SuperSQL の時間連結子による動的プレゼンテーション生成

笹田 麻衣子[†,☆1] 遠山 元道[††]

本論文では，SuperSQL を用いたデータベースコンテンツの動的プレゼンテーションについて述べる．Web の普及にともない，データベース検索結果のプレゼンテーションは静的な表構造の制約を脱し，スライドショーや動画などの形式が可能となった．しかし従来のレポートライタなどは静的な出力のみに対応し，データベース出力を動的に出力する一般的なフレームワークは存在しない．本研究では，SuperSQL に第 4 次元のレイアウト演算子，すなわち時間連結子を導入してこの問題を解決することを提案し，SWF（Flash ムービー）出力処理系の実装について報告する．

Generating Dynamic Presentation for Database Contents Using Sequencing Operators of SuperSQL

Maiko Sasada[†,☆1] and Motomichi Toyama[††]

Although diffusion of WWW has suggested movies and slideshows as forms of presentation of database contents, exisiting report writer for database does not support generating dynamic presentation. Consequently, no framework is capable of dynamic presentation of database contents at present. In this paper, we propose integration of sequencing operators, the forth dimensional operator, with SuperSQL, as a practical solution to this problem. Further, implementation of SWF (Flash Movie) generator will be discussed.

1. はじめに

近年インターネットの普及により，WWW（World Wide Web）を経由してデータベースシステムを利用することが標準となってきた．関係データベースを扱う場合，問合せ言語によってデータの検索を行った結果は 2 次元のフラットな表であり，それを WWW を通じてユーザに提示するには検索結果を HTML に変換するなどしなければならない．手作業でこの変換を行うとすれば，大量のデータを HTML に書き直すコストは膨大であり，データベースが更新されるたびに HTML ファイルも更新しなければ一貫性が失われるため，更新にかかるコストも大きい．そこで，データベースの検索結果を直接利用可能なメディアファイルとして出力する検索言語やレポートライタなどが数多く研究，実用化されている．

一方，ネットワーク環境の向上にともない，Web においてはストリーミング動画や広告バナーのように時間とともに刻々と表示内容が変化する動的なコンテンツが多用されるようになった．Macromedia® Flash®[13)] ☆2や QuickTime ☆3ムービー[4)]，SMIL (Syncronized Multimedia Integration Language)[15)] などが Web における動的コンテンツの代表格である．また，Microsoft® PowerPoint® ☆4プレゼンテーションの自動再生を展示するなど，Web 以外の環境でも動的なプレゼンテーションが用いられている場が数多くある．

こうした背景を受けて，データベースの検索結果を動的プレゼンテーションとして出力したいというニーズが高まっているが，現在のところそのようなフレームワークは存在しない．データベース出力において動的プレゼンテーションが必要な場合としては主に以下の 3 つがあげられる．

- **データが時間的な意味を含んでいる場合**

† 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University
†† 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University
☆1 現在，野村総合研究所
Presently with Nomura Research Institute

☆2 Macromedia および Flash は Macromedia, Inc. の米国またはその他の国における商標または登録商標である．
☆3 QuickTime は米国およびその他の国における Apple Computer Inc. の商標である．
☆4 Microsoft および PowerPoint は米国およびその他の国における Microsoft Corporation の商標である．

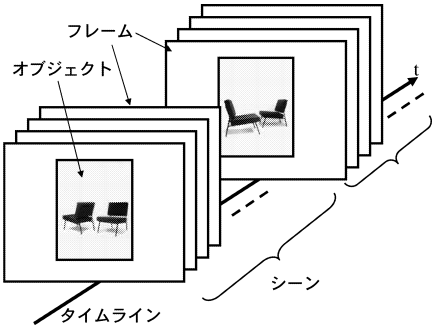


図 1 動的プレゼンテーションの構造
Fig. 1 Structure of dynamic presentation.

動画のキャプチャ画像のシーケンスや，気象衛星からの一定時間ごとの画像などを格納したデータベースなどを検索対象とする場合．

- **表示領域が狭い場合**
  限られた表示領域において多くの情報を提示したい場合，同じ領域を用いて異なるオブジェクトを表示することができる．
- **閲覧者の注目を集め効果的に見せたい場合**
  「動き」によって閲覧者の注目を集めやすく，視覚的に印象に残りやすい．

1.1 動的プレゼンテーションとは

動的プレゼンテーションの生成を目的とした時間連結モデル提案の前提として，動的プレゼンテーションの構成やプロパティについて定義を行っておく．動的プレゼンテーションの例を**図 1** に示す．なお，本論文でこれ以降に使用する用語の意味は以下のとおりである．

- **オブジェクト**
  プレゼンテーションを構成する 1 つ 1 つの要素．テキストや画像など．
- **タイムライン**
  プレゼンテーションが進行する時間軸．
- **フレーム**
  タイムラインにおけるプレゼンテーションの最小単位．
- **フレームレート**
  1 秒間に表示するフレームの数．
- **シーン**
  同じ場面を構成するフレームの集合．

動的プレゼンテーションを生成する手法として一般的に用いられているのは，Macromedia Flash などのムービーオーサリングツールを用いて 1 画面（シーン）ごとに画像やテキストオブジェクトを配置し，Flash ムービーを作成する方法や，ImageMagick[10] などの画像処理ソフトウェアを用いて，連続する複数の画像からアニメーション GIF を生成するといった方法である．しかし，こうした作業は非効率的であったり，メディアのフォーマットやシステムに対する知識が必須となったりするため，利用できるユーザが限られている．

そこで，本研究では Web やそれ以外の環境で利用可能な，汎用性の高い動的プレゼンテーションをデータベースから効率的に生成すること，またその手法に基づき，動的プレゼンテーションを表示領域の有効利用という観点で利用することを目的とする．具体的には，プレゼンテーションの内容となるオブジェクトをデータベースで管理し，簡単なクエリによって，データを検索した結果を動的プレゼンテーションとして生成するシステムの開発をめざした．

2. 関 連 研 究

本研究に関連する既存の研究を，以下の 2 つの観点からいくつかあげておく．

- 動的プレゼンテーションの効率的な生成
- 動的プレゼンテーションによる表示領域の有効利用

さらに，商用ツールやフリーソフトなどで動的プレゼンテーションと深い関係があるものについても述べておく．

2.1 動的プレゼンテーションの効率的な生成

データベースによってデータオブジェクトを管理し，効率的に動的プレゼンテーションを生成する手法については，マルチメディアデータベースを用いたマルチメディアドキュメントの自動生成という分野でいくつかの研究[1),7)~9)] が行われてきた．

Baral らは SQL を拡張し，マルチメディアデータベースの検索結果としてマルチメディアドキュメントを生成する SQL ＋ D という検索言語を提案した[5)]．X，Y の 2 次元だけでなく DISPLAY-WITH 句によってスクリーンの重なり（Z 方向）を指定し，SHOW 句，SYNC 句によってオブジェクトの時間的な配置を行う．

Gaggi らの schema modelling はマルチメディアオブジェクトの集合に対して，プレゼンテーションのテンプレートとなる XML スキーマを作成し，それを用いてマルチメディアドキュメントを自動生成するというものである[6),8)]．オブジェクトのレイアウトは座標による絶対指定でなく構造によって指定する．このシステムでは Web の階層構造を考慮し，ハイパーリンクを用いることができる．

また，Adiba らはプレゼンテーションの再利用を目

的とし，様々な環境のマルチメディアサーバから presentation warehouse を構成し，クエリによる検索結果をプレゼンテーションとした[2]．検索は OQL の拡張言語，OQLiST（Object Query Language integrating Spatial and Temporal aspects）[11] よって行う．オブジェクト指向のアプローチで，オブジェクトの配置は座標による絶対指定で行う．

2.2 動的プレゼンテーションによる表示領域の有効利用

動的プレゼンテーションを表示領域の節約に利用しようとするアプローチは，携帯電話や PDA など表示領域の小さい環境においていかに多くのデータを効率良く提示するかという研究で用いられている．

WebCarousel [20] は携帯電話や PDA など，インタラクションや表示領域が大幅に制限されている携帯端末上で Web の検索結果を受動的に閲覧させるシステムである．テキストを変換した音声と画像により，カルーセルコンポーネントを構成する．

Zancanaro らは，美術館においてガイド情報を PDA の小さい画面上に表示するため，動的にビデオクリップを生成する手法である cinematic techniques を提案し，実用化した[12]．ビデオクリップはガイドの一部を音声のコメントと同期する画像シーケンスとして構築される．また，次の画像に移動する際にトランジションが利用されている．

2.3 動的プレゼンテーションを生成する既存のツール

動的プレゼンテーションを生成する商用ツールやフリーソフトも多数存在する．ここではその一部を示しておく．

- Macromedia Flex
  ビジネス向けの「リッチインターネットアプリケーション」の生成を目的とした Macromedia のサーバ製品．XML 形式の言語 MXML（Macromedia-Flex Markup Language）によりムービーの内容を記述する☆[14]．
- PPT2SWF
  AJSoft 社による，PowerPoint スライドを Flash ムービーに変換するソフトウェア．Windows95 以降の環境で動作する[3]．
- ImageMagick
  画像処理ソフトウェア．連続する画像を読み込み，アニメーション GIF や mpeg に変換できる．

☆ Flex および MXML は Macromedia，Inc. の登録商標である．

これらのツールはいずれも，画像やテキストのオブジェクトから比較的簡単な操作で動的プレゼンテーションを生成できるものであり，広く知られている．

3. SuperSQL

SuperSQL は著者らが開発を行っている，SQL を拡張したワンソースマルチユースを実現するクエリ言語である．その質問文は SQL の SELECT 句を GENERATE <*medium*><*TFE*> の構文を持つ GENERATE 句で置き換えたものである．ここで <*medium*> は出力媒体を示し，HTML，XML，PDF，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 [17] 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式である．r_i を関係，P は SQL における条件文とすると，SuperSQL クエリは以下のように表せる．

GENERATE <*medium*><*TFE*>
FROM $r_1, r_2, ..., r_n$
WHERE P

3.1 TFE

TFE（Target Form Expression）は，SQL では SELECT 句に記述する属性名を，レイアウト演算子である結合子，反復子と組み合わせることで出力結果の構造を指定する式である．さらに装飾子によって表のセルの幅や背景色などの詳細なカスタマイズが可能である．

3.1.1 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る．**表 1** にその種類と意味，処理系におけるレイアウト式を示す．

3.1.2 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ことから，グルーピング演算子とも称される．たとえば

[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，

表 1 結合子の種類と意味
Table 1 Connectors in SuperSQL.

結合子	意味	クエリ表記	レイアウト式
,	横結合	A , B	[C1 [1] [2]]
!	縦結合	A ! B	[C2 [1] [2]]
%	深度結合	A % B	[C3 [1] [2]]

表 2 反復子の種類と意味
Table 2 Groupers in SuperSQL.

反復子	意味	クエリ表記	レイアウト式
[ ],	横反復	[A],	[G1 [1]]
[ ]!	縦反復	[A]!	[G2 [1]]
[ ]%	深度反復	[A]%	[G3 [1]]

[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．**表 2** に反復子の種類と意味を示す．

3.1.3 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@{ 装飾指定式 } の構文による装飾演算子によって指定することができる．装飾指定式は（項目名 = 値）として指定する．複数指定するときは各々を “,” で区切ったリストとする．装飾子は TFE の各オブジェクトに対して以下のように記述する．たとえば，$A@\{width = 300\}$ という装飾子を用いると，属性 A のセル幅を 300 px にする．

3.1.4 関 数

SuperSQL における関数は，データベース検索結果の文字列に対し，特定の処理を行うための機能である．関数の記述は

関数名 (TFE, オプションのリスト)

の形式で行う．オプションのリストは関数の実行に必要な条件やディレクトリパスなどの指定を “,” で区切ったリストである．現在のシステムでは画像名に対して用いる imagefile 関数，HTML 出力においてハイパーリンクを動的に構築する invoke 関数などが利用可能である．

3.1.5 レイアウト式

レイアウト式は，TFE によって指定されたオブジェクトの配置をポーランド記法を用いて表した式で，SuperSQL 処理系内部でデータを処理する順序を示すものである．TFE では属性名にあたる部分を，レイアウト式では任意の数字に置き換える．演算子については，結合子か反復子かによって決まる C もしくは G のアルファベット 1 文字と，結合次元を示す数字（1〜4）の組合せによって表現する．レイアウト式の例を以下に示す．

TFE: [name , photo]!
レイアウト式：[G2 [C1 [1] [2]]]

4. 時間連結モデルの提案

4.1 既存の動的プレゼンテーション生成およびデータベース出力における問題点

まず，2 章であげた動的プレゼンテーション生成手法については，マルチメディアデータベースのみを対象としているため，関係データベースなどに格納した既存のデータを活用することができず，テキストのみのデータを適切に扱えないという問題点がある．オブジェクトの配置は座標による絶対指定で行う場合が多く，未知のデータに対して適応できないため柔軟性に欠ける．また，Web 環境を想定したハイパーリンクに対応しているものが少ない．

また，動的プレゼンテーションの利用による表示領域の有効利用に関する既存の研究においては，テキストを音声に変換して扱うことから，プレゼンテーション構成に時間がかかるという点で利用環境によっては不利と考えられる．また，電車の車内における，ディスプレイを利用した情報サービスのように，音声を用いずに情報を伝えなければならない環境では情報を文字で伝えることが必須となる．

3 章で述べた SuperSQL は上記の問題をカバーしているものの，時間軸という概念を持っていないという問題がある．

4.2 時間連結モデルの概念

3 章で示したように，SuperSQL においてはデータオブジェクトの 3 次元方向への配置が可能であるが，4.1 節で述べた問題点を解決するためにはデータオブジェクトを時間軸方向に配置するという概念が必要である．そこで，本研究では新たに第 4 次元のレイアウト演算子，すなわち時間結合子と時間反復子を導入することによりこの問題に対応した[19]．時間結合子はデータオブジェクトどうしを時間方向に結合するものであり，時間反復子は検索結果のデータがあるだけ，一定時間ごとに時間方向に繰り返すものである．この 2 つのレイアウト演算子を総称して時間連結子とする．

なお，本研究では Web 基盤を利用することを想定し，出力メディアとして Macromedia Flash を採用した．Flash の出力ムービーの拡張子が．swf であることから，以降ではこれを SWF と呼ぶ．

4.3 時間結合子

時間結合子 # は 2 つ以上のデータインスタンスを時間方向に結合することを意味する．時間結合子を用いたクエリの例とその出力結果の概念を**図 2** に示す．

4.4 時間反復子

時間反復子 [ ]# は，囲んだ属性の値がデータベー

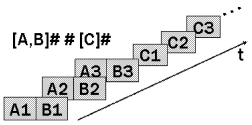


図 2 時間結合子の概念
Fig. 2 Concept of the sequencing connector.

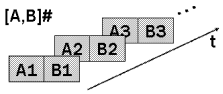


図 3 時間反復子の概念
Fig. 3 Concept of the sequencing repeater.

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を含むことができる．時間反復子を用いたクエリの例と出力結果のモデル図を**図 3** に示す．

4.5 時間連結における速度と繰返しの設定

時間連結における各データオブジェクトの表示時間と繰返しに関して必要なパラメータは以下の装飾子として指定することとする．

- **duration**
  時間結合する各データオブジェクトの表示時間を指定する．ミリ秒単位．
- **interval**
  時間反復するデータオブジェクトの表示時間を指定する．ミリ秒単位．
- **loop**
  最後のデータオブジェクトまで表示した後，再び最初から表示し始めるかどうかを true/false で指定する．

4.6 トランジションの設定

次のフレームに切り替わる際の視覚的効果であるトランジションを導入することで，データが切り替わったことを明示的に表現する．トランジションは装飾子によって種類とパラメータを設定する．たとえば，`@{transition=fade, offset=15}` と指定すると，各フレームの開始時はフェードインし，終了時にフェードアウトする．フェードする速度はパラメータ offset により調整可能とする．

4.7 生成モードの設定

様々なプレゼンテーションの目的や利用環境に対応できるよう，静的生成と動的生成の両方を可能にした．静的生成は SWF ファイルを静的に生成し，動的生成はファイル生成を行わず動的にムービーを表示する．指定の方法は装飾子 TFE の最も外側の要素に対し装飾子で `@{mode=static}` または `@{mode=dynamic}` と指定する．

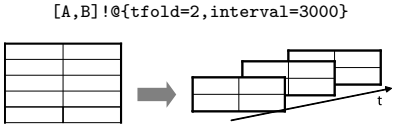


図 4 時間折り畳みの例
Fig. 4 An example of sequencing foldering.

4.8 時間折り畳みの導入

データオブジェクトを時間連結する場合，同じ領域を使用して表示するデータだけを入れ替えるため，必要とする画面領域が小さくて済むという特徴がある．これに基づき，表示領域を節約し，かつ利用可能な表示領域を最大限に利用し表示所要時間を短縮するため，縦横反復による連結方向に長い表を指定回数によって時間方向に折り畳む機能，時間折り畳みを実現する．時間折り畳みの例を**図 4** に示す．クエリでは装飾子によって以下のように時間折り畳みを指定する．

`<縦または横反復>@{tfold=回数, interval=間隔}`

5. SuperSQL による動的プレゼンテーション生成

5.1 時間反復による出力例

時間反復を入れ子にして用いたクエリとその実行結果を**図 5** に示す．クエリにおける色やサイズに関する装飾子は省略している．1 つの商品に登録されている画像を 4 秒ごとにすべて時間反復で表示した後，次の商品を表示するプレゼンテーションである．図 5 では最初の 2 商品までを示している．

5.2 時間折り畳みによる出力例

縦反復クエリによる縦に長い表（**図 6**）を数行ごとに区切り，それらを時間方向に連結する時間折り畳みを用いたクエリと出力例を**図 7** に示す．この例では検索結果が 60 タプルあるものを，10 秒ごとに 15 件ずつ折り畳み表示する．

6. SWF 生成系の実装

本研究の SWF 生成手法は SWF 生成ライブラリである ming [16] を利用するため，SuperSQL のメディア生成部から直接 SWF ファイルを出力するのではなく，ムービーの中身を記述した PHP コードを中間生成物として出力する．その PHP プログラムを実行（閲覧）することにより静的生成では SWF ファイルを生成，動的生成では SWF ムービーを表示する．

6.1 システム構成

本研究で実装したシステムの構成を**図 8** に示す．

```
GENERATE SWF [
 { r.name !
  {"designer" , r.designer} !
  {"company" , r.brand} !
  {"age" , r.age} !
  {"country" , r.country} !
  r.comment },
 [imagefile(h.name,path="photos")]#@{duration=4000}
]#
FROM product r, photo h
WHERE h.product = r.id
AND r.category like '%chair%';
```

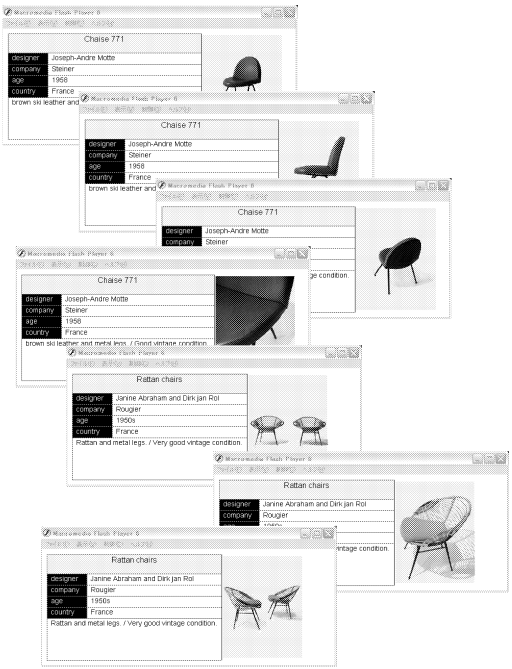


図 5 時間反復の実行結果
Fig. 5 A query and its output using sequencing repeater.

前述のとおり，SuperSQL 処理系で行う処理は PHP コードを出力するところまでである．SWF の生成には ming ライブラリ（0.3 以上）をインストールした PHP 処理系を必要とする．また，オンライン動的生成の場合，Java Servlet 技術を用いて SuperSQL 処理系を動的に呼び出すため，Tomcat などのアプリケーションサーバ上に SuperSQL 処理系の実行ファイルを配備する．

6.2 PHP コードの内容

SuperSQL により出力される PHP コードの内容について述べる．コードは大きく分けて 5 つのユニットから構成される．各ユニットの役割は以下のとおりである．

- **パラメータ設定部**
  ムービー全体に対するパラメータを設定する．ムービーのサイズ，背景色，フレームレート，Flash Player のバージョンなど．
- **共通関数宣言部**
  (1) セル（TextField）を生成・削除する関数
  ActionScript によって生成・削除を行う．生成の際，クエリで指定された装飾子の内容を反映したものを生成する．
  (2) 画像を追加・削除する関数
  ming の関数により追加・削除を行う．
  (3) フレームを進める関数
  指定された duration の時間だけフレームを進める．トランジションが指定された場合，その処理も含む．
  (4) 1 シーンを表示する関数
  上記のすべての共通関数を，処理の順序に従って呼び出す関数．処理の順序は，オブジェクトの生成（追加）→フレームを進める→オブジェクトの削除の順である．
- **シーン生成部**
  1 シーンに同時に表示するすべてのセル，画像に関する値を配列として宣言し，共通関数のシーン表示関数に引数として渡す．シーン生成部はシーンの数だけ記述する．
- **出力指定部**

```
GENERATE SWF
{"定刻" , "変更" , "便名" , "航空会社" , "目的地" , "状況"} !
[ f.appointed_time , f.changed_time , f.flight ,
  imagefile(f.airways_img,path="flightimg") ,
  f.airways , f.destination , f.status
]!
FROM flight_info f;
```

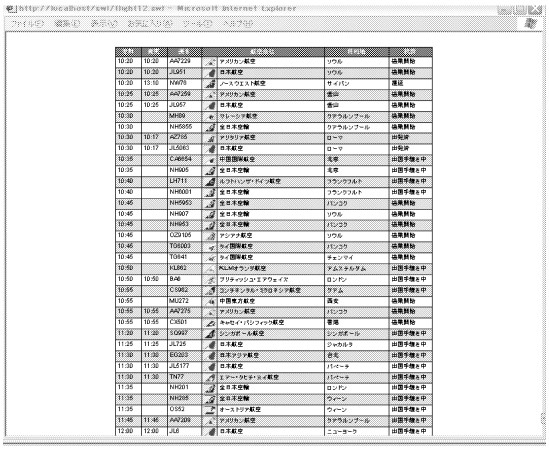


図 6 縦反復のみのクエリの実行結果
Fig. 6 A query and its output using vertical repeater.

```
GENERATE SWF
{"定刻" , "変更" , "便名" , "航空会社" , "目的地" , "状
況"} !
[ f.appointed_time , f.changed_time , f.flight ,
  imagefile(f.airways_img,path="flightimg") ,
  f.airways , f.destination , f.status
]!@{tfold=15,interval=10000}
FROM flight_info f;
```

図 7 時間折り畳みを用いた縦反復の実行結果
Fig. 7 A query and its output using sequencing foldering.

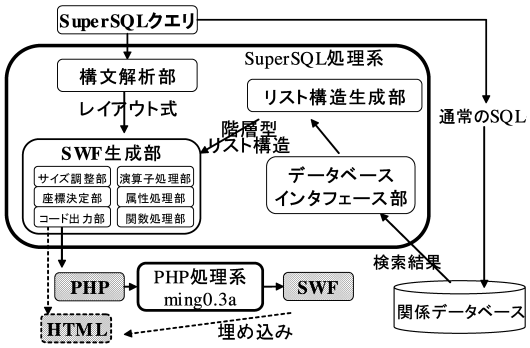


図 8 SWF 出力系のシステム構成
Fig. 8 System configuration of SWF generator.

(1) 静的生成の場合
SWF ムービーをファイルとして生成する save() 関数を記述.

(2) 動的生成の場合
SWF ムービーを表示する output() 関数を記述.

クエリ	`[product,[color]!,photo]#`
レイアウト式	`[G4 [C1 [1] [G2 [2]] [3]]]`
構造化された結果	`[[[M980J][[pink][blue][silver]] [m1.jpg]]]`

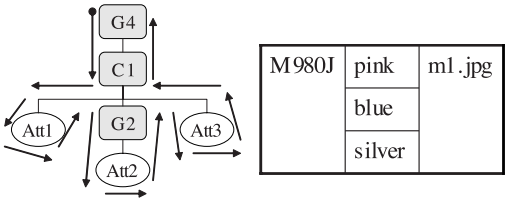


図 9 構造化された結果の処理順序
Fig. 9 Sequence of processing structured result.

- **埋め込みタグ出力部**

(1) 静的生成の場合
生成した SWF ファイルの内容を確認するため，生成と同時に画面に表示されるよう埋め込みタグを出力する．

(2) 動的生成の場合
何も出力しない．

6.3 PHP コード出力処理の流れ

6.2 節で示した PHP コードを出力する，SuperSQL の SWF 出力部の処理について説明する．SWF 出力部はデータベースからの検索結果をリスト構造にしたものと，3.1.5 項で述べたレイアウト式をもとに，オブジェクトのレイアウトや各プロパティを取得していく．すべてを設定し終えたら，同時に画面に表示するオブジェクトすべての値を配列し，シーン生成部として蓄積する．最後にムービー設定部，共通関数部，シーン生成部，出力関数部，埋め込みタグ出力部の順で PHP ファイルとして書き出す．また，日本語表示に対応するため，文字コードを UTF-8 で出力する．

処理の流れを以下に示す．

(1) 構造化された結果をレイアウト式に従って対応する処理部で処理し，value tree を生成.
(2) value tree をたどり，すべてのセルサイズを調整.
(3) value tree をたどり，すべてのセルの座標を決定.
(4) シーン生成用の配列を作成.
(5) すべての PHP コードを出力.

レイアウト式に従い，レイアウト演算子処理系では各次元への連結処理，属性・関数処理系ではセルサイズの初期値と装飾子で指定されたプロパティを構造化された結果から取得する．演算子処理系において，連結するオブジェクトがさらに演算子を含む TFE であった場合，その演算子処理系での処理を完了してからもとの処理系に戻り次のオブジェクトを処理する．**図 9** に処理の順序の例を示す．木構造を前順序でたどるの

表 3 SWFValue
Table 3 SWFValue.

メンバ	意味	初期値
type	要素の型	作成時に指定
inList	連結要素 (子)	none
name	セル識別 ID	none
width	幅	none
height	高さ	none
x	X 座標	10
y	Y 座標	10
bgcolor	背景色	0xffffff
bdcolor	枠線色	0x000000
color	文字色	0x000000
size	文字サイズ	14
align	セル内横位置	left
margin	セル内余白	3
data	データ文字列	none
imagepath	画像パス	none

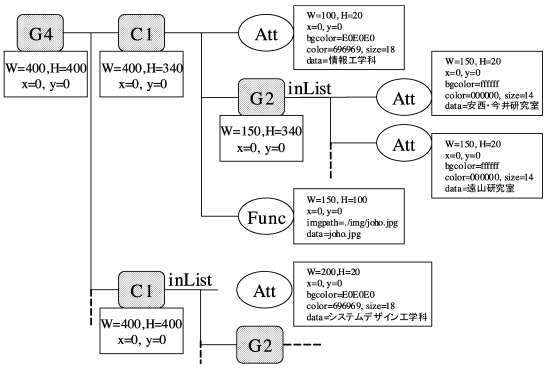


図 10 value tree の例
Fig. 10 An example of value tree.

と同じ順序で各処理系を呼び出している．

6.4 value tree の構築

各レイアウト演算子と属性または関数の処理系は，セルのプロパティをまとめた構造体のようなものを生成する．これを SWFValue（**表 3**）と定義する．TFE 中の属性名と関数の SWFValue は，前出の図 9 では木構造の葉にあたり，それ自身のプロパティのみを持つ．一方，レイアウト演算子の SWFValue はその演算子自身のプロパティを持ち，さらにその演算子によって連結するすべてのオブジェクトのプロパティを，ArrayList の拡張である inList という変数に格納する．レイアウト式のオブジェクトすべてについてこの SWFValue 生成処理が終了したときに完成する木構造を value tree と定義する．value tree はレイアウト式の構造と同じ階層型リスト構造を持つことになる．value tree の例を**図 10** に示す．value tree の根はレイアウト式で最初に現れる演算子となり，属性名や関数の値が葉となる．

```
function textFieldAS(name,x,y,w,h,bgcolor,color,
                          size,align,margin,str) {
  depth+=2;
  _root.createTextField(name,depth,x,y,w,h);
  with (_root[name]){
    multiline = true;
    wordWrap = true;
    border = true;
    background = true;
    backgroundColor = bgcolor;
  };

  format = new TextFormat();
  format.size = size;
  format.align = align;
  format.font = 'Arial';
  format.color = color;
  format.leftMargin = margin;
  format.rightMargin = margin;

  _root[name].setNewTextFormat(format);
  _root[name].text = str;

};
```

図 11 セルを生成する ActionScript の関数
Fig. 11 ActionScript function for creating cells.

6.5 表構造の生成方法

Flash においては HTML の TABLE のように階層的な構造を持たせることのできる「表」という形式のオブジェクトが定義されていない[13]．そのため，Flash における文字入力フォームである TextField オブジェクトを表のセルに見立て，隙間なく TextField を配置することで表構造を実現する．TextField によって表構造を生成する方法は以下の手順による．

(1) データ文字列の長さや画像サイズからセル固有の幅と高さを取得し，初期サイズとする．

(2) 他のセルのサイズに依存して変更すべきサイズを変更する．

(3) セルを配置すべき座標 (x,y) を計算する．

6.5.1 セルの生成とプロパティの設定

セルは Flash ActionScript の createTextField 関数を用いて動的に生成する．セルの大きさや座標などの各種プロパティはセル生成時に設定する必要があるため，生成の前にクエリで指定された装飾子の内容や，システム内部処理で自動的に与えられるプロパティの値を配列文字列として保存しておく．セルに対して装飾子で指定することができるプロパティは表 3 の width 以降となる．これらの値をシーンごと，項目ごとに配列に格納する文字列として出力し，この配列を引数としてシーン表示関数を実行することで，1 シーンに表示するすべてのセルを生成する．

Flash の TextField オブジェクトを生成する ActionScript の関数を**図 11** に示す．この TextField 生成関数は最初にムービーに付与され，シーンを切り替えるたびにシーン表示関数から呼び出される．

表 4 各トランジションにおけるマスクへの処理
Table 4 Mask processing for transition.

トランジション	in 時のマスクの処理	out 時のマスクの処理
fade	透明度を 100 まで上げる	透明度を 0 まで下げる
wipeLR	X 座標をムービーの右端まで移動する	X 座標を 0 まで戻す
wipeUD	Y 座標をムービーの下端まで移動する	Y 座標を 0 まで戻す

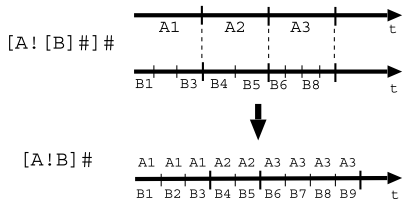


図 12 階層的時間構造の平坦化
Fig. 12 Planarization of hierarchical timeline.

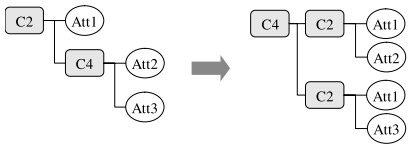


図 13 根以外に時間連結子を持つ木構造の変換
Fig. 13 Transformation of tree structure when existing sequencing operators except in root.

6.6 階層的時間構造への対応

図 12 に示すように，時間連結子を入れ子にして用いるとプレゼンテーションのタイムラインも階層的な構造を持つ．PHP によるムービー記述が煩雑になるのを避けるため，冗長ではあるが，これを平坦化してすべてを単一のタイムライン上に配置する．

時間構造平坦化処理は value tree の木構造を変換することで行う．階層的な時間構造を持つ value tree は根以外の節にも時間連結子が存在している．このような時間連結子を子要素として持つ親演算子の処理部において，**図 13** のように時間連結子を根に移動させる構造変換を行う．図 13 の例では，C4 の親である C2 の処理部でこの変換を行う．ただし，時間連結子の子が時間連結子となった場合は，子の子を直接親の子とする．このような変換を行うと，反復と結合の区別が失われる場合があるが，各要素が表示時間を保持しているため問題はない．時間連結子を子として持つ部分木すべてに対してこの処理を行うと，最終的には根のみに時間連結子が存在する木構造となる．

6.7 シーン間トランジションの実現方法

本システムで生成される SWF ムービーには ActionScript によって生成した Flash の TextField オブジェクトと ming の関数によってムービーに追加した画像が混在している．Flash における TextField は本来フォーム入力用のオブジェクトとされているため，ming においてはムービークリップ（sprite）として扱うことができない．よってセル自体に効果を与えることができない．本研究ではこの 2 種類のオブジェクトに対して同時にトランジション効果を与える方法を，マスクを用いた手法により実現した．

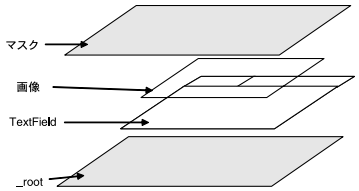


図 14 ムービーの階層とマスク
Fig. 14 Structure of the movie and mask.

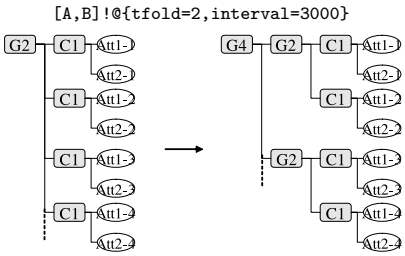


図 15 時間折り畳みにおける value tree の木構造変換
Fig. 15 Transformation of tree structure for sequencing foldering.

ムービーの階層図を**図 14** に示す．各オブジェクトは深度が衝突しないよう付加されている．マスクはすべてのオブジェクト最上位に配置し，初期状態はムービーの背景色と同色で透明度を 0 に設定しておく．トランジションの開始とともに，マスクのプロパティを変化させることでトランジション効果を実現する．今回実装した 3 種類のトランジションにおけるマスクへの処理を**表 4** に示す．

6.8 時間折り畳みの実現方法

時間折り畳みは value tree の木構造を変換することにより実現する．**図 15** の例では時間折り畳みを指定する装飾子 tfold の値として回数が 2 に指定されている．この場合，縦反復する要素の個数を 2 個ずつに区切り，それぞれ時間反復の要素とする．

表 5 他ツールとの比較結果
Table 5 Functional comparison of our system with other tools.

	SuperSQL	Macromedia Flex	PPT2SWF	ImageMagick
静的生成	○	×	○	○
動的生成	○	○	×	×
オブジェクト配置作業	SuperSQL クエリ記述	MXML 記述	PowerPoint	コマンド記述
外部プログラム	不要	必要	不要	不要
実行時間	○	△	×	×
SWF ファイルサイズ	○	-	×	×

7. 評価/考察

SuperSQL の時間連結を用いた動的プレゼンテーション生成手法について，既存の研究や他のツールとの比較を通じて有用性について考察を行った．

7.1 関連研究との比較

データベースのためのレポートライタ機能やスキーマによるマルチメディアプレゼンテーションの自動生成では，出力のためのテンプレートを必要としたが，本研究の手法では必要なく，簡単なクエリ記述のみで出力形式を指定可能である．また，一般的な動的プレゼンテーション生成ツールは，出力メディアに関する知識がなければオーサリングできない，デザイナ向けの環境であることが多いが，本研究の手法では，SQL の構文と簡単な TFE の構文を知っていれさえすれば，動的プレゼンテーションを作成することが可能である．さらに，オブジェクトの配置は構造による相対指定を用いているため，データの種類やサイズにかかわらず正しい位置関係でオブジェクトを配置することができる．これは HTML の TABLE タグ記述によってオブジェクトの配置を構造化する作業に似ている．また，SuperSQL では第 3 次元としてハイパーリンクも使用できるので，Web 環境に適したプレゼンテーションが生成できる．検索対象がマルチメディアデータベースだけでなく，テキストデータのみの関係データベースに対して適用できるため，過去のデータを活用することも可能である．

7.2 既存の動的プレゼンテーション生成ツールとの比較

以下の既存の動的プレゼンテーション生成ツールと本研究の比較を行い，**表 5** にまとめた．比較結果から分かるように，動的生成と静的生成の両方に対応し，簡単なクエリの記述のみで生成できる本研究の手法は他のツールに比べて汎用性が高く有用であるといえる．

7.3 トランジションの効果

本研究においては，装飾子で指定できるトランジションのプロトタイプとして 3 種類を実装した．プレゼンテーションの目的やデータの内容，レイアウトに応じてトランジションの有無や種類，パラメータ offset を適宜選択することにより，プレゼンテーションの閲覧しやすさや見た目の質を向上させることができる．トランジションの選択基準としては以下のものが考えられる．

- 目的や内容による選択
  気象衛星画像を時間反復して動画のように見せるような目的ではトランジションを用いると滑らかに見えないのでないほうがよい．文字データを中心とし，表構造やセルのサイズに変化がないデータのプレゼンテーションでは，できるだけ閲覧者の注目を喚起するため動きの大きいトランジションを用いると良い．offset の値を大きくするほどトランジションにかかる時間が長くなり変化が滑らかに見えるので，プレゼンテーションの速度と芸術性の優先度により調整する．
- レイアウトによる選択
  wipeUD と wipeLR のように方向性を持ったトランジションではデータの連結方向によって適するトランジションが異なる場合がある．たとえば，縦方向の連結が多い表レイアウトでは閲覧者の視線が上から下に移動するので，上下の動きをする wipeUD の方が適していると考えられる．

今回の実装では ActionScript により生成された TextField と ming によって追加された画像が混在する部分をムービークリップとして扱うことができないため，Flash 特有のトゥイーン機能を利用できない．そのため，表自体が形を変えたり移動したりするようなトランジションが実現できない．マスクにより実現できるトランジションの種類は限られており，選択肢が少ないことは改良すべき点である．

7.4 時間折り畳みの有用性について

時間連結の手法を画面領域の有効利用という目的に利用するため，タプル数件ごとの切替表示を可能とする時間折り畳みを実装した．時間折り畳みの有効性を示すため，データベースの検索結果 60 件に対して縦反復，時間反復，時間折り畳みの 3 つのレイアウトを用いた出力結果の比較を行った．各レイアウトの意味

表 6 縦反復と時間反復と時間折り畳みの比較
Table 6 Comparison of vertical repeating and sequencing repeating and sequencing foldering.

	意味	クエリ	表の高さ	画面充填率	所要時間
(a)	縦反復	`[A,B]!`	1,583 px	206.1%	0 秒
(b)	2 秒間隔で時間反復	`[A,B]#@{interval=2000}`	46 px	6.0%	120 秒
(c)	縦反復し 15 件で 10 秒ごとに時間折り畳み	`[A,B]!@{tfold=15,interval=10000}`	368 px	47.9%	40 秒

表 7 5 節で示したクエリ実行例の生成時間とファイルサイズ
Table 7 Execution time and output size of three examples in section 5.

	図番号	タプル数	フレーム数	文字属性数	画像属性数	実行時間	サイズ
(1)	図 5	13	60	10	1	2,243 msec	304 KB
(2)	図 6	60	1	6	1	2,013 msec	66 KB
(3)	図 7	60	4	6	1	2,293 msec	83 KB

とクエリ記述，出力結果の表の高さ，表による画面充填率[18]，すべてのデータを表示するのにかかった時間を**表 6** に示す．

今回の比較では縦反復を用いているため，サイズについては高さのみを考慮する．セル幅については 3 手法すべてに同じ幅指定を行った．画面充填率に関してはディスプレイサイズを 1,024 × 768 のディスプレイで Flash Player を全画面表示にした場合として計算している．なお，折り畳み時には閲覧者の視線が上から下に移動しながら必要としない情報を瞬時に読み飛ばすと考え，時間反復の場合より 1 タプル分の内容把握にかかる時間を短く設定した．出力されるムービーの縦の長さについては，データを 1 タプルずつ表示する (b) の時間反復が最も小さくなるが，時間折り畳みを用いた縦反復手法 (c) では，1 タプルの大きさがある程度把握できていれば適切なサイズに調整することが可能である．

画面充填率は出力されたムービーにおける表の高さが画面の高さをどれくらい満たしているかを表す指標であり，式 $S = H_{table}/H_{screen}$ で表せる．最も充填率が低くなるのはタプル 1 件ごとに表示する (b) である．この例では 1 タプルの高さが 46 px と小さいため，表示領域が極端に小さい場合には最適だが，ある程度の大きさの表示領域では余白部分が大きすぎて冗長なレイアウトとなる．(c) では折り畳み回数を 15 件ごとと指定したため画面サイズに対しては比較的小さな表となったが，折り畳み回数を適切な値に指定すれば表示領域に応じた画面の有効利用が可能となる．

全データ表示所要時間についていえば，1 画面に表示される情報が少ないにもかかわらず画面を見ていなければならない時間が長いのは閲覧者にとって煩わしいと感じることが多い．よって 1 シーンのオブジェクト数が多い場合や連結方向の長さによって表がある程度の大きさを持っている場合には時間反復が有効であるが，1 シーンの大きさが比較的小さい場合は時間折り畳みの方が有効であると考えられる．時間反復では所要時間が $interval$ × タプル数，時間折り畳みでは所要時間が $interval$ × タプル数/指定回数となる．

以上の結果より，表示領域の大きさやタプル数，1 シーンの表サイズ，検索結果の件数，閲覧スタイルなどを考慮し最適な折り畳み回数と時間間隔を指定することで画面の有効利用と全データ表示所要時間短縮が実現可能であり，時間折り畳みは有用であるといえる．今回の実装では折り畳み条件を反復回数によってのみ指定可能としているが，タプルごとにサイズが異なる場合，同じタプル数であっても 1 シーンの表サイズが一定にならない．そのため，表示領域が限られている場合はサイズによる折り畳みの指定が必要である．たとえば，`[A]!@{tfold=500px}` と記述すると 500 px に収まるタプル数だけ反復するという指定方法である．

7.5 実行時間とファイルサイズ

5 章で示した 3 つのクエリによるプレゼンテーション生成について，それぞれ静的生成で出力した際のクエリ実行時間☆とファイルサイズを**表 7** に示す．なお，フレーム数は階層的タイムラインを平坦化した後のタプル数を意味する．また，文字属性数と画像属性数はそれぞれ 1 タプルにおける数である．

表 7 に示した結果について考察を行う．

- **実行時間について**
  タプル数と画像の数に依存するといえる．特に画像関数の処理部では画像 1 つ 1 つに対して幅と高さの取得を行うため，多くの時間を要する．これはタプル数が多くても文字属性が多い (2) の場合はそれほど時間がかかっていないことからも分かる．また，時間折り畳みを使用した (3) は木構

☆ 動的生成モードを利用した場合の生成時間はネットワーク環境などに依存するため，今回は静的生成についてのみ評価を行っている．

造の変換処理を行うため同じデータを用いた縦反復のみの (2) より多くの時間を要したと考えられる．しかし平均して 2 秒程度の短い時間で生成されている．

- **ファイルサイズについて**
  画像を SWF ファイルに埋め込んでいるため画像のサイズと数に大きく依存している．また，シーン数やトランジションによるフレーム数増加の影響も受けていると考えられる．しかし画像を 60 枚埋め込んだ (1) でも 300 KB 程度と，十分実用に耐える軽量性を示している．

8. おわりに

本研究では，データベース出力に時間方向のレイアウトという概念を導入し，動的プレゼンテーションとして生成する手法を提案し，SuperSQL に SWF 出力系を実装した．これにより，プレゼンテーションのメディアに関する知識がなくても，簡単なクエリの記述のみによって動的プレゼンテーションを生成することが可能となった．さらに，いくつかのパラメータを調整することによりデータの種類やプレゼンテーションの目的に適した動的プレゼンテーションの生成が可能となった．

プレゼンテーションとしての品質向上のためデータが切り替わることを閲覧者に対し明示的に表現するシーン間トランジションを導入した．プレゼンテーションの目的やデータの種類，レイアウトに応じて適宜使用することで効果が得られた．

オフラインでのプレゼンテーション展示や Web ページへのプレゼンテーション埋め込みのために SWF ファイルを生成する静的生成モードと，頻繁に更新されるデータベースの内容をつねに反映したい場合に，Java サーブレットを用いて閲覧要求発生時に動的に SWF 生成処理を行う動的生成モードを選択可能にした．

時間連結の手法を表示領域の有効利用という目的に利用するため，縦横反復されたタプルを指定回数ごとに折り畳み時間方向に連結する時間折り畳みの機能を実現した．これにより縦横反復の利点である全データ表示所要時間の短さと，時間反復の利点である必要表示領域の小ささの双方を活かすことが可能となった．

今後の展望として，JavaScript との組合せによる文字テロップやページスクロールの導入，他のメディアとの組合せを視野に入れた開発をめざす．また，本研究で定義した SuperSQL による動的プレゼンテーション生成は SWF に依存するものではなく，コード生成部を追加すれば OpenGL などのメディアにも対応可能であり，システムの用途や環境に応じて SVG や XHTML を出力メディアとすることも視野に入れている．

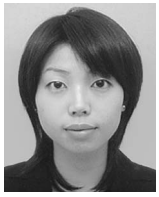
参考文献

1) Adiba, M.E. and Zechinelli-Martini, J.L.: Spatio-Temporal Multimedia Presentations as Database Objects, *Proc. 10th International Conference, Database and Expart Systems Applications* (*DEXA'99*), pp.974–985 (1999).
2) Adiba, M.E. and Zechinelli-Martini, J.L.: Building Spatio-Temporal Presentations Warehouses from Heterogeneous Multimedia Web Servers, *Proc. Advanced Information Systems Engineering, 14th International Conference* (*CAiSE '02*), pp.692–696 (2002).
3) AJSoft (Prv) Ltd.: PPT2SWF. http://www.geocities.com/ajsoftpk/ppt2swf.html
4) Apple Computer, Inc.: QuickTime. http://www.apple.com/quicktime/
5) Baral, C., Gonzalez, G. and Nandigam, A.: SQL+D: Extended Display Capabilities for Multimedia Database Queries, *Proc. 6th ACM International Conference on Multimedia 1998*, pp.109–114 (1998).
6) Celentano, A. and Gaggi, O.: Schema Modelling for Automatic Generation of Multimedia Presentations, *Proc. 14th International Conference on Software Engineering and Knowledge Engineering* (*SEKE'02*), pp.593–600 (2002).
7) Cruz, I.F. and Lucas, W.T.: A Visual Approach to Multimedia Querying and Presentation, *Proc. 5th ACM International Conference on Multimedia 1997*, pp.109–120 (1997).
8) Gaggi, O. and Celentano, A.: A Visual Authoring Environment for Prototyping Multimedia Presentations, *Prce. 4th International Symposium on Multimedia Software Engineering* (*ISMSE 2002*), pp.206–213 (2002).
9) Huang, C.M. and Jang, M.Y.: Interactive Temporal Behaviours and Modelling for Multimedia Presentations in the WWW Environment, *The Computer Journal*, Vol.42, No.2, pp.112–128 (1999).
10) ImageMagick Studio LLC.: ImageMagick. http://www.imagemagick.org/
11) Luis, J. and Martini, Z.: Integrating and Managing Multimedia Presentations by a Web-based Object Server, *Proc. EDBT PhD Workshop*, pp.584–586 (2000).
12) Zancanaro, M. Stock, O. and Alfaro, I.: Cinematic Techniques for Mobile Presentations,

Proc. 5th International Symposium (*Mobile HCI 2003*), pp.384–389 (2003).
13) Macromedia, Inc.: Macromedia Flash. http://www.macromedia.com/software/flash/
14) Macromedia, Inc.: Macromedia Flex. http://www.macromedia.com/software/flex/
15) of W3C., S.M.W.G.: Synchronized Multimedia Integration Language (SMIL) 2.0 Specification. http://www.w3.org/TR/REC-smil
16) Opaque Industries: Ming — A SWF output library and PHP module. http://ming.sourceforge.net/
17) Seto, T., Nagafuji, T. and Toyama, M.: Generating HTML Sources with TFE Enhanced SQL, *Proc. 1997 ACM Symposium on Applied Computing* (*SAC'97*), pp.96–100 (1997).
18) Maeda, Y. and Toyama, M.: ACTIVIEW: Adaptive data presentation using SuperSQL, *Proc. 27th International Conference on Very Large Data Bases* (*VLDB2001*), pp.695–696 (2001).
19) 笹田麻衣子，遠山元道：SuperSQL の HTML 出力における時間連結子の実装，日本データベース学会 DBSJ Letters，Vol.2, No.1, pp.131–134 (2003).
20) 灘本明代，近藤宏行，田中克己：携帯環境における 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(TOD12), pp.1–14 (2001).

(平成 17 年 3 月 20 日受付)
(平成 17 年 7 月 7 日採録)

（担当編集委員　浦本 直彦）



笹田麻衣子

平成 15 年慶應義塾大学理工学部情報工学科卒業．平成 17 年同大学大学院理工学研究科開放環境科学専攻修士課程修了．同年より，株式会社野村総合研究所に勤務．

遠山　元道（正会員）

昭和 54 年慶應義塾大学工学部管理工学科卒業．昭和 59 年同大学院博士課程修了．同年助手．現在情報工学科専任講師．博士（工学）．平成 8 年オレゴン大学院大学（OGI）客員研究員．平成 10〜13 年 JST さきがけ研究 21 研究者．主にデータベースの研究に従事．電子情報通信学会，ACM，IEEE 会員．