7 システムの適用と実験結果

今回の実験では，書誌情報検索システムへ適用した．そのシステムのアーキテクチャを図 11 に示す．利用者は，ブラウザのみが必要であり，書誌のタイトル，値段などを入力することによって検索をすることができる．その検索内容は，ユーザインタフェースエージェントによってメッセージへ変換される．メッセージは，KQML 形式であり，メッセージの content 部が XQuery で表される．メッセージの例を以下に示す．

```
prformative query:sender UIAgent:receiver Facilitator:id 1:
ontology UIAgent:content FOR $x IN /root/record
                         Where $x/Title="情報科学"
                         RETURN $x
```

さらに，このメッセージがファシリテータに送られる．ファシリテータは，トレーダサービスによって情報源側か

ら取得した宣伝情報よりあらかじめ作成しておいたブローカー機能によって検索対象となる情報源を特定する．さらに，この検索メッセージに対してオントロジ変換ルールを適用する．例えば，上の検索メッセージは，

```
performative query:sender UIAgent:receiver Facilitator:id 1:
ontology XMLWrapper_aquila :content FOR $x IN /root/record
                                Where $x/タイトル="情報科学"
                                RETURN $x
```

に変換されてアクセスエージェントに送られる．アクセスエージェントは，ORB を介して情報源に設置されたラッパーに検索を要求する．ラッパーは，メッセージを情報源固有の検索式へと変換して検索をする．すべての検索結果は，ファシリテータから統合エージェントに渡され，DublinCore の名前空間に沿った XML へと変換される．統合エージェントは，これらを統合してファシリテータへ返す．ファシリテータは，この XML を XML データベースへと格納すると同時にユーザインタフェースエージェントに結果を返す．ユーザインタフェースエージェントは，XML に対応していないブラウザでも閲覧可能にするために HTML へと変換してユーザに提示する．また，XML データベースに格納された XML 文書は，エージェント学習機能と組み合わせることで利用される．検索結果を図 12 に示す．

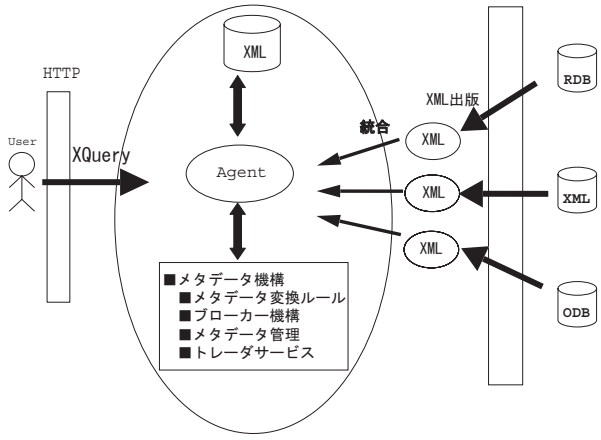


図 11: システム構成図

図 12: 検索結果

実装環境　今回の実験では，全体を通して RDB として PostgreSQL7.1.3，ODB として ObjectStore PSE for java2.0 を使用し，CORBA 環境は，JavaIDL によって実装した．書誌情報を，RDB，ODB，XML ファイルに蓄積し情報源とした．また，画像データを RDB に蓄積し書誌の画像データも検索対象とした．トレーダサービスによる，ブローカー機能とオントロジ変換ルールは RDB 上に構築した．

8　おわりに

本稿では，情報源間に存在するメタデータやデータ表現方法の異種性を解消するための一つの手段としてメタデータ機構を HI-AMS に構築することを提案した．検索結果の統合に関しては，異種分散情報源の共通モデルとして柔軟なデータ表現能力をもつ XML が最適であると考え，XML 出版，XML 情報統合を行い利用者には最適な結果を提示することができた．

今後の課題として，メタデータ機構をよりレベルの高いものにする必要がある．異種分散情報源には，今回の検索対象とした情報源より複雑な構造をしているものもあり，また簡単にオントロジを対応付けることが難しい場合もあると考えられる．これらの問題にも対応できるメタデータ機構にする必要がある．

参考文献

[1] 吉川正俊, ”Web データベースの基盤技術としての XML”, 情報処理, Vol.42, pp.638-642, July.2001.

[2] 吉野利明, 寺本良明, 川村　旭, 益岡竜介, 吉田武俊, ”ファシリテータを利用した商用データベースの統合”, 電子情報通信学会論文誌, D- , Vol.J81-D-I, No.5, pp.460-467, May.1998.

[3] 小西　修, ”異種情報源統合のためのアクティブ・メディエーション・システム － HI-AMS：High Intelligent - Active Mediation System －”, Mem.Fac.Sci.KochiUniv.(Inform.Sci), 17, March.1997.

[4] 村田　真, ”XML[I] － XML Schema と RELAX －”, 電子情報通信学会誌, Vol.84, No.12, pp.890-894, Dec.2001.

[5] J. Hammer, H. Garcia-Molina, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. ”Information Translation, Mediation, and Mosaic-Based Browsing in the TSIMMIS System”. In Exhibits Program of the Proceedings of the ACM SIGMOD International Conference on Management of Data, page 483, San Jose, California, June 1995.

[6] Bayardo Jr.R.J. and et al., ”InfoSleuth: AgentBased Semantic Integration of Information in Open and Dynamic Environments” SIGMOD 97, pp.195-206, May.1997.

[7] Mehmet Altinel, Michael J.Franklin, ”Efficient Filtering of XML Documents for Selective Dissemination of Information”, Proceedings of the 26th International Conference On Very Large Data Bases, pp.53-64, 2000.

[8] Efficiently Publishing Relational Data as XML Documents, Proceedings of the 26th International Conference On Very Large Data Bases, pp.65-76, 2000.

[9] Resource Description Framework(RDF) Model and Syntax Specification
http://www.w3.org/TR/1999/REC-rdf-syntax-19990222

[10] Resource Description Framework(RDF) Schema Specification 1.0
http://www.w3.org/TR/rdf-schema/

[11] Dublin Core Metadata Initiative (DCMI)
http://dublincore.org/