

レイアウト式 TFE の拡張

遠山元道 軽部和幸 指田英雄 佐藤 圭

実政宏幸 杉本晋司 瀬戸俊幸 中川裕二

慶應義塾大学理工学部管理工学科

e-mail: toyama@ae.keio.ac.jp

概要

関係データベースの質問言語の構成要素であるターゲットリストの拡張として、以前 TFE (Target Form Expression) を提案した。ターゲットリストと異なり、質問処理の結果を多次元の階層構造をもつ表に整形することができる。その後、その機能をさらに拡張し、また出力媒体も印刷物だけでなくマルチウィンドウや、WWW で提供されるハイパーテキストなども生成できるようにしたので、本稿で概略を紹介する。

The Enhancements on Layout Expression TFE

Motomichi Toyama, Kazuyuki Karube, Hideo Sashida, Kei Sato
Hiroyuki Sanemasa, Shinji Sugimoto, Toshiyuki Seto, Yuji Nakagawa

Department of Administration Engineering

Keio University

3-14-1, Hiyoshi, Yokohama, 223, Japan

We have introduced TFE (Target Form Expression), which is a generalization of target list, several years ago. Unlike an ordinary target list that produces a flat table as a query result, target form expression can specify the arrangement of result into a hierarchically structured table of one or more dimensions. Meanwhile, we have developed the enhancements on TFE's functionalities, as well as the additional output editing modules that produce non-paper outputs such as multi-window display or hypertext that can be viewed through WWW servers. In this paper, we introduce summaries of these enhancements.

1 はじめに

SQL[1]の質問文の基本構成は、SELECT - FROM - WHEREの三つの句からなる。さらにORDER BYにより、結果のタブルの並びを制御することができる。その目的は出力を「見やすく」することにあるが、情報の最終利用者にとって十分なレベルの「見やすさ」にはほど遠いものがある。情報の最終利用形態は千差万別だが、社内の帳票などを得るには、いわゆる4GLのレポートライターなどの外部ソフトの力を借りる。さらに高度な要求として、データブックの類、すなわち、名簿、時刻表、映画案内、統計集などをコンピュータ内のデータから編集することを考えると、汎用のレポートライターの能力を越え、専用の編集ソフトの開発が必要になる。

著者らは、[2, 3]においてSQLのSELECT句等にかかれるターゲットリストの概念を拡張し、タテ・ヨコ・深さ方向への結合とグルーピング(反復)の演算子をもつTFE(Target Form Expression)の概念と、これに基づく処理系の試作結果を発表した。これにより、関係データベース内の正規化された情報から、世の中に見られる各種のデータブックの類に匹敵する出力が得られることを示した。これを基本TFEと呼ぶことにする。

本稿では、その後著者の周辺で行った数々の拡張について概要を報告する。拡張は、新たな演算子の導入など、TFEの言語としての構造編集能力を高めるものと、その処理系の能力や可能性を高めるものに分けられる。後者に代表的なのは、新たな出力媒体への出力編集モジュールの追加である。例えば、出力媒体にHTML(Hyper Text Markup Language)を指定すれば、RDB内の情報から何レベルものメニュー階層をからなるハイパーテキストのソースが一回の検索の実行で得られるため、RDB / WWWの有機的結合という時代のニーズに大きな貢献が期待できる。

2 基本 TFE

TFEは、階層的な表の構造とレイアウトを規定する一種の式であり、オペランドと演算子から構成する。基本TFEにおいては、オペランドはSQL等のホストとなる質問言語におけるターゲットリストの要素(以降基本項目と呼ぶ)がこれにあたる。基本TFEの演算子は、結合子(二項演算)と反復子(単項演算)の2種類に分けられる。それぞれにおいてヨコ(第一次元)、タテ(第二次元)、深さ(第三次元)の三方向(次元)の演算子が定義される。

2.1 結合子

ヨコ、タテ、深さ方向の結合子は、それぞれ以下の記号で表す。結合子の両辺は任意のTFE式で、これらの整形結果を指定方向に連結する。優先度は同一で、結合の順序を明示するには中カッコ{ }をもちいる。

1. ヨコ結合 ,
2. タテ結合 !
3. 深さ結合 %

2.2 反復子

ヨコ、タテ、深さ方向の反復子は、それぞれ以下の記号で表す。大カッコの内側に任意のTFE式を置く。カッコ内で整形結果が複数生成されるとき、これらを指定方向に逐次連結する。

1. ヨコ反復 [],
2. タテ反復 []!
3. 深さ反復 []%

2.3 式の構成と出力生成規則

質問文において、従来のターゲットリストの代わりに以下で定義する目的式を記述する。

<目的式> ::= <反復式> | <反復式><結合子><目的式> | {<目的式>}
<反復式> ::= [<TFE式>]<結合子>

<TFE 式> ::= <基本項目> | {<TFE 式>} | <TFE 式><結合子><TFE 式>
 <結合子> ::= , | ! | %

目的式をもつ質問文を実行する際、結果として得られる構造化(レイアウト)された表は、以下の概略手順によって構成する。なお、目的式(および TFE 式)の次数を、それに含まれる基本項目の数と定義する。

[整形出力生成規則]

入力：目的式 出力：整形出力

1. 目的式 中の基本項目を、出現順にコンマで連結してターゲットリストを作り、ホストとなる質問言語の文法に基づいた質問文を構成する。その実行結果を 対象関係 とする。この対象関係と与えられた目的式に、規則 2 または規則 3 を適用し、結果を得る。
2. 目的式 1 が単独の反復式 (TFE 式 1 結合子 1) の場合。対象関係を、TFE 式 1 によって、規則 2、3、4 のいずれかで処理して得られる複数の出力を、結合子 1 で示される方向に逐次結合したものが結果である。
3. 目的式 1 が 反復式 1 結合子 1 目的式 2 の形の場合。反復式 1 の次数を k とする。対象関係 を、最初の k 個の属性と残りの属性にそれぞれ射影して得られる 対象関係 R1 と R2 について、R1 を 反復式 1 で、規則 2 によって整形した結果と、R2 を 目的式 2 で、規則 2、3、4 のいずれかで整形した結果を、結合子 1 で結合したものが結果である。
4. 規則 2、3 が当てはまらない場合。対象関係 R を反復式でない TFE 式 1 で整形する。このとき、一般に結果は複数の整形結果の集合となる。TFE 式 1 中の最上位レベルにある 基本項目 (複数可) に対応する属性の実現値の組合せを基準として R のタブルをグルーピングし、残りの属性からなる部分(商)関係の集合を、TFE 式 1 中の 反復式 のリストによって規則 2 または 3 によって整形した結果を、グルーピング基準項目の値とともに結合する。このような整形結果は、グルーピング基準項目の実現値の組合せ数だけ得られるので、その集合を結果とする。

3 拡張 TFE

過去 2 年間に行なった拡張は、言語仕様に関するものと、TFE の処理系に関するものに大別できる。この節では、前者に当たるものとして、GENERATE 句、装飾演算子(@)、関数(集約、グラフ生成、メディア変換)、複合反復子等の導入を概説する。

3.1 GENERATE 句

TFE はもともとターゲットリストの一般化であり、SQL[1] や QUEL[4] 等の SELECT 句、retrieve 句に置くものとして提案された。データベースからの出力を高品質の印刷物にする「データベース出版」がその主目的で、レイアウト編集の結果として L^AT_EX[5] のソースファイルを出力する。

しかしながら、次節で説明するように、TFE による構造化出力は多くの目的に対し、多種多様な出力を可能とすることが明らかになった。そこで、出力媒体(言語)を明示するために GENERATE 句を構文に加えることにした。GENERATE の直後に目的式を書いてもよいが、この場合出力媒体に latex が指定されたものとする。

構文： GENERATE <媒体指定> <目的式>
 FROM from 句 (以下の例では省略)
 WHERE where 句 (以下の例では省略)
 <媒体指定> ::= <空> | plain | latex | tcltk | HTML

参考： SELECT a, b, ... ≙ GENERATE plain [a, b, ...]!

3.2 装飾演算子 (@)

データベースから抽出された情報に、文字サイズ、文字スタイル、欄幅などの属性を与えて出力を生成するために、これらを指定する装飾演算子 (@) を導入する。

```
構文: <TFE式> @<装飾指定>
<装飾指定> ::= <装飾子> | {<装飾子のリスト>}
<装飾子> ::= <サイズ指定子> | <スタイル指定子> | <幅指定子>
<サイズ指定子> ::= tiny | small | normalsize | large | Large | LARGE | Huge
<スタイル指定子> ::= sl | it | tt | bf | sf
<幅指定子> ::= <整数>mm | <整数>cm
```

```
例: GENERATE latex [売場 @{Huge,bf} ! [氏名 @bf, 電話] !]!@large
```

幅指定のある欄で、実現値の幅が指定を越える場合、個別に文字サイズを下げて欄に納めるオプションを備えている。また、複数の欄を横方向に接合する式の全体にまとめて幅指定が行なわれた場合、幅指定のない欄には、実現値の最大幅に比例配分で幅が与えられる。

```
例: GENERATE [氏名, 電話, 住所 @105mm]!@220mm
```

上記の例で、氏名、電話の実現値の最大幅の比が3:2ならば、115(=220-105)mmのうち69mmが氏名、46mmが電話に配分される。

3.3 関数の導入

出力の表現力を高め、また様々なデータ型を取り扱うために、各種の関数を導入した。関数には集約関数、グラフ生成関数、メディア変換関数などがある。

3.3.1 集約関数

これは一般の質問言語において提供される集約関数と同様なものだが、TFEのグルーピング構文を利用し、階層的な集計分類基準についても直感的な記述を可能とする。複数の店舗にそれぞれ複数の売場のあるデパートのチェーンで、店舗毎の従業員の平均給与と売り上げ合計をもとめ、その下に明細として売場毎の平均給与と売上合計のリストを示すには以下のような質問を与える。店舗毎に改ページを行っている。

```
例: GENERATE [{店舗, avg(給与), sum(売上高)} ! [売場, avg(給与), sum(売上高)] !] %
```

3.3.2 グラフ生成関数

TFEによって生成される階層的な表の入れ子部分を、この入れ子部分から作られる折れ線グラフや棒グラフに置き換える関数である。店舗、売場別の月毎の売上を入れ子表にするには、次のように書く。

```
例: GENERATE [店舗, [売場, [月, sum(売上高)]] !] !] %
```

この月と売上計の入れ子表を折れ線グラフにするには、グラフ生成関数 `graph` (種別, X軸項目, Y軸項目, 系列項目リスト) を用いて、以下のように質問文を構成する。上記の質問と対比してほしい。 `graph` 関数の4番目の引数の系列項目リストは任意指定で、グラフに複数の系列を含める場合にこれを指定する。

```
例: GENERATE [店舗, [売場, graph("line", 月, sum(売上高))] !] !] %
```

このグラフを、売場毎にまとめ、各々のグラフに店舗ごとのデータ系列を複数の折れ線で描くには、質問文を以下のように変更すればよい。

```
例: GENERATE [売場, graph("line", 月, sum(売上高), 店舗) !]
```

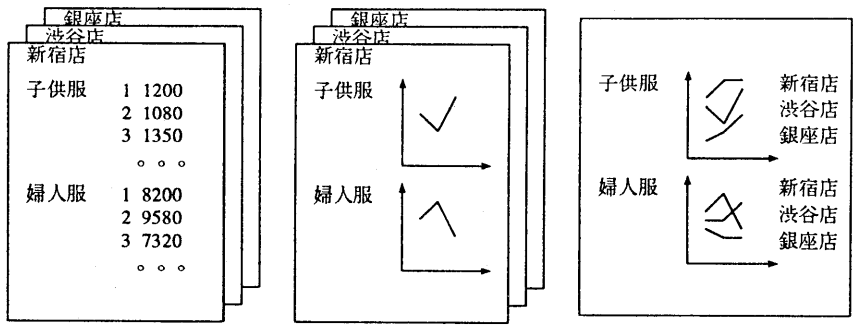


図 1: 入れ子表とグラフ生成

3.3.3 メディア変換関数

データベースから取り出されたスカラー値を他の型のデータに変換する関数。現在、1) イメージデータを含むファイルの名称を、これから取り込むイメージに変換する関数 `imagefile`(種別, ファイル名, パス, 幅, 高さ) と、2) 項目を出力から除去する関数 `omit`(<TFE 式>) がある。

`imagefile` 関数の第 1 引数は、"epsf", "gif", "jpg" などで、イメージデータの種別を示す。第 3 引数のパスは任意指定で、この文字列に第 2 引数を連結した文字列がパスを含むファイル指定となる。取込み後の表示サイズを指定する第 3、第 4 引数は任意。

例: `GENERATE [市!人口!市長], imagefile("gif", 市街図, "/home/city/images/")!`

`omit` 関数は、クロス集計表を作る際に有効。売場を縦、月を横方向に設定して集計するには、`omit` 関数なしでは以下のようにする。

例: `GENERATE [売場, [月, sum(売上)],]!`

これでは売り上げの数値の左方にそれぞれ月が表示されてしまう。これを以下のように変更すればすっきりとする。

例: `GENERATE [売場, [omit(月), sum(売上)],]!`

3.3.4 複合反復子

反復子は、インスタンス数に応じ、指定された方向に無制限にグループ項目を反復レイアウトする。しかし現実には紙面や画面のサイズの制約から、これでは困ることが多い。複合反復子は、ある方向(次元)での反復の上限を指定し、これを越える場合に次に指定した方向(次元)に伸長することを指示するものである。

構文: `[<TFE 式><結合子><回数><結合子><回数> ... <結合子>`

商品のカタログで、商品名、型番、価格、写真をまとめた枠を 3 行 2 列に配置し、さらに多ければページ替えをするように指定する。

例: `GENERATE [商品名! 型番! 価格! imagefile("gif", 商品写真)]!3,2%`

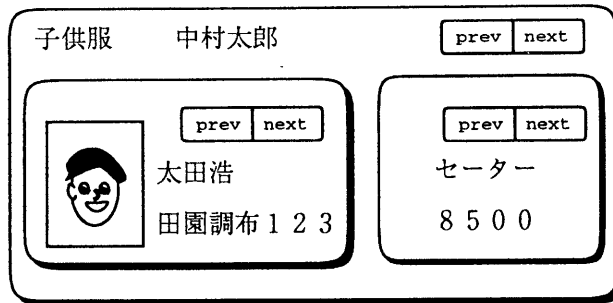


図 2: 深さ反復によるサブウィンドウの生成

4 処理系の拡張

前節では TFE の言語仕様の拡張について述べたが、処理系についても様々な拡張を試み、その可能性を拡大した。この節では、出力編集モジュールの追加により、 \LaTeX 以外の言語（媒体）の生成を行なう例、TFE に特有な状況に対処する最適化手法、そして WYSIWYG フロントエンドの作成について概要を述べる。

4.1 Tcl/Tk 生成によるマルチウィンドウ表示

Tcl/Tk は J. Ousterhout[6] によって開発されたスクリプト言語とウィンドウシステム用のツールキットである。これらを用いると、ウィンドウシステムを活用したアプリケーションや、アプリケーションのフロントエンドを容易に開発できる。

TFE の出力先としてこの Tcl/Tk のプログラム生成を選ぶにより、紙媒体の制約を逃れ、自由に深さ（第三次元）方向に仮想的な紙の束を、サブウィンドウの形で生成することができる。この場合、データベース出版というカテゴリからは離れ、いわゆるデータベース概視機能の実現ということになる。紙媒体では、第三次元の反復や結合は、最外レベルにおいてのみページ替えという形で実現できる。

ウィンドウシステム上では、仮想的な紙の束に `Prev`、`Next` ボタンを配し、これによって紙めくりをシミュレートすることにより、直感的かつ操作性の良いデータ提供が実現できる。

例：`GENERATE tcltk [{売場, 責任者名} ! {imagefile("gif", 写真), 社員名 ! 住所}%, [商品名 ! 価格]! %]%`

4.2 HTML 生成による RDB → WWW 変換

インターネットブームが巻き起こり、WWW (World Wide Web) サーバによる情報の公開が、ネットワーク透過的なハイパーテキストの形で盛んに行なわれている。それには HTML (Hyper Text Markup Language)[7] という言語によって内容情報を記述し、サーバに登録する必要がある。この際、情報の分量が多いと、HTML ファイルの作成・更新に大きな問題が生じる。一方、関係データベース (RDB) には多くの有用な情報が記録され、情報システムの運用に供されているが、インターネットからネットワーク透過的にその情報をアクセスする一般的なシステムは確立されておらず、両者を有機的に結べば大きなメリットが得られることは明らかである。

そこで、HTML を対象言語とした TFE の出力生成系を作成した。 \LaTeX や Tcl/Tk と異なり、HTML には階層的な入れ子構造を作成する機能がないため、タテヨコの二次元に関しては、複雑なレイアウトは期待できない。しかし、第三次元の深さ方向の結合 (%) を、ハイパーテキストにおけるリンクと解釈することにより、任意の深さの階層 (リンク) 構造をもつハイパーテキストを即時に生成することができる。深さ方向結合子 (%) の左辺をボタン、右辺をリンク先ページとするハイパーテキストを生成する。

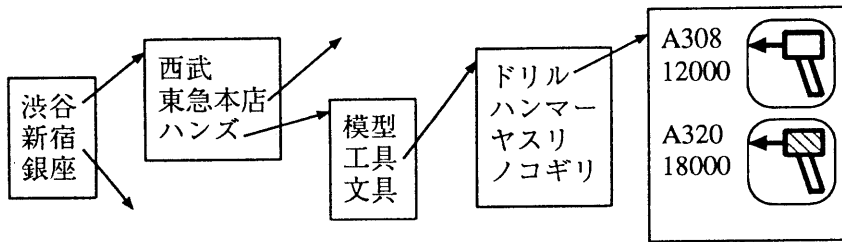


図 3: リンク階層をもつハイパーテキストの生成

例: GENERATE HTML [地域 % [店舗 % [売場 % [商品名 %
[型番 ! 価格 ! imagefile("gif", 商品写真)]!]!]!]!]!

4.3 直積生成の抑制による最適化

TFE を用いる環境では、従来の質問言語では考えられないような質問パターンが頻繁に使用される。特に、直積またはグルーピングされた部分関係どうしの直積が多用される傾向がある。セマンティクスの定義に忠実な安易な実装では、たちまち組み合わせ爆発が起こってしまう。

例えば、関係 R、S を 1 ページ目の左右に配置し、2 ページ目に関係 T を出力することは、TFE を用いれば以下のように容易に指定することができる (* の略記法は未定義未実装)。

例: GENERATE {[R.*], [S.*]} % [T.*]
FROM R, S, T

TFE のセマンティクスの定義通りに処理すれば、中間結果として R、S、T の直積をバックエンドの RDBMS に要求し、これが生成されたうえで、それぞれの射影を作って出力を編集する。いうまでもなく、最終結果がコンパクトであっても、中間結果は膨大なものになってしまう。

このような事態を避けるために、バックエンドへの質問を複数のサブ質問に分解し、得られた複数の結果を合成して用いることを基本とした最適化手法を開発し、実装した。これによって数メガバイトだった中間結果が数千バイトですむようになったケースもある。質問を複数に分けるために、当然オーバーヘッドも生じるので、どのレベルまでこれを適用するかなど、損益判断の難しい面もある。

4.4 WYSIWYG フロントエンドの作成

TFE は、プログラミングの経験のない人々には難解な部分も多く、だれでも使いこなせるものとは言い難い。また、装飾演算子 (@) によるフォントやサイズの指定は直感的ではなく、最終的に満足な出力を得るまでには試行錯誤を要する。そこで、GUI を用いた直接操作によって、TFE 式を自動的に生成するシステムを作成した。実験の結果、関係データモデルに関する最小限の知識をもつだけの利用者でも、出力見本と画面上のメタファの対比を行いながら画面上の操作を行うことによって、かなり複雑な構造をもつ出力を短時間の内に手にすることができた。

5 おわりに

関係データベースの質問処理結果の印字/表示能力の向上のため、ターゲットリストの拡張である TFE(Target Form Expression) をさらに強化した。この拡張によって、利用者は三次元の階層化された表を生成するばかりでなく、フォントや文字サイズの指定、欄幅の指定、表内のグラフ生成/埋め込み、外部の画像ファイルの取り込み

などを容易に行うことができる。この結果、編集能力は市販の汎用レポートライタ等の機能を大きく上回るものとなった。

処理系の実装においては、新たな出力編集モジュールとして Tcl/Tk 言語、HTML 言語のプログラムを生成するものを作成した。これにより出力を紙媒体の制約から解放し、ビットマップディスプレイ上でマルチウィンドウ画面やハイパーテキストの形でデータを閲覧できるようになった。第三次元の接続や反復がウィンドウの切替えや、ハイパーリンクのナビゲーションに変換され、操作性を高めている。

さらに、TFE を含む質問に特徴的な質問パターンに着目した新たな最適化アルゴリズムを提案し、実装した。質問が中間的に直積を生成するケースで大きな効果を発揮する。

GUI を用いて対話的に TFE によるレイアウト指定式を生成するフロントエンドを作成した。これを用いると、TFE の文法や詳しい機能を知らなくても、望む整形結果を手にすることができる。

ここで紹介した各種の拡張は個々に独立に実装したため、現在統合化の作業を進めている。今後の予定として、

- 第四次元以上の利用 (時間軸 ?)
- データの入力・更新への対応
- データベース設計への応用
- オブジェクト指向データベースへのインスタンス追加プログラムの自動生成

などを検討中である。

参考文献

- [1] Melton, J., Simon, A. R., *Understanding the New SQL: a Complete Guide*, Morgan Kaufmann, (1993)
- [2] Toyama, M., Three Dimensional Generalization of Target List for Simple Database Publishing and Browsing, in Proc. 3rd Australian Database Conference, *Research and Practical Issues in Databases*, World Scientific Pub. Co. (1992), pp. 139-153.
- [3] 遠山 元道, ターゲットリストの拡張によるデータベース出版と概視の実現, 情報処理学会データベースシステム研究会資料, 93-DBS-94 (1993), pp. 243-252.
- [4] Stonebraker, M., Wong, E., Kreps, P., and Held, G., The Design and Implementation of INGRES, *ACM TODS* 1, 3, (1976) pp. 189-222.
- [5] Lamport, L., *L^AT_EX: A Document Preparation System*, Addison-Wesley (1986)
- [6] Ousterhout, J.K., *An Introduction to Tcl and Tk*
- [7] Morris, M.E.S., *HTML for Fun and Profit*, SunSoft Press (1995)