半構造データに対応した例示オブジェクトに基づくWebビュー構築ツールの設計，実装，評価

小泉清一[†1,☆] 森嶋厚行[†2]
高野智[†3,☆☆] 北川博之[†4]

近年，多様な情報源中のデータに基づくWebサイト構築が重要なデータ操作の1つとなっている．この操作では次の2つの処理が必要である．(a) 情報源から必要なデータを抽出する，(b) 抽出したデータを埋め込むためのWebサイトのレイアウトを設計する．本稿では，これら2つの操作をシームレスに統合したWebビュー構築ツールAQUAについて述べる．AQUAはdrag-and-dropスタイルのGUIを持ち，通常のWebサイトオーサリングツールと同様の操作感を提供する．特徴は次の2つの要求を同時に満たすことである．(1) 情報源中の構造化文書のエレメント等の1つをExampleとして指定することにより，object-at-a-timeとset-at-a-time操作を一体化する．(2) XML等の半構造データの操作に対応する．本システムを実装し評価実験を行った結果，本システムは使いやすく，短時間で正確な操作が可能であることが示された．

Design, Implementation, and Evaluation of a Web View Authoring Tool for Semistructured Data Based on Example Objects

Seiichi Koizumi,[†1,☆] Atsuyuki Morishima,[†2] Satoshi Takano[†3,☆☆] and Hiroyuki Kitagawa[†4]

Recently, it has been a matter of great importance to publish the multimedia data objects stored in various information sources. The World Wide Web is often used as publication media. In such context, a crucial point is how to present the data extracted from various information sources. This paper proposes a visual user interface which amalgamates authoring, querying, and restructuring functions for multimedia Web view construction. The user is only required to drag and drop data objects, just like in typical authoring tools for Web pages. The evaluation result shows that our approach realizes intuitive, correct and efficient operations.

1. はじめに

近年，多様な情報源中のデータに基づくWebサイト構築が重要なデータ操作の1つとなっている．それらのデータはしばしば画像等を含むマルチメディアデータである．このデータ操作では次の2つの処理が必要である．(a) 情報源から必要なデータを抽出・再構成する．(b) 抽出したデータを埋め込むためのWebサイトのレイアウトを設計する．既存の枠組みでは，(a) と (b) に対して異なる操作系を採用している．一般に，(a) はset-at-a-timeの操作，(b) はobject-at-a-time操作となる．たとえば，市販されているWebアプリケーション構築ツールの主要な機能の1つは，データベースに対する問合せ結果をWebページとして出力することであるが，そこでは (a) をSQLで記述し (b) はGUIで行う[1]ことが一般的である．また，異種情報源上に様々なWebサイトビューを作成可能な研究レベルのWebサイト管理システムの例としてはStrudel[2]があるが，そこでは (a) を半構造データ操作系StruQLで記述し，(b) をHTMLテンプレート

†1 筑波大学工学研究科
Doctoral Program in Engineering, University of Tsukuba
†2 芝浦工業大学工学部情報工学科
Department of Information Science and Engineering, Shibaura Institute of Technology
†3 筑波大学第三学群情報学類
College of Information Science, University of Tsukuba
†4 筑波大学電子・情報工学系
Institute of Information Science and Electronics, University of Tsukuba
☆ 現在，日本電気（株）
Presently with NEC Corporation
☆☆ 現在，日立ソフトウェアエンジニアリング（株）
Presently with Hitachi Software Engineering Co., Ltd.

を作成することによって行う．

本稿では，これらの (a)，(b) の操作をシームレスに統合した Web ビュー構築ツール AQUA☆の設計，実装，評価について述べる．AQUA は drag-and-drop（以下，D&D）操作を基本とした GUI であり，通常の Web サイトオーサリングツールと同様の操作感を提供する．特徴は次の 2 つの要求を同時に満たすことである．(1) 情報源中の構造化文書のエレメントや RDB の値等（以下オブジェクト）の 1 つを Example として指定することにより，object-at-a-time と set-at-a-time のデータ操作インタフェースを統一する．具体的には，ユーザがマウスを用いて object-at-a-time の「お手本」操作を行うと，システムがユーザの意図を推論し，set-at-a-time の操作に拡張する．これにより，ユーザはレイアウト設計のためのオーサリング操作の感覚で，データ抽出・再構成操作を指定することが可能となる．(2) XML 等の半構造データの操作に対応する．操作対象のデータがリレーショナルデータベースにおけるような静的スキーマで規定された単純で均等な構造を持つ場合は，object-at-a-time の操作をスキーマ構造を手掛かりに対応する set-at-a-time の操作に拡張するのは比較的容易であるが，XML のような半構造データの場合には別の機構が必要である．これを実現するため，Example が代表するオブジェクト集合を，静的なスキーマ情報によらずにグラフ構造をベースに動的に決定する仕組みを持つ．

本稿の構成は次のとおりである．2 章で利用例を説明する．3 章で AQUA の基本概念を説明する．4 章では AQUA の操作を説明する．5 章で利用例の操作手順を示す．6 章でセマンティクスを説明する．7 章で実装について述べる．8 章で関連研究を説明する．9 章で評価実験について説明する．

2. 利 用 例

本章では，AQUA の利用例について述べる．XML 等の半構造データに対応していることから，**図 1** のような異種情報源統合環境の Web ビュー構築ツールとして AQUA を利用することが考えられる．もちろん，他のシナリオも考えられる．たとえば視覚的な XML 操作系として利用することもできるが，本章では図 1 のようなシステムで利用される場合の例について述べる．

例としては，**図 2** のような操作が考えられる．次の情報源が存在すると仮定する．(1) ある RDB に野球選手の現在の成績（打率等）が格納されたリレーション BATTING-STATS（P-Name，Hit，RBI，AVG）がある（図 2 (a)）．(2) 別の RDB（Video DB）には打席ごとの動画オブジェクトとその打席の情報（バッター名，打席の開始時間，終了時間等）を格納したリレーション VIDEO（VID，GID，Begin，End，Batter，Pitcher，Contents）がある（図 2 (b)）．(3) Web 上に，選手リストを含む各チームごとのページ（Team Page）が存在し，そのページから各選手ごとのページ（Player Page）にリンクが張られている（図 2 (c)）．ここで，これらのページは XML で記述されていると仮定し，さらに細部の構造が異なるバリエーションが存在するとする（**図 3**）．

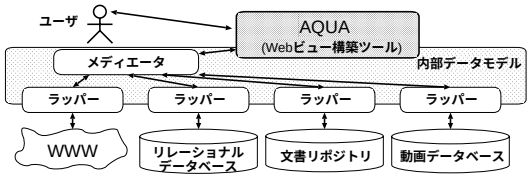

図 1 異種情報源統合環境
Fig. 1 Environment for integration of heterogeneous information sources.

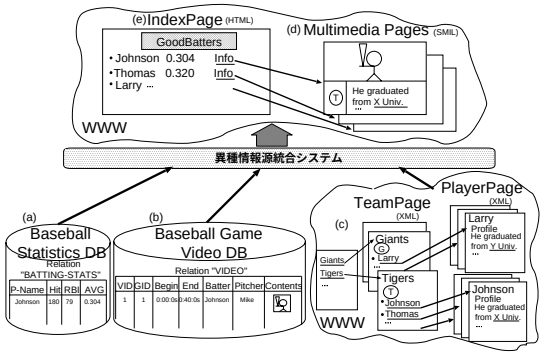

図 2 異種情報源を利用したマルチメディア Web ビュー
Fig. 2 Multimedia Web view on top of heterogeneous information sources.

このとき，「打率が 3 割より上の各選手ごとに打席の映像をまとめたマルチメディアページ（SMIL ページ[4]）を作成する（図 2 (d)）．そしてインデックスページ（HTML ページ）を作成し，そこで各選手ごとに選手名，打率，マルチメディアページへのリンクを示す（図 2 (e)）．インデックスページにはまた，「Good Batters という画像を貼り付ける.」という操作を行いたいとする．

3. 基 本 概 念

AQUA に現れる基本概念は次のとおりである．

☆ Amalgamation of QUerying and Authoring

```
<team>
  <tname>Tigers</tname>
  <logo><img src="TigersLogo.gif"/></logo>
  <players>
    <player ppage="http://..">Johnson</player>
    <player ppage="http://..">Thomas</player>
    ..
  </players>
<team>
```

(a) Tigers の Team page

```
<team>
  <tname>Giants</tname>
  <logo><img src="GiantsLogo.gif"/></logo>
  <players>
    <fielders>
      <player ppage="http://..">Larry</player>
      ..
    </fielders>
    <pitchers>
      <player ppage="http://..">Brian</player>
      ..
    </pitchers>
  </players>
</team>
```

(b) Giants の Team page

図 3 構造が異なる Team Page
Fig. 3 Team Page variations.

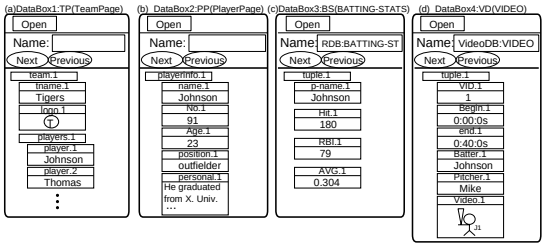

図 4 DataBox の例
Fig. 4 DataBox examples.

Window：次の 3 種類がある．

DataBox 情報源中の Web ページやリレーションのタプル（以下，**ページ**と総称）の集合を表示する．**図 4** は 2 章の例で利用される DataBox である．本稿では，各 DataBox には，Web 問合せ言語等の何らかの手段[5] を利用して，操作の対象となるページ集合がすでに集められていると仮定する．図 4 において，DataBox1（TP）には各チームごとのページ，DataBox2（PP）には各選手ごとのページ，DataBox3（BS）にはリレーション BATTING-STATS，そして DataBox4（VD）にはリレーション VIDEO が格納されている．画面には集められたページ集合中の 1 つのページのみが表示される．DataBox の「Next」「Previous」ボタンを押すことにより他のページをブラウジングすることができる．

Palette Web ページオーサリングのために Canvas に配置可能な汎用の部品が用意されている（**図 5**）．部品としては，HTML のレイアウトに利用する部品（リンクや List Item 要素）のほか，装飾用の画像ファイル等がある．

図 5 Palette の例
Fig. 5 Example of the Palette.

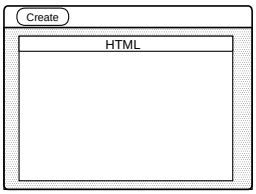

図 6 Canvas 例
Fig. 6 Example of the Canvas.

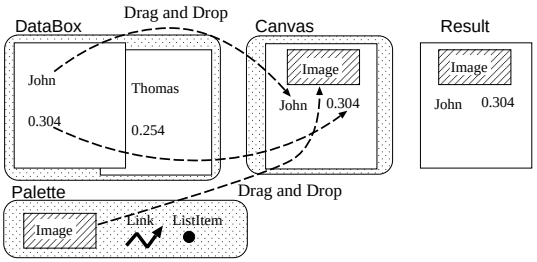

図 7 Drag-and-Drop 操作
Fig. 7 Drag-and-Drop operation and the result.

Canvas ユーザはここに DataBox および Palette からオブジェクトをドロップし，欲しい情報を描画する（**図 6**）．

Drag and Drop（D&D）：DataBox および Palette から Canvas へオブジェクトを D&D し，配置することにより，様々なマルチメディア Web ビューの生成が可能となる．**図 7** に単純な例を示す（図 7〜図 12 では説明のために図を簡略化し，DataBox 中の複数のページを同時に示している）．

D&D オブジェクト：D&D 対象の単位．構造化文書（XML 等）のエレメントや RDB 中の値が該当する☆．

Example：ユーザが，DataBox 中のオブジェクトを「Example である」と指定すると，そのオブジェクト（以下 Example オブジェクト）の D&D 操作は，その Example オブジェクトが代表となるオブジェクト集合に対する操作であると解釈される．

TargetSet：Example オブジェクトが代表するオブジェクト集合である（**図 8**）．情報源中の各ページに

☆ 本章ではオブジェクトと略記する．

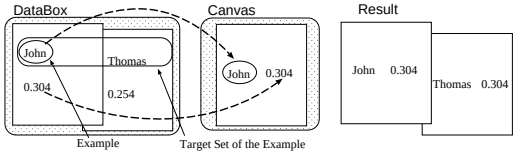

図 8 Example 指定操作
Fig. 8 Manipulation of an Example and the result.

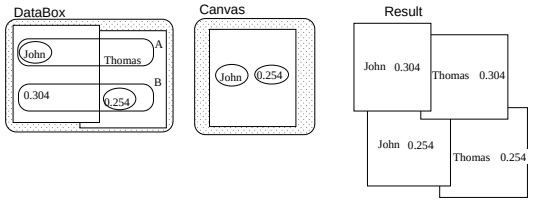

図 9 TargetSet A と B 間に S-Association がない場合
Fig. 9 No S-Association between TargetSets A and B.

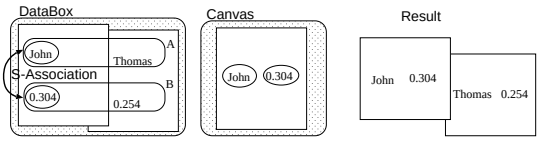

図 10 TargetSet A と B 間に S-Association がある場合
Fig. 10 S-Association between two TargetSets A and B.

おいて，Example オブジェクトと同じ位置にあるオブジェクトの集合がデフォルトの TargetSet となる．TargetSet の変更については 4 章で説明する．

Association：異なる TargetSet 中のオブジェクト間の関連を示す．Association には次の 2 種類がある．

Structural（S-）Association 2 つの Example オブジェクト間の相対的な位置関係により決まる．**図 9** のように 2 つの Example オブジェクトをそれぞれ別のページにおいて指定すると，AQUA は S-Association がないものと解釈し，結果は TargetSet の直積となる．これに対し，**図 10** のように指定すると，「同一ページから抽出されたオブジェクトを組み合わせる」という S-Association が発生し，一種の結合条件として働く．

Value（V-）Association Example オブジェクトの値が同一である場合に決まる．この Association は TargetSet を値により等結合することを意味する．**図 11** のように Example 指定を行うと，AQUA は V-Association がないものと解釈し，結果は TargetSet の直積となる．これに対し，**図 12** のように指定すると，値により等結合された結果が得られる．

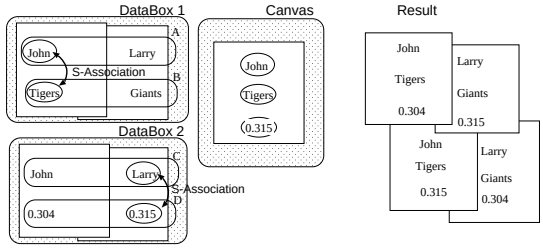

図 11 V-Association がない場合
Fig. 11 No V-Association.

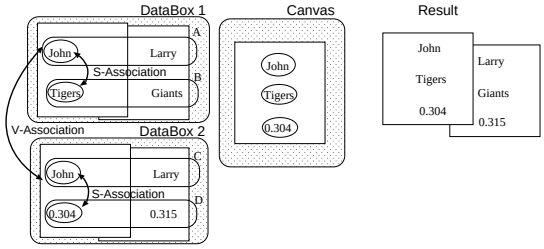

図 12 TargetSet A と C 間に V-Association がある場合
Fig. 12 V-Association between TargetSets A and C.

4. AQUA の操作

AQUA の操作は非常に単純である．すなわち，基本的に Palette および DataBox 中のオブジェクトを Canvas に D&D し，配置するだけである．Example の指定等は，マウスをオブジェクトの上で右クリックすると現れるメニュー（**図 13**）を通じて行う．そこで “Example” を選択することにより，そのオブジェクトが Example として指定できる．図 13 において ‘Johnson’ を Example と指定すると，リレーション BATTING-STATS の属性 P-NAME の値が TargetSet に含まれる．“Another” と “Clue” は TargetSet を変更するために用いられる．

一般的には，AQUA の操作手順は次の正規表現で表される．

(Example 指定 (Another 指定 | Clue 指定)* | D&D)*

直感的には，AQUA の操作は次の操作の任意の組合せである．(1) オブジェクトを D&D する．(2) オブジェクトを Example と指定する．(3) オブジェクトを Example と指定し，その後に Another もしくは Clue 指定を行うことにより，TargetSet を変更する．

Example でないオブジェクトの D&D は，単にそのオブジェクトを Canvas に配置することを示す．Example と指定すると，デフォルトの TargetSet が設定される（情報源中の各ページにおいて，Example オブジェクトと同じ位置にあるオブジェクトの集合がデ

フォルトの TargetSet となる．詳細は 6 章で説明する）．Example オブジェクトに対する操作は TargetSet 中のオブジェクト群の操作を代表した操作と解釈される．別のオブジェクトを Another と指定すると，そのオブジェクト（以下，Another オブジェクト）も含まれるように TargetSet を広げる．また，あるオブジェクトを Clue と指定すると，そのオブジェクト（以下，Clue オブジェクト）を手がかり（条件）として，TargetSet を狭めることができる．Clue を指定すると，そのオブジェクトが満たすべき条件を入力するための Condition Box が現れる．たとえば，**図 14** のように Johnson を Example と指定した後，0.304 を Clue と指定し，“>0.3” という条件を指定することによって，打率が 3 割より上の打者のみが Johnson が代表する TargetSet に含まれる．一般に Clue 指定は以下の条件をすべて満たすオブジェクト o_i のみが TargetSet に含まれるよう，TargetSet を縮小する．

- o_i は元の TargetSet に含まれる．
- o_i に対して，Example オブジェクトと Clue オブジェクトの相対位置関係と同一の位置関係を持つオブジェクト c_i が存在する[6]．
- c_i は Clue に対して指定された条件を満たす．

Example オブジェクトが左クリックにより選択されると，その Example の TargetSet に含まれるオブジェクトが強調表示される．Association は，3 章で述べたように Example オブジェクトの位置や値に応じて自動的に決定される．

通常 AQUA は，オブジェクトの組合せ 1 つに対し 1 ページを生成する（図 9〜図 12 参照）が，Canvas において repetition(“*”) を指定することによりグルーピングを行い，これを変更することができる．本質的には，repetition は入れ子型リレーショナル代数式[7]の Nest 演算子を意味する．グルーピングの利用例は次章中に示す．

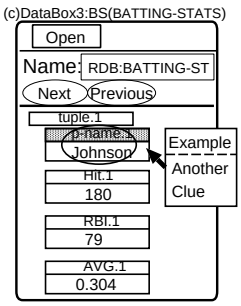

図 13 メニュー
Fig. 13 Menu.

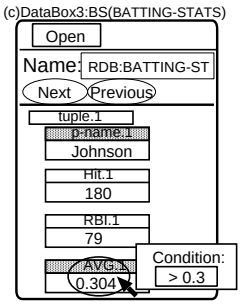

図 14 Condition Box の例
Fig. 14 Example of a Condition Box.

5. 操 作 例

図 2 に示された利用例は次の手順で実行できる．**図 15** の番号は各操作に対応する．

(1) Canvas 中に HTML 文書を作成するウィンドウを開く．
(2) Palette から Image “GoodBatters.gif” を D&D.
(3) Palette から ListItem を D&D する．
(4) 次の手順で全選手を含む TargetSet を作る．TargetSet を作るために，まず TP の “Johnson” を Example と指定する．このとき，DataBox 中の他ページにおいて Johnson と同じ位置にあるオブジェクトの集合が TargetSet TS に格納される．操作中の Example オブジェクトと，それが代表する TargetSet は，それぞれハイライト表示される．この時点ではまだ，Johnson の TargetSet には全選手が含まれない．その理由は，同じページの違う場所にある選手（たとえば Thomas）や，構造が異なるページ

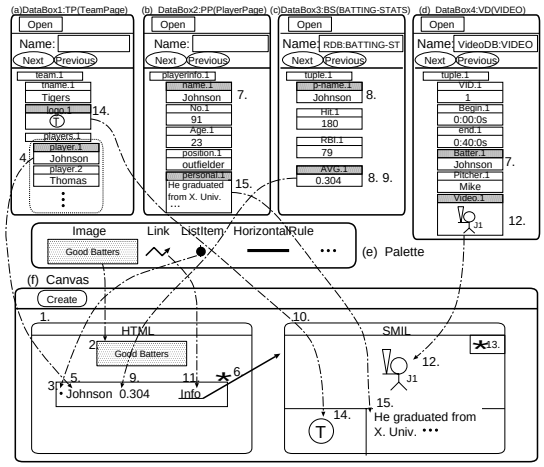

図 15 操作手順
Fig. 15 Specification for the example scenario.

の選手（たとえば Larry）が存在するからである．したがって，続けて “Thomas” を Another と指定し，次のページの “Larry” も Another と指定することによって TS を拡大する．結果的に “Johnson” が代表する TargetSet は全選手名を含むことになる．

(5) Example オブジェクト “Johnson” を D&D する．

(6) ListItem を repetition と指定（$*$ マークを追加）する．これにより，TS 中の選手名が 1 つのページに集められる．もし，この指定がなければ 1 つのページに 1 人の選手名を含んだページが TS 中の選手数分だけ作成されることになる．

(7) PP と VD の “Johnson” を Example とする．(4) の TargetSet を含めて，同一の値を持つオブジェクトを Example とする TargetSet 間には V-Association が生じる．この Association は，これらの TargetSet を等結合することを意味する．

(8) Batting-Stats の “Johnson” を Example とし，“0.304” を Clue とする．Clue の条件を [$>$ 0.3] とする．すなわち，打率が 3 割より上の選手名だけが TargetSet に含まれる．

(9) Batting-Stats の “0.304” を Example とし，D&D する．

(10) Canvas 中に SMIL 文書を作成するためのウィンドウを開く．

(11) Palette から Link を D&D する．このリンクにより，インデックスページから各選手ごとのダイジェスト映像へのリンクを作成する．

(12) VD の  を Example とし，D&D する．

(13)  を repetition とする．これによって各選手ごとの映像が 1 つの連続映像にまとめられる．

(14) TP の㋣を Example とし，D&D する．

(15) PP の “He graduated 〜” を Example とし，D&D する．

最後に Create ボタンを押すと，図 15 の Canvas で指定した HTML および SMIL の Web ページが得られる．

6. セマンティクス

AQUA の操作の意味は，次の 3 段階で定義する．(1) データをオブジェクト木としてモデル化．(2) ユーザの操作を元に，TargetSet と Association を表す Relation（Target Relation と呼ぶ）を作成．(3) Canvas での指定に基づいて Target Relation を変形し，要求されている構造に再構成する．

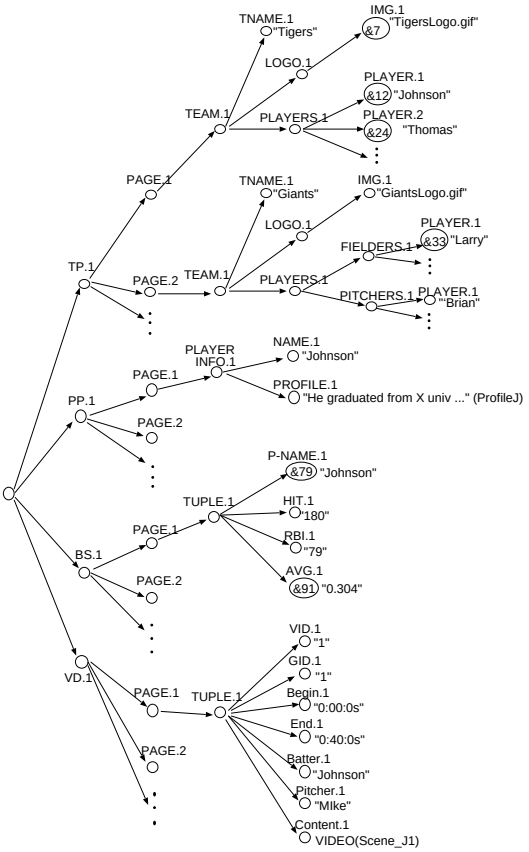

図 16 オブジェクト木
Fig. 16 Object tree.

図 16 は前節で示した操作例のデータに対応するオブジェクト木である．**オブジェクト** はルートオブジェクト，DataBox オブジェクト，ページオブジェクト，D&D オブジェクトのいずれかである．レベル 2 のオブジェクトを DataBox オブジェクト，レベル 3 のオブジェクトをページオブジェクト，レベル 4 以上の深さのオブジェクトを D&D オブジェクトと呼ぶ．ここで，DataBox オブジェクトをルートとした部分木は各 DataBox に対応している．このオブジェクトは ‘*DB*.1’ の形式でラベル付けされており，*DB* は DataBox 名を示す．ページオブジェクトをルートとする部分木は DataBox 中の各ページを示す．このオブジェクトは ‘Page.*i*’ とラベル付けされている．さらに深いオブジェクトは D&D オブジェクトである．これらをルートとする部分木はそれぞれ，XML ページ中の要素もしくはタプルの値と対応している．一

	Original Pred.	$Path(e)$	$Path(a)$	Modified Pred.
Rule 1	$p_1\ B.i\ p_2[o\langle v\rangle]$	$q_1\ B.i\ q_2$	$q_1'\ B.k(\neq i)\ q_2'$	$p_1\ B.?\langle B.i\rangle\ p_2[o\langle v\rangle]$
Rule 2	$p_1\ B.i\ p_2[o\langle v\rangle]$	$q_1\ B.i\ q_2$	$q_1'\ C(\neq B).k\ q_2'$	$p_1\ ?\langle B.i\rangle\ p_2[o\langle v\rangle]$
Rule 3	$p_1\ q_3\ p_2[o\langle v\rangle]$	$q_1\ q_3\ q_2$	$q_1'\ q_4(\neq q_3)\ q_2'$	$p_1? * \langle q_3\rangle\ p_2[o\langle v\rangle]$

図 17 Candidate-Predicate の変更規則
Fig. 17 Rules to modify Candidate-Predicates.

般的にオブジェクトは '*t.i*' とラベル付けされる．ここで，i は同じ親を持つ t が複数存在する場合に，それらの順序を表すために使われる自然数である．各オブジェクトは OID を持つ（&n と表記）．以下では，ルートからオブジェクトへのパスとオブジェクトの値をそれぞれ $path$(&n)，$value$(&n) のように表記する．たとえば，$path$(&12) = TP.1→PAGE.1→TEAM.1→PLAYERS.1→PLAYER.1, $value$(&12) = "Johnson" である．

6.1 Target Set の導出

6.1.1 Clue を持たない Target Set の場合

TargetSet は，Example 指定，Another 指定，Clue 指定により指定される．ここではまず，Clue 指定のない場合について説明する．

Target Set TS_e は以下のように定義される．

$$TS_e = \{o | o \in O \wedge Candidate\text{-}Pred_e(o)\}$$

ここで O は情報源中のすべてのオブジェクトの集合を示し，$Candidate\text{-}Pred_e(o)$ はワイルドカードを持ったパス式である．使用可能なワイルドカードには，同一のタグを持つノードを示す "*Tag*.?"，任意のラベルを持つノードを示す "?"，そして任意の深さのノードを示す "?*" がある．例として 5 章の操作 (4) で指定された $TS_{\&12}$ を示す．

$$TS_{\&12} = \{o | o \in O \wedge \text{TP.1}\rightarrow\text{PAGE.?}\rightarrow\text{TEAM.1}\rightarrow PLAYERS.1\rightarrow\text{?*}\rightarrow\text{PLAYER.?}[o]\}$$

$$= \{\text{"Johnson"}, \text{"Thomas"}, \ldots, \text{"Larry"}, \ldots, \text{"Brian"}, \ldots\}$$

$TS_{\&12}$ は全チームの全選手名の集合となる．

Candidate-Predicates の導出

$Candidate\text{-}Pred_e(o)$ は Example と Another 指定から導出される．この導出過程においては，述語に *annotation* を加えたものを利用する．これは，"⟨" と "⟩" によって囲まれた部分で，$path(e)$ と $value(e)$ を記録している．*annotation* は導出過程で利用するための参考情報であり，追加しても式の意味は同一である．次の式は，式の意味は上の式とまったく同じであるが，$path$(&12) と $value$(&12) の情報を annotation として追加したものである．

(ε は空ノードを示す)

$$TS_{\&12} = \{o | o \in O \wedge \text{TP.1}\rightarrow\text{PAGE.?}\langle\text{PAGE.1}\rangle\rightarrow\text{TEAM.1}\rightarrow\text{PLAYERS.1}\rightarrow\text{?*}\langle\varepsilon\rangle\rightarrow\text{PLAYER.?}\langle\text{PLAYER.1}\rangle[o\langle\text{Johnson}\rangle]$$

導出は次の手順で行われる．

(1) Example オブジェクトを e とする．まず，デフォルトの Candidate-Predicate として $p[e\langle v\rangle]$ が導出される．ここで，p は $path(e)$ の Page.i を Page.?⟨Page.i⟩ で置き換えたもの，v は $value(e)$ である．たとえば，5 章の操作 (4) においてユーザがオブジェクト &12 ("Johnson") を Example と指定したときには，TP.1→PAGE.?⟨PAGE.1⟩ → TEAM.1 → PLAYERS.1→ PLAYER.1[$o\langle Johnson\rangle$] がデフォルトの Candidate-Predicate として導出される．これによって定義される TargetSet が，デフォルトの TargetSet となる．この場合，すべてのページにある Johnson とまったく同じ位置にあるすべてのオブジェクトの集合となる．

(2) Another 指定が行われるたび Candidate-Predicate を変更し，指定されたオブジェクト a が TargetSet に含まれるようにする．変更規則を**図 17** に示す．各規則は，$path(e)$ と $path(a)$ を利用していかに Candidate-Predicate を変更するかを示している．ここで，B，C はラベル名を示し，p_i は元のパス式の一部である．q_i は $path(e)$ の一部であり，q_i' は $path(a)$ のうち p_i にマッチする部分である．基本的なアイデアは，$path(e)$ と $path(a)$ が一致しない場所に，ワイルドカードを挿入するということである．たとえば 5 章の操作例手順 (4) においてユーザがオブジェクト&24 ("Thomas") を最初の Another として指定したとき，$path(a)$=TP.1→PAGE.1→ TEAM.1→PLAYERS.1→PLAYER.2 となる．ここで，$path(e)$ = TP.1→PAGE.1→ TEAM.1→ PLAYERS.1→ PLAYER.1 と照合すると，PLAYER のラベル番号が異なっているので，Rule 1 が適用される．ここで，p_1 = TP.1→PAGE.?⟨PAGE.1⟩ →TEAM.1→ PLAYERS.1, q_1 = q_1' = TP.1→ PAGE.1→ TEAM.1 →PLAYERS.1, p_2 = q_2 = q_2' = ε となり，変更後の Candidate-Predicate は TP.1→PAGE.?⟨PAGE.1⟩→TEAM.1→ PLAYERS.1→PLAYER.?⟨PLAYER.1⟩

[o⟨Johnson⟩] となる．同様に，&33（“Larry”）が Another として指定されたときには Rule 3 が適用され，最終的に上記の $TS_{\&12}$ が得られる．

図 17 のルールには曖昧性があるため，複数のマッチング解釈がありうる場合がある．本システムでは付録 A.1 のアルゴリズムを用いてこの解決を行う．

6.1.2 Clue を持つ Target Set の場合

Clue が指定されたときの Target Set は以下のように定義される．

$$TS_e = \{o | o \in O \wedge Candidate\text{-}Pred_e(o) \wedge \exists c \in O(Clue\text{-}Pred_{cl}(c) \wedge SA\text{-}Pred_{e,cl}(o,c))\}$$

$Clue\text{-}Pred_{cl}(c)$ はワイルドカードを持ち，かつ $value(c)$ に対して条件が課されたパス式である．$SA\text{-}(S-Association)Pred_{e,cl}(o,c)$ は $Share_p(o,c)$ の形式で表され，o，c 間の相対的な位置関係を示す．ここで p はパス式である．例として 5 章の操作 (8) で指定された $TS_{\&79}$ を示す．

$$TS_{\&79} = \{o | o \in O \wedge p\text{→P-NAME.1}[o\langle\text{Johnson}\rangle] \wedge \exists c \in O(p\text{→AVG.1}[c > 0.3] \wedge Share_p(o,c))\}$$

where p = BS.1→PAGE.?⟨PAGE.1⟩→TUPLE.1.

“BS.1→PAGE.?⟨PAGE.1⟩→TUPLE.1→AVG.1[$c > 0.3$]” が Clue-Predicate であり，$value(c)$ には 0.3 より上という条件が課せられている．$Share_p(o,c)$ は $path(o)$ と $path(c)$ が部分パス式 “BS.1→PAGE.?→TUPLE.1” を共有することを示している（**図 18** 参照）．ゆえに，$TS_{\&79}$ には打率が 0.3 より上の選手のみが含まれる．

$$(\forall o_i \forall o_j \in X) Share_{BS.1\rightarrow PAGE.?\rightarrow TUPLE.1}(o_i, o_j)$$

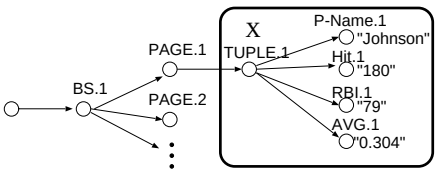

図 18 オブジェクト集合 X
Fig. 18 Object set X.

6.2 Target Relation

Target Relation は，Example による Target Set とそれらの間の Association を示す．

n 個の Clue を持たない Target Set が存在し，m 個の Clue を持つ Target Set が存在し，Target Set 間の Association が l 個存在すると仮定する．このとき，Target Relation は以下のように定義される．

$$\begin{aligned} TR = \{&(value(o_1), \ldots, value(o_{n+m})) \mid \\ &o_1 \in O \wedge Candidate\text{-}Pred_{e_1}(o_1) \\ &\wedge \ldots \\ &\wedge o_n \in O \wedge Candidate\text{-}Pred_{e_n}(o_n) \\ &\wedge o_{n+1} \in O \wedge Candidate\text{-}Pred_{e_{n+1}}(o_{n+1}) \\ &\wedge \exists c_1 \in O(Clue\text{-}Pred_{cl_1}(c_1) \\ &\wedge SA\text{-}Pred_{e_{n+1},cl_1}(o_{n+1}, c_1)) \\ &\wedge \ldots \\ &\wedge o_{n+m} \in O \wedge Candidate\text{-}Pred_{e_{n+m}}(o_{n+m}) \\ &\wedge \exists c_m \in O(Clue\text{-}Pred_{cl_m}(c_m) \\ &\wedge SA\text{-}Pred_{e_{n+m},cl_m}(o_{n+m}, c_m)) \\ &\wedge Association\text{-}Pred_1(o_{a_1}, o_{b_1}) \wedge \ldots \\ &\wedge Association\text{-}Pred_l(o_{a_l}, o_{b_l}))\} \end{aligned}$$

ここで Association-Pred は SA-Pred か VA-(V-Association) Pred によって示される．SA-Pred は $Share_p(o,o')$ の形式で示され，VA-Pred は $o \overset{v}{=} o'$ の形式で示される．$o \overset{v}{=} o'$ は $value(o) = value(o')$ を意味する．

例

次式は操作例から導出された Target Relation を示す．括弧内の数字は操作例における操作の順序を示す．式 (1) によって**図 19** の Target Relation が生成

Ex. Johnson in TP	Ex. (T) in TP	Ex. Johnson in PP	Ex. ProfileJ in PP	Ex. Johnson in BS	Ex. 0.304 in BS	Ex. Johnson in VD	Ex. 웃J1 in VD
Johnson	(T)	Johnson	ProfileJ	Johnson	0.304	Johnson	웃J1
Johnson	(T)	Johnson	ProfileJ	Johnson	0.304	Johnson	웃J2
Thomas	(T)	Thomas	ProfileT	Thomas	0.320	Thomas	웃T
Larry	(G)	Larry	ProfileL	Larry	0.315	Larry	웃L1
Larry	(G)	Larry	ProfileL	Larry	0.315	Larry	웃L2

図 19 Target Relation 例
Fig. 19 Target Relation.

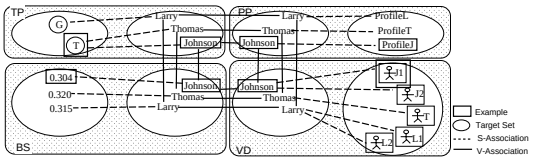

図 20 TargetSet と Association（TargetSet 中のオブジェクトは一部のみ）
Fig. 20 TargetSets and Associations.

Ex. Johnson in TP	Ex. 0.304 in BS	V Ex. J1 in VD	Ex. T in TP	Ex. ProfileJ in PP
Johnson	0.304	J1	T	ProfileJ
		J2		
Thomas	0.320	T	T	ProfileT
Larry	0.315	L1	G	ProfileL
		L2		

図 21 入れ子型リレーション例
Fig. 21 Result nested relation.

される．

$$
\begin{aligned}
&\{(value(o_1),\\
&\ldots, value(o_8))|o_1 \in O \wedge p_1 \rightarrow \text{PLAYERS.1} \rightarrow ?^*\langle\varepsilon\rangle\\
&\quad \rightarrow \text{PLAYER.?}\langle\text{PLAYER.1}\rangle[o_1\langle Johnson\rangle]^{(4)}\\
&\quad \wedge o_2 \in O \wedge p_1 \rightarrow \text{LOGO.1} \rightarrow \text{IMG.1}[o_2\langle \textcircled{T}\rangle]^{(14)}\\
&\quad \wedge o_3 \in O \wedge p_2 \rightarrow \text{NAME.1}[o_3\langle Johnson\rangle]^{(7)}\\
&\quad \wedge o_4 \in O \wedge p_2 \rightarrow \text{PROFILE.1}[o_4\langle ProfileJ\rangle]^{(15)}\\
&\quad \wedge o_5 \in O \wedge p_3 \rightarrow \text{P} - \text{NAME.1}[o_5\langle Johnson\rangle]^{(8)}\\
&\qquad \wedge \exists c_1 \in O(p_3 \rightarrow \text{AVG.1}[c_1 > 0.3]^{(8)}\\
&\quad \wedge Share_{p_3}(o_5, c_1)^{(8)})\\
&\quad \wedge o_6 \in O \wedge p_3 \rightarrow \text{AVG.1}[o_6\langle 0.304\rangle]^{(9)}\\
&\quad \wedge o_7 \in O \wedge p_4 \rightarrow \text{BATTER.1}[o_7\langle Johnson\rangle]^{(7)}\\
&\quad \wedge o_8 \in O \wedge p_4 \rightarrow \text{CONTENT.1}[o_8\langle \boxed{J1}\rangle]^{(12)}\\
&\quad \wedge Share_{p_1}(o_1, o_2)^{(14)} \wedge Share_{p_2}(o_3, o_4)^{(15)}\\
&\quad \wedge Share_{p_3}(o_5, o_6)^{(9)} \wedge Share_{p_4}(o_7, o_8)^{(12)}\\
&\quad \wedge o_1 \overset{v}{=} o_3^{(7)} \wedge o_1 \overset{v}{=} o_7^{(7)} \wedge o_1 \overset{v}{=} o_5^{(8)}\}
\end{aligned}
$$

where

p_1 = TP.1→PAGE.?⟨PAGE.1⟩→TEAM.1,

p_2 = PP.1→PAGE.?⟨PAGE.1⟩→PLAYERINFO.1,

p_3 = BS.1→PAGE.?⟨PAGE.1⟩→TUPLE.1, and

p_4 = VD.1→PAGE.?⟨PAGE.1⟩→TUPLE.1.

図 20 は操作例中のすべての Target Set と S-または V-Association を表している．図 19 は図 20 に対応した Target Relation を表す．

6.3 入れ子構造の生成と再構築操作

次に，Canvas から得られる grouping 指定に基づいて Nest 演算と Projection 演算を適用し，Target Relation を変形する．5 章の例では，次により**図 21** の入れ子型リレーションが得られる（ただし，TR は図 19 のリレーションを表す）．

$$Nest_{V=(\boxed{J1})}(Projection_{Johnson,0.304,\boxed{J1},\textcircled{T},ProfileJ}(TR))$$

最後に，その結果を Canvas で指定されたレイアウトにマッピングする（**図 22**）．

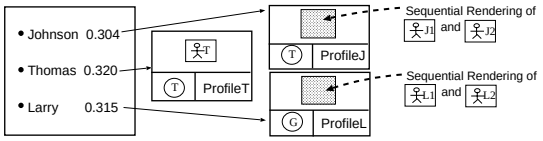

図 22 操作結果
Fig. 22 Result Web view.

7. プロトタイプシステムの実装

AQUA プロトタイプシステム実装を行った．システムの構成を**図 23** に示す．

本プロトタイプシステムの実装には Java（Java2 SDK 1.3）を用い，Drag and Drop API を利用した．コードサイズは約 21,000 行である．すべての情報源のページはラッパーを利用して XML にあらかじめ変換される．RDB 中のリレーションは 1 タプルずつ XML 文書に変換することによって取り扱われる．

AQUA はユーザの操作を基にした Target Relation を生成するための **Target Relation Specification** と grouping 指定やレイアウト情報を持つ **Result Specification** を基に，XML 操作問合せを作成し，異種情報源統合環境へと渡す．

異種情報源統合環境では AQUA から渡された XML 操作問合せを基に結果を生成し，AQUA に返す．最後に，AQUA はその結果を Result Specification で指定されるテンプレートに埋め込むことにより，ユーザの要求する Web ビューを生成する．

本プロトタイプシステムはほぼすべての機能を実装しているが，一部以下の制約がある．(1) Palette が未実装，(2) Drag&Drop の対象が末端の XML 要素のみ，(3) Canvas の Drop 位置が固定．

7.1 AQUA の構成

AQUA の主要なモジュール構成を図 23 の上部に示す．今回の実装においては Palette モジュールは実装せず，作成する Web ビューのテンプレートをあらかじめ用意することで対応している．

各モジュールの機能は以下のようになっている．

DataBox：`DataBox` は前述の DataBox を実装したモジュールであり，DataBox ウィンドウの表示と Tar-

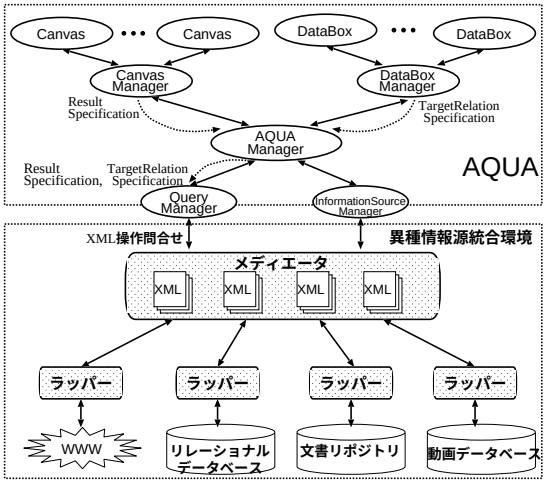

図 23 システム構成
Fig. 23 System architecture.

get Relation Specification の一部（TargetSet に対応する部分）の生成を担当する．ページを表示する領域，指定された Example に関する情報をリスト表示する領域，情報源をブラウジングするボタンを配置したツールバー，の 3 つのコンポーネントから構成される．1 つの情報源に対し 1 つの `DataBox` が生成され，そのウィンドウ内でのイベント（ユーザによる Example 指定等）に対応して部分式を生成し，それによる TargetSet を強調表示する．

DataBoxManager：`DataBoxManager` は `DataBox` を管理するモジュールである．1 つの `DataBox` において複数の Example が指定された場合は，Example オブジェクト間の S-Association を導出し，異なる `DataBox` において同じ値が Example 指定された場合は，V-Association を導出する．これらの Association と `DataBox` で生成された部分式より，Target Relation Specification を生成する．

Canvas：`Canvas` は前述の Canvas を実装したモジュールであり，Canvas ウィンドウの表示と grouping 情報，レイアウト情報の生成を行う．

CanvasManager：`CanvasManager` は `Canvas` を管理するモジュールである．各 `Canvas` で指定された grouping・レイアウト情報に `Canvas` 間のリンク情報を加え Result Specification を生成する．

AQUA Manger：`AQUA Manager` は本システムのメインモジュールであり，システム全体を管理する．主な機能は次の 2 つである．

- `InformationSourceManager` よりユーザの操作対象となる情報源のページ集合へのハンドラを受け取り，`DataBoxManager` へと渡す．これにより，情報源のページ集合が `DataBox` を通じてユーザに表示される．
- Web ビュー生成時に，`DataBoxManager` で生成された Target Relation Specification と `CanvasManager` で生成された ResultSpecification を `QueryManager` へ渡す．そして `QueryManager` から戻る結果に基づいて最終的な Web ビューを生成する．

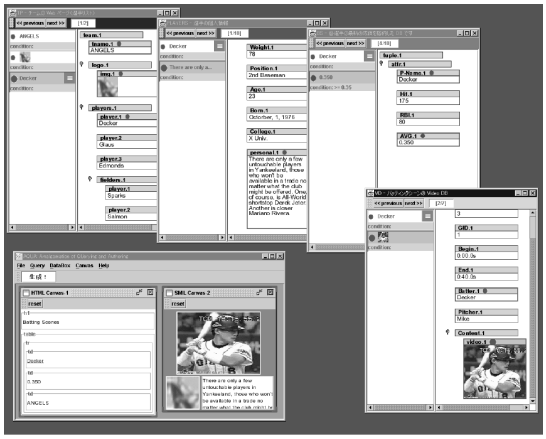

図 24 プロトタイプシステムの実行画面
Fig. 24 A screen shot from the prototype.

InformationSourceManager：
`InformationSourceManager` はユーザの操作対象となるページ集合へのアクセスを提供するモジュールであり，異種情報源統合環境における各情報源のページ集合へのハンドラを生成し，管理する．

QueryManager：`QueryManager` は `AQUA Manager` から受け取った Target Relation Specification と ResultSpecification から異種情報源統合環境の XML 操作問合せを作成し，メディエータへ渡す．そしてメディエータから戻る結果を `AQUA Manager` へ渡す．

利用する異種情報源統合環境に合わせて `InformationSourceManager` と `QueryManager` を実装することにより，AQUA を様々な異種情報源統合環境上で利用することが可能である．

図 24 はプロトタイプシステムの実行画面を示している．

8. 関連研究

本質的に AQUA は，例示オブジェクトに基づく GUI によって，Web ページオーサリング操作，DB 出版のためのデータ抽出・再構成操作，および XML（半構造データ）操作の融合を実現したものであるといえる（**図 25**）．以下に関連する研究について説明する．

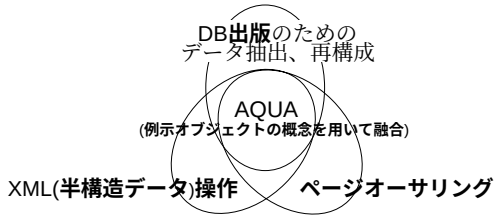

図 25 AQUA の位置付け
Fig. 25 Summary of AQUA's feature.

「DB 出版」システム：これらは，情報源中のページ上に，Web サイトや他の形式のビューを作成するものである．いずれも，(a) 情報源からのデータ抽出操作系と，(b) レイアウト指定操作系から構成されるが，既存システムではそれぞれ別の操作体系を利用する．Strudel[2] は (a) を半構造データ操作系 StruQL で記述し，(b) を HTML テンプレートを作成することによって行う．Delaunay[8] は，(b) に対しては D&D インタフェースを提供し，(a) に関しては SQL 風の問合せ言語利用する．RBE[9] では，(a) を GUI ウィジェットの D&D で行う．この GUI ウィジェットは，データベース中のデータを表示するコンポーネントである．(b) は，domain variables を用いて，そのウィジェットとデータの対応付けを指定することによって行う．Tiramisu[3] は Web サイト管理システムである．このシステムでは，FrontPage のような既存のオーサリングツールを用いて (b) を行う．(a) は，site-schema という図を用いて指定する．Tiramisu の基本概念では，Web サイトの論理設計と外観の設計を明確に切り分けており，(b) に対して他のツールを利用できるというのが特徴の 1 つである．SuperSQL[10] と SQL+D[11] は SQL に出力レイアウトの指定を加えた問合せ言語である．これによって (a) および (b) をともに指定する．これらは問合せを視覚的に示す枠組みを持たない．

Web オーサリングツール：HTML，SMIL のページ設計を行うオーサリングツールは多数存在する．本ツールの操作は，HTML オーサリングツールの DreamWeaver[12] のオーサリング操作系を参考に設計した．オーサリングツールの中には，データベースとの連携ができるものもある．たとえば，FrontPage[13] は ASP 形式の Web ページに SQL を埋め込むことでそれを実現する．

半構造データ，XML データ操作系：近年 XML をはじめとした半構造データの操作系が活発に研究されている．代表的なものとして，XQuery[14]，XQL[15]，XML-QL[16] 等がある．XML-GL[17] はグラフ構造をベースにした視覚的 XML 問合せ言語である．これらは結果のレイアウトという概念はないが，XML データの操作ができるという点で AQUA と強く関連する．しかし本質的に異なる点が 1 つある．それは，これらの言語では，パターンマッチに基づく問合せを「ユーザが記述する」のに対し，AQUA では，ユーザの「お手本」操作から問合せを「システムが推論する」ということである．

Query-by-Example パラダイム："Example" の概念は QBE[18] において最初に示された．入れ子構造の生成に関しては STBE[19] と RBE において示されている．これらが構造化されたデータのみを対象としているのに対し，DataGuide[20]，HQBE[5] は半構造データも対象としている．DataGuide は半構造データの抽象データ構造を示し，HQBE は Web，RDB，構造化文書等のビューを生成する．ただし，本ツールにおける Example は QBE 等におけるそれとは，(1) 具体的なオブジェクト自身であること，(2) ユーザ操作に基づき Example が表すドメインが動的決定されること，の 2 点でまったく異なる．

上記 (1) の特徴は，AQUA の使いやすさに貢献していると同時に次の問題点も持つ．すなわち，データ量に対して Example となるオブジェクトが少数であるとき，Example を発見するのが困難なことである．これに対しては文献 21) で，実データによる例示操作系に検索機能を追加する機構が説明されている．この機能は，QBE で利用されるような「人工的な例」の指定を許可することにより実現される．

Programming-by-Demonstration システム：例示による操作指示は，Programming by Demonstration の分野で広く研究されている．我々の知る限り半構造データ操作系に関する研究は存在しないが，本研究に関連する研究としては次のようなものがある．まず，問合せ生成に関するものとして，DADIE[24] がある．これは，RDB データのドラッグアンドドロップ操作から SQL 問合せを生成する．RDB 等の構造データは完全に規則的な構造を持つため，この枠組みでは XML 等の半構造データの操作は不可能である．Web ビュー生成に関連するものとしては，Internet Scrapbook[25] がある．これはドラッグアンドドロップ操作によって Web 版 "スクラップブック" を作成するものである．ポイントは，データそのものを保持するのではなく，Web ページからデータを抽出するためのパターンを作成することによって，Web の更新に対応することである．パターンは HTML に特化したヒューリスティクスを用いて作成される．そのパター

	CUI	GUI
パターンマッチ指定に基づく操作	XML-QL	XML-GL
例示オブジェクトを用いた操作		AQUA

図 26 実験 A における操作系間の関係
Fig. 26 Relationship among frameworks.

ンで対応できない更新に出会った場合，ユーザとの対話により，より適切なパターンに切り替える．ここでは AQUA のような set-at-a-time 操作の推論は行われない．

9. 評価実験

AQUA を用いたデータ操作に関する次の 2 つの評価実験を行った．(**実験 A**) 情報源中の情報，特に半構造データに対する操作に関する評価を行った．具体的には，同一の XML データに対する同一のデータ操作を，XML-QL，XML-GL，AQUA で記述し，比較した☆．これらの操作系間の関係は**図 26** のようになる．(**実験 B**) AQUA を用いてある程度複雑な Web ビューを構築した場合の使用感に関する評価実験を行った．

これらの実験の被験者としては，データベース初心者グループ（6 名，以下 A グループと呼称），データベースの基礎的知識を有するグループ（6 名，以下 B グループと呼称），データベースと XML-QL の基礎的知識を有するグループ（6 名，以下 C グループと呼称）の 3 グループ，計 18 名を対象とした．

9.1 実験 A（比較評価実験）

9.1.1 実験方法

比較評価実験として，XML 文書の再構成操作を AQUA，XML-GL，XML-QL で記述し，比較する試験を行った．本実験にあたっては，AQUA プロトタイプ処理系および AT&T が配布している XML-QL 実装システム[26] を利用した．XML-GL については実験時点で我々が利用可能な XML-GL 実装システムが存在しなかったため，市販の描画ツールを用いて XML-GL 操作に必要なコンポーネントを用意し，ドラッグアンドドロップで操作を記述できる環境を用意した．すべての操作系において操作記述条件を同一とするために，本試験においては操作の記述を終えた時点で解答終了とし，操作の実行は許可しなかった．

実験にあたっては各システムに対して同程度の解説資料を用意し，それぞれまず 20 分間のチュートリアルを行った後，試験を行った．

☆ StruQL も半構造データの操作が可能であるが，XML-QL と共通の概念も多く，XML データを使った実験は XML-QL の方が向いていると考えられるため，XML-QL のみを採用した．

表 1 実験 A の結果（問題 1）
Table 1 Result of Experiment A (question 1).

	A group	B group	C group	全被験者
AQUA	1 分 12 秒 (6/6)	36 秒 (5/6)	40 秒 (6/6)	49 秒 (17/18)
XML-GL	1 分 14 秒 (5/6)	1 分 1 秒 (6/6)	1 分 7 秒 (6/6)	1 分 7 秒 (17/18)
XML-QL	3 分 20 秒 (5/6)	1 分 14 秒 (6/6)	1 分 53 秒 (6/6)	2 分 9 秒 (17/18)

表 2 実験 A の結果（問題 2）
Table 2 Result of Experiment A (question 2).

	A group	B group	C group	全被験者
AQUA	1 分 27 秒 (5/6)	55 秒 (6/6)	55 秒 (6/6)	1 分 6 秒 (17/18)
XML-GL	3 分 17 秒 (5/6)	2 分 21 秒 (6/6)	3 分 3 秒 (6/6)	2 分 53 秒 (17/18)
XML-QL	5 分 43 秒 (3/6)	2 分 55 秒 (4/6)	2 分 56 秒 (6/6)	3 分 51 秒 (13/18)

表 3 実験 A の結果（問題 3）
Table 3 Result of Experiment A (question 3).

	A group	B group	C group	全被験者
AQUA	1 分 2 秒 (5/6)	46 秒 (6/6)	49 秒 (6/6)	52 秒 (17/18)
XML-GL	4 分 5 秒 (5/6)	3 分 58 秒 (5/6)	3 分 25 秒 (5/6)	3 分 49 秒 (15/18)
XML-QL	5 分 23 秒 (6/6)	2 分 40 秒 (5/6)	2 分 26 秒 (6/6)	3 分 30 秒 (17/18)

試験における操作対象としては，実論文データの XML 文書（DBLP[27] および SIGMOD Record[28] の XML 文書の一部）を利用した．これらは，タグ名に共通するものがあるが，構造が部分的に異なるため，全体としては半構造データを構成することになる（付録 A.1.1）．各システムにおいて次の問題群を出題した（付録 A.1.2）．

問題 1 単一項目のデータ抽出操作
問題 2 複数項目のデータ抽出操作
問題 3 抽出したデータのグルーピング操作

試験を行う操作系の順番が評価に影響を与える可能性があることを考慮し，被験者ごとに 3 つの操作系の試験の順序を変更した．それぞれ，1 問ごとに解答に要した時間の測定，正誤判定を行うが，解答の正答，誤答については被験者に通知しなかった．また，試験中の参考資料としてチュートリアル時の解説資料の閲覧を許可した．本試験終了後に使いやすさと使いたさの順序に関するアンケート調査を行い，正答率，解答に要した時間とあわせて比較評価の尺度とした．

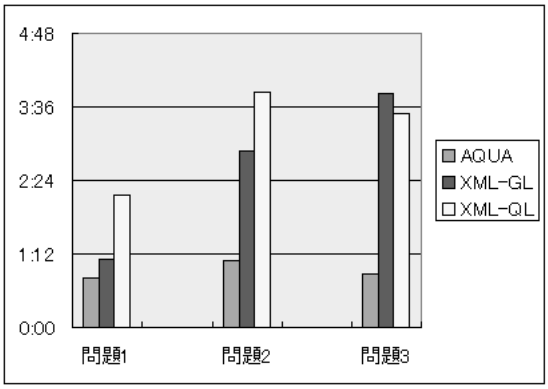

図 27 平均解答時間の比較
Fig. 27 Comparison in average answer time.

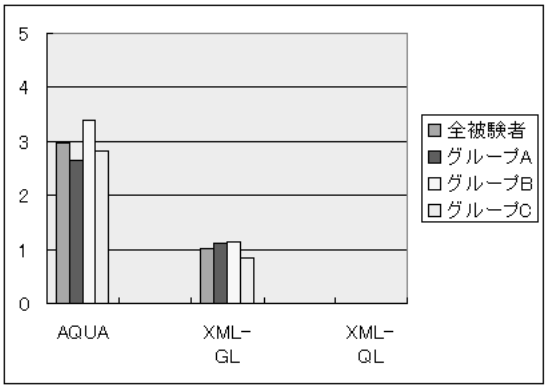

図 28 使いやすさの比較
Fig. 28 Comparison in easiness.

9.1.2 実 験 結 果

解答時間 解答時間は，操作記述時の「書きやすさ」を表す 1 つの尺度であると考えられる．この実験における各問題の平均解答時間を**表 1**，**表 2**，**表 3** に示す．括弧内は（正解者数/被験者数）を示す．問題 1，2 については，全被験者の平均解答時間は，XML-QL>XML-GL>AQUA の順に短くなっている（**図 27**）が，問題 3 については XML-QL の解答時間が XML-GL のものより短くなっている．この原因は，B，C グループの解答時間が長くなっているためである．一方，A グループについては XML-QL の時間の方が長い．これは，XML-GL によるグルーピング操作指定が SQL 等の方法と比べてかなり異なる形式であるために，B，C グループの被験者に戸惑いがあったためと思われる．解答時間に関する操作系間の有意差を検定するために，利用する操作系を要因とする分散分析と tukey の多重範囲検定を行った．その結果，AQUA と XML-GL 間，AQUA と XML-QL 間のそれぞれにおいて有意水準 1%の有意差が示された．これは，AQUA の解答時間が他の操作系より明らかに短いことを示している．

正答率 正解率は，「理解しやすさ」を表す 1 つの尺度であると考えられる．各システムごとの正答率に注目すると，AQUA>XML-QL>XML-GL の順となっている．全体的に正答率が高いのは，試験問題の難易度を低めに設定したためだと思われる．なお，今回の実験では，XML-QL と XML-GL における単純なスペルミス等の記述ミスのみによる不正解は正解としている．

相対的な使いやすさ AQUA，XML-GL，XML-QL それぞれの相対的な使いやすさの印象を調査するため，「本試験において XML-QL の使いやすさを “0” とした場合，AQUA，XML-GL の使いやすさをそれぞれ −5 以上 5 以下の実数で表してください．数字が大きいほど使いやすいことを示します」というアンケートを比較実験終了後に行った．相対的な使いやすさに関しては，AQUA>XML-GL>XML-QL という順序となっている（**図 28**）．相対的な使いやすさの値に関するグループ間と操作系間の有意差を，解答時間と同様の手法で検定した．その結果，AQUA と XML-GL 間，AQUA と XML-QL 間のそれぞれにおいて有意水準 1%の有意差が示された．これは，AQUA と XML-GL 間，AQUA と XML-QL 間の使いやすさの差が大きいことを示している．また，グループ間においては有意差が示されなかった．これは，いずれのグループに関しても AQUA の評価が高かったことを示している．

また，実験 A 程度の難易度における各システムの使いたさの順序を調査するため，「今回の試験問題のような課題を解く状況を想定した場合，各システムの使いたさの順序を教えてください」というアンケートを比較実験終了後に行った．以下にその結果を示す．

AQUA>XML-GL>XML-QL 6 名 (A:4 B:1 C:1)
AQUA>XML-QL>XML-GL 5 名 (A:0 B:4 C:1)
XML-QL>AQUA>XML-GL 4 名 (A:1 B:0 C:3)
XML-GL>AQUA>XML-QL 2 名 (A:1 B:1 C:0)
XML-QL>XML-GL>AQUA 1 名 (A:0 B:0 C:1)

これらの結果より，AQUA が最も好評を博していることが示された．

AQUA に対して「使いにくい」という印象をいだいた被験者の特徴は，次のように分類される．

(1) SQL 等のスキーマベースの問合せ言語を使い

表 4 AQUA 高度利用実験結果
Table 4 Result of Experiment B.

	A group	B group	C group	全被験者
問題 1	7 分 7 秒 (4/6)	6 分 58 秒 (6/6)	7 分 30 秒 (6/6)	7 分 12 秒 (16/18)

慣れており，「インスタンスベースの例による操作」に抵抗がある（1 名）

(2) XML-QL を使い慣れている（2 名）

(3) GUI よりもキーボード主体の操作を好む（1 名）

9.2 実験 B（高度利用実験）

9.2.1 実 験 方 法

次に，AQUA を利用してより高度な XML 文書の再構成（Web ビュー構築）操作を行う試験を行った．前述の比較評価実験終了後，AQUA の高度な利用法についてさらに 20 分間のチュートリアルを行った．試験では，2 章の利用例とほぼ同等の試験問題☆を出題した．

この XML 文書の一部と試験問題を付録 A.2 に示す．本試験では操作の実行を許可し，本人が正しいと思う結果が得られるまでに要した時間とアンケートを評価の対象とした．実験 A と同様に，試験中の参考資料としてチュートリアル時の解説資料の閲覧を許可した．

9.2.2 実 験 結 果

実験 B の解答時間に関するグループ間の有意差検定を行った．その結果，グループ間における有意差は示されなかった．これは，AQUA がどのグループに対しても同程度の「書きやすさ」を提供しているということを示している．

本試験終了後，「この試験問題程度の難易度の課題を解く状況を想定した場合，AQUA を使ってみたいと思うか」というアンケートを行った．試験問題の難易度が高かったにもかかわらず，18 名中 17 名の被験者から「使ってみたい」という回答を得た．この実験の平均解答時間を**表 4** に示す．

9.3 実験のまとめ

実験 A，B に関する考察：実験 A の結果を各システムごとに見てみると，正答率においては有意な差は見られなかったが，解答時間，相対的な使いやすさの値，使いたさの順序という各項目において AQUA が優れた値を示している．今回の比較評価実験の条件のもとでの AQUA の優位性は，AQUA の持つ以下の特長によるものと考える．

(1) DataBox に表示されている実際のデータを利用した操作指定と，その操作に対する応答を通じたインタラクティブ性が，操作系の直感的理解を促進する．

(2) 操作系固有の新たな概念の学習の必要がほとんどない．たとえば，パターンマッチのための正規表現等をユーザが直接意識する必要がない．

実験 A の結果をグループごとに見てみると，各グループ間の差異は明確でなかった．これは，実験 A の試験問題の難易度が低めだったことに起因すると思われる．

実験 B 終了後の被験者のコメントによると，「作りたい Web ビューのイメージがあれば作れる」「操作方法がうろ覚えでも操作できる」というように AQUA の操作感が直感的であることが示されている．

実験 A，B を通じて，試験時に参考資料としてチュートリアル時の解説資料の閲覧を許可したが，AQUA に関しては閲覧されることがほとんどなかったにもかかわらず，他システムと比較して同等以上の結果を示している．これは，AQUA の操作感が直感的であるためにユーザの理解を促したものと思われる．

しかし，被験者の多くが「複雑な操作を行った後の確認が大変」という印象を受けている．ユーザが行った操作を簡潔な形式でユーザに提示する機能の追加が必要であると考えられる．

AQUA に関する考察：難易度の低い課題（実験 A）に関しては，特に解答時間と使用感において高い評価が得られた．マルチメディアデータに対する操作を含む課題（実験 B）に関しては，かなり難易度が高くても高い正答率を得られることが示された．これらより，実験 B 程度までの難易度の操作は AQUA に向いていることが示された．また，データベースに関する知識の有無にかかわらず，高い評価が得られた．

10. お わ り に

本稿では，異種情報源統合環境におけるマルチメディア Web ビュー構築ツールの設計，実装，評価について述べた．Example の概念をドラッグアンドドロップに基づいた操作系に導入することにより，本ツールはオーサリングツールと同様の object-at-a-time の視覚的なページ設計操作と，set-at-a-time のデータ抽出操作を一体化した枠組みを提供する．また，Example が表すオブジェクト集合をグラフ構造をベースに動的に決定する仕組みを持つことにより，情報源の半構造性に対応した．

☆ 以下の 2 点が利用例と異なる．(1) Palette がない，(2) 打率ではなく年齢で選手を選択する．

本システムを実装し，評価実験を行った．具体的には，AQUA，XML-QL，XML-GL を比較対象とした XML 文書の操作に関する比較評価，AQUA を用いた複雑なマルチメディア Web ビューの構築に関する評価実験の 2 種類を行った．その結果，解答時間，正答率，使いやすさの各項目において高い評価が得られた．

今後の課題としては，(1) 動画，音声等連続メディアの再生範囲指定等，マルチメディアのより高度な操作への対応，(2) 操作の履歴表示等，有効性評価実験の結果を元にした改良，(3) Example として適当なオブジェクトを発見しやすくするための機構の導入，等があげられる．

謝辞 本研究の一部は，文部省科学研究費 基盤研究（B)(12480067）の助成による．実験用映像素材の使用を許可いただいた日本テレビ放送網（株）に感謝いたします．

参 考 文 献

1) Oracle Portal, Oracle Corporation homepage. http://www.oracle.com/
2) Fernández, M., Florescu, D., Kang, J., Levy, A. and Suciu, D.: Catching the Boat with Strudel: Experiences with a Web-Site Management System, *Proc. ACM SIGMOD '98*, pp.414–425 (1998).
3) Anderson, C.R., Levy, A.Y. and Weld, D.S.: Declarative Web-Site Management with Tiramisu, *Proc. ACM SIGMOD Workshop on the Web and Databases* (*WebDB '99*) (1999).
4) W3C, Synchronized Multimedia Integration Language (SMIL) 1.0 Specification, W3C Recommendation (1998). http://www.w3.org/TR/REC-smil
5) 森嶋厚行，北川博之：視覚的操作系による異種情報源統合利用支援，電子情報通信学会論文誌，Vol.J82-D-I, No.1, pp.315–326 (1999).
6) Morishima, A., Koizumi, S. and Kitagawa, H.: Drag and Drop: Amalgamation of Authoring, Querying, and Restructuring for Multimedia View Construction, *Proc. 5th IFIP 2.6 Working Conference on Visual Database Systems* (*VDB5*), pp.257–276 (2000).
7) Fischer, P.C. and Thomas, S.J.: Operators for Non-first-normal-form Relations, *Proc. IEEE COMPSAC83*, pp.464–475 (1983).
8) Cruz, I.F. and Lucas, W.T.: Automatic Generation of User-Defined Virtual Documents Using Query and Layout Templates, *Theory and Practice of Object Systems*, Vol.4, No.4, pp.245–260 (1998).
9) Kishnamurthy, R. and Zloof, M.: RBE: Rendering By Example, *Proc. ICDE '95*, pp.288–297 (1995).
10) Toyama, M.: SuperSQL: An Extended SQL for Database Publishing and Presentation, *Proc. SIGMOD '98*, pp.584–586 (1998).
11) Baral, C., Gonzalez, G. and Son, T.C.: Design and Implementation of Display Specifications for Multimedia Answers, *Proc. ICDE '98*, pp.558–565 (1998).
12) Macromedia Dreamweaver, Macromedia Corporation homepage. http://www.macromedia.com/
13) Microsoft Corporation, Microsoft Homepage. http://www.microsoft.com/
14) W3C, XQuery 1.0: An XML Query Language, W3C Working Draft (2001). http://www.w3.org/TR/xquery/
15) Robie, J., Lapp, J. and Schach, D.: XML Query Language (XQL), *The Query Languages Workshop* (*QL '98*) (1998). http://www12.w3.org/TandS/QL/QL98/pp/xql.html
16) Deutsch, A., Fernandez, M., Florescu, D., Levy, A. and Suciu, D.: A Query Language for XML, *Proc. 8th International World Wide Web Conference* (*WWW8*), Computer Networks, Vol.31, No.11–16, pp.1155–1169 (1999).
17) Ceri, S., Comai, S., Damiani, E., Fraternali, P., Paraboschi, S. and Tanca, L.: XML-GL: A Graphical Language for Querying and Restructuring XML Documents, *WWW8/Computer Networks*, Vol.31, No.11-16, pp.1171–1187 (1999).
18) Zloof, M.: Query By Example, *IBM Sys. J.* (1977).
19) Özsoyoglu, G., Matos, V. and Özsoyoglu, Z.M.: Query Processing Techniques in the Summary-Table-by-Example Database Query Language, *ACM TODS*, Vol.14, No.4, pp.526–573 (1989).
20) Goldman, R. and Widom, J.: DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases, *Proc. VLDB '97*, pp.436–445 (1997).
21) Morishima, A. and Kitagawa, H.: Design and Evaluation of an Example-based Graphical Manipulation Framework for XML, *Proc. Int'l Conf. on Web Information Systems Engineering* (*WISE2001*), pp.222–231 (2001).
22) Flickner, M., et al.: Query by Image and Video Content: The QBIC System, *IEEE Computer*, Vol.28, No.9, pp.23–32 (1995).
23) Ishikawa, Y., Subramanya, R. and Faloutsos, C.: MindReader: Querying Databases Through

Multiple Examples, *Proc. VLDB '98*, pp.218–227 (1998).
24) 杉浦　淳，古関義幸：例示によるデータベース問合せ文作成，インタラクティブシステムとソフトウェア III，pp.191–198, 近代科学社 (1995).
25) Sugiura, A. and Koseki, Y.: Internet Scrapbook: Automating Web Browsing Tasks by Demonstration, *Proc. ACM Symposium on User Interface Software and Technology*, pp.9–18 (1998).
26) XML-QL homepage. http://www.research.att.com/~mff/xmlql/doc/
27) DBLP homepage. http://www.informatik.uni-trier.de/~ley/db/about/dallas-usage.html
28) SIGMOD Record homepage. http://www.acm.org/sigmod/record/xml/
29) Kuntz, M. and Melchert, R.: Pasta-3's Graphical Query Language: Direct Manipulation, Cooperative Queries, Full Expressive Power, *Proc. VLDB '89*, pp.97–105 (1989).

付　　録

A.1 Candidate Predicate 導出アルゴリズムの詳細

ここでは，Candidate Predicate 導出アルゴリズムの詳細について説明する．特に，本アルゴリズムが，ルール適用時に生じる曖昧性をいかに解決するかを説明する．まず，説明を簡単にするため，パスを構成するラベルの番号を省略し，パス e_i をタグ名の列 $e_i = t_1 t_2 \ldots$ で表す．このとき，次のような曖昧性が生じる．たとえば，パス $p_1 = path(e_1) = t_a t_b t_c t_d$ と $p_2 = path(p_2) = t_a t_c t_b t_d$ が与えられたとき，ルール 3 で生成される述語は $t_a?*t_c?*t_d$ もしくは $t_a?*t_b?*t_d$ の可能性がある．

本システムでは Another Example e が指定されたびに，**図 29** の手続きを呼び出す．まず，1 行目と 2 行目で前処理を行う．P は，ある TargetSet を指定するためにこれまでに与えられた example（もしくは another example）へのルートからのパスの集合である．1 行目では，最新の $p_{latest} = path(e)$ が追加される．2 行目で作成される P' は P とほぼ同じであるが次が異なる．すなわち，必ずしも P 中のすべてのパスに現れないタグを特殊なタグ '.'（ピリオド）に置き換えたものである．ピリオドタグは，いずれのタグ（ピリオドを含む）ともマッチしない．以下ではピリオドタグ以外のタグを通常タグと呼ぶ．

```
P:{p_i|p_i は現在までに与えられた Example
もしくは Another Example へのパス }
#tags(pred): 述語 pred 中に存在するピリオドでない
タグの数
1.  P.add(p_latest);
2.  P' = P.ChangeUncommonTagsToPeriods();
3.  pred =default_pred(p'_latest);
4.  for each p'_i in P' {
5.      pred' =ApplyARule(pred, p'_i);
6.      if (!#tags(pred') < #tags(pred)) pred = pred';
7.  }
8.  pred = pred';
```

図 29　Candidate Predicate 導出手続き
Fig. 29　Procedure to derive Candidate Predicates.

Candidate Predicate の導出は，P ではなく P' に対して行われる．この前処理は，最終的に導出される述語には影響を及ぼさない．根拠は次が成立するからである．

性質 1　P から推論された Candidate Predicate が通常タグ t を含む $\Rightarrow \forall p_i \in P$（$p_i$ が t を含む）

ピリオドタグを導入した結果，性質 2 と 3 が保証できる．

性質 2　$\exists p'_i \in P'$（p'_i が通常タグ t を含む）$\Rightarrow \forall p'_i \in P'$（$p'_i$ が t を含む）

性質 3　Candidate Predicate の変更ルールを適用したとき，Candidate Predicate 中の通常タグが消え，ワイルドカードに変更される $\Leftrightarrow$ ルールの解釈が複数ある．

性質 3 が成立する理由は，次のとおりである．複数のパスをマッチしたとき通常タグ t が消えるのは，他のパスに対応する通常タグが存在しない場合のみであるが，性質 2 よりすべてのパスに同じタグが含まれているはずである．したがって，t がマッチングするような別の解釈が必ず存在する．また，いずれの通常タグも消えない場合は，かならず解釈は 1 通りしか存在しない．

性質 3 より，通常タグが消えたかどうかによって，複数の解釈が存在するかどうかを判定することができる．同時に，マッチングによって間違った解釈が行われた場合には，本来消えるべきタグ（上記の例では t_b もしくは t_c のいずれか）が必ず残る．間違えた場合に余計なタグが残ることから次がいえる．まず，本来欲しい Predicate を $Pred$ とし，$T = \{path(e) | Pred(e)\}$ とする．曖昧なルールを用いてシステムが間違って推測した Predicate を $\hat{Pred}$ とし，$\hat{T} = \{path(e) | \hat{Pred}(e)\}$ とする．このとき，$\exists p \in T - \hat{T}$ が成立する．

本アルゴリズムの 3 行目で，3 章で説明したデフォルトの TargetSet を求める．5 行目でルールの適用を行う（複数ある場合は任意の解釈が選ばれる）．6 行目

では，通常のタグがワイルドカードに変更されなかった場合（すなわち，あいまいな解釈が存在しない場合）のみ，変更を有効とする．8 行目は，あいまいな解釈がを引き起こすパスが存在する場合には，1 つだけそれを適用することを保証している．

システムはこのように生成された Candidate Predicate に基づき定義された TargetSet をユーザにハイライトして表示する．ユーザは Another を指定することにより再び図 29 の手続きを呼び出し述語の修正を行うが，そのとき，先ほど適用したルールの解釈の曖昧さが解決される．ここでポイントは，「間違った解釈の余地」が排除されることである．まず定義により，Another Example e 現在推測された TargetSet に含まれないものであるので，次が成立する．

$$path(e) \in T - \hat{T}$$

先の例では次のようになる．$P = t_a? * t_c? * t_d$ とする．間違って $\hat{P} = t_a? * t_b? * t_d$ が推測されたとする．このとき，Another として指定されるものは必ず $T - \hat{T}$ に含まれるので，「$t_a? * t_c? * t_d$」にはマッチするが「$t_a? * t_b? * t_d$」にはマッチしない．

したがって，この最新の Another 指定によって得られたパスは t_b を含まないので，2 行目により t_b はピリオドタグに変換され，$t_a * t_b * t_d$ のマッチングが選ばれる可能性が除去される．

A.2 比較評価実験に用いたデータと試験問題

A.2.1 デ ー タ

DBLP の XML 文書例

（1 文書につき 1 論文データ，2 文書を用意）

```
<article key="SuciuT97">
  <author>Dan Suciu</author>
  <author>Val Tannen</author>
  <title>A Query Language for NC.</title>
  <pages>299-321</pages>
  <year>1997</year>
  <volume>55</volume>
  <journal>JCSS</journal>
  <number>2</number>
  <url>db/journals/jcss/jcss55.html#SuciuT97</url>
</article>
```

SIGMOD Record の XML 文書例

（1 文書につき 3 論文データ，1 文書を用意）

```
<SigmodRecord>
 <issue>
  <volume>26</volume>
  <number>3</number>
  <articles>
   <article>
    <title>
     A Query Language for a Web-Site Management System.
    </title>
    <initPage>4</initPage>
    <endPage>11</endPage>
    <authors>
     <author position
       ="00">Mary F. Fernandez</author>
     <author position
       ="03">Dan Suciu</author>
     ...
    </authors>
   </article>
  </articles>
 </issue>
 ...
</SigmodRecord>
```

A.2.2 試 験 問 題

試験問題 1 全論文について，論文のタイトル（title）を載せたページを 1 枚ずつ作成せよ．

試験問題 2 SIGMOD Record に載っている論文のタイトル（title）と著者の名前（author）をそれぞれ 1 つずつ載せた HTML ページを作成せよ．

試験問題 3 SIGMOD Record に載っている論文のタイトル（title）と著者の名前（author）を抜き出し，その著者（author）ごとにタイトル（title）をまとめた HTML ページを作成せよ．

A.3 高度利用実験に用いたデータと試験問題

A.3.1 デ ー タ

野球チームページの XML 文書例

（2 文書を用意）

```
<team>
  <tname> 読売ジャイアンツ </tname>
  <logo src="logo.gif"></logo>
  <players>
    <player> 仁志　敏久 </player>
    <player> 松井　秀喜 </player>
    ...
    <infielders>
      <player> 元木　大介 </player>
      ...
    </infielders>
    <outfielders>
      <player> 清水　隆行 </player>
      ...
    </outfielders>
    ...
  </players>
</team>
```

選手ページの XML 文書例

（18 文書を用意）

```
<player>
  <name> 仁志　敏久 </name>
  <profile>
    <No.>8</No.>
    <Bats> 右 </Bats>
    <Throws> 右 </Throws>
    <Height>171</Height>
    <Weight>77</Weight>
    <Position> 内野 </Position>
    <Age>29</Age>
    <Born>1971 年 10 月 4 日 </Born>
```

```
    <College> 早大 </College>
    <Comment> 昨年は打率 3 割をかけて ...</Comment>
  </profile>
</player>
```

各打席の映像を格納した XML 文書例（10 文書を用意）

```
<tuple>
  <attr>
    <VID>3</VID>
    <GID>1</GID>
    <Begin>0:00.0s</Begin>
    <End>0:40.0s</End>
    <Batter> 仁志　敏久 </Batter>
    <Pitcher> 黒田 </Pitcher>
    <Content>
      <video src="1003CG1b.rm"/>
    </Content>
  </attr>
</tuple>
```

A.3.2 試 験 問 題

試験問題 25 歳以上の選手について，選手名のリストとチームの名前を載せた HTML ページを各チームごとに作成せよ．また，各選手について，その選手の打席の映像をまとめたもの，名前，所属しているチームのロゴ，をのせた SMIL ページを作成し，HTML ページの選手名からその選手の SMIL ページへのリンクを作成せよ．

(平成 13 年 9 月 10 日受付)
(平成 14 年 9 月 5 日採録)

小泉　清一（正会員）

1999 年筑波大学第三学群工学システム学類卒業．2001 年同大学大学院工学研究科修士課程修了．現在，日本電気（株）勤務．視覚的問合せ言語，ユーザインタフェース，XML 等に興味を持つ．

森嶋　厚行（正会員）

1993 年筑波大学第三学群情報学類卒業．1998 年同大学大学院工学研究科修了．博士（工学）．1998〜2001 年日本学術振興会特別研究員．1999〜2000 年 AT&T 研究所客員研究員．現在，芝浦工業大学工学部情報工学科講師．異種情報源統合，データ利用のためのユーザインタフェース等に興味を持つ．ACM，IEEE-CS，電子情報通信学会各会員．

高野　智

2001 年筑波大学第三学群情報学類卒業．現在，日立ソフトウェアエンジニアリング（株）勤務．オブジェクト指向プログラミング，XML，分散コンピューティングに興味を持つ．

北川　博之（正会員）

1978 年東京大学理学部物理学科卒業．1980 年同大学大学院理学系研究科修士課程修了．日本電気（株）勤務の後，1988 年筑波大学電子・情報工学系講師．同助教授を経て，現在，筑波大学電子・情報工学系教授．理学博士（東京大学）．異種情報源統合，文書データベース，WWW の高度利用等に興味を持つ．著書「データベースシステム」（昭晃堂），「The Unnormalized Relational Data Model」（共著，Springer-Verlag）等．電子情報通信学会，日本ソフトウェア科学会，ACM，IEEE-CS 各会員．