

# SuperSQL の HTML 出力における時間連結子の実装

## Implementation of the Sequencing Operator in HTML Generator of SuperSQL

笹田 麻衣子<sup>†</sup> 遠山 元道<sup>‡</sup>

Maiko SASADA Motomichi TOYAMA

出力媒体と出力構造を指定可能な SQL 拡張言語, SuperSQL によるデータベース問い合わせ結果の HTML 出力においてはこれまでに水平, 垂直, 深度演算子が実装されており, 空間的に多様なレイアウト指定が可能であった. しかし, 気象衛星から送られてくる 1 時間毎の画像など, 時間的につながりを持つデータの提示には限界があった. また, 狭い画面領域では全てのデータを見るために手動でスクロールする必要が生じる. 本論文では SuperSQL の HTML 出力においてデータを時間軸方向に配置する時間連結子を実装し, SuperSQL による問い合わせ結果の, アニメーションやスライドショーとしての提示を実現したことを報告する.

We have developed a SQL extension named SuperSQL, which can specify output structure and media such as HTML, XML and PDF. We have already implemented layout operators, connectors and repeaters, for each of three dimensions: the horizontal, the vertical and depth for layout specification in HTML outputs by SuperSQL. These are capable to specify various spatial layouts. However, these spatial operators are unsatisfactory in cases like presenting hourly images from weather satellite. Also, we often have to scroll the window manually in order to view all the data when display area is too narrow. In this paper, we introduced the sequencing operator, which arranges the data along the temporal axis in HTML, so that making the answers from database as animation HTML pages.

### 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に格納された情報を参照するニーズが高まっている. こうした背景を受け, 我々は出力媒体と多様なレイアウト表現を指定可能なSQL拡張言語, SuperSQLによるデータベース問い合わせ結果のHTML出力という研究を行ってきた[1][2]. SuperSQLのHTML出力については, 近年は実用化に向けた高品質化の研究が行われる一方, 複雑化に伴うユーザへの配慮としてGUIによるクエリ作成支援ツールの開発も行われてきた. これまではHTML出力のためのレイアウト指定演算子として, 水平連結子, 垂直連結子, 深度連結子が実装されており, これらの組み合わせによって空間的には多様なレイアウト

<sup>†</sup> 学生会員 慶應義塾大学大学院理工学研究科修士課程

[mai@db.ics.keio.ac.jp](mailto:mai@db.ics.keio.ac.jp)

<sup>‡</sup> 正会員 慶應義塾大学理工学部情報工学科

[toyama@ics.keio.ac.jp](mailto:toyama@ics.keio.ac.jp)

指定が可能である. しかし, データが動画のキャプチャ画像や気象衛星からの一連の画像である場合は, それらを連続したシーケンスとして扱い, 時間軸方向に連結して動画のように見せるのが効果的な提示方法である[3]. SuperSQLにおいてもデータの時間軸方向への連結という概念は存在し, X-Window上では実現されていた[4]が, 実用的に利用可能なHTML上での実現が望まれていた.

また, HTMLは紙面のように制限された領域ではないため, 出力するデータが多ければ多いほどサイズの大きなページが出力されてしまい, 画面が小さい場合には手動でスクロールする必要が生じていた. この問題については, 画面サイズに応じて最適なレイアウトに自動変換するACTVIEWの研究[5]などが行われてきた. しかし, SuperSQLを用いてデータを時間軸方向に連結し, スライドショーのように表示させることでWebの受動的視聴[6]を行うというアプローチも可能と考えられる.

こうした背景から, 本論文では新たにHTMLにおける時間連結子を提案し, その実現についての報告を行う.

### 2. SuperSQL とは

#### 2.1 システム概要

SuperSQL はワンソースマルチユースを実現する SQL 拡張言語である. 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, 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. なお, ここからは HTML 出力について述べる.

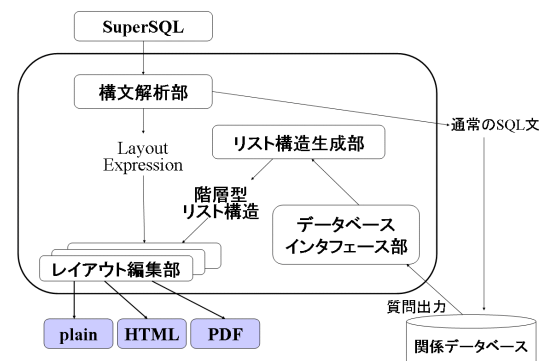


図1 SuperSQL 処理系  
Fig.1 SuperSQL System Configuration

#### 2.2 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.

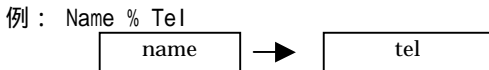
- 水平結合子( , : C1 ) データを横方向に結合.  
例: Name , Tel

name	tel
------	-----

- 垂直結合子( ! : C2 ) データを縦方向に結合.  
例: Name ! Tel

name
tel

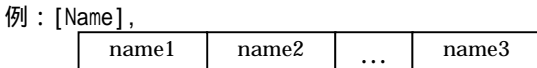
- 深度結合子( % : C3 ) データを奥行方向に結合 . 出力媒体が HTML の場合ハイパーリンクとなる .



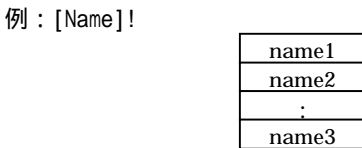
### 2.3 反復子

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出力する . 以下 , 反復子の種類について述べる .

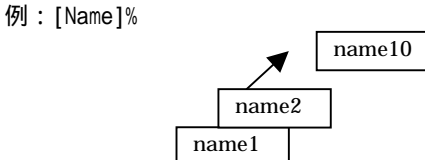
- 水平反復子( [ ] , : G1 ) データインスタンスがある限り , その属性のデータを横方向に繰り返す .



- 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方向に繰り返す .



- 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方向に繰り返す .



また反復子は単に構造を指定するだけでなく , その値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 .

### 2.4 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 , 配置 , 画像ディレクトリなどの情報を付加することができる . これらは装飾演算子( @ )によって以下のように指定する .

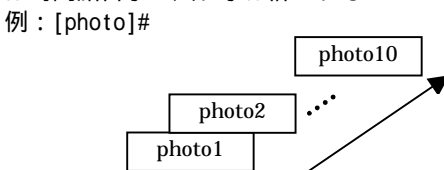
```
<TFE>@{<装飾指定>}
```

装飾指定は " 項目名 = 値 " として指定する . 複数の装飾指定を行うときは各々を " , " で区切る .

## 3. 時間連結子の提案

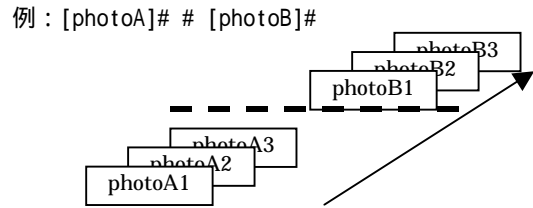
### 3.1 時間反復子と時間結合子の定義

- 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 , 今回の実装では時間結合子の入れ子は許されない .



- 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のに用いる .



### 3.2 時間反復に関するパラメータ

時間反復の詳細を決定するパラメータとして , 以下の2つをユーザが SuperSQL クエリ中で指定できるようにした .

- 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すパラメータで , 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 .

- loop  
最後までデータを表示し終え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 . また , SuperSQL クエリの中で時間反復が一番外側の反復子である場合 , 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 , loop=no , border=1 }
```

## 4. 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 が広く用いられている . また , Macromedia Flash によるアニメーションや 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ため , JavaScript によるアニメーションを用いた .

- 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

具体的には , 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 .

### 4.1 時間反復子の実現

時間反復子を含む SuperSQL クエリを実行した場合 , 図 2 のような構造を持つ HTML ページを生成させる . 図 2 は JavaScript によって画像アニメーションを行う HTML ページの簡単な例を示したもので , <head></head> の内部に JavaScript 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 内のアニメーション関数を呼び出して実行する .

他の反復子を用いた場合に出力される HTML では , 表示する全ての値を <body></body> の内部に記述するが , JavaScript によるアニメーションを行う場合には表示する最初の値のみを <body></body> の内部に記述し , 2 番目以降の

値は JavaScript 内の配列に文字列として記述する必要がある。ここでいう値とは、単にデータベースから抽出したデータインスタンスではなく、クエリで指定された文字スタイルや画像パスを反映したタグを含む文字列である。そこで既存のコード生成部が生成する HTML タグからこうした値部分だけを抽出する処理部を実装した。

SuperSQL によって生成する HTML ではテーブルの 1 つのセル、つまり<td></td>タグで囲まれた部分に 1 つの値を表示する。個々のセルを span という領域であるとみなす。時間反復子の内部に他の結合子によるレイアウトを含む場合は span が複数になるため、spanId という識別子を割り当てて区別する。このとき、セル内のデータが画像ならば図 2 のように直接画像に name として識別子を付与する。

```
<html>
<head>
<title>sample.html</title>
<script langage="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];
document.s1.src = img1[num].src;
num++;
if(num > (max-1)) num = 0;
else setTimeout("anime()",delay);
}
//-->
</script>
</head>
<body OnLoad="anime()">
<table border=1 cellspacing=2><tr>
<td><span id=s0>
<font color=blue>名前 1</font>
</span>
</td>
<td><img src=img/写真 1.jpg name=s1></td>
</tr></table>
</body>
</html>
```

図 2 時間反復子を用いた HTML の例

Fig.2 A Sample of HTML code using Temporal Repeater

### 4.2 時間結合子の実現

時間反復したデータインスタンス同士を時間軸方向に結合するため、時間反復された各データインスタンスを図 2 のような個別の HTML ファイルとして出力し(子ページ)、親ページ(図 3)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。これも時間反復と同様に JavaScript を用いた時間切り替え表示を用いるが、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></body>内部の記述はインラインフレームのみであるという点が異なる。

```
<html>
<head>
<title>sample.html</title>
<script language=JavaScript>
<!--
var page = new Array();
var pageDelay = new Array();
var pageNum = 0;
page[0] = 'sample0.html';
page[1] = 'sample1.html';
var pageMax = page.length();
pageDelay[1] = 2160;
pageDelay[0] = 2160;
function changePage(){
document.getElementById("iframe").src = page[pageNum];
pageNum++;
if(pageNum > (pageMax - 1)) pageNum = pageMax - 1;
else setTimeout('changePage()',pageDelay[pageNum]);
}
//-->
</script>
</head>
<body OnLoad="changePage()">
<iframe src="sample0.html" width=100% height=100% id="iframe">
iframe is not available.
</iframe>
</body>
</html>
```

図 3 時間結合子を用いた HTML の例

Fig.3 A Sample of HTML code using Temporal Connector

### 4.3 時間連結における構文上の制約

4.1 で示したように、JavaScript を用いたアニメーション HTML において<body></body>の内部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。例えば、

```
[dept ! [name, photo]!] #
```

のような SuperSQL クエリを実行した場合、各 dept に対する name と photo の組み合わせの数が異なるため、最初の値だけをテンプレートとして用いると値を表示すべき span が存在しない、またはいつまでも切り替わらずに表示され続けるといった状況が起こりうる。また、2.2 で示した現時点での時間結合子の意味付けも含めたくうえで、時間連結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むことはできない
- 時間反復子の内部に時間結合子を含むことはできない

### 5. 出力例

まず、既存の水平反復を用いたデータベースからの出力結果を図 4 に示す。次に本論文で提案、実装した時間反復を用いた出力結果を図 5 に示す。実際の HTML は以下の URL で閲覧可能である。

<http://ssql.db.ics.keio.ac.jp/DEMO/animation/dbsj.html>



図 4 水平反復による出力

Fig.4 An Example of Output using Vertical Repeater

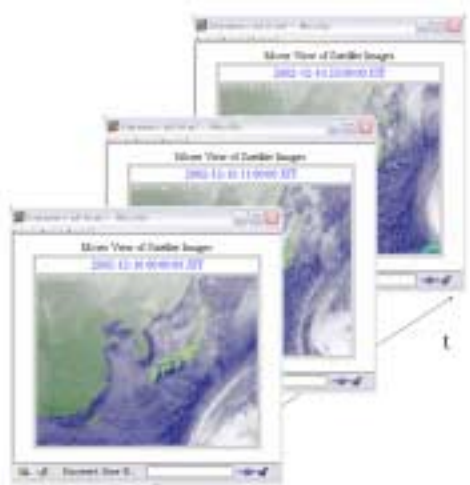


図5 時間反復による出力例

Fig.5 An Example of Output using Temporal Repeater

## 6. 評価・検討

### 6.1 表現力に関して

SuperSQL の HTML 出力における既存の水平連結子等を用いた出力では、時間的なつながりを持つデータも空間的に配置せざるを得なかった(図4)。一方、本論文で提案した時間連結子を用いた出力(図5)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、ばらばらに格納されていた静止画像を1つの動画として見せることができ、元のデータの持つ時間的な意味まで表現可能となった。また、時間反復子は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 クエリの記述も行える。

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

なお、これらの出力例も以下の URL にて閲覧できる。

<http://ssql.db.ics.keio.ac.jp/DEMO/animation/dbsj.html>

### 6.2 表示領域に関して

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方向に長い表となって出力されるのに対し、時間反復を用いた場合はデータベースからの結果のうち1タプル分のデータを表示する領域さえあればよい。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ようになったことは、表現力の向上とともに時間反復の重要な特徴といえる。この点を活かせば携帯電話や 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であろう。

### 6.3 改善すべき点

時間反復したものとを時間結合する場合、本研究ではインラインフレーム内のページ切り替えを用いた。出力 HTML において JavaScript 中の関数を呼び出す部分では、<body OnLoad="anime()">のように記述し、ページの読み込み完了まで待つ関数を実行する。親ページ、子ページ共にこのような指定を行っているにもかかわらず、親ページの delay 指定ではページ切り替え時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。こうした状況は、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、代替手段を検討する必要があると考えている。

## 7. まとめと今後の課題

本研究では SuperSQL の HTML 出力においてデータを時間軸方向に連結するレイアウト指定演算子、時間反復子と時間結合子を提案し、その実装を行った。その結果、画像処理等のプロセスを経ることなくデータベースの問い合わせ結果をアニメーションもしくはスライドショー形式で表示する HTML ページとして出力することを実現した。これにより、データの持つ時間的な関連の適切な表現を可能にしたばかりでなく、表示領域を大幅に削減することができた。

今後の課題としては、時間折り畳み、複数のアニメーション間での同期、情報量によるフレームレートの自動設定などの実現を検討している。

### [謝辞]

本研究の一部は、文部科学省の世界的研究教育拠点の形成のための重点的支援 21 世紀 COE プログラム「アクセス網高度化光・電子デバイス技術」の支援によるものである。

### [文献]

- [1] SuperSQL: <http://ssql.db.ics.keio.ac.jp/>
- [2] 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).
- [3] 菊池時夫, 喜連川優, "GMS-5 気象衛星画像データベースと統合視覚化システムの構築", 情報処理学会論文誌: データベース, Vol.42, No.SIG08 (TOD 10), pp.148-155 (2001).
- [4] 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).
- [5] Y. Maeda, M. Toyama: "ACTIVIEW: Adaptive data presentation using SuperSQL", Proceedings of the 27th International Conference on Very Large Data Bases(VLDB '01), pp.695-696 (2001).
- [6] 灘本明代, 近藤宏行, 田中克己, "携帯環境のための Web 検索結果の動的再構成と受動的視聴", 情報処理学会論文誌: データベース, Vol.42, No.SIG15 (TOD 12), pp.1-15 (2001).

### 笹田 麻衣子 Maiko SASADA

慶應義塾大学大学院理工学研究科修士課程在学中。2003 慶應義塾大学理工学部情報工学科卒業。データベースシステムの研究に従事。情報処理学会学生会員。日本データベース学会学生会員。

### 遠山 元道 Motomichi TOYAMA

慶應義塾大学理工学部情報工学科専任講師。博士(工学)。1984 慶應義塾大学大学院博士課程単位取得退学。主にデータベースシステムの研究に従事。IEEE Computer Society, ACM, 日本ソフトウェア科学会, 電子情報通信学会, 日本データベース学会会員。