

# SuperSQLにおける時間連結子の実装

笹田麻衣子<sup>†</sup> 遠山 元道<sup>††</sup>

<sup>††</sup> 慶應義塾大学理工学部情報工学科 〒 223-8522 神奈川県横浜市港北区日吉 3-14-1

E-mail: <sup>†</sup>mai@db.ics.keio.ac.jp, <sup>††</sup>toyama@ics.keio.ac.jp

**あらまし** 近年インターネットの普及により、WWWを通じてデータベースを参照するニーズが高まっていることから、出力媒体と多様なレイアウト表現の指定を可能にするSQL拡張言語、SuperSQLによるHTML出力の研究を行ってきた。SuperSQLによるHTML出力においてはこれまでレイアウト指定のために、水平連結子、垂直連結子、深度連結子の実装されていた。しかしデータの連結方向に対して表示領域が狭い場合には画面を手動でスクロールする必要があった。また動画のキャプチャ画像など、時間方向に連結することによって意味を持つデータの適切な提示が困難であった。そこで本論文では新たに時間連結子の実装を行い、データベース問い合わせ結果を画像アニメーションやスライドショーを行うWWWページとして出力可能にしたことを報告する。

**キーワード** SuperSQL, データベース出版, HTML, 動画

## Implementation of the sequencing operator in SuperSQL

Maiko SASADA<sup>†</sup> and Motomichi TOYAMA<sup>††</sup>

<sup>††</sup>Department of Information and Computer Science, Faculty of Science and Technology,  
Keio University

Hiyoshi3-14-1, Kouhoku-ku, Yokohama-shi, Kanagawa, 223-8522 Japan

E-mail: <sup>†</sup>mai@db.ics.keio.ac.jp, <sup>††</sup>toyama@ics.keio.ac.jp

**Abstract** Since demand of referring to a database via WWW is increasing, we have developed HTML outputs in SuperSQL. SuperSQL is the SQL extensible language which can specify output structure and media. Three operators, which are horizontal, vertical and depth connectors have already implemented for layout specification in HTML outputs. However, when display area is too narrow against the direction of data connection, we have to scroll window manually. Moreover, we were not be able to express the time relationship between screen shots of movies with current three operators. In this paper, we report that we have implemented sequencing operator, which enable to output the animation HTML pages.

**Key words** SuperSQL, Database Publishing, HTML, Movie

### 1. はじめに

近年インターネットの普及により、WWWを通じてデータベースを参照するニーズが高まっていることから、出力媒体と多様なレイアウト表現の指定を可能にするSQL拡張言語、SuperSQLによるHTML出力の研究を行ってきた [2]。HTML出力においては実用化に向けた高品質化の研究を行っており、冗長なネストの除去や多様な装飾子の実装により、幅や背景色の指定、画像データの表示などが可能となっている。これまではレイアウト指定演算子として、横連結子、縦連結子、ハイパーリンク連結子の実装されていた。

しかし、動画のキャプチャ画像など、時間方向に連結することによって意味を持つデータの適切な提示が困難であった。ま

た、データ量が多く、データの連結方向に対して表示領域が狭い場合には画面を手動でスクロールする必要があった。これまでのSuperSQL処理系ではデータベース問い合わせ結果をX-window上に動画像として出力するための時間連結子やSMILなど動画像メディアに対する時間連結子は実装されている [4] [5]。

本論文ではSuperSQLによるHTML出力の表現力向上のため、HTMLにおける時間連結子を提案する。以下、2.章ではSuperSQLについて述べ、3.章では本論文で提案する時間連結子の定義、実現の方法を述べる。4.章では実装した時間連結子を用いた出力結果を示し、5.章でその評価と今後の課題について述べる。最後に6.章で結論を述べる。

## 2. SuperSQL とは

SuperSQL は SQL を拡張したワンソースマルチユースを実現する言語である。その質問文は SQL の SELECT 句を GENERATE< media >< TFE > の構文をもつ 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 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。なお、ここからは HTML 出力について述べる。

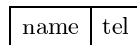
### 2.1 結合子

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 3 種類がある。括弧内は左がクエリー中の演算子を示し、右がレイアウト式を示す。

- 水平結合子 ( , : C1)

データを横に結合して出力。

例： Name , Tel



- 垂直結合子 ( ! : C2)

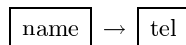
データを縦に結合して出力。

例： 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



### 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返して表示する。また反復子は単に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]

とすると、単に各々の一覧が表示されるだけで互いの関連は失われるが、

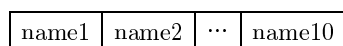
[科目名![学籍番号, 評点]]!

とした場合には、その科目における学生の評点一覧が表示される。以下、反復子の種類について述べる。

- 水平反復子 ( [ ] : G1)

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す。

例： [Name],

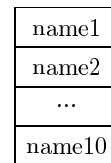


- 垂直反復子 ( [ ] ! : G2)

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す。

返す。

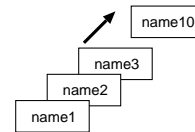
例： [Name]!



- 深度反復子 ( [ ] % : G3)

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に繰り返す。

例： [Name]%



### 2.3 装飾子

SuperSQL では関係データベースにより抽出された情報に、文字サイズ、横幅、画像ディレクトリなどの情報を付加できる。これらは装飾演算子 (@) によって指定することが可能である。

< TFE > @ { 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. 時間反復子と時間結合子

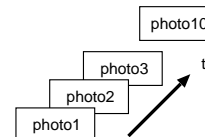
この章では本論文で新たに実装を行った HTML の時間連結子について述べる。

### 3.1 時間反復子と時間結合子の定義

- 時間反復子 ( [ ] # : G4)

時間反復子は囲んだ属性の値がデータベースにあるだけ、一定時間ごとに繰り返すことを指定する。大括弧の中には単一の属性だけでなく、他の結合子によって結合された複数の属性から成る TFE を含むことができる。ただし、時間結合子の入れ子は許されない。

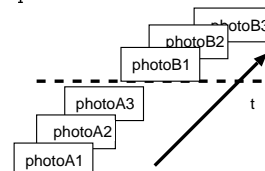
[photo]#



- 時間結合子 ( # : C4)

時間結合子は時間反復子で囲まれたデータを 1 つのインスタンスと考え、そのインスタンス同士を時間軸方向に結合する。

[photoA]# # [photoB]#



### 3.2 時間反復に関する装飾子

時間反復を行う SuperSQL クエリーにおいて、反復子と共に以下の装飾子を用いることによってアニメーションに関する各パラメータの指定を行うこととする。

#### 3.2.1 delay

1 データ、もしくは 1 画面に同時に表示するデータの集合あたりの表示時間を表し、動画においてはフレームレートとも呼ばれる。1/1000 秒単位で指定し、デフォルト値は 1000 とする。

#### 3.2.2 loop

データをあるだけ表示し終わった場合に最初のデータに戻って繰り返しアニメーション表示を行わせるかどうかを指定するパラメータ。繰り返す場合は loop="yes" , 繰り返さない場合は loop="no" とする。デフォルト値は yes とする。

複数のパラメータ指定を行う場合は以下のように","で区切る。

```
[name, photo]#@{delay=600, loop=no}
```

また、SuperSQL クエリーの中で時間反復が一番外側の反復子である場合は、以下のように時間反復に関する装飾子とテーブル全体に関する装飾子を併記することが可能である。

```
[name, photo]#@{delay=600, border=1}
```

### 3.3 実現方法

WWW ページにおいてアニメーションを行う方法として、データが画像の場合はアニメーション GIF に変換するという方法がある。また、Flash アニメーションや SMIL による記述がある。しかし今回の SuperSQL 処理系の開発においては以下の条件を満たすことを優先した。

- 他のアプリケーションによる画像処理等が不要
- 画像だけでなく文字のアニメーションも可能
- 閲覧に特別なプラグインを必要としない
- 既存のコード生成部からの拡張で実装が可能

以上の点から、本研究ではアニメーション手法として JavaScript を用いた。つまり、これまでの SuperSQL 処理系で出力されていた HTML のヘッダ部分にアニメーションを行う JavaScript を埋めこみ、本体部分をもそのスクリプトが実行可能なように書き換えるという処理を追加することで実現した。

#### 3.3.1 時間反復子

ここではまず、時間反復の実現方法について述べる。例として、SuperSQL で時間反復子を用いた場合に生成される HTML ファイルの構造を図 1 に、そのソースを図 2、ヘッダに埋めこむ JavaScript を図 3 に示す。この例では、GENERATE 句が以下のようなクエリーを実行したとする。

```
[date ! photo]#@{delay="1000"}
```

図 2 の HTML の <HEAD></HEAD> 内部に図 3 のスクリプトを記述し、<body OnLoad="anime()"> とすることで、ページの読み込み完了時に JavaScript 内の anime() という関数を呼び出して実行させる。HTML の <body> 部分には表示する最初の値のみを記述し、これを表示のテンプレートとする。

テーブルの各セル (<td></td>), もしくは画像表示領域

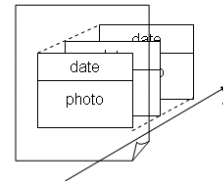


図 1 時間反復に相当する HTML の構造

```
<html>
<head>
  <title>sample.html</title>
  この部分に JavaScript を記述する
</head>
<body OnLoad="anime()">
  <table border=1 cellspacing=2><tr>
    <td><span id=s0>
      <font color=blue>名前 1</font>
    </span>
  </td><img src=img/写真 1.jpg name=s1</td>
</tr></table>
</body>
</html>
```

図 2 出力 HTML ファイルの構造

(<img src=...>) の 1 つ 1 つを span とよび、各 span に識別子として spanId を割り当てる。SuperSQL によるデータベースからの出力を考えた場合、span はクエリーに記述した属性の数と同数になる。この例ではクエリーが [name , photo]# であったので span が 2 個存在し、それぞれ s0, s1 という spanId が割り当てられている。スクリプトの配列や関数はこの span ごとに作成されることになる。また、spanId の割り当てに関しては、span が画像データの場合と文字データを表示する通常のセルの場合では指定の方法が以下のように異なる。

- 画像の場合

```
<img src=画像ファイル名 name=spanId>
```

- 文字表示セルの場合

```
<td><span id=spanId>文字データ</span></td>
```

図 3 のスクリプトではまずデータベース問い合わせ結果からデータの属性の種類(つまり span) ごとに配列、delay などの宣言を行う。次に各 span に相当する配列に全てのデータの表示タグを文字列として格納し、配列の要素数からデータの個数を得る。データが画像の場合には先行読み込みを行う。anime() はアニメーションを行う関数であり、配列に格納したデータを delay ごとに、対応する span に表示させる。最初の 2 行がデータを span の表示させる処理である。この部分の記述はデータが画像と文字の場合で以下のように異なる。

```
document.spanId.src = 画像配列 [num].src;
```

```

<script language="JavaScript">
  <!--
    o0 = new Array();
    o1 = new Array();
    img1 = new Array();
    var num = 0;
    var delay = 1000;
    o0[0] = '<font color=blue>名前 1</font>';
      :
    o0[9] = '<font color=blue>名前 10</font>';
    o1[0] = 'img/写真 1.jpg';
      :
    o1[9] = 'img/写真 10.jpg';
    var max = o0.length;
    for(i=0;i<max;i++){
      img1[i] = new Image();
      img1[i].src = o1[i];
    }
    function anime() {
      document.getElementById('s0').innerHTML = o0[num];</html>
      document.s1.src = img1[num].src;
      num++;
      if(num > (max-1)) num = 0;
      else setTimeout("anime()",delay);
    }
  //-->
</script>

```

図 3 出力 HTML のヘッダ中に埋め込む JavaScript

```

document.getElementById('spanId').innerHTML = 文
字配列 [num];

```

setTimeout("anime()",delay) という記述は、delay の時間だけ待ってから anime(), つまり自分自身を実行するという再帰的な関数呼び出しである。何番目のデータであるかを表す変数 num が最大値に達したとき、num を初期化して最初から再び表示を繰り返すか、表示を終了するかはクエリーでの loop の指定によって変える。

### 3.3.2 時間結合子

時間反復したデータインスタンス同士を時間結合するという概念に相当するのは図 4 のような表示を行う HTML である。その仕組みは、時間反復された各データインスタンスを個別の HTML ファイルとして出力し(子ページ)、親ページの中のインラインフレーム内に子ページを順に表示させていくというものである。

このような HTML のソースを図 5 に示す。図 5 を見ると、JavaScript を用いるという点は先に述べた時間反復と同様だが、JavaScript 内で配列に格納するのが子ページの HTML ファイル名である点と、<body>タグ内部はインラインフレームの記述のみであるという点が異なる。

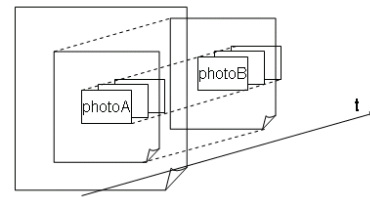


図 4 時間結合に相当する HTML の構造

```

<html>
<head>
<title>sample.html</title>
この部分に JavaScript を記述
</head>
<body OnLoad="changePage()">
  <iframe src="sample0.html" width=100% height=100%
    id="iframe">
    iframe is not available.
  </iframe>
</body>

```

図 5 時間結合に相当する HTML の構造

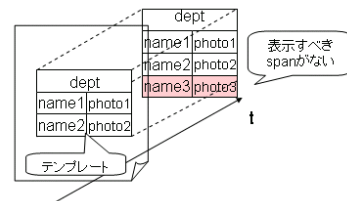


図 6 反復子が原因で正しく表示されない例

スクリプトの流れを述べると、まず最初にページ名、各ページの delay(pageDelay) を格納する配列を宣言する。pageDelay とは子ページがアニメーションし終わるのにかかる時間、つまり”子ページの delay × データ数”であり、子ページによって異なるためこのように配列を用いる。また、何番目の子ページかを表す pageNum も宣言しておく。次に、ページ名の配列に、全ての子ページのファイル名を格納する。このとき子ページのファイル名は自動的に付与される。また、子ページの pageDelay を計算し、配列に格納する。changePage() はページ切り替え関数であり、時間反復のアニメーション関数とほぼ同様である。

### 3.4 時間反復子における制約

3.3.1 で示したように、JavaScript を用いたアニメーション表示において<body>部分には最初に表示するデータ(または結合されたデータの集合)のみを記述し、spanId を指定する。データベースから SuperSQL によって時間反復を用いた HTML を生成する場合、このことが原因となって正しく表示されないケースが考えられる。(図 6)

また、3.1 で示した時間結合子の意味付けも含めたうえで、時間反復子の使用に際して以下の制約を設けることとする。

- 時間反復子の内部に反復子を含めることはできない

[photo],

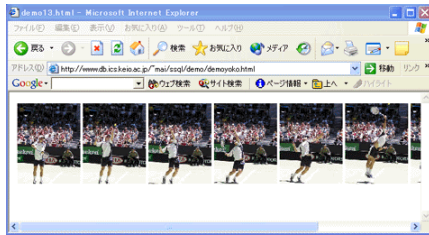


図 7 従来の水平反復による出力結果

[photo]#

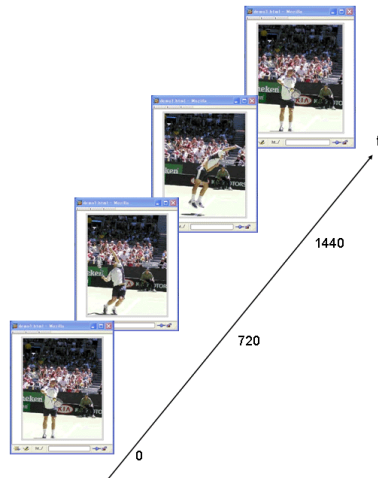


図 8 本研究における時間反復による出力結果

- 時間反復子の内部に時間結合子を含めることはできない

#### 4. 実行結果

本研究における実行結果の例として、動画のキャプチャ画像に対する SuperSQL 問い合わせ結果を以下に示す。図 7 は従来の水平反復を用いた SuperSQL の出力結果である。図 8 は本研究において実装した時間反復を用いた出力結果のスナップショットである。また、図 9 は時間反復と時間結合を用いた出力結果である。

#### 5. 評価・検討

##### 5.1 表現力について

まず、元のデータの持つ時間的なつながりをいかに適切にわかりやすく表現できるかといった表現力の観点から評価を行う。

SuperSQL の HTML 出力における既存の水平、垂直、深度の 3 種類の連結子のみを用いた出力では、時間的なつながりを持つデータもこれらの方向に連結して空間的に配置せざるを得なかった。(図 7) 一方、本論文で実装を行った時間連結子を用いた図 8 では、データを一定時間ごとに切り替えてアニメーション表示させることができ、ばらばらのデータとして格納されていた画像があたかも 1 つの動画であるかのように見せることができ、元のデータの持つ時間的な意味まで表現可能となったといえる。

[photo]# # [photoB]#

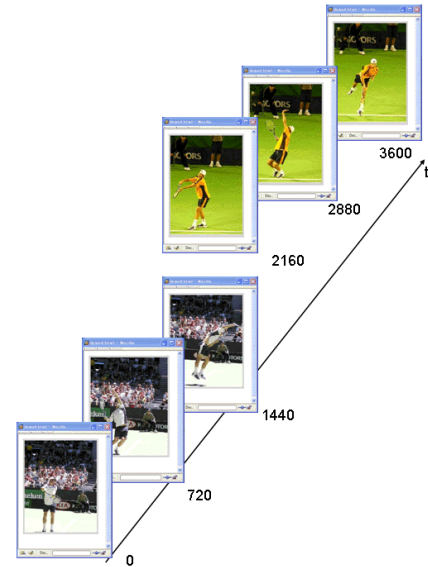


図 9 本研究における時間結合による出力結果

また、時間反復子については 2. 章で述べた他の結合子との組み合わせが可能であり、GENERATE 句が以下のような SuperSQL クエリーの記述も行える。

```
[date ! {title , photo}]#
[name ! [photo]#!
[photoA]#@[delay=200] # [photoB]#@[delay=300]
```

以上のことから、本研究による時間連結子の実装により、SuperSQL の HTML 出力における表現力が高まったといえる。

##### 5.2 表示領域について

従来の SuperSQL ではデータを空間的に配置して出力する際に多くの表示領域を必要としていたことから、時間連結子の実装によってそれがどのように改善されたかについて述べる。

従来の横反復、縦反復を用いた場合、データが多いほどその反復方向に長い表となって出力される。これに対し、時間反復を用いた場合はデータベースからの結果として 1 タプル分のデータを表示する領域さえあればよい。つまり、横(縦)反復で必要とした領域 / タプル数 (ピクセル) で済む。このことはデータベースからのタプル数が多ければ多いほど時間反復が有効であることを示す。画面が小さい環境でも手動でスクロールすることなくデータの閲覧が行えるということは、表現力の向上とともに時間反復の重要な特徴といえる。この点を活かして携帯電話や PDA 等の携帯小型端末でのデータの受動的閲覧といった応用も可能と考える。

##### 5.3 時間反復同士の時間結合について

これは、時間反復したものを 1 つのインスタンスと考え、そのインスタンス同士を時間結合するというものであり、クエリー表記は

```
[photoA]# # [photoB]#
```

のようなものである。

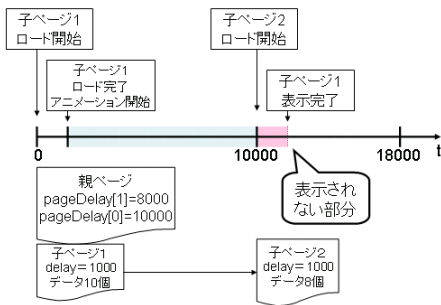


図 10 時間結合における遅延の原因

前述の通り、本研究ではこのような出力を実現するためにインラインフレーム内のページ切り替えを用いており、出力 HTML から JavaScript 中のアニメーション関数を呼び出す部分では、`<body OnLoad="anime()">`のように記述している。これはページの読み込みが完了した時にアニメーション関数を実行するという指定である。

ここで問題となるのは、インラインフレーム内の切り替えを行う親ページにおいても、時間反復インスタンスのアニメーションを行う子ページにおいても`<body OnLoad="anime()">`としてしまうためにページ切替時に遅延が発生するという点である。親ページの delay 指定ではこの遅延を考慮していないため、結果としていくつかのデータが表示されずに次の子ページの表示が始まってしまうということが起こる。これを図 10 に示す。

この状況は、特に表示するデータがサイズの大きい画像であり、delay が小さい場合 (1000ms 以下程度) に顕著に現れる。この問題はユーザに大きな制約をもたらすものであり、データベース出力の手段として十分であるとは言い難い。よって、インラインフレームを用いた方法は妥協策とし、これに代わる方法を検討する必要がある。

#### 5.4 ブラウザへの対応について

本研究で出力した JavaScript の各処理に対するブラウザ対応状況は以下の通りである。

表 1 主要ブラウザの JavaScript 対応状況

	画像切替	文字切替	インラインフレーム
Internet Explorer	4 以降	4 以降	4 以降
IE(MAC)	4 以降	4 以降	4 以降
Netscape	3 以降	6 以降	6 以降
Mozilla	○	○	○
Opera	×	×	×

本研究では実際に Internet Explorer 5.5 以降と Mozilla 1.0 で出力 HTML の動作確認を行ったところ、現在使用されているほとんどのブラウザでは問題なく動作した。古いバージョンのブラウザへの対応は特に考えていない。

#### 5.5 今後の課題

##### 5.5.1 時間結合の代替策について

セクション 5.3 で述べた遅延の問題を解決するためには、ページ delay にページの読み込みが完了するまでの時間を含めな

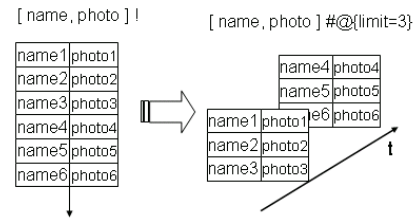


図 11 時間折り畳みの概念

ればならない。もしくは最初の時間反復が全ての値を表示し終えた瞬間に次のページが読みこまれるような方法が必要である。現段階ではこれらを実現する方法を検討中である。

##### 5.5.2 表示領域に対する適応について

データベースからの検索結果のタプル数が多い場合、SuperSQL で生成したページの読み込み時間がかかってしまう。また、それを閲覧する側にとっても手動で画面をスクロールする必要があり不便である。

そこで、まずページの時間折り畳みの導入を考える。例えば「結果が 20 行を超えたら時間折り畳みを行う」という形でユーザがクエリー中に記述し、結果を 1 ページに 20 行ずつというように分割して生成し、それをスライドショーのように時間切り替え表示するというものである。その概念を簡単に図 11 に示す。

また、ページの自動スクロールを導入し、これらを組み合わせることによって、さらに視聴型のデータベース出力が可能になるであろう。これらはいずれもクエリー中では装飾子として実装するのが望ましいと考える。

##### 5.5.3 時間連結子におけるユーザ制約について

本研究において実装した時間連結子は、セクションで述べたようにユーザ制約を設けている。しかし、視聴型のデータベース出力を実現するためには、時間反復、時間結合が自由に組み合わせ可能である必要がある。

まず、時間反復の内部に反復子が含まれていたとしてもアニメーション可能であれば、スライドショーとして出力する際にデータの構造まで考慮する必要がなくなり、自由度が高まるだろう。

また、本研究では時間結合の意味付けとして、時間反復したインスタンス同士を時間軸方向に並べるのみとしている。これを更に拡張し、時間反復以外のインスタンスを時間結合することができるになれば、やはりスライドショーとしての表現力が高まるであろう。例えば、GENERATE 句が以下のような SuperSQL クエリーが実行可能になることが望ましい。

```
[title # [photo]##
```

## 6. 結 論

本論文では SuperSQL の HTML 出力においてデータを時間軸方向に連結し、データの持つ時間的なつながりを表現するための時間連結子を実装した。これにより、画像処理等のプロセスを経ることなくデータベースからの結果をアニメーション表

示する HTML ページとして出力することが可能となった。今後は時間結合のための新たな手法を検討し、また自動スクロールや時間折り畳みなどの機能を追加することによって時間連結子を拡張することを考えている。

## 文 献

- [1] SuperSQL: <http://ssql.db.ics.keio.ac.jp/>
- [2] Motomichi 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
- [3] T. Seto, T. Nagafuji and M. Toyama, "Generating HTML Sources with TFE Enhanced SQL," *Proc. ACM Symp. on Applied Computing(SAC '97)*, pp. 96-105, 1997
- [4] 多田光伸, 遠山元道, "SuperSQL を用いた動画プレゼンテーションの再構成提示" 電子情報通信学会データ工学ワークショップ, 2002
- [5] 田中宣行, 遠山元道, "TFE を用いた動画再生とレイアウト" データベースシステム研究報告, No.109, pp. 99-110, 1996
- [6] 近藤宏行, 灘本明代, 田中克己, "モバイル環境における検索エンジンの出力結果の再構成と提示" 情報処理学会研究報告, No.69, 2000-DBS-122-26, 2000
- [7] 半場方人著:『詳解 JavaScript 辞典』, 秀和システム, 2002
- [8] ハイパーテニスインフォメーション  
<http://hti.jp/>