

異種情報源統合環境における例示オブジェクトを用いた マルチメディア Web ビュー構築ツールの設計と実装

5A-3

小泉 清一[†] 森嶋 厚行^{††} 佐藤 朗^{†††} 北川 博之^{††}

[†] 筑波大学 工学研究科 ^{††} 筑波大学 電子・情報工学系 ^{†††} 筑波大学 第三学群 情報学類

〒 305-8573 茨城県つくば市天王台 1-1-1

筑波大学 電子・情報工学系 データベース研究室

Tel (0298)53-5385 Fax (0298)53-5051

e-mail {izumi, mori, satou, kitagawa}@dblalab.is.tsukuba.ac.jp

近年、様々な情報源中のデータを Web ページとして出力することが重要なデータ処理の一つとなっている。これらのデータはしばしば画像などを含むマルチメディアデータである。この処理においては、情報源から抽出されたデータを、どうレイアウトし表示するかといったページ設計が大きな問題となる。既存の枠組のほとんどでは、情報源に対する問合せ等のデータ抽出操作とページ設計操作を異なる様式で行なっている。本稿では、これらの操作をシームレスに統合した Web ビュー構築ツールの設計と実装について述べる。具体的には、情報源中のデータをドラッグアンドドロップして配置することにより、情報源からのデータ抽出操作においても、通常の Web ページオーサリングツールと同様の操作感を提供する。本研究の特徴は、次の二点である。(1) データオブジェクトを Example として指定することにより、通常のオーサリングツールと同様の object-at-a-time のページ設計操作と、set-at-a-time のデータ抽出操作を一体化している。(2) 情報源の異種性や半構造化に対応するため、Example が表すデータ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。これによって、ユーザはマウスを用いたインスタンススペースの操作を行うだけで、システムが操作の意味を推論し、ユーザの要求に合った問合せを生成することを目指す。

Design and Implementation of a Multimedia Web View Authoring Tool Based on Example Objects for Heterogeneous Information Integration Environments

Seiichi Koizumi[†] Atsuyuki Morishima^{††} Akira Satou^{†††} Hiroyuki Kitagawa^{††}

[†]Doctoral Program in Eng., Univ. of Tsukuba ^{††}Institute of Info. Sci. and Elec., Univ. of Tsukuba

^{†††}College of Information Sciences, Univ. of Tsukuba

Database Lab., Inst. of Info. Sci. and Elec., Univ. of Tsukuba

1-1-1, Tennoudai, Tsukuba City, Ibaraki Pref. 305-8573, Japan

Tel +81-298-53-5385 Fax +81-298-53-5051

e-mail {izumi, mori, satou, kitagawa}@dblalab.is.tsukuba.ac.jp

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 first feature is that the user can designate an existing data object as an example. Manipulation of an example is interpreted as manipulation of a set of data objects. Therefore, the object-at-a-time authoring framework and the set-at-a-time data extraction framework are integrated in a seamless way. The second feature is that our framework can cope with heterogeneity of information sources and semi-structured data.

1. はじめに

近年、様々な情報源に格納されているデータを Web ページとして出力することが重要なデータ処理の一つとなっている。それらのデータは、しばしば、画像、音声などを含むマルチメディアデータである。研究レベルおよび実用レベルにおいて、そのような処理を行うシステムが数多く存在する。例えば、Web サイト管理システム¹⁾²⁾では、異種情報源上に様々な Web ビューを作成可能である。また、実用化されている Web アプリケーション構築ツールの主要な機能の一つは、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。これらでは、情報源に対するデータ抽出操作と、それによって得られるデータをどのようにレイアウトし表示するかというページ設計の二点が大きな問題となる。既存の枠組みの多くは、情報源からのデータ抽出とページ設計を異なる操作系を用いて支援している。例えば、ページ設計は視覚的操作系で行い、RDB に対する問合せ指示は SQL を埋め込むといった具合である。

本稿では、我々が開発している異種情報源統合システム (図 1) における Web ビュー構築ツールの設計と実装について述べる。本ツールにおける基本操作は、各情報源中のデータオブジェクトをドラッグアンドドロップ (D&D) する事である。その際、情報源中の既存のデータオブジェクトを Example として指定することにより、通常の HTML や SMIL³⁾ ページオーサリングツールと同様の object-at-a-time のページ設計操作と、set-at-a-time のデータ抽出操作を一体化している。また、情報源の異種性や半構造化に対応するため、Example が表すデータ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。これによって、ユーザはマウスを用いたインスタンススペースの操作を行うだけで、システムが操作の意味を推論し、ユーザの要求に合った問合せを生成することができる。これらの特徴を有する本ツールにより、各種情報源上に様々なマルチメディア Web ビューを構築することが可能になる。

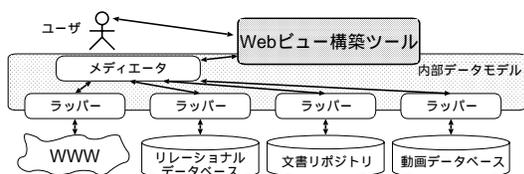


図 1 異種情報源統合システム

2. 基本概念

本ツールに現れる基本概念は次の通りである。各基本概念の具体例は 4 節に示す。

Object: D&D 対象の単位。構造化文書 (HTML 等) のエレメントや RDB 中の値が該当する。

Window: 次の 3 種類がある。

- **DataBox:** 情報源中のデータ集合 (例えば Web ページやリレーションのタブルの集合) を表現する。画面には集合中の 1 つのデータ (例えば 1 つの Web ページや 1 タブル) のみが表示される。
- **Palette:** 汎用のページ設計用部品が表示されている。
- **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。

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。

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、そのオブジェクトの D&D 操作は、その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。

Target Set: Example オブジェクトが代表するオブジェクト集合。

Association: 異なる Target Set 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。これらがあると、その関連をもつ特定の組み合わせだけが操作の対象となる。

- **S-Association:** 2 つの Example 間の相対的な位置関係により決まる。
- **V-Association:** Example として指定されたオブジェクトの値が同一である場合に決まる。

3. 基本操作

基本操作は、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。ただし、マウスをオブジェクトの上で右クリックすると、[Example, Another, Clue] というメニューが出現するので、ドラッグ前にそのオブジェクトの役割を指定することにより、多様な操作が可能となる。操作手順を次に示す。

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

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 と指定した場合は、Target Set 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。最初はデフォルトの Target Set が設定される。Another は、Another と指定されたオブジェクトも含まれるように Target Set を広げる。オブジェクトを Clue として指定すると、そのオブジェクトの存在が、Example が Target Set に含まれるための手がかり (条件) であることを示す。これは、Target Set を狭める働きをする。Clue を指定するときには、そのオブジェクトが満たすべき条件をより詳細に記述することも可能である。Association は、暗黙に決定される。

4. 操作例

本ツールを用いた操作例の概要を図 2 に示す。(1) ある RDB に野球選手の現在の成績 (打率など) が格納されたリレーション BATTING-STATS(P-Name, Hit, RBI, AVG) がある。(2) 別の RDB (Video DB) には打席毎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, 終了時間など) を格納したリレーション VIDEO (VID, GID, Begin, End, Batter, Pitcher, Contents) がある。(3) Web 上に、選手リストを含む各チーム毎のページ (XML ページ) が存在し、そのページから各選手毎のページにリンクが張られている。このとき「打率が 3 割より上の各選手毎に打席の映像をまとめたビュー (SMIL ページ) を作成する。そしてインデックスページ (HTML ページ) を作成し、そこで各選手毎に選手名, 打率, ビューへのリンクを示す」という例について考える。

データ操作の元となる DataBox と Palette は図 3(a) ~ (e) のように示される。各 DataBox には、他の操作機能¹³⁾を利用して、操作の対象となるデータ集合が既に集められていると仮定する。このデータ集合は DataBox の「Next」「Previous」ボタンでブラウジングできる。図 3 において、DataBox1 (TP) には各チーム毎のページ、DataBox2 (PP) には各選手毎のページ、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。

操作手順を以下に示す。図 3 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) TP の “Johnson” を Example と指定する。この

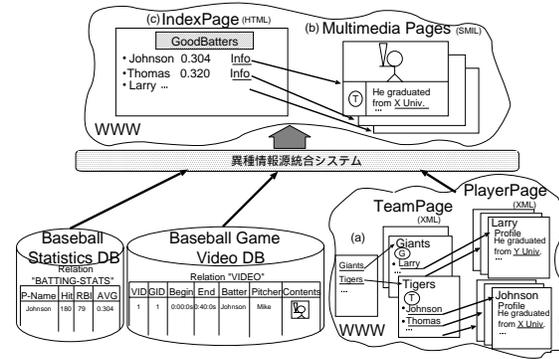


図 2 操作例

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 Target Set $TS_{\&12}$ に格納される。操作中の Example と、それが代表する Target Set は、それぞれハイライト表示される。続いて “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 $TS_{\&12}$ は拡大され、結果的に “Johnson” が代表する Target Set は全選手名を含むことになる。(Target Set の具体的決定法は 5 節で述べる。)

- (5) Example “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、 $TS_{\&12}$ 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 $TS_{\&12}$ 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 Target Set を含めて、同一の値を持つオブジェクトを Example とする Target Set 間には V-Association が生じる。この Association は、これらの Target Set 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 Target Set 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毎のダイ

- ジェスト映像へのリンクを作成する。
- (12) VD の V を Example とし, D&D する。
 - (13) V を repetition とする。これによって各選手毎の映像が 1 つの連続映像にまとめられる。
 - (14) TP の T を Example とし, D&D する。
 - (15) PP の “Profile ~” を Example とし, D&D する。

最後に Create ボタンを押すと, 図 3 の Canvas で指定した HTML 及び SMIL の Web ページが得られる。

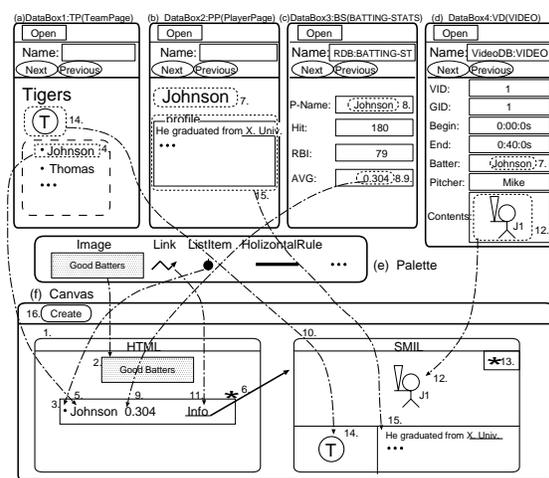


図 3 操作手順

5. セマンティクス

本ツールにおける操作の意味を, 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) Target Set と Association を表す Relation (Target Relation) を作成。(3) Canvas での操作に基づいた入れ子構造の生成と再構築操作。

5.1 データ構造

操作対象のデータをオブジェクト木として表す。図 4 は前節で示した操作例のデータの一部を示している。ここで, レベル 2 のノードをルートとした部分木は DataBox に対応している。このノードは 'DB.1' の形式でラベル付けされており, DB は DataBox 名を示す。レベル 3 のノードをルートとする部分木は XML ページもしくはタプル (ここでは統一のためタプルも XML ページとみなして表記する) を示す。このノードは 'Page.i' とラベル付けされている。更に深いノードをルートとする部分木はそれぞれ, XML ページ中の要素と対応している。これらのノードは 't.i' とラベル付けされ, t は要素を囲っているタグ名であり, i は

自然数である。同じノードを親とする同一のタグがあるとき, i によって順序付けされる。各オブジェクトは &n の形式で OID を持つ。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。

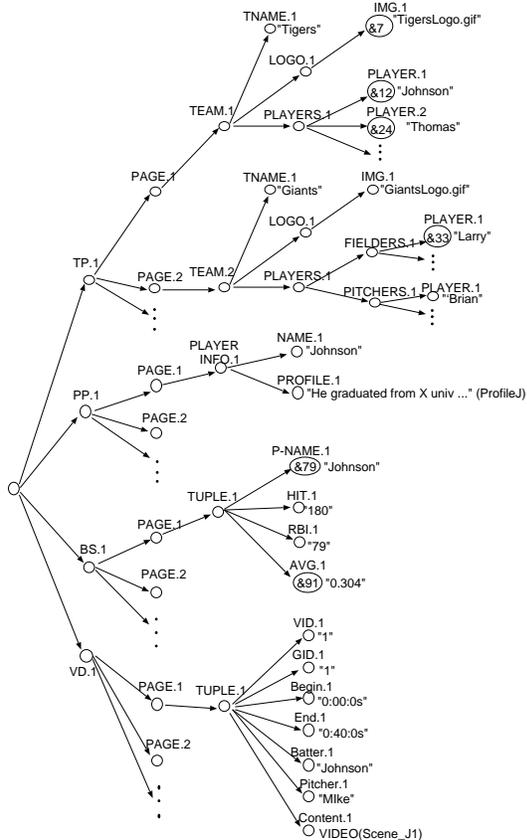


図 4 データ構造

5.2 Target Set

5.2.1 Clue を持たない Target Set の場合

Example(e) のみ, または Example と Another が指定されたときの Target Set TS_e は以下のように定義される。

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

ここで O は情報源中の全てのオブジェクトの集合を示し, $Candidate-Pred_e(o)$ 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 4 節の操作 4. で指定された $TS_{\&12}$ を示す。

$$TS_{\&12} = \{o \mid o \in O \wedge TP.1 \rightarrow PAGE.? \rightarrow TEAM.1 \rightarrow$$

$$PLAYERS.1 \rightarrow ?* \rightarrow PLAYER.?[o]\} \\ = \{ \text{"Johnson"}, \text{"Thomas"}, \dots, \text{"Larry"}, \dots, \text{"Brian"}, \dots \}$$

$TS_{\&12}$ は全チームの全選手名の集合となる.

Candidate-Predicates の導出

$Candidate-Pred_e(o)$ は Example と Another の指定から導出される. この導出過程において, Predicate に *annotation* を加える. これは, $path(e)$ と $value(e)$ を記録し, " " と " " によって囲まれる. 例として *annotation* を加えた $TS_{\&12}$ を示す. (ε は空ノードを示す)

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

導出過程を以下に示す.

- (1) まず, オブジェクト e を Example と指定すると, デフォルトの *Candidate-Predicate* は $p[e(v)]$ のように定義される. ここで, p は $path(e)$ の $Page.i$ を $Page.?(Page.i)$ で置き換えたものであり, v は $value(e)$ である. 操作例の 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, $TP.1 \rightarrow PAGE.?(PAGE.1) \rightarrow TEAM.1 \rightarrow PLAYERS.1 \rightarrow PLAYER.1[o(\text{Johnson})]$ がデフォルトの *Candidate-Predicate* として導かれる.
- (2) デフォルトの *Candidate-Predicate* は Another 操作によって指定されたオブジェクトを受け入れることによって, 拡張される. 図 5 に導出規則を示す. これは, Example と Another オブジェクトをそれぞれ e, a としたとき, $path(e)$ と $path(a)$ よりどのように *Candidate-Predicate* が変更されるかを示す. B, C はラベルネームを示し, p_i は注釈付きのパス式の一部である. q_i は Another オブジェクトへのパスの一部であり, q'_i は p_i に含まれる. 操作例の 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, $path(a) = TP.1 \rightarrow PAGE.1 \rightarrow TEAM.1 \rightarrow PLAYERS.1 \rightarrow PLAYER.2$ となる. ここで, $path(e) = TP.1 \rightarrow PAGE.1 \rightarrow TEAM.1 \rightarrow PLAYERS.1 \rightarrow PLAYER.1$ と照合すると, *PLAYER* のラベル番号が競合しているので, Rule 1 が適用される. ここで, $p_1 = TP.1 \rightarrow PAGE.?(PAGE.1) \rightarrow TEAM.1 \rightarrow PLAYERS.1$, $q_1 = q'_1 = TP.1 \rightarrow PAGE.1 \rightarrow TEAM.1 \rightarrow PLAYERS.1$, $p_2 = q_2 = q'_2 = \varepsilon$ となり, 変更後の Predicate は $TP.1 \rightarrow PAGE.?(PAGE.1) \rightarrow TEAM.1 \rightarrow$

$PLAYERS.1 \rightarrow PLAYER.?(PLAYER.1)[o(\text{Johnson})]$ となる. 続いて $\&33$ ("Larry") が 2 目の Another として指定されたとき, 同様に Rule 3 が適用され, 上記の $TS_{\&12}$ が得られ, $TS_{\&12}$ には全選手名が含まれる.

5.2.2 Clue を持つ Target Set の場合

さらに, Clue が指定されたときの Target Set は以下のように決定される.

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

$Clue-Pred_{cl}(c)$ はワイルドカードを持ち, $value(c)$ に対して条件が課されたパス式である. $SA-Pred_{e,cl}(o, c)$ 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 4 節の操作 8. で指定された $TS_{\&79}$ を示す.

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

where $p = BS.1 \rightarrow PAGE.?(PAGE.1) \rightarrow TUPLE.1$.

" $BS.1 \rightarrow PAGE.?(PAGE.1) \rightarrow TUPLE.1 \rightarrow AVG.1[c > 0.3]$ " 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 " $BS.1 \rightarrow PAGE.?(PAGE.1) \rightarrow TUPLE.1$ " を共有することを示している.(図 6 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.

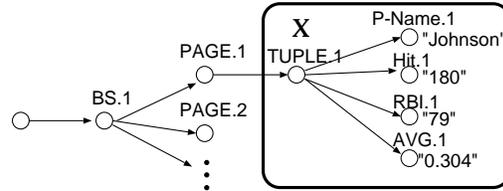


図 6 オブジェクト集合 X

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

5.3 Target Relation

Target Relation は, Example による Target Set とそれらの間の Association を示す. 詳細な定義はフルペーパーにおいて記述する.

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

例

式 1 は操作例の Target Relation の明細を示す. 括弧内の数字は操作例における操作の順序を示す. 式 1

	Original Pred.	$Path(e)$	$Path(a)$	Modified Pred.
Rule 1	$p_1 B.i p_2[o(v)]$	$q_1 B.i q_2$	$q'_1 B.k(\neq i) q'_2$	$p_1 B.?(B.i) p_2[o(v)]$
Rule 2	$p_1 B.i p_2[o(v)]$	$q_1 B.i q_2$	$q'_1 C(\neq B).k q'_2$	$p_1 ?(B.i) p_2[o(v)]$
Rule 3	$p_1 q_3 p_2[o(v)]$	$q_1 q_3 q_2$	$q'_1 q_4(\neq q_3) q'_2$	$p_1 ? * (q_3) p_2[o(v)]$

図 5 Candidate-Predicate 導出規則

Ex. Johnson in TP	Ex. \textcircled{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. $\textcircled{\text{タ}}$ in VD
Johnson	\textcircled{T}	Johnson	ProfileJ	Johnson	0.304	Johnson	$\textcircled{\text{タ}}_1$
Johnson	\textcircled{T}	Johnson	ProfileJ	Johnson	0.304	Johnson	$\textcircled{\text{タ}}_2$
Thomas	\textcircled{T}	Thomas	ProfileT	Thomas	0.320	Thomas	$\textcircled{\text{タ}}_T$
Larry	\textcircled{G}	Larry	ProfileL	Larry	0.315	Larry	$\textcircled{\text{タ}}_L$
Larry	\textcircled{G}	Larry	ProfileL	Larry	0.315	Larry	$\textcircled{\text{タ}}_L$

図 7 Target Relation 例

によって図 7 の Target Relation が生成される.

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

where

$p_1 = TP.1 \rightarrow PAGE.?(PAGE.1) \rightarrow TEAM.1,$
 $p_2 = PP.1 \rightarrow PAGE.?(PAGE.1) \rightarrow PLAYERINFO.1,$
 $p_3 = BS.1 \rightarrow PAGE.?(PAGE.1) \rightarrow TUPLE.1, \text{ and}$
 $p_4 = VD.1 \rightarrow PAGE.?(PAGE.1) \rightarrow TUPLE.1.$

式 1: 操作例に対応した

Target Relation Specification

5.4 入れ子構造の生成と再構築操作

Canvas でのユーザの操作を基に、次の情報を反映させた Result Specification(図 9) を導出する。(1) Canvas での Example の位置と繰返し構造の指定、(2) パレットから選択されたオブジェクト、(3) 各オブジェクトの配置位置などのレイアウト情報、(4) ハイパーテキストリンク構造. Target Relation に対し、(1) の繰返し構造に従って Projection 演算と Nest 演算を行うことにより、入れ子型リレーションを構築する.

この例において、以下の式は図 8 の入れ子型リレーションを生成する.

$$Nest_{V=(\textcircled{\text{タ}})}(\pi_{Johnson, 0.304, \textcircled{\text{タ}}, \textcircled{T}, ProfileJ}(TR))$$

最後に、(2)~(4) の情報を基に再構築操作を行うことにより Web ビューが構築される.

Ex. Johnson in TP	Ex. 0.304 in BS	V		Ex. ProfileJ in PP
		Ex. $\textcircled{\text{タ}}$ in VD	Ex. \textcircled{T} in TP	
Johnson	0.304	$\textcircled{\text{タ}}_1$	\textcircled{T}	ProfileJ
Thomas	0.320	$\textcircled{\text{タ}}_T$	\textcircled{T}	ProfileT
Larry	0.315	$\textcircled{\text{タ}}_L$	\textcircled{G}	ProfileL

図 8 入れ子型リレーション例

```

{ROOT:( " <HTML><BODY><IMG SRC= "goodbatters.gif "><UL> ",
*( " <LI>Johnson 0.304 <A HREF=&URL(P1)>Info</A> ",
</UL></BODY></HTML> " ),
P1:( " <SMIL><HEAD><LAYOUT>
<ROOT-LAYOUT HEIGHT= "373 " WIDTH= "506 "/">
<REGION ID= "R1 " LEFT= "0 " TOP= "0 "
HEIGHT= "215 " WIDTH= "346 "/">
<REGION ID= "R2 " LEFT= "0 " TOP= "216 "
HEIGHT= "157 " WIDTH= "167 "/">
<REGION ID= "R3 " LEFT= "168 " TOP= "216 "
HEIGHT= "155 " WIDTH= "338 "/">
</LAYOUT></HEAD>
<BODY><PAR><SEQ> ",
*( " VIDEO(Scene1)[tag='REF',+REGION='R1' " ],
" </SEQ>
TigersLogo.gif[tag='REF',+REGION='R2']
<TEXTSTREAM SRC=&URL(P2),+REGION='R3' />
</PAR></BODY></SMIL> " ),
P2:( " <WINDOW>ProfileJ</WINDOW> " )}

```

図 9 Result Specification 例

6. プロトタイプシステムの実装

異種情報源統合システム上にプロトタイプシステムの開発を行った. このシステムは、Web ビュー構築ツールでのユーザの操作を基にビューの再構築を行う. システムの構成を図 10 に示す.

本プロトタイプシステムの実装には Java を用い、Windows98 上で開発を行った. 情報源としては XML

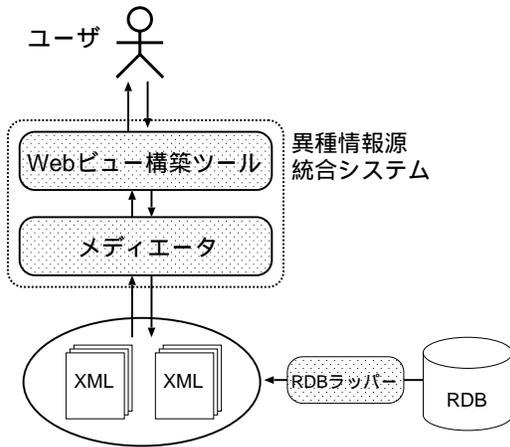


図 10 システム構成

文書集合と RDB を対象としている。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。

このシステムにおいて、Web ビュー構築ツールは、ユーザによる問合せの生成を視覚的に支援する。ユーザは本ツールにおいて、クリック後のメニュー選択や D&D 操作をマウスを用いて行い、目的とする Web ビューを生成する。本ツールはユーザの操作を基に Target Relation Specification そして Result Specification を導出し、メディエータへと渡す。本ツールの詳細は次節で述べる。

メディエータは以下の機能を持つ。

情報源の管理 情報源として指定された XML 文書集合を Document Object Model(DOM)⁴⁾ オブジェクト集合へと変換し、管理する。ここで、DOM とは HTML・XML 文書のための Application Programming Interface(API) である。

Web ビュー生成 まず、Web ビュー構築ツールより渡された Target Relation Specification を基に、必要なデータ集合を DOM オブジェクト集合より抽出、結合し Target Relation を生成する。次に、その Target Relation に対して Result Specification 中の指定を基に Projection, Nest 演算を行い、入れ子リレーションを生成する。最後に入れ子リレーションに対して Result Specification の構造をテンプレートとして適用することにより、ユーザの要求する Web ビューを生成する。

6.1 Web ビュー構築ツールの実装

Web ビュー構築ツールの主要なモジュール構成を図 11 に示す。

各モジュールの機能は以下のようにになっている。

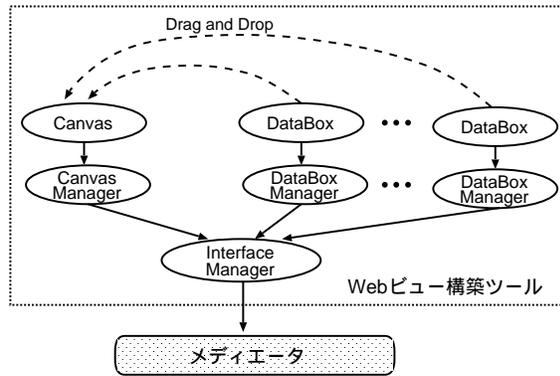


図 11 本ツールの構成

DataBox, Canvas モジュールはそれぞれ DataBox, Canvas ウィンドウの表示を行う。そして、そのウィンドウ内でイベント(ユーザによる D&D 操作など)が発生した場合は、そのイベントをそれぞれ DataBox Manager, Canvas Manager へと渡すと同時に、その結果を描画する。

各 DataBox Manager モジュールは受け取ったイベントから Candidate Predicate または Clue Predicate を導出する。さらに、1 つの DataBox において複数の Example が指定された場合は、Example 間の S-Association を導出する。これらの Predicate は Canvas の“Create”ボタンが押されたときに Interface Manager へと渡される。

Canvas Manager モジュールは受け取ったイベントから Result Specification を導出し、“Create”ボタンが押されたときに Interface Manager へと渡す。

Interface Manager モジュールは受け取った Predicate のオブジェクトの値から V-Association を導出し、Target Relation Specification を完成させ、Result Specification と共にメディエータへと渡す。

そしてメディエータにおいてユーザの要求した Web ビューが生成される。

本ツールは、Target Relation Specification と Result Specification を変換モジュール等の追加によって他の操作系の問合せに変えることにより、他の統合環境へも適用可能となる。

図 12 はプロトタイプシステムの実行画面を示している。

7. 関連研究

本ツールの特徴は、マルチメディア Web ビュー構築時のページ設計操作と、情報源に対するデータ抽出操作をシームレスに統合している点と、Example に

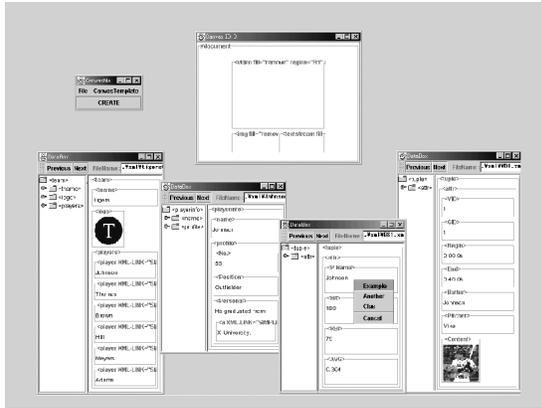


図 12 プロトタイプシステムの実行画面

基づくデータ抽出操作において対象データの異種性や半構造化性を考慮している点である。以下では、関連の深いいくつかの研究について述べる。

Delaunay⁵⁾ は、ユーザがアイコンを D&D するインタフェースを提供する。このアイコンは、様々な情報源への問合せ結果であるマルチメディアオブジェクトを示している。しかし、その問合せには SQL 風問合せ言語を入力しなくてはならない。RBE⁶⁾ では、ユーザは様々な GUI ウィジェットの D&D を行う。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。ウィジェットとデータの対応は domain variables によって指定される。Tiramisu²⁾ は Web サイト管理システムである。このシステムにおいて生成されるページの設計は FrontPage のようなオーサリングツールによって行われる。Tiramisu の基本概念では、Web サイトの論理設計と外観の設計を切り分けている為、情報源に対するデータ抽出操作の枠組みは、ページ設計の操作の枠組みとは異なっている。SuperSQL⁷⁾ と SQL+D⁸⁾ は SQL に表示の指定を加えた問合せ言語である。これらは問合せを視覚的に示す枠組みを持たない。

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。FrontPage⁹⁾ は ASP 形式の Web ページに SQL を埋め込むことが可能である。

“Example”の概念は QBE¹⁰⁾ において最初に示された。入れ子構造の生成に関しては STBE¹¹⁾ 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、DataGuide¹²⁾、HQBE¹³⁾ 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。本ツールにおける Example は QBE 等における

それとは、(1) 具体的なデータオブジェクト自身であること、(2) グラフ構造に基づき動的にそれが表す Target Set が決定されること、の二点で全く異なる。

XML-GL¹⁴⁾ はグラフ構造をベースにした視覚的 XML 問合せ言語であり、QBE に似た枠組みを持ちながらデータの半構造化性に対応している。しかし、レイアウト等の視覚的なページ設計を行う枠組みを持たない。

また、多くの視覚的問合せ言語があり¹⁵⁾、それらのうちのいくつかは D&D 形式の問合せをサポートしている。これらの言語は皆、スキーマのようなメタデータに基づいた問合せを要求する。これに対し、本ツールの枠組みではインスタンススペースの操作例から、問合せを推論する。

8. 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と実装について述べた。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 が表すデータ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、情報源の異種性や半構造化性に対応した。これは、インスタンススペースの操作を基にシステムが操作の意味を推論することにより、ユーザの要求に合った問合せの生成を可能にする。

今後の課題としては、動画、音声等連続メディアの再生範囲指定への対応、より高度なページレイアウト設計支援、本ツールの有効性評価実験、等が挙げられる。

謝辞 本研究の一部は、文部省科学研究費、筑波大学特別プロジェクト、次世代情報放送システム研究所の助成による。

参考文献

- 1) Fernández, M., Florescu, D., Kang, J., and Levy, A. Catching the Boat with Strudel: Experiences with a Web-Site Management System. *Proc. ACM SIGMOD'98*, 1998.
- 2) 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.
- 3) W3C. “Synchronized Multimedia Integration

- Language (SMIL) 1.0 Specification,” W3C Recommendation, <http://www.w3.org/TR/REC-smil>, 1998.
- 4) W3C. “Document Object Model(DOM) Level 1 Specification” W3C Recommendation, <http://www.w3.org/TR/REC-DOM-Level-1>, 1998.
 - 5) Cruz, I. F. and Lucas, W. T. Automatic Generation of User-Defined Virtual Documents Using Query and Layout Templates, 1998. *Theory and Practice of Object Systems*, Vol. 4, No. 4, pp. 245-260.
 - 6) Kishnamurthy, R. and Zloof, M. RBE: Rendering By Example. *Proc. ICDE'95*, pp. 288-297, 1995.
 - 7) Toyama, M. SuperSQL: An Extended SQL for Database Publishing and Presentation. *Proc. SIGMOD '98*, pp. 584-586, 1998.
 - 8) Baral, C., Gonzalez, G., and Son, T. C. Design and Implementation of Display Specifications for Multimedia Answers. *Proc. ICDE'98*, pp. 558-565, 1998.
 - 9) Microsoft Corporation. Microsoft Homepage. <http://www.microsoft.com/>.
 - 10) M. Zloof, Query By Example. *IBM Sys. J.*, 1977.
 - 11) 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.
 - 12) Goldman, R. and Widom, J. DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases. *Proc. VLDB '97*, pp. 436-445, 1997.
 - 13) 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.
 - 14) 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.
 - 15) 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.
-