

ACTIVIEW : SuperSQL を用いた 適応型 Web ビューの実現

慎 祥 撰^{†1} 前 田 葉 子^{†2} 遠 山 元 道^{†3}

本稿では、関係データベースと WWW との連係システムである SuperSQL を利用することで、データベースの情報をビューとして表示する際に、端末の表示画面サイズに動的に適応することを実現した ACTIVIEW を提案する。さらに、提案する ACTIVIEW によってユーザ環境に最適なレイアウトを持つ Web ビューを提供するため、レイアウトの評価基準を定め、より良いレイアウトへの変換の実現を目指す手法として最適化制約と目標指標を提案する。ACTIVIEW によって Web ビューを生成する際、ユーザ環境に最適なレイアウトと構造的に一貫性のあるレイアウトを生成するため、2 つの制約と 3 つの目標を提案する。

ACTIVIEW : Implementation of Adaptive Web View Using SuperSQL

SANG-GYU SHIN,^{†1} YOKO MAEDA^{†2}
and MOTOMICHI TOYAMA^{†3}

In this paper, we propose ACTIVIEW to achieve dynamic adjustment to the terminal display size when information retrieved from a database is displayed as a view using SuperSQL, a query language which connects relational database to WWW. In addition, the criterion for the layout is proposed to offer the Web view with the optimized layout for the user environment by using ACTIVIEW. The restrictions and objective functions are proposed for achievement of the optimized layout conversions. For the Web View generated by ACTIVIEW to be the optimized for user environment and structurally consistent layout, we propose two restrictions and three objective functions.

1. はじめに

近年、WWW の普及とともに、WWW を利用したビジネスアプリケーションや、利用者への情報提供、企業内での情報開示がさかんに行われ、ユーザの Web ページに対する要求も多様化している。ページ提供側も、それにともない、1 人 1 人のユーザの環境に合わせたビューの提供や、個人向けに差別化された情報を提供することが望まれている¹⁾⁻³⁾。そのため、ユーザごとに異なるコンテンツのページを提供することや、近年増加している小型携帯端末からの WWW の利用を考慮した専用のページを別に作ることが求められている⁴⁾⁻⁶⁾。

本研究では SuperSQL システムを利用することにより、関係データベースからの検索結果を、ユーザごとに異なる Web アクセス環境に適応するアダプテーションの技術を提案する。本稿では、表示画面サイズに合わせて、表示する表のレイアウトを変換することで適応化を行うことをアダプテーションと呼ぶ。アダプテーションはパソコンに限らず、小型携帯端末や携帯電話の爆発的増加にともない注目されてきている。

本研究では、WWW からのデータベースの利用という観点から、アダプテーションの問題に取り組んでいる。データベースは、各種情報システムの中核として、多くの情報を格納している。データベースと WWW を統合することで、あらゆる場所から多くの情報を多くの方が利用できるようになる。Web を通して多くのユーザがデータベースに格納されている情報を共有する状況において、大部分はデスクトップコンピュータ環境が、特定のユーザアクセス端末機に固定された Web ページである。これらの手法では、変化するユーザの端末環境への動的な対応ができない。

最近、急速に増加している携帯端末を利用した Web データアクセスを見ても、端末機によって表示画面が異なるだけでなく、表示画面を縦や横に変えながらアクセスすることも可能になっている。このように、変化する端末のサイズや、発表される多様な端末機に 1 つずつ対応した（ユーザ環境に適応した）Web ページを提供するのは不可能である。

そこで、本研究では様々な端末からのデータのアクセスに対応し、ユーザ端末環境に適応したビューを動的に提供することに着目し、適応型表示ビュー ACTIVIEW を提案する。

†1 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University

†2 UBS 証券会社
UBS Securities Japan Ltd.

†3 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University

ACTIVIEW⁷⁾⁻¹⁰⁾ は、関係データベースと WWW の連携システムである SuperSQL¹¹⁾⁻¹³⁾ の基礎の上に開発したアダプテーション技術である。関係データベースからのフラットな検索結果を構造化する SuperSQL の機能を応用し、この構造を再構成することで様々なユーザ環境、すなわちユーザ表示画面に適応化した Web ビューの提供を行う。HTML の作成は SuperSQL システムを利用しているため、ACTIVIEW は SuperSQL 問合せ文をユーザの表示画面に合わせた表を生成するように書き直すことによってユーザの表示画面に適応した表を提供することが可能になる。

本研究では、環境に最適なレイアウトを持つ Web ビューを提供するため、レイアウトの評価基準を定め、より良いレイアウトへの変換の実現を目指す手法として 2 つの最適化制約と 3 つの目標指標を提案する。

以下、2 章では、関連研究の紹介とこれらの従来の研究と比較した本研究の特徴を述べる。3 章では、SuperSQL を用いて適応型表示ビューを自動生成する方法である ACTIVIEW について述べ、4 章では ACTIVIEW における適応化について、5 章では ACTIVIEW のレイアウト変換手法を、この変換手法の適用戦略による適応型表示ビューの生成に関しては 6 章で述べる。この実験と考察を 7 章で述べ、最後に 8 章で結論を述べる。

2. 関連研究と比較

本研究は、データベースからの検索結果を限られたサイズのユーザ表幅に収まるように、オリジナルのレイアウト指定に近い形式で自動変換するシステムの提案を目的としている。このため関連研究としては、制限された表示領域における適切な情報提示があげられる。

限られたユーザ表示画面サイズに適したレイアウトへの変換については様々な手法が研究されている。

まず Web コンテンツの省略・小型化を行い、ユーザに提示する Digestor システムが Bickmore らによって提案された¹⁴⁾。Digestor システムは、発見的手法によって求められた優先順位に基づいて Outlining、フォントの縮小、さらに画像の縮小または削除といった処理を行うため、オリジナルのイメージを壊すことなく Web コンテンツをユーザに提供できる。

Buyukkokten らは、単語の重要度を用いて文章の要約を行いテキストのみでユーザに提示する Power Browser システムを提案した¹⁵⁾。Power Browser システムでは、ユーザの必要としない情報はなるべく表示せず、ユーザの必要に応じて簡単な操作でより多くの情報を表示することができる Accordion Summarization という手法を用いている¹⁶⁾。

このほかに、Web コンテンツを PDA などの小さなページに適用するためにブラウザを提供する手法は、Bjork らの WEST¹⁷⁾ と、Bruijn らの RSVP Browser¹⁸⁾ などがある。WEST は、コンテンツの省略と、focus+context 視覚化手法による小さなターミナルへの Web コンテンツ提供技術を研究した。また、RSVP Browser はコンテンツを評価し、小型化されたイメージと簡略化されたテキストでページを作成する手法によって、小さな画面の端末に Web コンテンツを提供する技術を研究している。

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、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¹⁹⁾。

他に、XML と XSLT の適用による Web ページの提供や iPod などの特定の端末向けに開発されたアプリケーションの利用による Web アクセスなどがある。

しかしながら、これらの研究は、Web コンテンツを縮小や要約、リンクを用いるなど様々な手法で制限された領域に情報を収めているものの、レイアウト構造の変換は行っていない。多様な端末にエンベデッドされたアプリケーションからの Web アクセスも、一般性を持たずに特定な端末に限られている。

これらの研究と比べて本研究で開発した手法には次のような特長がある。

- (1) 関係データベースの検索結果を対象にする。
- (2) ユーザ環境に対して、表示画面サイズに適したレイアウトへの変換を動的に行うことで、特定の端末環境に限られず、多様なレイアウトの生成が可能である。

SuperSQL 処理系を利用するため、本研究では一般的な Web コンテンツではなく、データベースからの検索結果を対象にする。これは ACTIVIEW が SuperSQL に従属することによる制約であるが、その代わりに、SuperSQL の機能により HTML 以外の PDF などの多様なメディアに出版することが可能になるという利点にもなる。

ここで紹介した研究は、携帯端末や PDA などの限定されたユーザ環境を対象にしている。しかし、ユーザの環境は変化するもので、この変化するユーザ環境に動的に適応できる技術が必要とする。

ACTIVIEW は、レイアウトを生成する SuperSQL の問合せ文を書き換え、ユーザ環境に合わせたレイアウトを再構成することで、変化するユーザ環境への動的な適応が可能となる。他に、パソコンから同じ Web ページを見る際も、人によってモニタの使い方が異なる場合も考えられる。たとえば、モニタの全画面で Web ページを見る人がいれば、半分は Web ページを開き、もう半分は他の作業をする人もいる。また、作業する途中で、Web プ

ラウザの幅を変換する場合もありうる。このように、関係データベースの検索結果を多様な人の都合に従って Web データテーブルに生成し、提供することについては従来の研究ではなされてこなかった。

本稿で紹介する ACTIVIEW は、前田らによって提案された先行研究を基にしている⁸⁾⁻¹⁰⁾。前田らは SuperSQL 問合せ文の演算子を変換することで元のレイアウト指定からユーザ表示画面幅に合うレイアウトに再構成するシステムを Lisp ベースで実現した。しかし、レイアウト変換を単純な手法で行っていたため、ユーザに望ましいレイアウトとならない場合もあった。

SuperSQL システムが Java ベースになることとともに、ACTIVIEW システムも再作成する必要やより効率的なレイアウト変換のためにレイアウト変換手法を拡張する必要性が生じた⁷⁾。そのほか、再構成されるレイアウトの最適化問題やレイアウト変換の基準、再構成されたレイアウトを評価する手法や効率的な変換手法などの必要性が生じた。

亀岡ら²⁰⁾は SuperSQL におけるレイアウトの自動修正手法を提案している。この研究では生成対象を PDF として、固定サイズの紙面に対して、レイアウトを自動変換する手法を提案している。基本概念は ACTIVIEW と同様に SuperSQL の演算子の変更によるレイアウトの変換を基にしている。変換手法として、反復子によって反復される出力を折り返す装飾子の定義やラベル付けアルゴリズムの提案による長さ調整によって、幅と長さに制約がある出版物に対応している。しかし、変換するユーザ環境や Web 上での動的な対応、PDF より縦方向の制約がゆるい HTML を対象とすることなど、ACTIVIEW は前提条件が大きく異なっている。

3. ACTIVIEW の実現

ACTIVIEW に関して述べる前に、その基盤技術である SuperSQL について述べる。SuperSQL とは SQL を拡張した問合せ言語であり、データベースに問い合わせ得た平坦な構造の出力結果を構造化し、指定したメディアである XML, HTML, PDF²⁰⁾, Excel, SWF (Shockwave Flash) などへ変換するシステムである。SuperSQL のメディア生成技能を基礎にした ACTIVIEW の特徴として、問合せ文の出力結果が Web ページの表示環境に応じて変化することがあげられる。

3.1 SuperSQL の概要

SuperSQL とは、TFE (Target Form Expression) を用いて SQL を拡張した問合せ言語で、データベースに問い合わせた平坦な構造の出力結果を構造化し、指定されたメディア

の応用データへ変換するシステムである。TFE は、SQL のターゲットリストを拡張したもので、TFE 特有のオペレータ (結合子と反復子) を用いてレイアウト (生成されるテーブルの構造) を指定することができる。各結合子と反復子は次元に対応している。たとえば、HTML メディアを生成する場合 1, 2 次元は、表の行と列に対応し、3 次元はハイパーリンクに対応する。

SuperSQL の問合せ文は、SQL の SELECT 節を GENERATE < *medium* > < TFE > の構文を持つ GENERATE 節で置き換えたものである。ここでは < *medium* > は出力媒体を示し、XML, HTML など出版メディアの指定ができる。

出版メディアは、SuperSQL を通じ、データベースからの検索結果から生成できるメディアを示す。以下に、結合子と反復子について簡単に述べる。

- 水平結合子 (,) : 両オペランドのデータを横に連結して出力。図 1 の A。
- 垂直結合子 (!) : 両オペランドのデータを縦に連結して出力。図 1 の B。
- 深度方向への結合子 (%) : 両オペランドのデータをリンクを利用することで連結して出力。図 1 の C。
- 水平反復子 ([;]) : オペランドのデータがある限り、そのデータを横に繰り返し結合する。図 1 の D。
- 垂直反復子 ([!]) : オペランドのデータがある限り、そのデータを縦に繰り返し結合する。図 1 の E。

垂直反復子と水平結合子を組み合わせて用いると、図 1 の F のような出力が得られる。このような反復子のネストによってグルーピングを直感的に指定できる。

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、文字列の出力領域 (セル) の横幅、セル内での文字列の位置などの情報を付加する装飾子がある。これらは、装飾演算子 “@{ 装飾指定式 }” 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 “,” で区切る。たとえば、チーム名の背景を赤にして、セル内の横幅を 120 pixel にしたい場合は以下のように記述する。

```
チーム名@{bgcolor=red, width=120}
```

装飾子は、このように属性に指定するだけでなく、反復子や式の範囲、画面全体などに対して指定するものもある。

3.2 システム構成

全般的なシステムの流れを図 2 に示す。図 2 において、太線で囲まれた部分が SuperSQL

A	p.team , p.name	Japan Koji NAKATA	結果テーブル
	クエリーの例		
B	p.team ! p.name	Japan Koji NAGATA	結果テーブル
	クエリーの例		
C	p.team % p.name	Japan → link → Koji NAKATA	結果テーブル
	クエリーの例		
D	[p.name],	Koji NAKATA Shinji ONO ...	結果テーブル
	クエリーの例		
E	[p.name] !	Koji NAKATA Shinji ONO ... KAWAGUCHI	結果テーブル
	クエリーの例		
F	[p.team , [p.name] !] !	Japan Koji NAKATA Shinji ONO ... England BECKHAM	結果テーブル
	クエリーの例		

図 1 各結合子と反復子のクエリーの例と結果テーブル
Fig. 1 Connectors, repeaters and resulting tables.

処理系であり、点線で囲まれた部分が追加された ACTIVIEW を実現するための処理部分である。

まず、フロントエンド (Front End) がブラウザからユーザ表示画面の幅を受け取り、この情報を適応化処理部に渡す。

適応化処理部 (Adaptation Processor) は、フロントエンドが受け取ったユーザ幅に従って元の問合せ文をユーザ表示画面に収まるように書き換える処理を行う。この処理によって、ユーザの表示画面サイズに対応する。

書き換えた SuperSQL 問合せ文に基づき、SuperSQL の HTML 生成部は関係データベースから Web 上で利用可能な HTML を生成する。

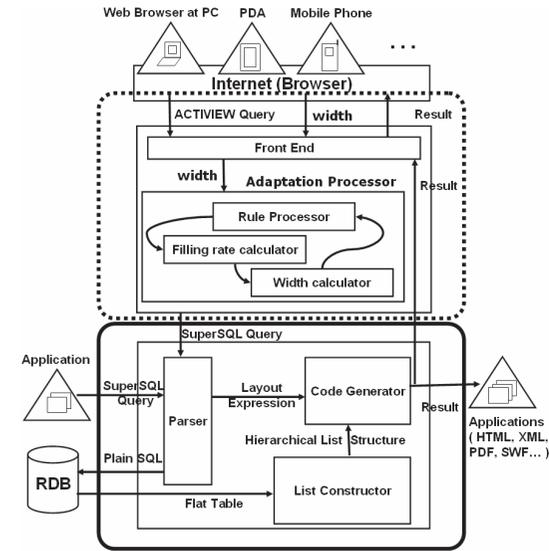


図 2 システムの概要図
Fig. 2 Overview of architecture.

3.3 ACTIVIEW の実装

ACTIVIEW の問合せ文 (図 3) は、フロントエンドが受け取ったユーザ表示画面サイズの情報を基に、適応化処理部によって、ユーザ表示環境の画面サイズに適応するように変換される。変換の手法は以下に示す。

- (1) フロントエンドからユーザ表幅の値 (表示画面サイズ) を受け取る。
- (2) 表の構造に関する情報と、装飾情報の“幅”に関する情報が混在する式 (レイアウト式とサイズ情報) を生成し、表の幅を計算する。
- (3) 生成されるレイアウトが本稿で提案する制約と目標を満たしたレイアウトではない場合は、提案手法に従ってレイアウト構造の変換を行う。
- (4) 各制約と目標を満たし、ユーザ表示画面に適応されたレイアウトが生成されるまで、(2), (3) を繰り返す。

本稿で提案する制約と目標に関しては、次章で詳細を述べる。

レイアウト情報を視覚的に理解しやすいように木構造を用いて表現すると、問合せ文 (図 3) から得られたレイアウト情報は図 4 のように表現できる。

幅制約とする .

ACTIVIEW システムが生成した SuperSQL の問合せ文によって再構成されたレイアウトの中で、ユーザ表示画面の幅に収まる幅を持つレイアウトを、幅制約を満たしたレイアウトとする . すなわち、 q を問合せ文、 n を属性の数、 e_i を属性、 i を属性の順序、 $W(e_i)$ を属性 e_i の幅、 $W(U)$ をユーザ表示画面の幅とすると次の式に従って、問合せ文に指定された各属性の幅の合計がユーザ表示画面の幅値、 $W(U)$ より小さい幅値を持つレイアウトが幅制約 $RW(q)$ を満たしたレイアウトになる .

$$RW(q) = \begin{cases} true, & \text{if } \sum_{i=1}^n W(e_i) \leq W(U) \\ false, & \text{if } \sum_{i=1}^n W(e_i) > W(U) \\ exception, & \text{if } \exists i W(e_i) > W(U) \end{cases}$$

条件を満たすまでレイアウトの変換を行う . しかしながら、属性の中に、ユーザ表示画面の幅より広い幅を持つ属性が存在する場合があります . この場合は、レイアウトを変換してもユーザ表示画面の幅に収まるレイアウトは生成できない . これは例外として、本システムではユーザ表示画面の幅に合わせて属性の幅値の変換を行う .

図 4 のレイアウト情報とテーブルの幅 (width) に関する装飾情報を合わせて得られる木構造表現を図 6 に示す .

図 6 はレイアウトの変換によって、レイアウトの幅値が変化する例の 1 つである . ユーザの表示画面の幅が 950 であると、図 6 の左側に示した元のレイアウト幅 (1,260) では幅制約を満たせない . 幅制約を満たすために、変換を行ったレイアウト (右側) の幅値は 940 になり、幅制約を満たしたレイアウトになる .

表 1 に示す計算方法を用いてレイアウト式から生成されるレイアウトの幅が求まる . 計算方法は表 1^{*1}に示す . また、 $sum()$ 、 $max()$ は関数であり、以下の計算をする .

$sum()$: () 内の値の合計 .

$max()$: () 内の最大値 .

この計算方法を図 6 に適用すると

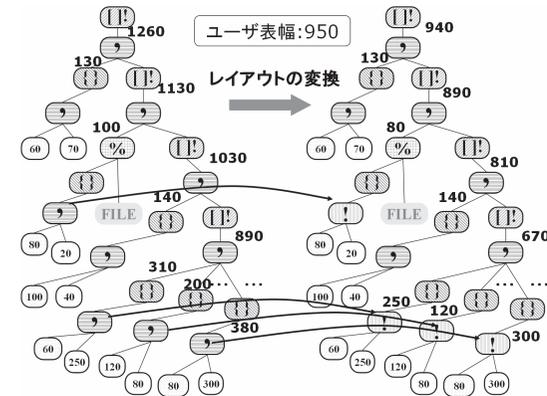


図 6 レイアウト-サイズ式の木構造表現
Fig.6 Tree of layout-size expression.

表 1 結合子ごとの計算

Table 1 Calculation of width for connectors.

次元	結合子	計算法
1	E_i, E_j	$(sum\ width(E_i)\ width(E_j))$
2	$E_i ! E_j$	$(max\ width(E_i)\ width(E_j))$
3	$E_i \% E_j$	$width(E_i)$

● レイアウト変換の適用前 :

$$\begin{aligned} & (sum (sum 60 70) (sum (sum 80 20) \\ & (sum (sum 100 40) (sum \\ & (sum 60 250) (sum 120 80) (sum 80 300)))) \\ & = 1260 \end{aligned}$$

● レイアウト変換の適用後 :

$$\begin{aligned} & (sum (sum 60 70) (sum (max 80 20) \\ & (sum (sum 100 40) (sum \\ & (max 60 250) (max 120 80) (max 80 300)))) \\ & = 940 \end{aligned}$$

と計算でき、レイアウトの変換後の幅が小さくなっていることが分かる .

*1 File はハイパーリンク先のファイル、width() は指定された TFE の表示幅を求める関数

4.2 開発者制約

動的なレイアウト変換によって、データを提供する側、すなわちページのデザイナー（開発者）が初期的に与えたレイアウトとは意図の異なるレイアウトが提供されることは望ましくない。そこで、開発者制約を以下のように定義する。

定義 2（開発者制約） 初期問合せ文における構造化指定による開発者の意図の反映を開発者制約とし、レイアウトの変換を行う際に尊重する。

開発者制約では、まず開発者が与える問合せ文をどこまでレイアウトに関する制約ととらえるかが問題となる。開発者が与えた入れ子構造をシステムが変更してしまえば、意味的に開発者が意図していないレイアウトができあがってしまう可能性がある。

開発者は初期問合せ文を作成する段階で、以下の 3 つの事項をレイアウトに盛り込むことで、レイアウトの構造に意図を指定できる。

属性の順序、ネスト、グルーピング

ネストは、SuperSQL の反復子のネスト指定によって、同じ属性値を持つタプルをひとまとめにすることである。SuperSQL のネストの適用によってひとまとめになる属性と、これらの属性によって入れ子化されたレイアウトが生成できる。

また、グルーピングは中括弧（{ }）で複数の属性を含む部分式を括ることで、括られた属性の関連性が他の属性より高いことを指示する。続いて、この 3 つの開発者制約について詳細を述べる。

4.2.1 開発者制約—順序

順序制約は次のように定義する。

定義 3（開発者制約—順序） 元の問合せ文に現れる属性の順序を開発者制約の 1 つである順序制約とする。

たとえば、図 7 のように選手と国の順に定義された問合せ文から、システムが属性の順序を変更した場合、開発者が意図した意味とは異なるレイアウトが生成される恐れがある。これによって、開発者が意図した属性間にあった関連性が崩れてしまう。

4.2.2 開発者制約—入れ子

SuperSQL では問合せ文に反復子のネストを指定することによって、同じ属性値を持つタプルをひとまとめにすることができる。すなわち、この指定により入れ子構造のレイアウトが生成できる。

SuperSQL のネストの適用によってひとまとめになる属性と、これらの属性をまとめて入れ子構造にする属性は強い関連性を持っている。図 8 のように同じデータに対するもの

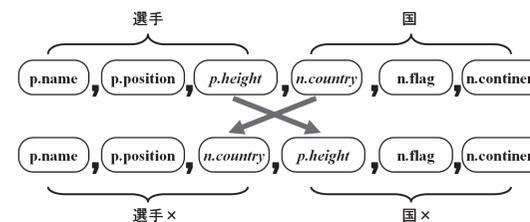


図 7 開発者による属性の順序定義
Fig. 7 Developer defined order of attributes.

[team_country ,
[player, position]!
]!

England	Michael OWEN	FW
	Peter CROUCH	FW
	David BECKHAM	MF

Japan	Shinji ONO	MF

...

(a) チームによるネスト

[[team_country ,
player]! , position
]!

England	Michael OWEN	FW
England	Peter CROUCH	FW
England	David BECKHAM	MF
England	...	MF
...
Japan	Shinji ONO	MF
Japan	...	MF
...

(b) ポジションによるネスト

図 8 表ネストの違いによる意味の違い

Fig. 8 Difference of meanings by different nesting of tables.

であっても、どの属性を用いてネストを行うかによって異なる意味を持つレイアウトが生成される。図 8 (a) は、チームごとにネストされた各選手の名前とポジションを表示するレイアウトであり、(b) は各ポジション別にネストされたチーム名と選手を表示している。したがって、レイアウトの変換を適用する際、開発者の意図、すなわち、あらかじめ開発者によって与えられたネスト構造に関する条件は満たすことを前提とする。

したがって、入れ子制約の定義は次のようになる。

定義 4（開発者制約—入れ子） 元の問合せ文に現れる反復子によるネスト指定に従って入れ子構造を維持する条件を、開発者制約の 1 つである入れ子制約とする。

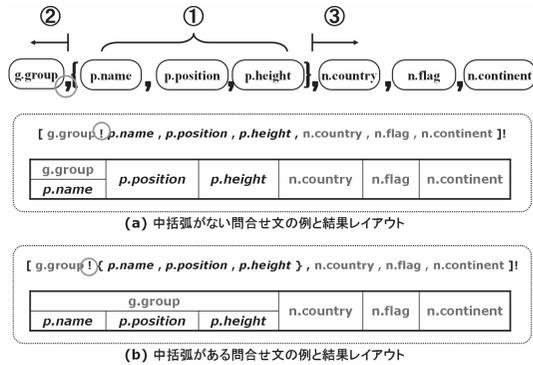


図 9 SuperSQL 問合せ文中の中括弧の役割
Fig. 9 Role of braces in SuperSQL query.

4.2.3 開発者制約—グルーピング

SuperSQL では演算子の適用順序を明示する目的で問合せ文にグルーピング ({ } 演算子) を指定することができる。ここで、中括弧で括られた属性の間は他の属性との間の関係より高い関連性を持っていると判断することができる。

たとえば、図 9 の①のように中括弧で括られた属性、p.name、p.position、p.height の 3 つの属性の関係は、他の g.group ②や n.country ③などの属性との関係より関連性が高いことを意味する。SuperSQL システムでは要素が中括弧でくくられると 1 つの要素としてまとめて処理する。図 9(a) で示すように横連結子 (,) が縦連結子 (!) に変換される際 (図 9 の丸の中の連結子)、問合せ文に中括弧を使用することによって、異なるレイアウト指定が可能である。図 9(b) の場合は中括弧が使用された場合であって、3 つの属性が同時に移動する。

定義 5 (開発者制約—グルーピング) 属性を中括弧で括ることによって、元の問合せ文に指定されているグルーピング定義による構造制約を、開発者制約の 1 つであるグルーピング制約と定義する。グルーピングされた属性の間は高い関連性を持っていると判断し、レイアウトの変換を行う際に尊重する。

4.3 幅占有率目標

初期問合せ文に対し構造の変換を行うことで生成されたレイアウトの中で、制約を満たしたレイアウトが複数存在する場合、ユーザ表示画面に最適化されたレイアウトを選択するためには複数存在する候補レイアウトから結果レイアウトを決定するための指標が必要に

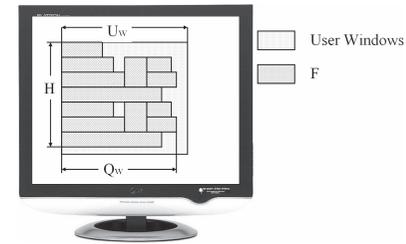


図 10 ユーザ表示画面に対する結果レイアウト
Fig. 10 Result layout on user window.

なる。本稿では、今まで述べた 2 つの制約とともに適応化の指標として「幅占有率目標」、「充填率目標」、「長さ目標」の 3 つの目標を提案している。まず、幅占有率を次のように定義する。

定義 6 (幅占有率) ユーザ表示画面に対して構成されたレイアウトの幅の割合を幅占有率とする。

すなわち、幅占有率はユーザ表示画面の幅 $W(U)$ に対する結果レイアウトの幅 $W(R)$ の比率 (図 10) である。制約を満たしたレイアウトが複数存在する場合は、指標の 1 つである幅占有率を基準とした評価関数によるランキングを行う。問合せ文 q から生成されたレイアウトの幅占有率は次のような評価関数 $TW(q)$ によって求める。

$$TW(q) = \frac{W(R)}{W(U)} \times 100$$

4.4 充填率目標

複数存在する候補レイアウトから充填率が高いレイアウトは表示空間を有効に利用していると考えられる。充填率は以下のように定義する。

定義 7 (充填率) ユーザ表示画面に対して構成されたレイアウトの占める面積の割合を充填率とする。

すなわち、ユーザ表の大きさである { 表示画面の横幅 ($W(U)$) × 表の長さ (H) } の内部に実質データが表示に要する面積 (F) の比率を充填率 (図 10) とし、候補レイアウトの中からより充填率が高いレイアウトを優先する。充填率の評価関数 $TF(q)$ は以下のように定める

$$TF(q) = \frac{F}{W(U) \times H} \times 100$$

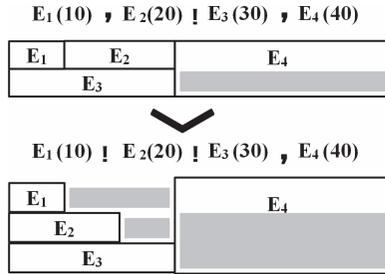


図 11 充填率の比較
Fig. 11 Comparison with filling rate.

図 11 は充填率による評価の 1 つの例である．網かけ部分はデータの表示されていない空白を表示している．図 11 のように同じ幅を持つレイアウトであってもレイアウトの構造によって異なる充填率を持つ場合がある．この場合はより高い充填率を持ち，効率的に空間を利用していると判断される上のレイアウトが優先となる．

4.5 長さ目標

幅制約を満たしたレイアウトを生成する際，縦に非常に長いレイアウトのテーブルが生成されてしまうおそれがある．この問題を解決するため，長さ目標を定義し，この条件を満たしたレイアウトを候補レイアウトとする．

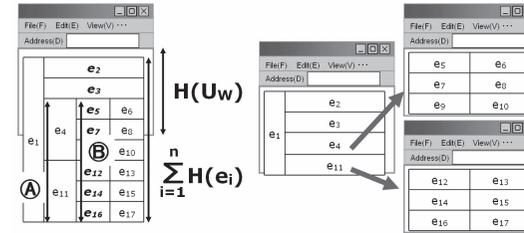
定義 8 (長さ目標) 構成されたレイアウトの長さがユーザ表示画面の長さとの割合である式 $TH(q)$ を満たしている場合，長さ目標を満たしたレイアウトとする．

問合せ文 q に対する長さ目標関数 $TH(q)$ は，次のように定義する．

$$TH(q) = \begin{cases} true, & \text{if } \sum_{i=1}^n H(e_i) \leq H(U) \times d(q) \\ false, & \text{if } \sum_{i=1}^n H(e_i) > H(U) \times d(q) \end{cases}$$

$H()$ はレイアウトの長さを計算する関数である．生成されるレイアウトの長さは，縦に連結されている各要素のセルの長さ（高さ，HTML の height で指定する値で以下「長さ」に統一する）を足しあわせることで計算できる（図 12 (a) のボールドのイタリック体で示した要素）．

構成されたレイアウトの長さ（ $\sum_{i=1}^n H(e_i)$ ）がユーザ表示画面の長さ（ $H(U)$ ）に，ある



(a) 長さ目標を満たす前のレイアウト (b) 長さ目標を満たしたレイアウト
図 12 長さ目標
Fig. 12 Length objective.

倍率 $d(q)$ を掛けた長さより短いレイアウトでないと長さ目標は満たさない． $d(q)$ は次のように定義する．

$$d(q) = \begin{cases} \frac{lst(H(n_x))}{H(U) \times IC_x}, & \text{if } \frac{lst(H(n_x))}{H(U) \times IC_x} \leq y \\ y, & \text{if } \frac{lst(H(n_x))}{H(U) \times IC_x} > y \\ 1, & \text{if } \frac{lst(H(n_x))}{H(U) \times IC_x} < 1 \end{cases}$$

$lst(H(n_x))$ は入れ子構造を構成している各ネスト n_x （図 12 (a) の A と B の）の中で一番長い長さを持つネストの長さである．図 12 (a) の A と B の長さは構造化を行った後，実際現れる属性の長さを合計して計算する．すなわち，ネスト A の長さは e_4 と e_{11} が縦に連結されるレイアウトで，セルの長さが 30 pixel であるとしたら，サブレイアウトの長さは $2 \times 30 = 60$ になり，ネスト B の長さは $6 \times 30 = 180$ になってネスト B が $lst(H(n_x))$ になる．各ネスト別に計算することは長さ目標を満たすため，リンク変換が適用されると，リンクが付けられた各ネスト別のレイアウトが生成されるからである．

問合せ文が

$[A, [B]!]!$

のようになっていると，B の構造は A の値の数だけ反復される．そこで，B の反復回数 (IC) は A の数^{*1}になる．すなわち，ネストと属性の間の連結子にリンク変換が行われると，ネストの長さを属性の反復回数で割った長さのレイアウトが生成される．例の図 12 で示すよ

*1 ここでの数は属性に対する検索結果の数ではなく，データ反復回数 (IC_x) (6.2.1 項) によって計算された値である．

うに, (a) では 180 pixel だったサブレイアウト B の長さが (b) では 2 つのページに分けられ, 90 pixel になるのが分かる.

y は開発者が指定した値である. 一番長いサブレイアウトをユーザ表示画面の長さとして反復回数で割り算した結果が, 開発者指定の変数 y より長い場合は開発者指定の変数 y が基準になる. この場合は, リンク指定で複数になったレイアウトの長さが開発者指定の変数 y を満たさない場合がありうるため, 可能な範囲での長さ目標を許容する.

たとえば, 開発者が指定した y が 3 であれば, ACTIVIEW システムは開発者指定に従って結果レイアウトの長さがユーザの表紙画面の長さより 3 倍以内であれば, 長さ目標を満たしたレイアウトであると判断する. $lst(H(n_x))$ が 1,200 pixel で, ユーザの表示画面の長さが 100 pixel, 反復回数が 3 である場合はリンク変換によって生成されるレイアウトの長さが 400 pixel と予想され, 開発者の基準 300 pixel では (100×3) 長さ目標は満たされない. この場合の $d(q)$ は定義のとおり, 計算結果ではなく開発者指定の 3 になり, 例外のサブレイアウト以外のサブレイアウトは開発者が指定した基準で再構成を行う. これは, サブレイアウトの長さを短くしても目標基準を満たすことができなくなるという問題が生じてしまうからであり, 例外として扱う. しかし, $lst(H(n_x))$ が 600 pixel であれば開発者指定は超えるが, 反復回数で計算すると 200 pixel になり, 開発者指定より短くすることが可能なレイアウトであることを意味する. この場合は定義のとおり, 200 pixel が長さ目標の基準になる.

問合せ文からの各ネストの長さの予想する手法については 6.2.1 項で詳細を述べる.

5. ACTIVIEW によるレイアウト変換

本章では, 元の間合せ文によって生成される初期レイアウトから, 制約や目標を満たすレイアウトを生成するレイアウト変換手法について述べる.

5.1 レイアウト変換の基本概念

SuperSQL の問合せ文によって生成される Web ビューは, 3.1 節で説明したとおりに問合せ文に指定する TFE の定義によって構造化される. ACTIVIEW による構造の変換は, 元の間合せ文に指定されている TFE の変換による 2 つの手法が考えられる⁸⁾.

- (1) 横連結 (,) を縦連結 (!) に変換する.
- (2) 横連結 (,) か縦連結 (!) をハイパーリンク連結 (%) に変換する.

(1) の手法は, 横に並んで表示されるべきデータが縦に並べられるので, 表としてデータ間の関係が理解しやすく, ハイパーリンクで辿るよりも手間が少ない. しかしながら, 大量

- ① GENERATE ACTIVIEW [group , country , player]! FROM... WHERE...
- ② GENERATE ACTIVIEW [group ! Country , player]! FROM... WHERE...
- ③ GENERATE ACTIVIEW [group ! Country ! player]! FROM... WHERE...

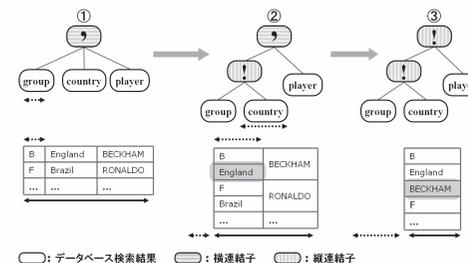


図 13 横縦変換

Fig. 13 Conversion from width to length.

のデータによって非常に長いレイアウトが生成される場合や携帯端末のように表示画面のサイズが極端に小さいものでは, (2) の手法も採用する. まず, 携帯端末に代表される小型端末の特徴を以下にあげる.

- 小さい表示画面
- 上下左右ボタンによるスクロール操作

これらの特徴を考慮し, (2) の手法を採用する理由は, まず, ユーザが必要とする情報を素早く提供するため, 限られた画面に多くの種類のデータを表示する必要がある点である.

次に, スクロールに大きな手間がかかる場合, ハイパーリンクで情報に飛ぶときの手間の方が少なくなることがあげられる.

この基本概念に従ってレイアウト変換手法を「横縦変換」と「リンク変換」の 2 つの具体的手法として定義する. これらの発見的なレイアウト変換手法を用い, ユーザ表示画面に適応化されたレイアウトを提供する手法を提案する.

5.2 横縦変換による幅変化

横連結から縦連結にレイアウトを変換することで「幅制約」を満たしたレイアウトが生成できる. さらに, この変換の適用によって「幅占有率目標」や「充填率目標」に従ったレイアウトの生成が可能になる. 横縦変換を次のように定義する.

定義 9 (横縦変換) ACTIVIEW 問合せ文に指定されている横連結 (,) を 1 つ, もしくは複数選び, 縦連結 (!) に変換する手法を横縦変換とする.

すなわち, 図 13 のように, 元の間合せ文にある横連結 (,) が 1 つずつ縦連結 (!) に

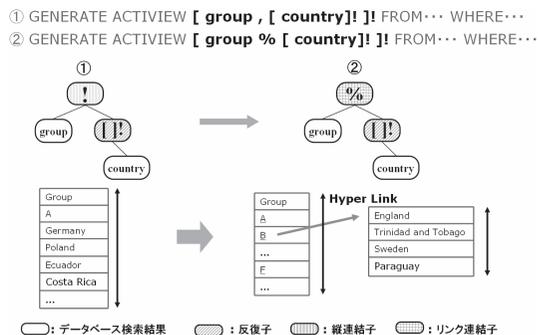


図 14 リンク変換
 Fig.14 Conversion to link.

変換されることで、縦に変換される連結子の右辺の分だけ幅が狭いレイアウトが生成される。

5.3 リンク変換による長さ変化

「長さ目標」を満たしたレイアウトを生成するため、リンク変換を定義する。

定義 10 (リンク変換) ACTIVIEW 問合せ文に指定されている結合子を 1 つずつ選び、リンク連結 (%) に変換する手法をリンク変換とする。

図 14 のようにリンク変換を適用することによって、リンク元の Web ページとリンク先の Web ページの複数のレイアウトが生成される。生成された複数のレイアウトはそれぞれ制約と目標を満たさなければならない。

6. レイアウト変換適用の戦略

ACTIVIEW では、レイアウトの変換手法に従いつつ、制約と目標に合わせて変換することで、ユーザの表示画面への適応化を実現する。ここで問題となるのが、どの連結子を縦連結またはリンク連結に変換するのかという点である^{9),20)}。

ACTIVIEW では多様なユーザのアクセス環境に対して、属性間の関連性に基づいたユーザに見やすいレイアウトの動的な生成を目指す。まず、計算コスト削減による動的なユーザ対応を実現するため、ACTIVIEW システムで前処理を行うことによってレイアウト変換の候補数を減らす手法を提案する。さらに、属性間の関連性を基に制約と目標に従うレイアウトの変換戦略による、ユーザに見やすいレイアウトの生成を実現する。

6.1 簡易なレイアウト変換

簡易なレイアウト変換手法⁷⁾として、結合子が現れる順序どおりにレイアウト変換を適用する「単純順序探索」と、適用可能なすべての結合子に対して変換を行うことによって生成される、全レイアウトを対象にする「全数探索」の 2 つの手法がある。

「単純順序探索」によるレイアウト変換手法は、問合せ文に属性が現れる順どおりにレイアウト変換を適用する手法である。m 個の属性に対し、レイアウト変換の適用が可能なのは m - 1 の連結子であり、生成可能なレイアウトの数は m - 1 個になる。これは、必ず満たすべきである制約条件を満たしたレイアウトを生成可能だが、目標に従った最適化は行われない。そのため、「単純順序探索」によるレイアウト変換手法では、ユーザに望ましくない(見にくい)レイアウトが生成され⁸⁾、ユーザ表示画面に最適化されたレイアウト変換手法ではない。

「全数探索」の手法は元の問題文から生成可能な全レイアウトパターンを対象にしているため、最適な目標の結果値を持つレイアウトを生成することができる。しかしながら、属性数が多いときに処理時間が非常に長くなるという問題があり、Web 上でのユーザに対する動的な対応が困難になる。

そのうえ、属性間の関連性を考えていないレイアウトは、高い結果値で目標を満たしたレイアウトを生成していても、ユーザに見やすいレイアウトであるとはいえない。n 個の演算子を持つ問合せ文から構成されるレイアウトの数 $P_L(n)$ は

$$P_L(n) = 3^n$$

になる。そこで、計算コストを下げながら、制約と目標に従った変換を求める適用手法が必要になる。

6.2 ACTIVIEW のレイアウト変換

ACTIVIEW では、計算コストを下げる手法として、長さ目標を満たさないレイアウトに対して「リンク変換」を先に行う手法と、「統一変換」が可能な連結子に対しては同時に変換を行う手法を提案する。そして、一貫性のあるレイアウトを生成するために、5 つの発見的な戦略を提案する。

6.2.1 リンク変換の導入による計算コスト削減

初期レイアウトにリンク変換を先に適用することによって、変換の対象になる連結子の数を分ける。リンク変換の適用は次の優先順位に従う。

- (1) 反復子との連結子に適用することを前提にする。
- (2) 多くの子要素を持つネストからの変換を優先する。

id	position	name	birth	height	weight
1	GK	Yoshikatsu KAWAGUCHI	15/08/1975	179	78
2	DF	Teruyuki MONIWA	08/09/1981	181	77
3	DF	Yuichi KOMANO	25/07/1981	171	74
4	DF	Tsuneyasu MIYAMOTO	07/02/1977	176	72
5	DF	Koji NAKATA	09/07/1979	182	74
6	FW	Naohiro TAKAHARA	04/06/1979	180	74
7	FW	Selichiro MAKI	07/08/1980	184	77
8	FW	Atsushi YANAGISAWA	27/05/1977	177	75
9	MF	Yasuhito ENDO	28/01/1980	177	75
10	MF	Hidetoshi NAKATA	22/01/1977	175	72
...
23	MF	Shunsuke NAKAMURA	24/06/1978	178	69

4 23 23 14 13

図 15 データ反復回数
Fig. 15 Number of repeated occurrences.

- (3) 深いレベルにあるネストからの変換を優先する (図 18 の LEVEL 参照).
- (4) 中括弧内のリンク変換は禁止する.

反復子の入れ子指定によるネスト構造とこれと連結されている属性は、他の属性の間より関連性が高く、関連性が高い属性が同じ入れ子に含まれることからこの関連性を維持するため、(1) の戦略を前提とする。これによって、関連性が高い属性どうしのレイアウトが生成されてユーザにとって直感的なレイアウトの提供ができる。

入れ子構造から長いネストを優先することで少ない変更によって長さ目標を満たす確率が高くなる。多くの子要素を持つネスト構造からリンク変換を行うためには、初期問合せ文から生成される各ネストの長さを計算しなければならない。そこで、データ反復回数 (IC_t) を導入する。リレーション r にある属性 t におけるある値が、どれだけ重複して出現しているか (図 15) を示す IC_t は

$$IC_t = G_{count-distinct(t)}(r)$$

となる。初期問合せ文の各ネストに含まれている属性 t の IC_t を計算することで各ネストの長さが予想できる。たとえば、図 3 から反復子と連結されている各属性のデータ反復回数を計算すると、表 2 ようになる*1。表 2 の長さは生成されるレイアウトのセルの長さを 30 pixel と仮定したうえの結果である。

リンク変換戦略に従って $p.name$ の方に最初にリンク変換が適用され (図 16 の A の連結子) 図 16 の B のようなレイアウトが生成される。22,080 pixel の長さを持つネストがリンク変換によって 690 pixel の長さを持つレイアウトに変換されると予想できる。優先順位

*1 FIFA ホームページ²¹⁾ のデータに基づく。

表 2 各ネストにある属性 t の IC_t 計算結果
Table 2 Calculation results of elements in each nest.

属性 t	検索結果	IC_t	長さ
h.uear	2006	1	30
t.group	A, B, ..., H	8	240
p.position	GK, MF, FW, DF	4	120
p.name	NAKATA, ...	736	22,080

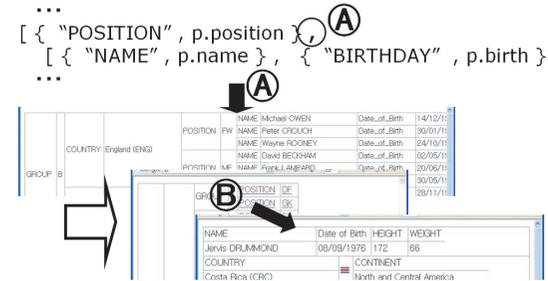


図 16 リンク変換の適用
Fig. 16 Conversion to hyperlinks.

によって次は $t.group$ と反復子による入れ子指定の間の連結子にリンク変換を行う。

開発者は問合せ文を作成する際、重要度が高いコンテンツを持つ属性を前面に指定し、重要度が低い属性は奥の方に、あるいはネスト構造ではネストされる属性を含む上位カテゴリの属性をネスト構造の親ノートの方に指定するのが一般的である。したがって、戦略 (3) を設定した。これに基づくのが戦略 (4) である。

中括弧内の属性は関連性が高いため、この間にはリンク変換を行うと開発者が意図した意味的な構成が崩れる。

6.2.2 統一変換可能な連結

元の問題文に統一変換が可能な連結が存在すれば、同時に変換を適用することで、実質的に変換の対象になる結合子の数を減らすことができ、計算コストの削減が可能になる。

経験的に属性間の関連性が高い連結に対して統一変換を行うことにより、一貫性を持つレイアウトを生成する可能性が高いと考えられる。

たとえば、同一中括弧内の属性のように関連性が高い連結において、同時に変換を行って生成されたレイアウトが、連結子それぞれに対し、全数探索によって目標の結果値を最大化

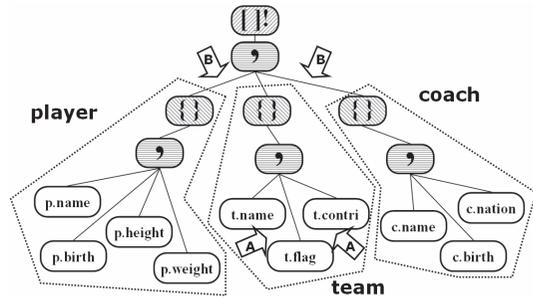


図 17 統一変換可能な連結
Fig. 17 Consistent conversions.

するよりも、構造的には一貫性のあるレイアウトである場合がある。

図 17 の場合、属性間の関連性を考えず全数探索の手法でレイアウトを生成すると、player 領域に含まれている 4 つの属性を連結する 3 つの連結子、team の 2 つ、coach の 2 つと、3 つの中括弧の間の 2 つの連結子に対して変換を行い、 3^9 個のレイアウトパターンが生成される。6.1 節では n 個の演算子を持つ問合せ文から構成されるレイアウトの数 $P_L(n)$ は $P_L(n) = 3^n$ になるが、提案手法では % 連結子の適用は特別の戦略に従うため、変換パターン数は横と縦変換の 2 種類として 2^n にする。また、中括弧で含まれている領域を 1 つのグループとして扱おうと、各中括弧で含まれている領域に統一的に変換を適用するかしないかの各 2 通りが 3 つと、中括弧の間の 2 つの連結子に対する変換を含む 2^5 個でレイアウトの変換を行うため、全数探索より計算コストの削減が可能になるとともに、一貫性のあるレイアウトが生成可能である。

6.3 一貫性のあるレイアウトの生成

ACTIVIEW での一貫性のあるレイアウトは、次の優先順位に従い、6 つの戦略によって生成される構造のレイアウトであるとする。

- (1) 複数のネスト内で行う変換より同一ネスト内での変換を優先する。
- (2) 各属性間の変換より入れ子や中括弧との連結子への変換を優先する。
- (3) 同一入れ子内では親ノードの方を優先する。
- (4) 変換の数が少ないレイアウトを優先する。
- (5) 中括弧内の連結子は統一変換を優先する。
- (6) これ以外は、全数探索によって生成されるレイアウトのうち、高い目標の結果値を持つ

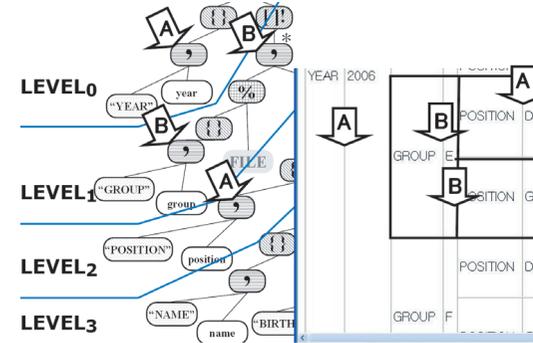


図 18 一貫性のあるレイアウトの生成の例
Fig. 18 Example of generating consistent layout.

つレイアウトを優先する。

これらの戦略は制約を満たしながら適用される。

6.3.1 同一入れ子内での変換優先

制約を満たしたレイアウトであれば、複数のネスト内で行った変換によるレイアウト結果より、同一ネスト内での変換で構成されるレイアウト結果を優先する。たとえば、目標の結果値が同じであるレイアウト結果が複数あれば、図 18 の A のように複数のネスト内で変換が行われたレイアウト結果より、B のように同一ネスト内で変換を行って構成されたレイアウトを結果レイアウトとして提供する。ACTIVIEW では、同一領域内での変換が複数の領域での変換より、全体レイアウトに与える影響が少ないと仮定している。

6.3.2 反復子 (ネスト指定), グルーピングとの結合優先

属性と属性間の連結子で行われた変換より、反復子と連結子やグルーピングされた属性を意味する中括弧との連結子に変換を行うことを優先する。これは、入れ子内の属性の構造が変換され、レイアウト全般に反復されながら影響を与えるより、反復子と中括弧との連結子に変換が適用され、入れ子や中括弧で括られた属性は全体的に変換される方が全体的にはレイアウトへの影響は少ないという経験則に基づく。ACTIVIEW では、属性間の連結子より入れ子や中括弧との連結子が構造的には弱いと判断することで、これを実現する。

6.3.3 上位ノード優先

候補レイアウトが複数存在する場合、より上位のノードの演算子に変換を行ったレイアウトの方を優先する。たとえば、図 17 の A と B で変換を行って再構成されたレイアウト結

果の幅目標の評価関数の結果値が同じであれば、B に変換が適用されたレイアウトを結果レイアウトとする。同一ネスト内でのノード関係は、6.3.2 項と同じく、上位ノードの方が弱い結合であると判断し、上位ノードの変換を優先する。

ネスト間での変換も、同様に上位のネストで行う変換が下位ネスト（深いネスト）で行う変換より優先される。

6.3.4 少ない変換適用を優先

ACTIVIEW では、全体レイアウトに与える影響が少ないレイアウトの方が元の開発者が指定したレイアウトに近いレイアウトであると仮定している。開発者によって定義された初期レイアウト構造が一番ユーザに直感的で見やすいレイアウトであるという前提に基づいている。したがって、変換の数がより少ないレイアウトを元の構成により近いレイアウトであると判断している。

6.3.5 結果レイアウトの選択

複数生成される候補レイアウトから、最もユーザ表示画面に適したレイアウトを提供するため、判断指標が必要となる。ACTIVIEW では、今まで述べた各目標の結果値の比較と連結子の変換が行った数、変換優先順位（6.3 節）の比較によって結果レイアウトを選択する。各条件を結合し、長所が多いレイアウトが選ばれる。

7. 実験および評価

提案手法により得られる結果レイアウトの妥当性と、処理時間の測定のために行った実験およびその結果について示す。

7.1 実験方法

ACTIVIEW は、従来の研究と比べ、データの内容ではなく構造そのものを対象にしているため、今回の実験では直感的に構成の意味を理解しやすい World Cup のデータ²¹⁾ を利用している。FIFA ホームページからの実際のデータをベースに図 19 のようなデータテーブルを作成した。

元となる問合せ文は図 3 の問合せ文をそのまま使う。図 3 の問合せ文から、変換を適用する前に生成されるオリジナルレイアウト（図 20）の幅は 3,030 pixel である。図 20 は図 3 の問合せ文から生成されたレイアウトの結果である。

図 3 の問合せ文から生成されるレイアウトは「2006 年」で大きくネストされた各グループにあるポジション別の選手とコーチの情報を表するテーブルである。すなわち、選手の情報は、まず対戦グループ別、そしてポジション別にネストされた深さ 4 のテーブル（図 18

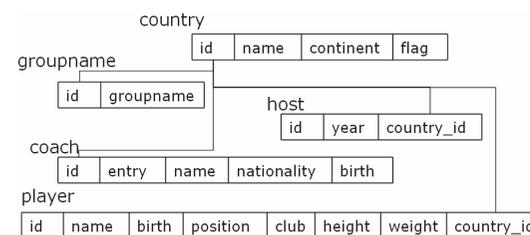


図 19 データベースのテーブルスキーマ
Fig. 19 Schema of database table.

YEAR	GROUP	POSITION	NAME	Date of Birth	HEIGHT	WEIGHT	COUNTRY	CONTINENT
2006	D	GK	PAULO FERREIRA	18/01/1979	181	76	Portugal (POR)	Europe
			Ebrahim MIRZAPOUR	16/09/1978	192	85	Iran (IRN)	Asia
			JOAO RICARDO	07/01/1970	180	76	Angola (ANG)	Africa
		MF	Oswaldo SANCHEZ	21/09/1973	184	79	Mexico (MEX)	North and Cent
			RICARDO	11/02/1978	188	80	Portugal (POR)	Europe
			Mehdi MAHDAVIRIA	24/07/1977	176	73	Iran (IRN)	Asia
	E	DF	Chris ALBRIGHT	14/01/1979	185	83	USA (USA)	North and Cent
			Cristian ZACCARDO	21/12/1981	184	77	Italy (ITA)	Europe
			Hans SAPPEI	28/06/1976	178	68	Ghana (GHA)	Africa
		GK	Zdenek GRYGERA	14/05/1980	185	78	Czech Republic (CZE)	Europe
			Gianluigi BUFFON	28/01/1978	190	83	Italy (ITA)	Europe
			Petr CECH	20/05/1982	197	87	Czech Republic (CZE)	Europe
			Sammy ADJEI	01/09/1980	188	80	Ghana (GHA)	Africa
			Tim HOWARD	06/03/1979	187	90	USA (USA)	North and Cent

図 20 元の問合せ文から生成されるオリジナルレイアウト
Fig. 20 Original layout generated from original query.

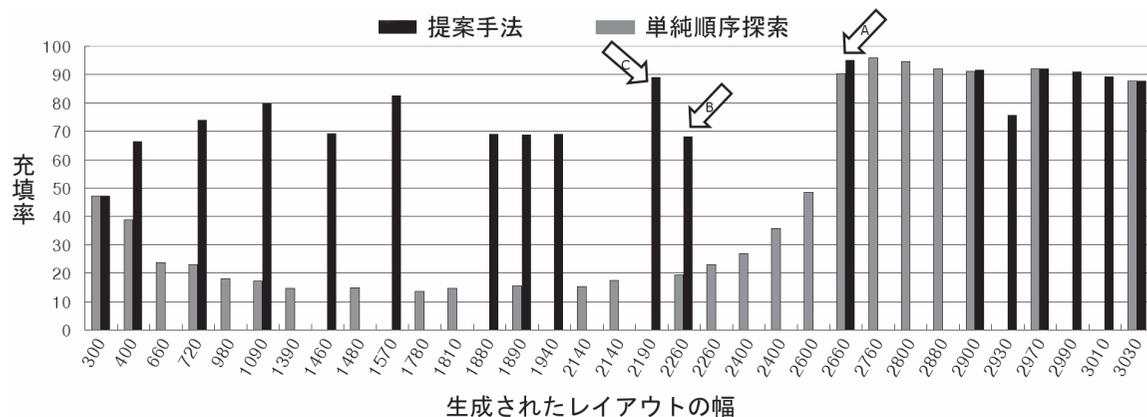


図 23 変換されたレイアウトの充填率
Fig. 23 Filling rate of the converted layout.



図 21 結果レイアウトの例 1
Fig. 21 Example 1 of result layout.

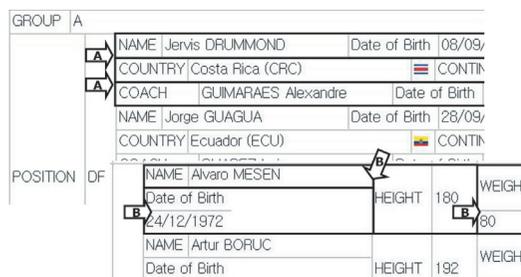


図 22 結果レイアウトの例 2
Fig. 22 Example 2 of result layout.

参照) である*1。

7.2 実験結果・評価

提案手法が一貫性のある構造のレイアウトを生成することを示すため、従来の手法によって生成されるレイアウト結果と比較を行う。

図 21 の結果レイアウトは 6.3.2 項の戦略に従い変換された結果である。上位のネストと

これに含まれる下位ネストの間の連結子が横から縦に変換された結果レイアウトを示す。

6.3.2 項で述べたように、開発者が望んだ意味的なネスト関係の守られたレイアウトが生成されていることが分かる。

図 22 の結果レイアウトは 6.3.3 項の戦略が適用された結果である。上位ノードに変換を適用する(図 22 の A)方が、下位ノード(図 22 の B)に変換を適用するより一貫性のあるレイアウトを生成していることが分かる。

レイアウト変換戦略により生成されるレイアウトの幅の種類を図 23 に示す。図 23 は

*1 データの量は 6.2.1 項で述べたようになる。

3,030 pixel から 300 pixel までの間で提案手法と単純順序探索によって生成可能なレイアウトの幅の全種類とこれの充填率の比較を示す。図 23 から見るように単純順序探索と比べ、提案手法の方が生成されるレイアウトに対してより高い充填率を示している。

たとえば、ユーザ表示画面の幅が 2,700 pixel である場合は、単純順序探索と提案手法ともに、2,660 pixel の幅を持つレイアウトを結果レイアウトとして生成する。しかし、提案手法の方が高い充填率を持つレイアウトを提供している（図 23 の A にあたる）。

単純順序探索によって生成されるレイアウトは、幅が縮められるほど充填率が極端に悪くなっていく。これに比べ、提案手法によって再構成されるレイアウトは全般的に良い充填率を示している。



図 24 幅 2,260 ピクセルの結果レイアウト
Fig. 24 Result layout of 2,260 pixel layout.

図 24 は、単純順序探索と提案手法による結果レイアウトの充填率の差が非常に大きな 2,260 pixel の幅を持つレイアウトの生成結果を比較している（図 23 中の B にあたる）。単純順序探索によるレイアウト変換手法は統一性がない変換を行うため、変換を行えば行うほど図 24 (a) の結果レイアウトのように、それと連結される構造に大きい影響を与え、充填率が低くなる。これに比べて提案手法による変換では、提案手法によって生成されるレイアウトは（図 24 (b)）ほぼ無駄なスペースがない構造の結果レイアウトを生成している。

図 25 は各結果レイアウトを生成するために適用された変換の数を示す。提案手法によるレイアウト変換では結合子の変換数が多い場合（図 25 の B）であっても、統一変換を行うために計算コストは小さい。そのうえ、属性間の関連性に基づいた変換を優先しているため（6.3 節）、単純順序探索に比べて少ない数の結合子の変換（図 25 の A）で制約を満たしたレイアウトの生成が可能になっている。

たとえば、ユーザの表示画面の幅が 2,300 pixel である場合、提案手法では最適の充填率を持つ 2,190 pixel（図 23 の C）の幅のレイアウトと、最適の幅占有率を持つ 2,260 pixel の幅を持つレイアウト（図 23 の A）が候補レイアウトとしてあげられる。6.3.5 項で述べたように、ユーザに提供される結果レイアウトは、この 2 つの候補レイアウトから表 3 のように各目標の結果値の比較と連結子の変換を行った数（図 25）、変換優先順位（6.3 節）の比較によって選択される。この結果、2,260 pixel の幅を持つレイアウトが結果レイアウト

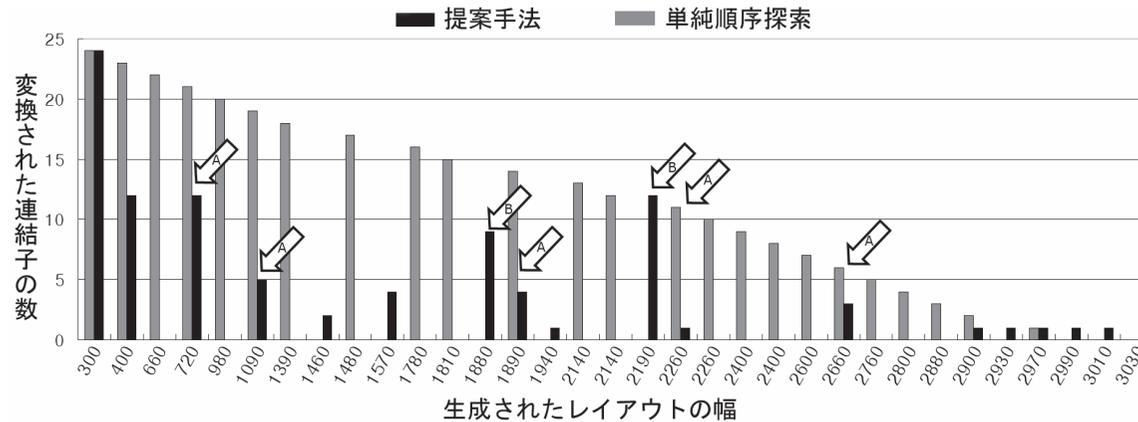


図 25 変換された結合子の数
Fig. 25 Number of converted connectors.

表 3 結果レイアウトの選択
Table 3 Selection of resulting layout.

基準	2,190	2,260
幅占有率	95%	98%
充填率	90%	68%
連結子変換の数	12 → 2	1
戦略優先順位	2	2

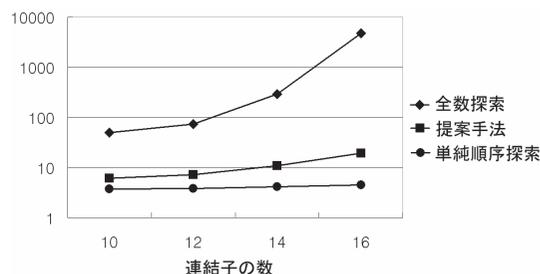


図 26 実行時間
Fig. 26 Execution time.

トとして選ばれる．2,190 pixel 幅のレイアウトを生成するために 12 カ所の連結子が変換されたが，統一変換が適用されたので比較する際の変換数は 2 回（統一変換が行われたのが 1 回，それ以外に 1 回）とする．

提案手法によるコスト削減の結果で処理が非常に速くなり（図 26），Web 上での動的な対応が可能になった．全数探索と比べて本提案手法は，連結子が増えるほど効率的である．

8. まとめ

本稿では，関係データベースと WWW との連係システムである SuperSQL の関係データベースからの結果テーブルの構造化機能を利用して，データベースの情報を WWW ビューとして表示する際に，端末の表示画面サイズに動的な適応を実現する ACTIVIEW を提案した．

ACTIVIEW を利用することで，1 つのビュー定義から多様なユーザ表示画面サイズに適応化されたレイアウトを動的に生成することを可能とした．さらに，提案する ACTIVIEW によってユーザ環境に最適なレイアウトを持つ Web ビューを提供するため，レイアウトの

評価基準を定め，より良いレイアウトへの変換の実現を目指す手法として最適化制約と目標指標を提案した．

ACTIVIEW はまず，ユーザ表示画面幅に収まるレイアウトであることを判定する幅制約，元の問合せ文に開発者によって指定され，基本構造化情報として守るべきである開発者制約を満たした候補レイアウトを生成する．この 2 つの制約は ACTIVIEW が結果レイアウトを生成する際，必ず満たすものである．これらの制約を満たしたレイアウトの候補からどれくらい効率的に幅を利用しているかを判定する幅占有率目標と，ユーザ表示画面に対する空間の効率的な利用度を判定する充填率目標，生成されるレイアウトの長さを調節する長さ目標によってユーザ環境に最適化された一貫性のあるレイアウトを提供することを可能とした．さらに，本稿で提案するレイアウト生成手法と，単純順序探索，全数探索によるレイアウト生成手法とを生成時間，変換される結合子の数，生成されたレイアウトの構造的な一貫性について評価実験によって比較し，本提案手法の有効性を示した．

今後，構造化情報がない問合せ文，すなわち構造を指定しないで通常の SQL で書かれた問合せ文に対して，適切なレイアウト構造を自動的に提供できる手法と ACTIVIEW が提供するレイアウトの認知的な理解度を数値的に評価できる評価関数に関して研究を続けたいと考えている．

参考文献

- 1) Adomaviciu, G. and Tuzhilin, A.: Personalization technologies: A process-oriented perspective, *Comm. ACM*, Vol.48, No.10, pp.83–90 (2005) .
- 2) Abiteboul, S., Cluet, S. and Milo, T.: Active Views for Electronic Commerce, *Proc. VLDB*, pp.138–149 (1999) .
- 3) Ceri, S. and Franternali, P.: Data-Driven, One-To-One Web Site Generation for Data-Intensive Applications, *25th VLDB*, pp.615–626 (1999) .
- 4) 清光英成, 竹内淳記, 田中克己: ActiveWeb: アクティブルールに基づく Web コンテンツの個別化とアクセス管理, 情報処理学会論文誌: データベース, Vol.42, No.SIG8 (TOD10), pp.139–147 (2001).
- 5) 平林真美, 大月一弘, 清光英成, 森下淳也, 北村新三, 絹川達也: リンクに基づいた明示的 Web ページ評価法, 情報処理学会論文誌: データベース, Vol.43, No.SIG12 (TOD16), pp.92–102 (2002).
- 6) 竹内 淳, 清光英成, 田中克己: アクセス履歴の集約とメタルールに基づく Web コンテンツのアクセス管理機構, 情報処学会データベースシステム研究会報告, Vol.2000, No.96, pp.315–322 (2000).
- 7) Shin, S.G., Maeda, Y. and Toyama, M.: ACTIVIEW: Implementation of Adaptive

Web View Using SuperSQL, *The 6th International Conference on Informatics and Systems*, Cairo, IEEE (2008).

- 8) Maeda, Y. and Toyama, M.: ACTIVIEW: Adaptive data presentation using SuperSQL, *27th VLDB*, pp.695-696 (2001).
- 9) 前田葉子, 遠山元道: ACTIVIEW : SuperSQL を利用した適応型表示ビューの実現, 電子情報通信学会データ工学ワークショップ, 5B-7 (2001).
- 10) 前田葉子, 遠山元道: ACTIVIEW における適応型表提示の最適化, 電子情報通信学会データ工学ワークショップ, A2-6 (2002).
- 11) Toyama, M.: SupserSQL: An Extended SQL for Database Publishing and Presentation, *Proc. ACM SIGMOD*, pp.584-586 (1998).
- 12) Shin, S.G., Arisawa, T. and Toyama, M.: The Integration of Media Generators in SuperSQL Query Processor, *3rd International Conference on ELPIT2003*, pp.72-76, KIPS&KMMS (2003).
- 13) Seto, T., Nagafuji, T. and Toyama, M.: Generating HTML Sources with TFE Enhanced SQL, *ACM Symposium on Applied Computing*, pp.96-100 (1997).
- 14) Bickmore, T., Girgensohn, A. and Sullivan, J.: Web Page Filtering and Re-Authoring for Mobile Users, *The Computer Journal*, Vol.42, No.6, pp.534-546 (1999).
- 15) Buyukkokten, O., Kaljuvee, O., Garcia-Molina, H., Paepcke, A. and Winograd, T.: Efficient Web Browsing on Handheld devices Using Page and Form Summarization, *ACM Trans. Inf. Syst.*, Vol.20, No.1, pp.82-115 (2002).
- 16) Buyukkokten, O., Garcia-Molina, H. and Paepcke, A.: Accordion Summarization for End-Game Browsing on PDAs and Cellular Phones, *SIGCHI '01*, Vol.3, No.1, pp.213-220 (2001).
- 17) Bjork, S., Holquist, L-R., Redstrom, J., Bretan, I., Danielsson, R., Karlgren, J. and Franzen, K.: WEST: A Web Browser for Small Terminals, *ACM Symposium on UIST '99*, pp.187-196 (1999) .
- 18) Bruijn, D.O., Spence, R. and Chong, M.Y.: RSVP Browser: Web Browsing on Small Screen Devices, *Personal and Ubiquitous Computing*, Vol.6, Issue 4, pp.245-252 (2002).
- 19) 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).
- 20) 亀岡慎平, 遠山元道: SuperSQL を用いた組版におけるレイアウト自動修正, 情報処

理学会論文誌: データベース, Vol.46, No.SIG13 (TOD27), pp.78-93 (2005).
21) <http://www.fifa.com/worldcup/index.html>

(平成 21 年 3 月 20 日受付)

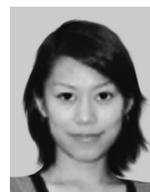
(平成 21 年 7 月 16 日採録)

(担当編集委員 今村 誠)



慎 祥禎 (学生会員)

平成 12 年東義大学工学部コンピュータ工学科卒業。平成 15 年慶應義塾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。現在, 同大学院理工学研究科開放環境科学専攻課程在学中。データベースと構造化文書, データマイニング, XML に関する研究に従事。電子情報通信学会, IEEE Computer Society, ACM 各会員。



前田 葉子

平成 13 年慶應義塾大学理工学部情報工学科卒業。平成 15 年慶應義塾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 UBS 日本入社。IT 債券部に在籍中。システムアナリスト。



遠山 元道 (正会員)

慶應義塾大学理工学部情報工学科准教授。昭和 54 年慶應義塾大学工学部管理工学科卒業。昭和 59 年同大学大学院博士課程修了後, 管理工学科助手, 専任講師を経て現職。博士 (工学)。平成 8 年 Oregon Graduate Institute 客員研究員。平成 10 ~ 13 年科学技術振興事業団さきがけ研究 21 「情報と知」領域研究員。主にデータベースの研究に従事。電子情報通信学会, 日本ソフトウェア科学会, IEEE, ACM 各会員。