データ特性に基づくSuperSQL質問文の自動生成手法

根本　潤†　遠山　元道††

† 慶應義塾大学大学院 理工学研究科 開放環境科学専攻　〒 223–8522 横浜市港北区日吉 3–14–1
†† 慶應義塾大学 理工学部 情報工学科　〒 223–8522 横浜市港北区日吉 3–14–1
E-mail: †jun@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

あらまし　近年，ショッピングサイトやニュースサイトなどのデータ集約型 Web サイトが一般的に見られるようになった．そうしたサイトにおいて閲覧者が欲しい情報を得るためには，サイト作成者側がデータを分類表示したり，検索機能を提供したりする必要がある．それと同時に，閲覧者は欲しい情報を得られるような検索を行わなければならない．そこで，こうしたサイト作成者側の負担を軽減し，かつ閲覧者が情報を得やすいようなデータの表示形式を提供するため，本研究ではデータ特性に基づく SuperSQL 質問文の自動生成手法を提案する．本手法では，まず，サイト作成者側の用意した閲覧者がデータを検索するための通常の SQL 質問文を仮に問い合わせる．そして，得られた結果の値の統計情報等をもとに SuperSQL 質問文を生成する．最後に，SuperSQL 質問文の問合せ結果を表示する．
キーワード　SuperSQL, 問合せ処理, DB 言語

Automatic Generation of SuperSQL Query based on Data Characteristic

Jun NEMOTO† and Motomichi TOYAMA††

† School for Open and Envionmental Systems, Graduate School of Science and Technology, Keio University
Hiyoshi3–14–1, Kouhoku-ku, Yokohama-shi, 223–8522 Japan
†† Department of Information and Computer Science, Faculty of Science and Technology, Keio University
Hiyoshi3–14–1, Kouhoku-ku, Yokohama-shi, 223–8522 Japan
E-mail: †jun@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

Abstract　Recently, data-intensive web sites such as shopping sites are becoming more and more popular. It is difficult for users to get wished information in these web sites unless Web designers and programmers lay out web pages properly according to categories and provide a function to search data easily. For the purpose of reducing these burden in designing and programming web sites and providing interfaces by which users get information without difficulty, we propose an automatic SuperSQL query generation method based on data characteristics. In our method, first, web designers and programmers prepare and query normal SQL. Then, we generate a SuperSQL query based on statistical information of values in the result. Lastly, the result of querying SuperSQL is returned to end users as a final result.
Key words　SueprSQL, Query Processing, DB Language

1. はじめに

近年，ショッピングサイトやニュースサイトなどのデータ集約型 Web サイトが一般的に見られるようになった．そうしたサイトにおいて閲覧者が欲しい情報を得るためには，サイト作成者側がデータを分類表示したり，検索機能を提供したりする必要がある．それと同時に，閲覧者は欲しい情報を得られるような検索を行わなければならない．

そこで，こうしたサイト作成者側の負担を軽減し，かつ閲覧者が情報を得やすいようなデータの表示形式を提供するため，本研究ではデータ特性に基づく SuperSQL 質問文の自動生成手法を提案する．本手法では，まず，サイト作成者側の用意した閲覧者がデータを検索するための通常の SQL 質問文を仮に関係データベースに問い合わせる．そして，得られた結果の値の統計情報等に基づいて SuperSQL 質問文を生成する．なお，本研究では，この統計情報のことをデータ特性と呼ぶこととす

る．そして最後に，生成された SuperSQL 質問文の問合せ結果を表示する．本手法の処理の流れを図 1 に示す．

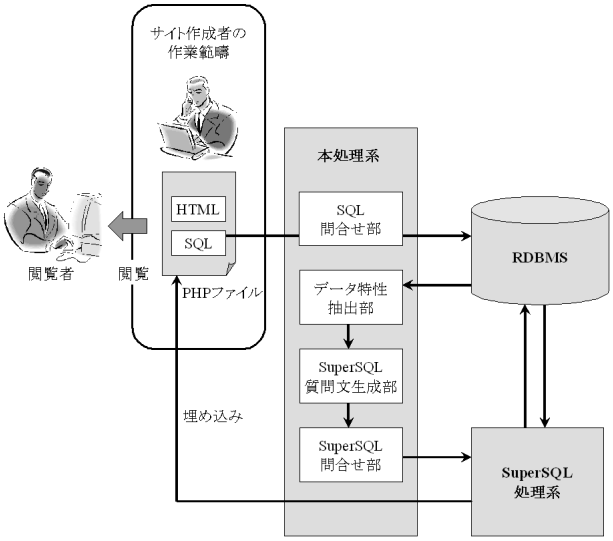


図 1　SuperSQL 質問文生成の流れ

図のように，サイト作成者は関係データベース内に格納されたスキーマ情報と HTML に関する知識さえあれば，SuperSQL を用いて得られる多彩なレイアウトの表を含んだページを提供できる．さらに，サイト作成者にプログラミングの知識があれば，検索インターフェースを付加することで，閲覧者の検索結果に応じてより見やすいレイアウトで情報を提供するといった利用方法も考えられる．

以下，本稿の構成を示す．まず，2. 章で SuperSQL の概要について述べる．次に，3. 章でデータ特性の定義について述べる．そして，4. 章では，それに基づいて SuperSQL 質問文を生成する手法について述べる．これらを実装，評価，検討した結果を 5. 章で述べる．さらに，6. 章で関連研究について触れた後，最後に 7. 章でまとめを述べる．

2. SuperSQL

本章では，最終的な出力結果を得るための問合せ言語である SuperSQL について簡単に述べる．SuperSQL は関係データベースの出力結果を構造化し多様なレイアウト表現を可能とする SQL の拡張言語であり，慶應義塾大学遠山研究室で開発されている [4] [5]．そのクエリは SQL の SELECT 句を GENERATE< $media$ >< TFE > の構文を持つ GENERATE 句で置き換えたものである．ここで < $media$ > は出力媒体を示し，HTML，XML，Excel，LaTeX，PDF などの指定ができる．また < TFE > はターゲットリストの拡張である Target Form Expression を表し，結合子，反復子などのレイアウト指定演算子を持つ一種の式である．

2.1 結　合　子

結合子はデータベースから得られたデータをどの方向 (次元) に結合するかを指定する演算子であり，以下の 4 種類がある．括弧内はクエリ中の演算子を示している．

- 水平結合子 (,)

データを横に結合して出力．

例：Name, Tel

name	tel

- 垂直結合子 (!)

データを縦に結合して出力．

例：Name! Tel

name
tel

- 深度結合子 (%)

データを 3 次元方法へ結合．出力が HTML ならばリンクとなる．

例：Name % Tel

name	→	tel

- 時間結合子 (#)

時間結合子は後述する時間反復子で囲まれたデータを 1 つのインスタンスとし，そのインスタンス同士を時間軸方向に結合．

2.2 反　復　子

反復子は指定する方向に，データベースの値があるだけ繰りかえして表示する．また反復子はただ構造を指定するだけでなく，そのネストの関係によって属性間の関連を指定できる．例えば

[科目名]! , [学籍番号]! , [評点]!

とした場合には各属性間に関連はなく，単に各々の一覧が表示されるだけである．一方，ネストを利用して

[科目名！[学籍番号 , 評点]!]!

とした場合には，その科目毎に学籍番号と評点の一覧が表示されるといったように，属性間の関連が指定される．以下，その種類について述べる．

- 水平反復子 ([ ],)

データインスタンスがある限り，その属性のデータを横に繰り返し表示する．

例：[Name],

name1	name2	…	name10

- 垂直反復子 ([ ]!)

データインスタンスがある限り，その属性のデータを縦に繰り返し表示する．

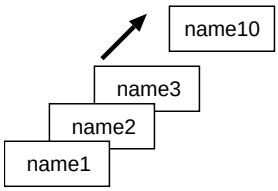
例：[Name]!

name1
name2
…
name10

- 深度反復子 ([ ]%)

データインスタンスがある限り，その属性のデータを奥行き方向 (例：HTML ではリンク，PDF ではページ変換) に繰り返し

表示する．
例：[Name]%



- 時間反復子 ([ ]#)

データインスタンスがある限り，その属性のデータを一定時間ごとに繰り返し表示する．

2.3 装 飾 子

SuperSQL では関係データベースより抽出された情報に，文字サイズ，文字スタイル，横幅などの情報を付加できる．これらは装飾演算子 (@) によって指定する．

```
<属性名>@{ <装飾指定> }
```

装飾指定は” 装飾子の名称 = その内容” として指定する．複数指定するときは各々を”,” で区切る．

3. データ特性

データ特性とは，先にも述べたように始めに関係データベースへ問い合わせた結果の統計情報のことをいう．本研究ではこのデータ特性に基づいて TFE を決定する．本章では，現在，利用を想定している 2 つのデータ特性について述べる．

3.1 インスタンス特徴度

インスタンス特徴度とは，ある属性におけるある値がどれだけ重複して出現しているかを示すものである．問い合わせる SQL のターゲットリストの各属性を a_i(ただし，$i = 1, 2, \cdots m$ で m は自然数) とすると，インスタンス特徴度 IC_i は，次のように表される．

$$IC_i = |distinct(V_i)|$$

なお，V_i は問合せ結果における属性 a_i の実際の値の集合 $V = \{v_1, v_2, \cdots, v_n\}$(ただし，$n$ は結果の全タプル数) であり，$distinct$ はその重複を取り除く関数とする．このインスタンス特徴度が小さい属性ほど，カテゴライズして表示することが望ましいと仮定できる．

しかしながら，この仮定が妥当でないケースは多々ありうる．例として，学生の成績データベースを考える．学籍番号や所属学科，科目，評点，性別などを結果として返す問合せを行ったときに，性別はそもそも 2 通りしかなく，インスタンス特徴度は小さくなるが，必ずしも性別でカテゴライズして表示してほしいとは限らない．属性の意味的な性質に関するこのような問題に対しては，3.3 節で対応方法を述べる．

3.2 数値特徴度

属性のデータ型が数値や時刻などである場合も，3.1 節で述べたインスタンス特徴度の仮定が当てはまらない．数値型や時刻型のデータの場合，むしろ値が多岐にわたっているほど，一定の範囲でカテゴライズして表示してほしい可能性もありうる．そこで，数値特徴度を次のように定める．

$$NC_i = \frac{1}{\bar{v_i}} \sqrt{\frac{1}{n} \sum_{j=1}^{n} (v_{ij} - \bar{v_i})^2}$$

これは，v_i の変動係数を表す．変動係数は，データの標準偏差を平均値で割ったものであり，単位に依存しない実質的なデータのばらつきを評価できる尺度である．この値が閾値以上のとき，区間 $[min(V_i), max(V_i)]$ を一定区間に分割し，分類表示するものとする．このとき，変動係数のための閾値が適切に定められていれば，狭い範囲に集中しているようなデータを分類表示してしまうようなケースは避けられる．

3.3 ユーザインタラクション

データベースへの問合せ結果は目的に応じて多数の視点で捉えられることが可能であるケースは多い．そして，閲覧者はそうした視点に基づいて自分が情報を得やすいようなレイアウトで結果を閲覧できることを望む．しかしながら，サイト作成者がデータインスタンスに合わせた構造を考慮したり，多数のレイアウトをあらかじめ用意するのは大きな負担であるといえる．そこで，本研究のようなレイアウトの自動決定が期待されるが，データ特性のみに基づく決定はユーザの多様な要求に対しては不完全である．よって，ユーザとのインタラクションによりこの不完全性を解決する．

3.3.1 注 視 属 性

一般的に，エンドユーザがデータベースから出力された結果をもとに作成された Web ページを閲覧する際，ある属性に着目して閲覧することは多い．例えば，ショッピングサイトについて考えてみると，ある閲覧者はメーカ別に並ぶ商品の中から欲しいものを選択したいかもしれないし，またある別の閲覧者は価格ごとに分類された商品の中から選択したいかもしれない．すなわち，各閲覧者ごとに閲覧の際に着目している属性が 1 つもしくは複数存在する．本研究では，閲覧者が最も重きをおいている属性を注視属性とする．この注視属性は TFE を決定する際に反映される．

3.3.2 注視属性の選択履歴

注視属性は，閲覧者によって異なるが，どの注視属性がよく選ばれているのかという選択履歴を TFE の決定に反映することで，意味的に考えて本来は分類表示すべきでない属性を分類対象として選ぶ確率を下げることができる．例えば，3.1 節における学生データベースの例では，所属学科や科目名が注視属性として頻繁に選ばれる一方で，性別が選ばれることが稀ならば，デフォルトのレイアウトとして性別で分類表示することを避けることができる．

4. SuperSQL 質問文の生成

本章では，SueprSQL 質問文の生成について述べる．本研究では，出力メディアとして HTML を想定し，HTML においての構造決定を考えるため，以下では TFE を中心に考える．FROM 句，WHERE 句については基本的に最初の問合せに用いられた通常の SQL のものを利用する．

4.1 集約優先度

TFE を決定する際にまず始めに考えるのは，どの属性について分類表示するかということである．これは，以下に定める集約優先度によって決まる．

$$GP_i = \frac{n}{IC_i}\ g_i\ m\ h_i$$

ここで，g_i は注視属性を反映させるための値で，a_i が注視属性であるならばパラメータ α をとり，そうでなければ 1 をとる．また，h_i は他のユーザによって蓄積された注視属性の履歴を反映させるための値である．全履歴数に対して，各属性が注視属性として何回指定されたのかを比率で表す．なお，h_i に属性の数 m を乗じているのは，属性の数によって尺度が偏ってしまうことを避けるためである．同様に，結果タプルの数 n を乗じているのは，結果タプル数によって尺度が偏ってしまうことを避け，この集約優先度をどのような SQL 質問文に対しても共通の指標として利用するためである．本手法では，この集約優先度が閾値以上の属性を分類表示の対象とする．

4.2 TFE の決定

分類表示の対象として選ばれた属性は，2.2 節の反復子の例のように，それ以外の属性をネストするように配置する．例えば，最初に問い合わせる SQL の SELECT 句が

SELECT 学籍番号 , 所属学科 , 科目名 , 評点

で，閾値以上の集約優先度をもつ属性が所属学科であった場合，生成される TFE は，

[所属学科 , [学籍番号 , 科目名 , 評点]!]!

となる．また，さらに科目名も集約の対象であった場合には，

[所属学科 ! [科目名 , [学籍番号 , 評点]!]!]!

となる．このときに出力される結果は以下のようになる．

所属学科 1		
科目名 1	学籍番号 1	評点 1
	学籍番号 2	評点 2
	学籍番号 3	評点 3
所属学科 2		
…	…	…

例からわかるように集約対象が複数あった場合は，より集約優先度が高いほうの属性が大分類となるように外側に配置する．また，集約対象の属性とそれ以外の属性との結合は，内側から順に 1 次元方向，2 次元方向，3 次元方向の結合子を用い，それを繰り返す．これは，単一方向への結合が視認性を下げてしまうことを避けるための対応である．ただし，この対応とその視認性に関しては議論の余地を残すところであり，評価実験の結果と併せて 5.2 節および 5.3 節で検討する．

4.3 数値特徴度を考慮する場合

3.2 節で定めた数値特徴度を考慮する場合，若干 SuperSQL 質問文の生成プロセスが異なる．まず，注視属性として選ばれた属性のデータ型が数値型もしくは時刻型である場合は数値特徴度が閾値以上かを判断する．もし，閾値以上であれば，注視属性の値がとりうる範囲を一定区間に分割し，それぞれの区間の開始値・終了値を一時テーブルに格納する．そして，その一時テーブルを他のテーブルと結合し，区間ごとに集約する SuperSQL 質問文を生成する．一方，数値特徴度が閾値未満である場合には，通常の集約優先度を求めるプロセスに入る．また，数値型，時刻型のデータ型の属性が注視属性として選ばれなかった場合においても，数値特徴度が閾値以上の場合には，代替レイアウトとして数値による分類が存在することをユーザに対して示すこととする．

5. 実装・評価・検討

5.1 実　　装

データ特性に基づいて，SuperSQL 質問文の生成処理系を PHP の関数として実装した．本節では実装した関数の仕様と利用例について述べる．

5.1.1 関 数 仕 様

以下は実装した PHP の関数の仕様である．

int **dcssql** (string *sql*,

boolean *style* [, string *gazedAttribute*])

dcssql 関数は，注視属性を *gazedAttribute* として SQL 質問文 *sql* から生成される SuperSQL 質問文の問合せ結果を宣言箇所に挿入する．*sql* および *gazedAttribute* はいずれも文字列型であり，*gazedAttribute* は省略可能である．

style はブール型であり，SuperSQL 質問文の問合せ結果にシステムが自動的に決定するスタイル情報を付加するかどうかを指定可能である．スタイル情報は，2.3 節の装飾子によって指定されるものである．現在の実装では，次の 2 点のみを考慮している．

（1） 集約優先度が閾値以上で，分類表示の対象として選ばれた属性の背景色指定

（2） 各属性の最大文字列長に応じた横幅指定

スタイル情報の付加の必要性に関しては 5.2.2 節で，評価実験の結果と併せて後述する．

dcssql 関数の戻り値は整数型で，すべての処理が問題なく終了した場合に 1 を返し，エラーが生じた場合には −1 を返す．また，4.3 節で述べたような，数値型の属性が注視属性として選ばれなかったけれども，数値特徴度が閾値以上の場合には 2 を返し，代替レイアウトが存在することを示すために利用する．

5.1.2 利　用　例

5.1.1 節のとおり，dcssql 関数は通常の SQL 質問文とスタイルフラグおよび注視属性を引数として渡すと，そこに SuperSQL 質問文の問合せ結果を挿入するので，その他の PHP コードの部分と併せて最終結果とし，閲覧者に提供するといった利

```
1   /* sample.php */
2   <html>
3   <head>
4   <title>書籍リスト</title>
5   <meta http-equiv="Content-Type"
        content="text/html;charset=EUC-JP" />
6   </head>
7   <body>
8   <form method="get" action="sample.php">
9   <input type="radio"
        name="gazed" value="name">名前
10  <input type="radio"
        name="gazed" value="author">著者
11  <input type="radio"
        name="gazed" value="publisher">出版社
12  <input type="radio"
        name="gazed" value="category">分類
13  <input type="radio"
        name="gazed" value="price">価格
14  <input type="radio"
        name="gazed" value="pdate">出版日
15  <input type="submit" value="注目！">
16  </form>
17  <?php
18      include("dcssql.inc");
19      $sql = "SELECT b.name, b.author, b.publisher,
           b.category, b.price, b.pdate FROM book b";
20      dcssql($sql, TRUE, "$_GET[gazed]");
21  ?>
22  </body>
23  </html>
```

図 2　PHP ファイル例

用方法が考えられる．なお，閲覧者が注視属性を選択し，それを本処理系に引き渡す部分に関してはサイト作成者側がコーディングする必要がある．しかし，それはほぼ HTML だけで記述可能で，それ以外の部分についてはプログラミングに関する知識を要求しない．例として，図 2 にサイト作成者が用意すべき PHP ファイルの内容を示す．また，そのファイルに閲覧者がアクセスした際に表示される画面の例を図 3 に示す．

以下，図 2 のファイルについて解説する．まず，閲覧者が注視属性を指定するインターフェース部分が，8-16 行目に記述されている．例では，ラジオボタンにより提供しているが，排他的に注視属性を選択できればどのようなインターフェースでもかまわない．

18 行目の include 関数は dcssql 関数の実体などを含むファイルを取り込むためのものである．

19 行目では dcssql 関数に与えるための通常の SQL 質問文を設定している．例における SQL 質問文が想定しているテーブルのスキーマについては，表 1 に示す．また，そのテーブルには 100 件の書籍データが格納されており，問合せ結果としてこ

図 3　sample.php の実行例

表 1　book テーブルのスキーマと
全タプルに対するインスタンス特徴度

属性名	データ型	内容	IC
isbn	文字列型	ISBN	100
name	文字列型	書籍名	100
author	文字列型	著者	74
publisher	文字列型	出版社	30
category	文字列型	分類	12
price	整数型	価格	43
pdate	日付型	出版日	78
img	文字列型	画像ファイル名	100

れら全てのデータが返ってきた場合のインスタンス特徴度も併記しておく．

そして，20 行目で閲覧者のブラウザから GET メソッドで送信された注視属性に関するフォームデータを`$_GET[gazed]` によって PHP ファイルが受け取り，さらにそれを dcssql 関数に引き渡している．

閲覧者が図 2 の sample.php にアクセスすると，最初は注視属性が選択されていない場合の結果が表示される．図 3 の結果は，表 1 にある通り，インスタンス特徴度の低い順に，分類，出版社で集約されている．なお，このときには以下のような SuperSQL 質問文が生成されている．ただし，見やすさのために装飾に関する指定は省略している．

```
GENERATE HTML
[ b.category !
[ b.publisher ,
[ b.name , b.author , b.price , b.pdate ]! ]! ]!
FROM book b
```

5.2　評　　価

ここでは，本システムの有用性を検証するための実験と評価について述べる．

5.2.1　実 験 環 境

コンピュータリテラシーの高い被験者，そうでない被験者を

表 2　スタイル情報付加の有無による見やすさの比較 (書籍)

	スタイルあり	スタイルなし
平均	1.38	3.75
t 値	-5.80	
t 境界値	1.81	

含めた 8 人に本システムを用いた Web ページを閲覧してもらった．Web ページに用いたデータベースは書籍，ニュース，都道府県の地理統計情報である．簡略化のため，いずれのデータベースも単一のテーブルから構成されている．

書籍テーブルに関しては 5.1.2 節の利用例で用いたものと同様なので省略するが，それ以外の各データの特徴について簡単に述べると，まず，ニューステーブルは ID，タイトル，本文，ニュースソース，大分類，小分類，到着時刻という属性をもち，100 件のニュースが格納されている．本文のデータ型はテキスト型となっており，容量は 200 バイトから 6000 バイト程度である．

また，統計情報テーブルは ID，都道府県名，地域，人口，面積，事業所数，固定電話加入者数，携帯電話加入者数，PHS 加入者数という属性からなる．都道府県名，地域以外はいずれも数値型である．

以降の各実験における閾値やパラーメータについては，経験的に数値特徴度の閾値を 0.3，集約優先度の閾値を 3，注視属性を反映させるためのパラメータ α を 5 とした．

5.2.2　予 備 実 験

実験を始めるにあたって，生成する SuperSQL 質問文に対するスタイル情報の付加に関して予備実験を行った．予備実験では，まず，被験者にスタイル情報のない SuperSQL 質問文によって得られた結果と，スタイル情報のある SuperSQL 質問文によって得られた結果を閲覧してもらった．その上で，それぞれに対し見やすさの観点で 5 段階評価 (1:非常に見にくい，2:見にくい，3:ふつう，4:見やすい，5:非常に見やすい) で点数をつけてもらった．見やすさというのは，構造の妥当性を無視した外見的な観点である．これらを，書籍，ニュース，統計情報の 3 つ全てについて行った．

予備実験の結果を表 2 に示す．表 2 は，書籍データ対して，スタイル情報がある場合とない場合の評価点数を比較し，t-検定を行ったものである．t-検定は帰無仮説を「スタイル情報付加の有無に関わらず評価点数の平均に差はない」，対立仮説を「スタイル情報を付加したときの評価点数の平均の方が高い」として，有意水準 5%の片側検定を行った．

表 2 にある通り，t 値の絶対値が t 境界値を上まわっているため，帰無仮説は棄却される．したがって，書籍データにおいて，システムが自動的に決定したものではあるけれでも，スタイル情報を付加したほうが閲覧者にとって見やすい結果となることがわかった．また，ニュース，統計情報のデータに対する検定結果も書籍データの場合と同様な結果が得られた．以上をふまえ，今後の実験では全てスタイル情報を付加することを前提に行うこととした．

表 3　フラットな表結果とシステム生成結果の見やすさの比較

		フラット	システム
書籍	平均	1.50	3.75
	t 値	-5.46	
	t 境界値	1.81	
ニュース	平均	1.36	3.00
	t 値	-3.87	
	t 境界値	1.81	
統計情報	平均	2.75	3.00
	t 値	-0.32	
	t 境界値	1.77	

表 4　フラットな表結果とシステム生成結果の構造の妥当性の比較

		フラット	システム
書籍	平均	2.38	4.25
	t 値	-6.05	
	t 境界値	1.77	
ニュース	平均	2.00	3.38
	t 値	-3.66	
	t 境界値	1.76	
統計情報	平均	3.38	3.50
	t 値	-0.18	
	t 境界値	1.77	

5.2.3　実　　験

実験では，まず，被験者に通常のフラットな表構造をもつ結果と，データ特性に基づいて生成した SuperSQL 質問文によって得られた結果を閲覧してもらった．その上で，それぞれに対し見やすさの観点と構造の妥当性の観点で 5 段階評価をしてもらった．構造の妥当性とは，外見的な美しさではなく，表構造そのものがそのデータ (書籍，ニュース，統計情報) を閲覧するに際して妥当であるかという観点であり，1:非常に不適切，2:不適切，3:妥当，4:適切，5:非常に適切，として点数をつけてもらった．これらの手順を書籍，ニュース，統計情報の 3 つ全てについて行った．

実験の結果を表 3 と表 4 に示す．表 3 は，フラットな結果とシステムが生成した結果の見やすさにおける評価点数を比較し，t-検定を行ったものであり，表 4 は構造の妥当性について同様のことを行ったものである．なお，t-検定は帰無仮説を「フラットな結果とシステムが生成した結果の評価点数の平均に差はない」，対立仮説を「システムが生成した結果の評価点数の平均の方が高い」として，有意水準 5%の片側検定を行った．

表 3 および表 4 からわかるのは，書籍，ニュースに関しては見やすさ，構造の妥当性のいずれにおいても t 値の絶対値が t 境界値を上まわっているため，帰無仮説は棄却される，ということである．すなわち，書籍データ，ニュースデータに関してはシステムが生成した結果の方がフラットな結果よりも，見やすさ，およびデータ閲覧の際の構造としての妥当性において優れているといえる．一方で，統計情報に関しては見やすさ，構造の妥当性のいずれにおいても帰無仮説を棄却するにいたらなかった．これは，統計情報はほとんどの属性が数値型であり，注視属性を指定しないかぎり分類表示されないため，フラット

表 5　リンクを用いた結果と用いない結果の見やすさの比較

		リンクあり	リンクなし
書籍	平均	3.62	3.25
	t 値	1.42	
	t 境界値	1.70	
ニュース	平均	3.13	2.00
	t 値	3.92	
	t 境界値	1.71	

表 6　リンクを用いた結果と用いない結果の構造の妥当性の比較

		リンクあり	リンクなし
書籍	平均	3.56	3.38
	t 値	0.44	
	t 境界値	1.70	
ニュース	平均	3.31	2.38
	t 値	2.89	
	t 境界値	1.71	

な表とほぼ同じ構造の結果が生成されたからであると考えられる．

続いての実験として，被験者に注視属性を選んでもらい，表のレイアウトを再構成した後，その結果についてこれまでと同様に見やすさと構造の妥当性の観点から点数をつけてもらった．さらに，表示結果にリンクが含まれている場合については，同じ注視属性を選択した状態でリンクを用いないで生成した結果に対しても点数をつけてもらった．

その後，先の実験と同様にフラットな表構造をもつ結果とシステムが生成した結果について比較を行った．その結果，先の実験と同様に書籍データおよびニュースデータでは有用性を示せたが，統計情報データに関しては通常のフラットな表構造と比べて統計的に有意な違いを見出せなかった．したがって，統計情報データを除いては注視属性を導入した場合でもフラットな表より見やすく適切なレイアウトで情報を提供できているといえる．しかしながら，実験と同時に行ったアンケートでは一部の注視属性を指定した際に，意図しない結果になったという報告を受けた．こうした検討項目については 5.3 節で述べる．

表示結果をリンクを用いて生成した場合と，リンクを用いず生成した場合の比較については，表 5 と表 6 に示す．これらは，それぞれの場合において評価点数を比較し，t-検定を行ったものである．t- 検定は帰無仮説を「リンクを用いた結果と用いない結果の評価点数に差はない」，対立仮説を「リンクを用いた結果の評価点数の平均の方が高い」として，有意水準 5%の片側検定を行った．

結果として，書籍データに関しては見やすさ，構造の妥当性のいずれにおいても帰無仮説は棄却されなかった．一方，ニュースデータに関しては逆に見やすさ，構造の妥当性のいずれにおいても帰無仮説が棄却され，リンクを用いたほうが視認性の高いという結果となった．これらの原因としては，書籍データは一冊の本データに必要な領域自体がそこまで大きくないために，カテゴリなど集約に用いる集約対象の属性との関係が把握できる一方で，ニュースデータでは本文の長さにともなって一件のニュースに大きな領域を必要とするために，集約に用いる属性との関係を把握できないレイアウトになってしまう，ということがあげられる．このようなケースの一例として，注視属性を大分類に設定しニュースデータ表示した場合の画面例を図 4 と図 5 に示す．

図 4　ニュース (リンクあり)

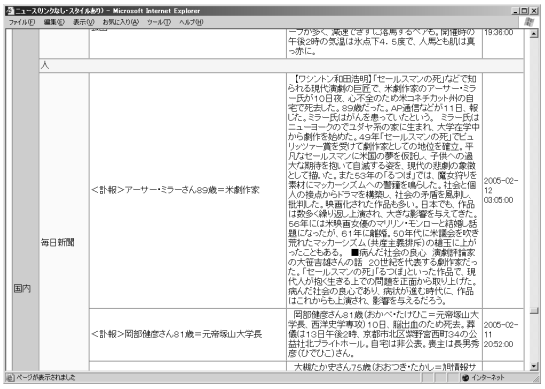


図 5　ニュース (リンクなし)

5.3　検　　討

以下では，実験の際に被験者から得られた意見や現在考えられる本手法の問題点や検討すべき点について議論を行う．

5.3.1　注視属性の反映方法

本手法では，注視属性を集約対象選択の際に反映させるため，パラメータ α を用いて重みをつけているが，注視属性として選んだにも関わらず，最も外側で集約して表示されないのは不自然であるという意見があった．さらには，注視属性として選んだ以外の属性が最も外側でリンクになった場合，意味性が大きく崩れてしまった状態となる．したがって，注視属性が選ばれたときは強制的に分類表示を試み，それ以外の属性に関して集約優先度を求めて構造を決定していくという対策を取るべきかもしれない．

5.3.2　リンクの必要性と対象

集約優先度が閾値以上となる属性が 3 つ以上ある場合，出力結果にリンクが含まれることになるが，そのリンクの位置が不適切であるという意見があった．ニュースデータなど，本文や詳細項目まで一度に表示してしまうと視認性が低下してしまうのである．したがって，単純に外側で集約するためにリンクを

用いるのではなく，意味的に分割可能なところでリンクを作り，詳細情報をその先で表示するという方法も検討する必要がある．

また，そもそもリンクを用いること自体が視認性の低下につながるという意見もあった．確かにリンクを用いないでも画面に優に収まるにもかかわらず，リンクを用いて表示してしまうと視認性を下げかねない．しかし，5.2.3 節の実験の際に述べたように，集約した属性ごとに必要な表示領域が大きい場合，すなわち，ある一つの値に対して多くの件数のデータが集まってしまうか，一件のデータの表示に要する領域が大きい場合，リンクの利用は視認性の低下に大きく貢献する．したがって，この表示領域の如何によってリンクを用いるか用いないかを決定するという手法の導入が今後の課題になると思われる．

5.3.3 最適なパラメータの決定

現在は数値特徴度，集約優先度の閾値や注視属性に対して適用するパラメータ α を経験的に決定してしまっている．実験の際に定めた閾値とパラメータは結果を見るかぎり概ね機能していたと思われるが，これらパラメータの最適な値の導出に関してさらに追求する必要がある．あるいは，この閾値とパラメータに関しては，サイト作成者の裁量にまかせられるような仕様にしてしまうといった方法も考えられる．

5.3.4 FROM 句の利用

本手法は，問合せ結果のインスタンスに着目し，TFE を決定している．しかしながら，単にそれだけにとどまらず，属性がどのテーブルのものなのか，またそのテーブルではどのような意味をもつかといった情報を FROM 句や関係スキーマから推測することで，構造決定に生かすことも考えられる．例えば，5.3.2 で述べたような，リンクを用いて分割する場所を特定する場合に，その属性がどのテーブルのものであるかを考慮することは意味のあることであると思われる．

5.3.5 パフォーマンス

現在の実装では，閲覧者がサイト作成者の用意した本処理系を含む PHP ファイルにアクセスするたびに，SuperSQL 質問文を生成し，さらにそれを問い合わせて結果を得ている．これは非常に非効率であり，生成される SuperSQL 質問文が同一で，かつデータベースに変化がなければ，新たに問い合わせる必要はない．したがって，一度生成された SuperSQL 質問文をキャッシュとして保持したり，データベースの状況を監視したりするような管理機構が必要であるが，これについては，[6] や [7] で提案されている手法を応用することで解決されると思われる．

6. 関 連 研 究

まず，SuperSQL を利用して，閲覧者の画面表示領域に応じて動的にレイアウトを決定する ACTIVEW があげられる [2]．ACTIVIEW は，動的に SuperSQL の TFE を決定するという点では本研究と類似するが，あくまで携帯端末や PC の画面といった様々なデバイスに適した表示を行うためにレイアウト構造を変換しているのであって，本研究のようにデータの特性に応じてそれに適したレイアウト構造を決定するアプローチとは異なる．

次に，XML データに対して問合せとレポート出力を行う QURSED とその作成ツールである QURSED Editor について述べる [3]．このシステムでは，開発者が GUI の QUSERD Editor を用いて，XML Schema と HTML ページからレポート出力のための質問文のセットを作成する．このとき，質問文のセットの作成には Tree Query Language と呼ばれるモデルに基づいて行われ，これが XQuery に変換されて HTML のレポート出力が行われる．QURSED Editor は本研究の処理系と異なり，開発者に多くの裁量権が与えられていて，様々な質問文を実行可能なフォームを用意できたり，またそれを多様に表示することができたりする．しかしながら，逆にそれが大きな負担となるケースも多々あるため，本システムのようにデータの特性に応じて自動的にレイアウトを構成するアプローチには意義がある．

そして，様々なテーブル操作を GUI システム上で行い，データベースのレポートや Web ページを出力する Table Presentation System について述べる [1]．TPS では数学的なモデルに基づき，通常のフラットな表を好みのレイアウトに変換していく *programming by example* というアプローチを取っている．これは，通常のプログラミング作業や先の QURSED Editor に比べると開発者の負担は軽減されていると思われる．しかし，エンドユーザである閲覧者にはレイアウトを変更する手段が提供されておらず，構造自体は静的なものしか出力することができない点で本研究とは異なる．

7. おわりに

本研究では，通常の SQL 質問文の問合せ結果のデータ特性を定義し，それに基づいて SuperSQL 質問文を自動生成する手法を提案した．評価実験を通して，サイト作成者がデータを構造化するための負担なしに，閲覧者の期待するレイアウトでデータを出力することが可能になったことを示した．

文　献

[1] W. Chen, K. Chung, “A Table Presentation System for Database and Web Applications”, *Proceedings of IEEE EEE ’04 International Conference on e-Technology, e-Commerce and e-Service*, pp. 492-498, 2004

[2] Y. Maeda, M. Toyama, “ACTIVIEW: Adaptive data presentation using SuperSQL”, *Proceedings of the VLDB ’01*, pp.695-696, 2001.

[3] Y. Papakonstantinou, M. Petropoulos, V .Vassalos, “QURSED: Querying and Reporting Semistructured Data”, *Proceedings of ACM SIGMOD ’02 International Conference of Management of Data*, pp. 192-203, 2002

[4] SuperSQL: http://ssql.db.ics.keio.ac.jp/

[5] M. Toyama, “SuperSQL: An Extended SQL for Database Publishing and Presentation”, *Proceedings of ACM SIGMOD ’98 International Conference on Management of Data*, pp. 584-586, 1998

[6] 有澤達也, 石川恭子, 遠山元道, “SuperSQL 処理系における INVOKE 関数に対するキャッシュ機構”, 情報処理学会研究報告 *DBSJ-131-037*, 2003

[7] 有澤達也, 石川恭子, 遠山元道, “SuperSQL の INVOKE 処理における中間データのキャッシュ”, データ工学ワークショップ *DEWS2004*, 2004