

芸術情報のデジタルアーカイビングにおけるXMLの利用

高橋 悠子† 河村 嘉之‡ 遠山 元道§

†慶應義塾大学大学院 理工学研究科 管理工学専攻 ‡日立ソフトウェアエンジニアリング (株) インターネットビジネス推進部
§慶應義塾大学 理工学部 情報工学科/アートセンター JST 科学技術振興事業団 さきがけ研究 21 「情報と知」領域研究員

神奈川県横浜市港北区日吉 3-14-1 慶應義塾大学情報工学科遠山研究室

045-563-1141 内線 43244

† yuu@db.ics.keio.ac.jp, ‡ kaw@tir.hitachi-sk.co.jp, § toyama@ics.keio.ac.jp

あらまし

近年、芸術アーカイブをデジタル化し計算機上に表すという試みが行われている。ここで、芸術に関する研究に於いて研究作業の効率化のみならず、いまだあまり扱われなかったインターアートの相関など新たな事実を発見する可能性を持っている。しかし、扱われる芸術情報は格納される構造が不均一であり、データの構造が予測しにくいといった特徴をもつ。

本論文では、このような芸術情報の格納に半構造データモデルを適用し、XML を用いて、芸術情報に関するデータベースを作成し、検索システムを提案する。

キーワード

デジタルアーカイブ、半構造データモデル、XML

Using XML for Digital Archiving of Art Information

Yuko Takahashi † Kazuyuki Kawamura ‡ Motomichi Toyama §

†Department of Administration Engineering, Keio University.

‡Hitachi Software Engineering Co.,Ltd., Internet Business Promotion Dept.,

§Department of Information and Computer Science, Faculty of Science and Technology, Keio University, Art Center,
PRESTO, JST

3-14-1, Hiyoshi Kouhoku-ku Yokohama-shi, Kanagawa-ken

+81-45-563-1141 ext.43244

† yuu@db.ics.keio.ac.jp, ‡ kaw@tir.hitachi-sk.co.jp, § toyama@ics.keio.ac.jp

Abstract

A number of the art archives have been digitalized on computers in recent years. It is not only meant for the efficiency of the art studies, but also aimed at the possible discovery of new facts lie between several art areas. However, the structure of each record representing art information is irregular, and it is difficult to foresee a generic structure.

In this paper, we apply semistructured data model to record art information, which is represented in XML. We have implemented a database of the art information and the query.

key words

Digital Archive, Semistructured Data Model, XML

1. はじめに

WWWの普及により、多種多様な情報がネットワーク上で扱われるようになった。そこで、取得できる情報が増加していく中でその情報を管理する手法が求められるようになったが、多様な情報を扱う際に従来の関係データベースなどで用いられている明確な型定義を用いる手法では表現が難しいものが多い。例えば、構造が不規則なデータ集合で、スキーマが無く、非常に緩やかなスキーマしか定義されていないようなデータや、データベース以外の形で蓄積されているため、従来のデータベースのようにあらかじめデータにスキーマが提示されていないデータなどが挙げられる。

このような特徴を持ったデータについて、近年さかんに研究が行われているデータモデルが半構造データモデル(semistructured data model)である。

半構造データモデルに関する研究として、Stanford大学のOEM¹⁾(Object Exchange Model)、Pennsylvania大学およびのAT & T研究所のエッジラベル付きグラフ²⁾などがある。

半構造データを対象とした問い合わせ言語は、OEM上の検索言語として開発されたLorel³⁾、エッジラベル付きグラフモデルの操作言語として開発されたUnQL⁴⁾等が研究されている。また、XMLを対象とした問い合わせ言語については、XML-QL⁵⁾、XQL⁶⁾がある。その他、構造化文書データベースに対する基本的な処理記述言語としてW3QL⁷⁾やX²QL⁸⁾が提案されている。

一方、近年作品やアーティストに関する資料を収集する芸術アーカイブ*をデジタル化し計算機上に表しているという試みがすすめられている。

そこで本論文では、アートに関する研究資料を格納するために適したデータモデルと、このモデルを用いたデータベースの作成を行い、条件に適合したオブジェクトを得るだけでなく、検索結果に構造を定義することが出来る検索言語を開発し、これを用いた検索システムを提案する。

そして、実際に本システムを、慶應義塾大学アートセンターにおける土方 巽¹⁵⁾に関するアーカイブのデジタル化に実際に適用する。

2. 芸術アーカイブ

2.1 Genetic Archive Engine

”芸術アーカイブ”とは、絵画、彫刻などの芸術作品を制作してゆく過程でその作品に関連する作者の手紙

やメモといった、芸術分野の研究者がその作品の制作過程を知る上で重要な資料を収集し学術的に体系化したものである。

現在、慶應義塾大学アートセンターでは、平成10年度から、「創造的デジタルメディアの基礎と応用に関する研究」COEの一環として、「ジェネティックアーカイブエンジン」という、芸術アーカイブをデジタル化するプロジェクトが現在も進行中である。この「ジェネティックアーカイブエンジン」で構成されるアーカイブの基本コンセプトは「ジェネティック」(作品の結果ではなくプロセス、生成過程)であり、土方 巽アーカイブ、イサム・ノグチアーカイブ、稀観書アーカイブの3つのアーカイブが作成されている。

2.2 芸術データの特徴

図1は土方 巽演出の〈バラ色ダンス〉の為に製作されたポスターであるが、あるが、このように、作者、題名、製作年、舞台芸術等の情報に加えて、舞台芸術に関する説明などが含まれる。



図1 芸術データの例

情報の多くは、作者、題名、舞台芸術といった変換することの無い不変情報と、舞台芸術にあるコメントのような可変情報が含まれるが、芸術情報を扱う上でこのような可変情報は特に重要である。この例では舞台芸術に関しての説明が付けられたが、題名を付ける経緯が説明される可能性や、製作年に関するコメントが付けられる可能性もあり、おなじ作品を表していても同様なデータ構造を持つとは限らない。

また、情報の単位決定も芸術情報には重要である。図2の土方 巽の上演作品を写した写真では、この情報を表す際に、写真を上演作品の一部の情報であるという捉え方と、写真と上演作品は別のものと扱い、上演作品に対して写真は上演写真という関係があるという捉え方ができる。しかし、この写真が写真家の細江 英公氏の撮影であったとすると、写真の情報の一部として題材である上演作品があるのと捉えることもできる。

このようにデータを表現する際の特徴としては、その芸術作品を研究する人物の視点によって中心となる対象が変わることがいえる。

* 以下「アーカイブ」と省略する。



図2 データ表現上の特徴

2.3 芸術データの表現方法

代表的なデータモデルとして、関係データモデル、オブジェクト指向データモデル、半構造データモデルが挙げられるが、関係データモデル、オブジェクト指向データモデルは、最初に定義された明確な型に適合しない情報を追加することが困難であるという共通の性質がある。

しかし、半構造データモデルではデータの内部にその構造を記述し、データの構造定義がデータ記述時に行われるため、データを蓄積する前の構造定義を必要としない。このことによって不均一な構造を持つ芸術情報にも柔軟に対応することができる。

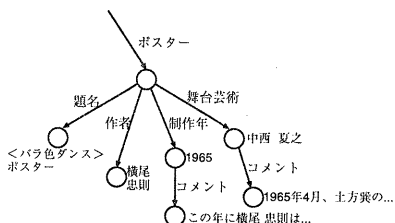


図3 ラベル付き有向グラフを用いた例

ここで、図1で用いた例を、ラベル付き有向グラフ(図3)で表現する。このようにデータ内にその構造を記述することができるため、必要なときにその構造を変更することができる。

そこで、本論文ではデータ記述言語としてXMLを用いて芸術データ(半構造データ)を表現する。

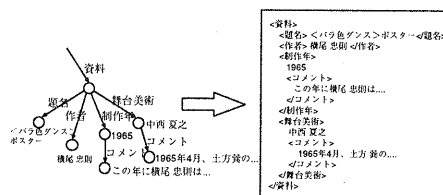


図4 XMLを用いた半構造データの表現

図4で示されるように、グラフ中の各エッジとXML文書内のタグが対応し、エッジに付けられたラベルがタグの要素名となる。XML文書中では、グラフ中の任意の枝に対し枝以下の部分木に対応した要素が、その枝に付けられたラベルと同じ要素名のタグによって

囲まれる。

芸術情報のもう一つの特徴として、その芸術作品を研究する人物の視点によって中心となる対象が変わることが挙げられる。対象となるオブジェクト(公演、写真、新聞記事など)を表現するとき、オブジェクト間の関係に上下をつけて表すのではなく、それぞれを独立したオブジェクトとして表現する。また、オブジェクト間の関係を表すためにそれぞれのオブジェクトを名前をつけたリンクによって結ぶ。

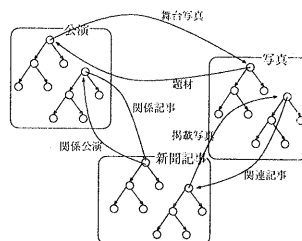


図5 オブジェクト間のリンク

図5では、公演、写真、新聞記事はそれぞれ独立したオブジェクトの集合としてデータベースで管理される。ある公演とある写真の間には、公演から写真に対しては舞台写真、写真から公演に対しては題材というようにそのオブジェクト間の関係を名前つきリンクで表現する。

以上より、以下の二つの情報を記述することによってデータを蓄積していく。

- XMLで表現された個々のオブジェクト
- オブジェクト間の関係を表したリンク

3. システムの構成・実装

3.1 XML文書による芸術データの格納

本論文において、データベースに格納されるデータは、XMLで記述された各オブジェクトの情報と、オブジェクト間を結ぶリンクの情報となる。ここではそれぞれの格納方法について述べる。

3.1.1 オブジェクト情報の格納

オブジェクトの構造は、各エッジに分割して格納する。

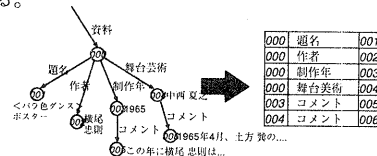


図6 エッジの分割格納

また、エッジの情報に加え、各オブジェクトの情報、

文字列を持つノードの情報、外部ファイルへの参照を持つノードの情報も整理し格納する。

現在の実装では、この分割された情報の格納先として関係データベースを使用する。関係データベース内には以下のようなスキーマを定義し格納する。

- オブジェクト全体の情報

オブジェクト ID	対象名	ファイル名
アクセス許可	root ノード ID	コメント

個々のオブジェクトに対する情報として、そのオブジェクトの ID、そのオブジェクトが属する対象名、そのオブジェクトが記述された XML 文書のファイル名、公開する際に公開かが記述される。root ノード ID は分割されたエッジ情報の root にあたるノードの情報が入る。コメントはリンクを作成する際の補助的な情報である。

- エッジの情報

親ノード ID	ラベル	子ノード ID
---------	-----	---------

オブジェクト内のノードとノードの間を結ぶエッジの情報を表す。

- 文字列を持つノードの情報

ノード ID	データタイプ	実データ
--------	--------	------

オブジェクト内の文字情報を持つノードの情報を表す。XML 文書では全ての情報は文字列になる。しかし、整数、実数などの値も表現するため、付加的な情報としてデータタイプを加えて表現する。

- 外部ファイルへの参照を持つノードの情報

ノード ID	データタイプ	ファイル名
--------	--------	-------

画像ファイルなどの外部に実際のデータがあるノードの情報を表す。データタイプとしては、画像のフォーマットなどが入る。

そして、各スキーマ間の関係は、図 7 のとおりである。

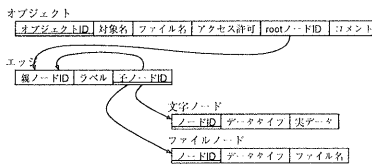


図 7 スキーマ間の関係

3.1.2 リンク情報の格納

リンクの情報も同様に保持していく。実装では、この情報も関係データベースに格納する。

- リンクの情報

参照元オブジェクト ID	ラベル	参照先オブジェクト ID
--------------	-----	--------------

参照元からラベルで表される関係で指し示される参照先へのリンクを表す。

ここで表されるリンクの情報は、参照元から参照先への一方方向のリンクである。双方方向のリンクを表すには、参照元と参照先を入れ換えたリンクを他に生成する。また、ある参照元から参照先への関係が複数あるときは、そのオブジェクト間に持たれる関係の個数のリンクを生成する。

3.1.3 データの保存

芸術データは、関係データベースサーバ、文書サーバ、マルチメディアデータサーバに格納される。

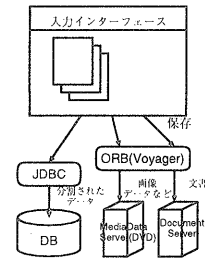


図 8 データの保存

3.2 芸術情報の検索

半構造データモデルで多く用いられる検索は、再構成する検索であるため、本論文でも検索結果として XML 文書を生成する検索を提案する。

3.2.1 検索言語

検索言語について以下の例を用いて説明する。

```
CREATE [a. 制作. 制作年, [a//]]
FROM 公演 a
WHERE a.制作. 演出 = '土方 巽'
ORDER BY a.制作. 制作年;
```

この質問文は、土方 巽が演出した公演の情報の概要を制作年ごとにまとめ、制作年の順番に並べて出力するという意味になる。

CREATE 句と FROM 句は必須であり、WHERE 句と ORDER BY 句は必要に応じて省略可能である。

FROM 句では、選択される対象を指定し、質問文内でその対象を特定するための変数を割り当てる。

CREATE 句では、選択される項目、選択されるオブジェクトと、結果として得られるオブジェクトの構造を指定する。

項目の指定は、その項目が属するオブジェクトの root から取り出す情報のあるノードまでの path を指定する。path の先頭は選択するオブジェクトが属す

る対象に対して FROM 句で割り当てられた変数を指定する。これ以下の path は、その項目に到達するまでに通るエッジに割り当てられたラベルを記述する。

また、オブジェクトの指定は、選択する文書(オブジェクト)が属する対象に対して FROM 句で割り当てた変数を指定する。加えて文書の選択の際にその詳細度を指定することが出来る。ある文書を表す変数 a に対して詳細度の指定方法は以下になる。

a	a で指定される文書の完全な情報を表示
a/	a で指定される文書の主要な情報を表示
a//	a で指定される文書の概略を表示

現在の実装では文書の詳細度を 3 段階で指定することが可能である。この詳細度の指定によって表示される情報の定義は、情報作成者が事前にスタイルシート指定言語を用いて行う。

構造情報の指定は、一つ以上の項目を [と] で括ることによって指定する "a, [b]" と記述することによって、b の項目は a によって群化される。

WHERE 句では、選択されるオブジェクトが適合する条件を指定する。条件は、オペランド、"=", "<", ">" などの比較演算子、AND、OR の論理演算子で構成される。オペランドとして CREATE 句で説明した path による項目指定、リンク参照、オブジェクトの ID 参照、定数を用いることができる。

リンク参照は "a->関係公演" のように記述し、リンク元のオブジェクト (この例では a) から->の後ろで指定された関係 (この例では関係公演) で関連付けられたリンクによって参照されるオブジェクトを表す。

オブジェクトの ID 参照は a# のようにオブジェクトに対し付与された変数に # をつけることによって記述する。

ORDER BY 句では、出力される結果の順番の基準となる項目を指定する。ここも path による項目指定を用いる。

3.2.2 質問処理

処理系では以下のようなプロセスで質問文から結果を生成する。

- (1) 質問文をパースし、構造情報と取得する項目/オブジェクト情報を取り出す。
- (2) 取得する項目/文書情報を SQL 質問文に変換する。
- (3) 変換した SQL 質問文を用いて関係データベースから必要な情報を獲得する。
- (4) 関係データベースから取得した表を構造情報をもとに構造化する。

- (5) 構造化された情報を XML に変換して出力する。

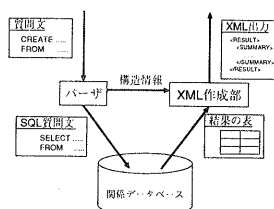


図9 質問処理

ここで、次の質問文を用いて質問処理を説明する。

```

CREATE [a. 制作, タイトル, [b//]]
FROM 公演 a, 作品 b
WHERE a->関係作品 # = b#;
  
```

項目/オブジェクト情報、構造情報

この質問文は、パーザで項目/オブジェクト情報を抜き出して以下の SQL 質問文に変換する。

- 項目/オブジェクト情報
(項目)a. 制作, タイトル, (オブジェクト)b
- 構造情報
[text, [obj]]
(ここで text は項目、obj はオブジェクトを表す。)

SQL 質問文への変換

この質問文は、パーザで項目/オブジェクト情報を抜き出して以下の SQL 質問文に変換する。

```

SELECT i0.value, o1.objid
FROM オブジェクト o0, エッジ n0, エッジ n1,
     文字ノード i0, オブジェクト o1,
     link r0, オブジェクト o2
WHERE o0.dtdname = '公演' AND
      o0.rootnid = n0.pnid
AND n0.label = '制作' AND
      n0.nid = n1.pnid
AND n1.label = 'タイトル' AND
      i0.nid = n1.nid
AND o1.dtdname = '作品'
AND o0.objid = r0.fromnode AND
      r0.relation = '関係作品'
AND r0.tonode = o2.objid AND
      o2.objid = o1.objid
  
```

XML の生成

生成された SQL 質問文を関係データベースで実行

することによって、質問結果としてフラットな表(表1)を得る。

表1 関係データベースから得られる出力

value	オブジェクト ID
四季のための二十七晩	thwork0000
四季のための二十七晩	thwork0001
四季のための二十七晩	thwork0002
四季のための二十七晩	thwork0003
四季のための二十七晩	thwork0004

構造情報

[text, [obj]]

この表と構造情報をもとに、この結果を構造化(表2)する。この例では、二つ目の項目の値は一つ目の項目の値によって群化される。つまり、ここでは「四季のための二十七晩」という情報で以下の「thwork0000」などの情報がまとめられることになる。

表2 構造化された出力

value	オブジェクト ID
四季のための二十七晩	thwork0000
	thwork0001
	thwork0002
	thwork0003
	thwork0004

構造化された情報は、XML へと変換される。

文字列で表される情報は、<TEXT>タグで囲まれ、その内側にその項目の path の最後の要素をタグとして付けて出力される。例えば以下ようになる。

<TEXT><タイトル>

四季のための二十七晩

</タイトル></TEXT>

オブジェクトを表す情報は、その詳細度を表すタグで囲まれ、内側にそのオブジェクトの対応する XML 文書が出力される。詳細度が異なっても、ここで出力される XML 文書は共通である。

出力の変換

質問を実行した結果として XML 文書を得たが、XML には表示情報が含まれないため、スタイルの指定が必要である。本研究では、XML に対するスタイルの指定はスタイル指定言語 XSL¹¹⁾を用いた。

XSL を用いたスタイル指定は以下のように記述する。

```
<xsl:template match="制作年">
  <xsl:value-of expr="attribute(NAME)"/>
  <xsl:text>: </xsl:text>
  <xsl:process-children/>
</xsl:template>
```

```
<xsl:text>年</xsl:text>
```

```
<br>
```

```
</xsl:template>
```

この例では、XML 文書中の要素名が制作年である要素に対して、その属性値「NAME」の値と:をプレフィックスとして出力し、その要素の内容を出力した後で、HTML の改行タグ
を出力することとなる。このスタイル指定を用いることによって、以下のように XML 文書が HTML 文書に変換される。

XML: <制作年 NAME="上演年">1972</制作年>

↓

HTML: 上演年: 1972 年

また、別のスタイル指定を用いることによって XML 文書中から特定の要素のみを変換することも可能である。

XML 文書それぞれに対して必要となる XSL を対象ごとに情報提供側で用意しておく。この XSL を用いて詳細度ごとの出力を調整する。文書変換時には、変換処理に必要な XSL を結合する。

生成された XML と XSL を用いて質問結果を他のメディアへ変換する。この変換にはフランス INRIA 研究所の Koala¹³⁾を用いた。

3.2.3 ナヴィゲーション

情報検索の手法として、3.2.2 節で質問文を用いた方法について説明してきたが、質問文を用いる方法では、質問について指定された条件に対して適合した情報を返す。その他の検索方法として、WWW などで用いられる関連する情報を順次たどり必要な情報を得るナヴィゲーションがある。

アーカイブは、関連する情報を網羅的に管理するため、ナヴィゲーションによる検索が有効であると考えられる。ナヴィゲーションではリンクを辿ることによって関連する情報を取得する。一方、本論文で情報を取得する方法は質問文を実行する方法のみである。そこで、本論文では、ナヴィゲーションを実現する方法として、リンクによって質問文を呼び出す方法を用いることにより、質問文中に他の質問文を呼び出すリンクを埋め込むように質問文を拡張する。

この方法を用いることによって、リンクを辿ることにより、順時質問文が呼び出される。これによって閲覧者は、情報の提供者によって用意された質問群を適時呼び出していくこととなる。

そこで SuperSQL で用いられてきた手法¹⁰⁾を応用し、質問文中に関数を埋め込むことによってリンクの生成を実現する。その関数として、invoke 関数と

linkat 関数を用いる。

invoke 関数

一つの項目/オブジェクトに対して、指定した質問文を実行するリンクを生成する為の関数。

項目/オブジェクト%**invoke**(呼び出す質問文,
条件 1, 引数 1,
条件 2, 引数 2...)

%の左側にくる項目/オブジェクトがリンク元となる。
invoke 関数では、%の左側には項目、オブジェクトと
もにとることができる。しかし、invoke 関数による
指定は、%の左側に指定された項目/オブジェクト全体
がリンク元となるため、オブジェクトを指定するときは
注意が必要である。条件 1 と引数 1、条件 2 と引数
2... はそれぞれペアで扱われ、引数 1 で指定された項
目の値が条件 1 の文字列と連結され、呼び出される質
問文に連言で追加される。この条件と引数のペアは複
数記述可能である。

```
b//%invoke(work.qry, a# = , b#)
```

上の例では質問文 **work.qry** を呼び出すリンクをオ
ブジェクト **b** の概要に対して生成する。質問を呼び出
すとき、質問文の WHERE 句に **a# =** オブジェクト
b のオブジェクト ID が追加される。

linkat 関数

一つのオブジェクト内で、指定した項目に適合する文
字列から指定した質問文を実行するリンクを生成する
為の関数。

オブジェクト%**linkat**(リンク元となる項目,
呼び出す質問文,
条件 1, 引数 1,
条件 2, 引数 2...)

invoke 関数との違いは、invoke 関数は%の左側にく
る項目/オブジェクト全体をリンク元とするが、linkat
関数は左側にくるオブジェクト内にリンクを生成する
ことである。linkat 関数の第一引数は、%の左側で指
定したオブジェクト内でリンク元となる項目の指定と
なる。オブジェクト中でこの項目の値と同じ文字列に
対して、第二引数で指定された質問文を呼び出すリン
クを生成する。

```
a%linkat(b. 名前, person.qry, a# = , b#)
```

この例では質問文 **person.qry** を呼び出すリンク
をオブジェクト **a** の内の **b. 名前** の値にマッチする文

字列上に生成する。質問を呼び出すとき、質問文の
WHERE 句に **a# =** オブジェクト **b** のオブジェクト
ID が追加される。ここで、**b. 名前** の値として、“土方
巽”、“芦川 羊子”など複数の人物が含まれるときはそ
れぞれについてのリンクが作成され、呼び出される先
の質問文の WHERE 句に付加される条件もそれに対
応したものとなる。また、現在の実装では単純な文字
列による適合のため、“土方 巽”と“土方巽”は別のも
のと扱われる。

出力される XML 文書中には、リンク元となる項目
に対して以下のようなコードが追加される。

```
<演出>土方 巽</演出>  
↓  
<link href="http://localhost/servlet/invoker  
?person.qry+%61%23%20%3d+%74%68%70%72">  
<演出>土方 巽</演出>  
</link>
```

このように、リンク先となる要素に対し、<link>タ
グと共にリンク先として指定された URL とその後ろ
に呼び出される質問文、その条件とそれに対応する値
が追加される。

この link タグは、以下のスタイル指定によって、
HTML のアンカータグに変換される。

```
<xsl:template match="link">  
  <a href="{attribute(href)}">  
    <xsl:process-children/>  
  </a>  
</xsl:template>
```

これによって変換された HTML は以下のようにな
る。

```
<a href="http://localhost/servlet/invoker  
?person.qry+%61%23%20%3d+%74%68%70%72">  
  土方 巽  
</a>
```

3.2.4 リンクによる質問文の動的呼び出し

リンクを用いた質問文の呼び出しは、リンク先とし
て、質問文を読みだし処理を行い結果を返すプログラ
ムを指定し、そのプログラムが質問文を実行すること
によって行われている。

生成されるコードによって呼び出されるリンクのリ
ンク先は、そのプログラムの URL であり、その引数
に質問文および条件が与えられる。

質問処理の結果生成された XML 文書はスタイル
シートによって HTML に変換され Web ブラウザに

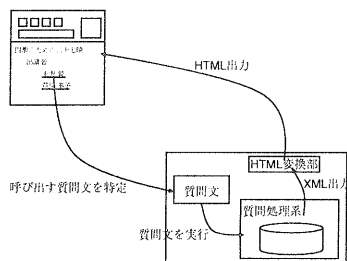


図10 リンクによる質問の呼び出し

表示される。

4. 結 論

本論文の目的として、芸術情報の格納に適したデータモデルを作成し、このモデルを用いたデータベースを作成し、これらを用いた検索システムを提案した。

芸術アーカイブに格納される芸術情報は、格納される情報の構造が不均一であることやデータの構造が特定しにくいといった構造上の特徴と、研究者の視点により作品の捉え方が異なるといった表現上の特徴を持つ。本論文では、半構造データモデルを応用しデータをXMLを用いて記述することによってこのような不均一な情報を表現した。また、作品などの認識できる情報を独立したオブジェクトと捉え、その間を関係を記したリンクで表現することによって様々な研究者が利用することのできる検索システムを作成した。

半構造データモデルなどの自己記述的なデータモデルにおいて、検索結果も構造を持つデータであることは重要である。本論文では、条件に適合したオブジェクトを得るだけでなく、XML生成部でデータの再構成を行うことのできる検索言語を開発し、これを用いた検索システムを作成した。

また、質問言語の拡張によりナビゲーションによる情報検索を可能にした。予め定義された質問を呼び出すことにより検索言語を記述せずに芸術アーカイブ内の情報を得ることが可能になった。ナビゲーションでは、質問文をアーカイブ作成者側で用意するため、質問の呼び出しによるストーリー付けが可能となる。これにより、今まで実現が難しかったアーカイブへの意味付けの実現が期待できる。

また、開発した検索言語は、今後普及することが確実視されているXMLの基づく情報検索ツールとして幅広い応用が可能である。

参 考 文 献

- 1) R. Goldman, S. Chawathe, A. Crespo, J. McHugh. A Standard Textual Interchange Format for the Object Exchange Model (OEM).

- Technical Report, October, 1996.
- 2) P.Buneman, S.Davidson and D.Suciu: Programming Constructs for Unstructured Data, Proceedings of Int. Workshop on DB-PL, electronic Workshops in Computing, '96, pages 506-516, Motreal, 1996
- 3) Jason McHugh, Serge Abiteboul, Roy Goldman, Dallen Quass, Jennifer Widom: Lore: A Database Management System for Semistructured Data. SIGMOD Record 26(3), pp54-66, 1997
- 4) P.Buneman, S.B.Davidson, G.G.Hillebrand and D.Suciu: A Query Language and Optimization Techniques for Unstructured Data, Proceedings of ACM-SIGMOD '96, pages 506-516, Motreal, 1996
- 5) A.Deutsch, M.Fernandez, D.Florescu, A.Levy and D.suciu: A Query Language for XML, Proceedings of the Eighth International World Wide Web Conference (WWW8), Computer Networks, Vol.31, No.11-16, pages 1155-1169, 1999
- 6) J.Robie, J.Lapp and D.Schach: XML Query Language, The Query Languages Workshop, <http://www12.w3.org/TandS/QL/QL98/pp/xql.html>, 1998
- 7) Dabid Konopnicki, Oded Shimueli: Information Gathering in the world wide web: The W3QL Query Language and the Q3QS System: ACM Transactions on Database System, Vol.23, No.4, pages 369-410, December 1999
- 8) 品川 徳秀, 北川 博之, 石川 佳治: 『拡張可能XML 問い合わせ言語 X^2QL とその処理系』, DEWS, 2000
- 9) 田島 敬史: 『半構造データののためのデータモデルと操作言語』, 情報処理学会論文誌, Vol.40, pages 152-170, Feb 1999
- 10) 遠山 元道: 『TFE による HTML 生成質問文における質問分割の効果と等価性』, Proceedings of Advanced Database Symposium '97, pages 87-94, Tokyo, Japan, December 1997
- 11) Extensible Stylesheet Language (XSL) <http://www.w3.org/Style/XSL/>
- 12) Voyager ObjectSpace Inc. <http://www.objectspace.com/products/voyager/index.html>
- 13) the Koala XSL engine for Java <http://www.inria.fr/koala/XML/xslProcessor/>
- 14) The jdbc Data Access API <http://java.sun.com/products/jdbc/>
- 15) 鷲見 洋一, 前田 富士男: ジェネティック・アーカイブ・エンジン〜デジタルの森で踊る土方異〜, 慶應義塾大学アートセンター, 2000