

## 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

遠山 元道

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部分であるターゲットリストの拡張として TFE(target form expression) を導入する。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。第3次元を厚み方向次元とすると、これによって VDT 上にウィンドウの階層構造を作り、概視における視認性と操作性を向上させることができる。INGRES 上での実現について報告している。

## Implementation of Database Publishing and Browsing by Generalizing Target List Syntax and Semantics

Motomichi Toyama

Department of Administration Engineering

Keio University

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

This paper introduces a generalization of target list, the *target form expression*. Unlike an ordinary target list that produces a flat table as a query result, target form expression can specify the layout of result as a hierarchically structured table of one or more dimensions. The third depth dimension is useful for showing results on VDT displays to improve readability and operability. An implementation upon INGRES database management system is reported.

## 1 はじめに

*Target Form Expression* (TFE) はターゲットリストの一般化である。SQL[1] をはじめとする関係型データベースの問い合わせ言語において、ターゲットリストは関係代数の射影操作と、結果テーブル内での各属性の順序とを規定する。ターゲットリストが文字通りリストである限り、結果テーブルも単純な値のリストとなり、問い合わせの結果は当然にフラットな表となる。TFE は問い合わせ言語の結果表示に関する表現能力を広げるもので、問い合わせの結果として次元を含む多次元の階層的構造表を提供する。

データベース管理者の観点からすると、正規化されたフラットな関係には多くの利点がある。たとえば意味制約、整合性の強制、更新セマンティクス等の単純化、問い合わせ言語の構文・意味の単純化などである。しかしながら、データベースシステムのユーザにとって、始めから最後までをフラットな表で扱うことが常によいとは限らない。特に、データベースからの出力から帳票を作成するために各種の4GL言語が開発され、広く用いられていることは、情報の最終的な表現形態としてフラットな表が不十分なことを如実にあらわしている。

結合を含む問い合わせの結果の関係には部分従属や推移従属が発生しやすく、一部の属性に共通の値を持つタプルが多く生じる。図1(b)は、(a)のサンプルデータベース中のdept関係とemployee関係の結合の結果である。タプルはソートされて結果が分かりやすくなっている。しかし、冗長な値がディスプレイのスクリーン域を大きく占有している。連続する同一の値の表示を抑止すれば、より直観的にはなるが、それでもスクリーン域の浪費は変わらない(図1(c))。もし概略の情報(deptの情報)と詳細の情報(employeeの情報)を水平ではなく、垂直にレイアウトすれば、表示が直観的になるばかりでなく、スクリーン域をより効率的に利用できる。

2次元のTFEはターゲットリストの直接的な拡張で、多様なレイアウトを容易に実現する。図2は、TFEでフォーマットした表の例である。TFEは2次元に限らず、線形のレイアウトや3次元の階層表をも扱う。第3次元は深さ(厚み)を表す。これをX-windowやMS-Windowsのようなマルチウィンドウシステムの子ウィンドウに対応させることにより、VDTディスプレイの上の結果表示の可読性と操作性の両方を大きく改善することができる。

データベース情報を階層テーブルとして取り扱う試みは、*Non-First Normal Form* (NF<sup>2</sup>)あるいは、*Nested Relation* データモデルの研究においてなされている。しかし、値の表示に関心が向けられたことはあまりなく、主にデータモデルや問い合わせ言語などが研究されている[2]。NF<sup>2</sup>モデルのためのそれぞれの問い合わせ言語は通常nest/un-nestの操作によって階層を生成したり、逆に平坦化したりする。TFEは、nest演算子の機能に、空間内での方向性を加味したものと見ることがもできる。ただし、TFEではデータベース自体は基本的な関係データベース、すなわちフラットな関係の集まりであることを前提とし、出力結果を階層化することを目的としているので、前提・目的の根本的な点で両者は異なっている。

ORACLEの拡張SQLのような製品は、繰り返しの値を除去することを許すフォーマット指定などの機能を備えている。これによって、図1(c)のような結果表示が可能である。4GLシステムとして知られている商品の多くは洗練されたスクリーン定義ツールとレポート定義ツールを備えている。しかし、ユーザは実際の検索がなされるより前に、独立のツールを利用してこれらの定義をしなければならない。これらは繰り返し利用するアプリケーションの開発には適しているが、質問ごとに手軽に利用するには不向きである。TFEは手軽なばかりでなく、構造の表現力においていわゆる4GLよりはるかに強力である。

次節以降に述べるように、TFEは「式」であり、連結演算子(連結子)によって要素同士の3次元の各方向への連結を表現し、全体として表示を構成する。この概念によれば、一般のターゲットリストにおいて、項目を分離しているコンマは、水平方向にオペランドを連結する演算子と考えることが

```

employee(number,name,salary,manager,tel,address,face)
dept(number,name,store,floor,manager)
store(number,city,state)

```

(a) Sample Database Schema

department	store	manager	employee	tel
Book	San Francisco	James	Ferro	321-4567
Book	San Francisco	James	Choy	312-0987
Candy	San Francisco	Edward	Onstad	312-5555
Children's	El Cerrito	Smythe	Bailey	313-2353
Children's	El Cerrito	Smythe	Evans	555-6666
Children's	San Francisco	Ross	Collins	322-2356

(b) Repetitive Values in Joined and Ordered Relation

department	store	manager	employee	tel
Book	San Francisco	James	Ferro	321-4567
			Choy	312-0987
Candy	San Francisco	Edward	Onstad	312-5555
Children's	El Cerrito	Smythe	Bailey	313-2353
			Evans	555-6666
	San Francisco	Ross	Collins	322-2356

(c) Duplicates Eliminated

図 1: 一般的な検索結果出力

できる。各次元ごとに連結子を用意して多次元の表示オブジェクトを構成することができる。

SNOBOL4B[3] は、古典的な文字列操作言語 SNOBOL4 に縦と深さ方向の連結を導入することによって 3 次元の文字オブジェクトを扱えるように拡張したものである。本研究の着想は、この SNOBOL4B のアイデアを、時を隔ててデータベースの出力整形に応用することに結び付けて得られたものである。

なお本報告は、著者による文献 [4] に基づくものである。

## 2 2次元 TFE

この節ではまず 2 次元の TFE を導入する。次元の制限を除き、本質的な構文要素はすべてここに現れる。

### 2.1 連結子

TFE は、普通の数式と同様に、演算子とオペランドから構成する。基本的なオペランドは、基本項目と呼ばれ、通常の問い合わせ言語のターゲットリストに現れる項目と同様である。属性名、数式、文字定数、集約関数の呼び出しなどが、これに含まれる。TFE の実験システムは INGRES の上に実装しているため、例題はすべて、QUEL 言語に基づいている。しかし、ターゲットリストは、異なる言語間でもほとんど差異がないため、ここでの議論は、SQL など、関係データベースの他の

department	employee	
store	tel	
manager	address	

Book	Ferro, Tony	Choy, Wanda
San Francisco	321-4567	321-0987
James, Mary	888 North St.	4567 South St.

Candy	Onstad, Richard	
San Francisco	312-5555	
Edwards, Peter	555 West St.	

Children's	Bailey, Chas M.	Evans, Michael
El Cerrito	313-2353	555-6666
Smythe, Carol	8 North St.	6 West St.

Children's	Collins, Joanne	
San Francisco	322-2356	
Ross, Stanley	12 East St.	

Furniture	Raveen, Lemont	Wallace, Maggie J.	Schmidt, Herman
Oakland	395-5898	495-5821	485-4365
Thompson, Bob	99 East St.	123 West St.	90 East St.

図 2: TFE による出力例

問い合わせ言語にも適用可能である。

連結子は、オペランドを「接着」するための演算子のクラスである。得られるオブジェクトをボックスと呼ぶ。連結子は基本項目とボックスとの任意の組合せをそのオペランドとしてとることがある。

最初の連結子は、コンマによって表現される水平連結子である。これによって、オペランドは水平方向に（左から右へ）連結される。したがって、ある TFE において、すべてのオペランドが単純な基本項目であり、さらに式中のすべての演算子がコンマであれば、式全体は通常のターゲットリストと同様になり、同様に解釈される。

EXAMPLE 1 水平連結子

```
range of e is employee
retrieve (e.name, e.tel)
```

```
Jones 111-2222
Smith 333-4444
Evans 555-6666
.....
```

EXAMPLE 2 垂直連結子

```
range of e is employee
retrieve (e.name ! e.tel)
```

```
Jones
111-2222
Smith
333-4444
Evans
555-6666
.....
```

垂直連結子は、感嘆符 (!) によって表わされる。これは、オペランドを垂直方向に連結するものであ

る。次の例は、垂直連結子の単純な応用で、結果は縦方向の線形な配置である。

水平連結と垂直連結の優先順序は同等で、左方優先連結である。連結順序を明示的に変更するには、部分式を中括弧でくくる。

### EXAMPLE 3 結合順序の変更

```
range of e is employee
retrieve (e.name, {e.tel ! e.address})
```

```
Jones  111-2222
        1234 North St.
Smith  333-4444
        55 East St.
Evans  555-6666
        6 West St.
.....
```

## 2.2 反復群

反復群は、ボックスを導入する TFE のもうひとつの構文要素である。構文は、左大括弧に続いて、TFE、右大括弧、後続連結子である。生成されるボックスには、括弧内の TFE により特定されるサブタブルを、その TFE によって1つずつ整形し、それらを連結した結果がはいる。後続連結子は、整形されたサブタブル同士を互いに連結する方向を指定する。以下の2例は、反復群の基本的な用法を表している。

### EXAMPLE 4 水平反復

```
range of e is employee
retrieve ([e.name ! e.salary],)
```

```
Jones  Smith  Evans  Thomas  .....
12000  6000   5000   10000  .....
```

### EXAMPLE 5 垂直反復

```
range of e is employee
retrieve ([e.name,e.tel]!)
```

```
Jones  111-2222
Smith  333-4444
Evans  555-6666
.....
```

この例から推測されるように、([e.name,e.tel]!)のように垂直反復群を最も外側にもつ TFE は、従来のターゲットリスト(e.name,e.tel)と同様の意味を持つことになる。そこで、従来のターゲットリストをそのまま TFE として解釈することができるように、最外殻に反復群を持たない TFE には垂直反復群を補うことにする。

反復群が一つ以上の基本項目に連結されている場合には、これらの基本項目は反復群の群化規準項目となる。次の例では、反復群のサブタブル(name, tel)は、部門名の値によって群化している。

### EXAMPLE 6 反復群

```
range of d is dept
range of e is employee
retrieve (d.name,[e.name,e.tel]!)
where d.manager=e.manager
```

```
Book      Ferro  321-4567
          Choy   321-0987
Candy     Onstad 312-5555
Children's Collins 322-2356
          Bailey 313-2353
          Evans 555-6666
```

.....

反復群式は任意の深さの入れ子を構成できる。以下の問い合わせの結果は、全体を部所ごとに群化し、さらにその中で各従業員を店舗によって群化している。

### EXAMPLE 7 反復群の入れ子

```
range of d is dept
range of e is employee
range of m is employee
range of s is store
retrieve (d.name,[s.name,m.name,[e.name,e.tel]!]!)
where d.manager=e.manager and d.manager=m.number
      and d.store=s.number
```

```
Book      San Francisco  James  Ferro  321-4567
          Choy     312-0987
Candy     San Francisco  Edward  Onstad 312-5555
Children's El Cerrito  Smythe  Bailey 313-2353
          Evans 555-6666
          San Francisco  Ross   Collins 322-2356
```

.....

## 3 第3次元

深度連結子は、第3次元の連結を指示するもので、パーセンテージの記号(%)で表す。

SNOBOL4Bにおいては第3次元の連結は *normal concatenation* [3] と呼ばれていて、ラインプリンタによって二重印字による強調、下線、フォームオーバーレイなどを実行する目的で導入されている。TFEにおける深度連結は、対象となる出力装置によってその意味づけが異なる。

深度連結を2項連結子として使用することはまずない。これは、反復群の後続連結子として、厚み方向の反復を指定するのに有効である。深度連鎖子をトップレベルの反復群に利用すると、印刷装置における改ページを指示するものと解釈される。VDTディスプレイにおいては、ウィンドウの「次画面」と解釈することができる。次の問い合わせは、従業員の名簿の出力において、部署の変わり目

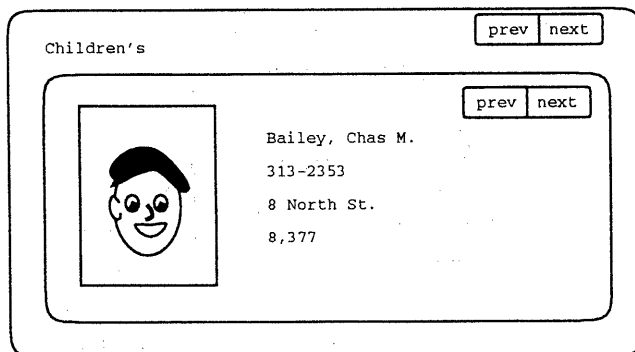


図 3: 深度反復によるサブウィンドウの生成

で改ページをおこなう。

EXAMPLE 8 ページ換え出力

```
range of d is dept
range of e is employee
retrieve ([d.name ! [e.name, e.tel, e.address]!])%
where d.manager=e.manager
```

トップレベル以外で TFE に現れる深度連結子による反復群は、画面上のサブウィンドウと解釈される。図 3 に、検索結果の employee レコードの部分サブウィンドウ化する例を示す。face は顔写真のイメージ属性である。prev と next というラベルのついたボタンの組が、サブウィンドウごとと右上方にある。ユーザーは、これらのボタンでページの前後を操作することができる。

内側のウィンドウにある next ボタンは、表示中の部所に所属する次の従業員の情報を呼び出す。外側のウィンドウにあるボタンによって、部所を順に (前後に) 呼び出すことができる。このような画面レイアウトも、次のように簡単な指示で作成できる。

EXAMPLE 9 サブウィンドウの生成

```
range of d is dept
range of e is employee
retrieve ([d.name![e.face,{e.name!e.tel!e.address!e.salary}]])%
where d.manager=e.manager
```

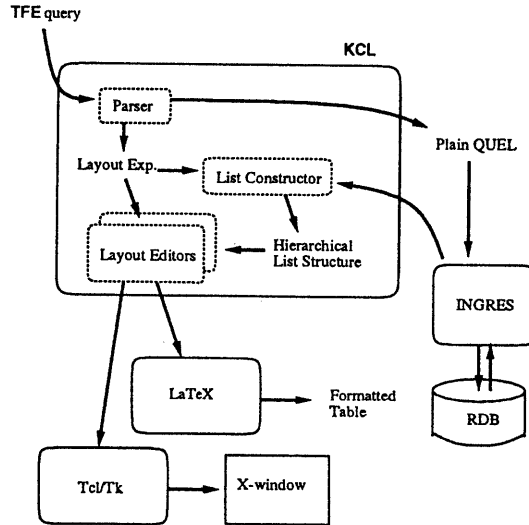


図 4: システム構成

## 4 実現

### 4.1 システム構成

図 4に見られるように、TFE プロセッサの実現は Sun の Sparc Station の上で利用可能な、INGRES[5], kyoto Common Lisp(kcl)[6],  $\text{\LaTeX}$ [7] の3つのフリーソフトに依存している。TFE プロセッサの対話バージョンは、 $\text{\LaTeX}$ の代わりに Tcl/Tk [8] と X-Window System を利用する。INGRES にとって TFE プロセッサは完全に独立なフロントエンドなので、INGRES に対しては何も修正を加えていない。

TFE プロセッサは Common Lisp [9] で記述している。これは三つ主要なモジュール：パーザ、リスト構造生成部、レイアウト編集部 からなる。パーザは、TFE 拡張 QUEL で書かれた質問を受け、これを純 QUEL 質問とレイアウト式に分ける。レイアウト式は、出力される表の構造を表すリストである。図 2 を生成した質問のレイアウト式は次の通りである。

```
(g2 (c1 (c2 1 2 3) (g1 (c2 4 5 6))))
```

TFE 中の属性名などは、レイアウト式中では整数に置き換えられている。g は反復群 (grouping)、c は接続子 (connector) を表し、付随する数は次元を指定する。

リスト構造生成部は、バックエンドの関係データベース (INGRES) の出力したフラットな表を受けとり、レイアウト式の構造を反映する値のリスト構造に変換する。反復群のサブリスト化が主な仕事である。レイアウト式の上位レベルの属性によってタブルをソートし、同一値を括り出してできる部分関係を、レイアウト式の部分でさらに構造化する、という再帰的な処理を行なう。

リスト構造生成部の出力は、レイアウト式とともにいずれかのレイアウト編集部に渡される。印字媒体への出力の場合、これらは印字に適したデータに変換される。試作システムでは、ターゲットを  $\text{\LaTeX}$  ソースとしている。図 5 が出力された  $\text{\LaTeX}$  ソースの例で、これを  $\text{\LaTeX}$  で処理することによって、図 2 のような印字結果が得られる。



```

\documentstyle{article}
\begin{document}
\noindent
\begin{tabular}[t]{|l|}
\hline
\begin{tabular}[t]{|l|}
\hline
\verb|department      | \\\hline
\verb|store           | \\\hline
\verb|manager         |
\\\hline
\end{tabular} &
\begin{tabular}[t]{|l|}
\hline
\verb|Ferro, Tony     | \\\hline
\verb|321-4567| \\\hline
\verb|888 North St.  |
\\\hline
\end{tabular} &
\begin{tabular}[t]{|l|}
\hline
\verb|Choy, Wanda     | \\\hline
\verb|321-0987| \\\hline
\verb|4567 South St. |
\\\hline
\end{tabular}
\\\hline
\end{tabular} &
\begin{tabular}[t]{|l|}
\hline
\verb|employee       | \\\hline
\verb|tel             | \\\hline
\verb|address         |
\\\hline
\end{tabular}
\\\hline
\end{tabular}
\\\hline
\end{tabular}

\noindent
\begin{tabular}[t]{|l|}
\hline
\begin{tabular}[t]{|l|}
\hline
\verb|Book           | \\\hline
\verb|San Francisco  | \\\hline
\verb|James, Mary    |
\\\hline
\end{tabular} &
\begin{tabular}[t]{|l|}
\hline
\verb|Candy           | \\\hline
\verb|San Francisco  | \\\hline
\verb|Edwards, Peter |
\\\hline
\end{tabular} &
\begin{tabular}[t]{|l|}
\hline
\end{tabular}
\\\hline
\end{tabular} &
\begin{tabular}[t]{|l|}
\hline
\end{tabular}
\\\hline
\end{tabular}

```

図 5: 生成された L<sup>A</sup>T<sub>E</sub>X ソースコード

VDT 画面への出力は、レイアウト式と値のリスト構造を参照しながら動的に描画する方法と、これらの値を織り込んだプログラムを生成し、それを実行する方法とが考えられる。我々は、現在 Tcl/Tk 言語を利用して、後者のアプローチをとる実験を行なっている [8]。

## 4.2 質問処理

現在の処理系は、関係データベースをそのまま用いているが、TFE 特有のセマンティクスを利用して効率を上げるように質問処理機構に手を加えることが可能である。

例えば、結合操作の結果を群化することがあらかじめ分かっていたら、重複値のバックエンドからの転送を抑制することができる。

## 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、従来のターゲットリストを拡張し、表現力を飛躍的に高めた TFE(target form expression) を提案した。この拡張によって、利用者は 3次元の階層化された表を容易に手に入れることができる。このような表は平坦な表に比べ、はるかにコンパクトで、内容を直観的に理解しやすい。3次元構造は VDT 表示ではサブウィンドウの

作成を表し、やはり直観に訴える表示を容易に得ることができる。

試作システムは、INGRES の出力を  $\text{\LaTeX}$  のソースに変換することができる。さらに、X-window システム上でのマルチウインドウ表示系を作成中である。

## 謝辞

協力していただいた 陳 文潔、楊 青、李 林、伊波 亨の各氏に感謝します。

## 参考文献

- [1] ISO, Database Language SQL, Document ISO/TC97/SC21/WG3 N117, (1987)
- [2] Special Issue on Nested Relations, *Data Engineering 11*, 3, IEEE Computer Society, (1988)
- [3] Gimpel, J.F., Blocks — A New Datatype for SNOBOL4, *CACM 15*, 6, (1972) pp. 438-447.
- [4] 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.
- [5] Stonebraker, M., Wong, E., Kreps, P., and Held, G., The Design and Implementation of INGRES, *ACM TODS 1*, 3, (1976) pp. 189-222.
- [6] Yuasa, T., Hagiya, M., *Kyoto Common Lisp Reference Manual for SUN Workstation*, Kyoto University (1985)
- [7] Lamport, L.,  $\text{\LaTeX}$ : A Document Preparation System, Addison-Wesley (1986)
- [8] Ousterhout, J.K., *An Introduction to Tcl and Tk*, 未刊.
- [9] Steele, G.L., *COMMON LISP: the Language (second edition)*, Digital Press (1990)