

TFE を用いた動画再生とレイアウト

田中 宣行† 遠山 元道‡

†慶應義塾大学 理工学研究科 管理工学専攻

‡慶應義塾大学 理工学部 情報工学科

電話 : 045-563-1141 内線 3606

E-mail:{nori,toyama}@ae.keio.ac.jp

要約

TFE(Target Form Expression) は、質問言語におけるターゲットリストの概念を拡張した一種の式であり、階層的な表の構造とレイアウトを規定するオペランドと演算子からなる。演算子を自由に組み合わせることによって多様なレイアウトを指示できる。本研究で、TFE の時間軸方向の反復を利用して、関係データベースに格納されている複数の画像の動画としての再生を可能にした。また、動画を水平、垂直方向に連結して出力する際には、“間引き”によってより自然な出力を得ることができるようにし、時間軸を含めたレイアウトに対しても TFE が有効であることを確認した。

キーワード TFE, SQL, 質問言語, 関係データベース, 動画, マルチメディアデータベース

Animated picture layout using TFE

TANAKA Noriyuki, TOYAMA Motomichi

Department of Administration Engineering, Faculty of Science and Technology, Keio University.

TEL:+81-45-563-1141 ext. 3606

E-mail:{nori,toyama}@ae.keio.ac.jp

Abstract

TFE(Target Form Expression) is an extension of the target list in query language, which specify a structure of a stratified table and a layout. It consists of operands and operators. Various combinations of operators enable us to make enormous varieties of layouts. In this paper, we have introduced the fourth dimensional layout operators into TFE, that compiles a set of images in RDB into an animated movie. When the images in an animation are connected along with horizontal or vertical axis, thinning out technique makes more natural output. We have concluded that TFE provides good basis for the layout of image data in a RDB with time axis.

keywords TFE, SQL, Query Language, RDB, animation, multimedia database

1 はじめに

現在の情報化社会では、文字情報だけでなく音声、画像といった情報も提供されるようになり、関係データベースにもこれらの情報への対応がせまられている。例えば、気象衛星からの静止画像を関係データベースに格納すれば、質問言語 (SQL 等) を用いて簡単に各種の条件に基づく検索ができる。しかし、関係データベースの質問言語はデータの検索を主な目的としており、その出力結果をレイアウトするためには、外部の編集ソフトの力などを借りて整形しなおさなければならない。

そこで著者等のグループは、質問言語にレイアウト機能を取り入れた TFE (Target Form Expression) を用いたシステムを提案し [1]、その後、数々の拡張を行なってきた [2]。これらの拡張で、関係データベースからの出力結果のレイアウトや多様な媒体への出力ができるようになったが、新たにひとまとまりの画像 (気象衛星の静止画像など) を動画として再生させる要求が浮上してきた。

本研究では、TFE の拡張の一環として、関係データベースに格納されている複数の画像を自由にレイアウトしたり、それらを時間軸方向にまとめて動画として再生できるようにした。このように動画として再生できることで、一連の静止画像の時間的な変化や流れを理解するのに大きく役立つこととなる。

2 TFE

従来、関係データベースからの結果出力には“見やすさ”の点で問題があった。そこで、そのデータベースの“弱点”を補うために、質問言語におけるターゲットリストの概念を拡張したものが TFE (Target Form Expression) である。

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**からなる。

オペランドは、質問言語 (SQL 等) におけるターゲットリストの要素にあたる。また、**演算子**は、**連結子** (二項演算子) と**反復子** (単項演算子) の 2 種類に分けられ、それぞれについて**水平** (第一次元)、**垂直** (第二次元)、**深度** (第三次元)、**時間** (第四次元) が定義されている。さらに、装飾演算子や各種の関数があり、これらの演算子・関数を自由に組み合わせることによって多様なレイアウトの表現が可能になる。

2.1 GENERATE 句

TFE 処理系では、出力媒体 (言語) を明示するために SQL の SELECT 句の代わりに GENERATE 句を用いる。

```
GENERATE <媒体指定> <TFE>
FROM      from 句
WHERE     where 句
```

媒体指定には、latex、html、tcltkなどを記述することができる。これにより、指定した媒体の言語 (latex なら \LaTeX のファイル) を出力する。今回はこれに新たに“movie”を加えた。

2.2 TFE の構文

```
<TFE> ::= <項目> | <項目> <連結子> <TFE>
<連結子> ::= , | ! | % | %% | #
<項目> ::= <基本項目> |
          <基本項目> @ { <装飾子リスト> }
<基本項目> ::= <属性名> | <関数表現> |
              { <TFE> } | [ <TFE> ] <複合反復子>
<反復子> ::= , | ! | % | %% | #
<複合反復子> ::= <反復子> |
                <複合反復子> <数値> <反復子>
<関数表現> ::= <関数名 1> ( <TFE> ) |
              <関数名 2> ( " <文字列> " )
<関数名 1> ::= sum | ave | null | ...
<関数名 2> ::= verb | ...
<装飾子リスト> ::= <装飾子> |
                  <装飾子> , <装飾子リスト>
```

<装飾子>::=<装飾種類>=<装飾指定子>
 <装飾種類>::= font | width | …
 <装飾指定子>::="<文字列>"|<数値>|
 <数値><単位>

2.3 連結子

水平(,), 垂直(!)、深度(%)、時間(#) 方向の連結子(カッコ内がその記号)がある。連結子の両辺は任意の TFE で、これらの整形結果を指定方向に連結する。すなわち、水平連結子なら左から右へ横方向に、垂直連結子なら上から下へ縦方向に並べて出力する。深度連結子は指定媒体によって解釈が異なり、例えば latex を指定した場合、ページ替えを行なう。また、時間連結子は、整形結果を時間方向に連結する。

2.4 優先順位

通常、連結子における優先順位は左が優先であり、数式のように左から右へつなげていく。この優先順位を変更する場合には、優先的に処理したい部分を中カッコ{ }で括ればよい。

2.5 反復子

連結子と同様に、水平([],), 垂直([]!), 深度([]%), 時間([]#) 方向の反復子(カッコ内がその記号)がある。大カッコ[]の内側に任意の TFE を置く。大カッコ内で整形結果が複数生成される時、これらを指定方向に逐次連結する。特に、TFE の一番外のレベルで時間方向の反復子が使われた場合、大カッコの内側で整形された結果を時間方向に連結し、一つの動画再生画面を出力することになる。

2.6 群化基準項目

TFE において反復群が一つ以上の項目に連結されているとき、これらの項目は群化基準項目になる。問い合わせ言語の出力結果がこれら

の群化基準項目により群化(グルーピング)され、冗長な群化基準項目の表示が抑制される。

d.name, e.name [d.name, [e.name]!]!

name	name		name	name
紳士服	佐藤	➡	紳士服	佐藤
紳士服	下谷			下谷
紳士服	瀬戸			瀬戸
書籍	永藤		書籍	永藤
書籍	指田			指田
家具	加藤		家具	加藤
家具	軽部			軽部
家具	実政			実政

図 1: グルーピング

2.7 装飾演算子

出力の際に、様々な装飾を施すための演算子が装飾演算子(@)である。基本項目の後に@をつけ、中カッコの中で(,)で区切って、装飾子を並べる。

[d.name@[fontsize=60,font=Bold],[e.name]!]!

name	name
紳士服	佐藤 下谷 瀬戸
書籍	永藤 指田
家具	加藤 軽部 実政

図 2: 装飾

2.8 関数

関係データベースから取り出された値を集計したり、他の型のデータに変換するためのものである。例えば、imagefile 関数は、関係データベースからの出力に画像のファイル名がある場合、そのファイル名の画像に変換し出力する。他に、集約関数・グラフ生成関数などがある。

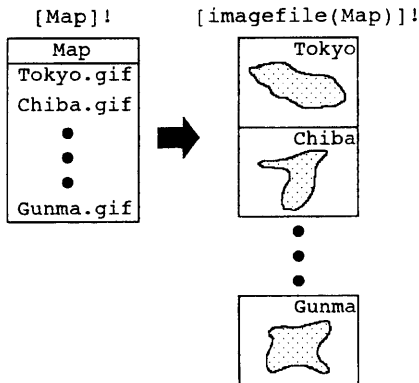


図 3: imagefile 関数

2.9 TFE 処理系

TFE 処理系に入力された質問文は、構文解析部で通常の SQL 文とレイアウト式に分離される。SQL 文はそのままデータベースに入力され、その結果とレイアウト式とを組み合わせ、実際の属性値間のつながりを表現した階層型リスト構造が生成される。こうして得られた階層型リスト構造とレイアウト式を用いて、指定した媒体の言語で書かれたファイルを出力する。

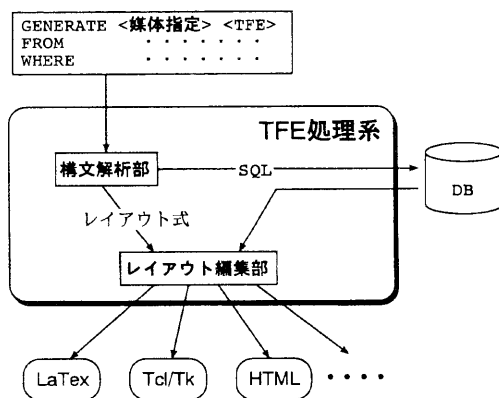


図 4: TFE 処理系

3 動画と第四次元反復

TFE において時間 (第四次元) 方向の反復子 ([]#) を用いることで、大カッコの内側にある複数の整形結果を時間方向に連結することができる。もし、大カッコの内側が画像ファイルであった場合、これらを時間方向に連結して再生させることで動画として利用できることになる。ここでは、その動画の定義と第四次元反復子の持つ意味を説明する。

3.1 動画

動画は、MPEG¹や QuickTime Movie²といった複数のフレームを圧縮して一つのファイルとして扱うのが普通である。しかし、本研究では前提条件として、静止画像が複数枚集まったものであるとして扱っていく。この時、画像一枚一枚をフレームと呼び、1秒間に表示するフレームの数をフレーム数と呼ぶ。そしてこれらのフレームのファイル名が関係データベースに格納されているものとする。

3.2 第四次元反復

第四次元反復 ([]#) の大カッコの内側が、imagefile 関数でファイル名から画像に変換された複数の整形結果である場合、それらを時間方向に連結することになるが、もし、この第四次元反復子が TFE の一番外側のレベルである場合、時間的な変化を見ることのできる動画再生画面が出力される (図 5)。

画面の下にあるボタン類 (これを操作パネルと呼ぶ) は、再生、停止、一時停止、巻き戻し、コマ送りなどを行うためのもので、装飾リストの中にフレーム数を記述するとそのフレーム数で動画を再生することになる。

¹Moving Picture Experts Group. メディア統合系動画圧縮の国際標準

²アップルコンピュータ社によるデジタルムービーファイルのためのフォーマット

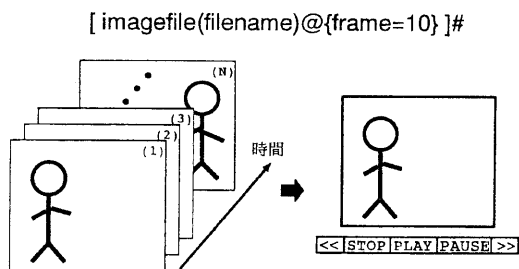


図 5: 四次元反復

4 動画再生とレイアウト

N 枚の画像ファイルを時間方向に連結して出力をすることで、動画としての再生が可能になる。これは、画像中にあるオブジェクトの動きや変化をリアルタイムで見ることができるという点で、非常に有効である。また、これらの画像ファイルを水平、垂直方向に連結して出力することで、より細かくオブジェクトの変化を見ることができる。

本研究では、動画として幼児の歩行を横と斜めから撮影したものを使用し、X-Windows 上に出力するよう実験的に拡張してみた。

4.1 試作システムのながれ

動画の再生やレイアウトを行う場合、指定媒体に“movie”と記述をする。今回の実装では、TFE 処理系は質問を処理し、結果として X-Window に表示するための Motif によるソースファイルを生成する。このソースファイルをコンパイル、実行し結果を X-Window 上に表示する。

4.2 実行例

関係データベース中に image と framedata という 2 つの関係があるとする。image の属性には、画像の番号、名称、作成日があり、画像の番号が主キーである。また、framedata の属

性には、スキーマ image の外部キーである番号と、フレーム番号、ファイル名がある。

image			framedata		
number	name	date	number	frame	name

このとき、次の質問文を TFE 処理系に通して得られた結果を再生させた様子が図 6 である。

```
GENERATE movie [ imagefile(f.name)@
                  {frame=10} ]#
FROM image i, framedata f
WHERE i.name='Side' and i.number=f.number
```

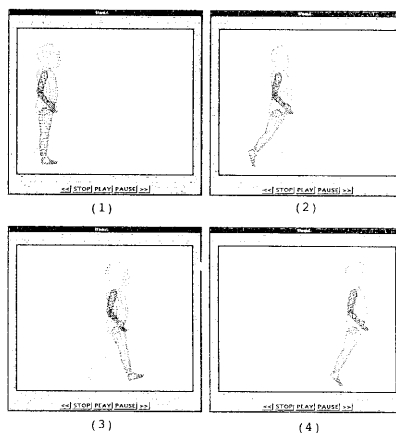


図 6: 出力結果と再生の様子

また、次のような質問文があった場合、出力結果は図 7 のようになる。

```
GENERATE movie [ imagefile(f1.name)@
                  {width=50%,height=50%} !
                  imagefile(f2.name)@
                  {width=50%,height=50%} ]#
FROM image i1, image i2,
      framedata f1, framedata f2
WHERE i1.name='Side'
      and i2.name='SideLong'
      and i1.number=f1.number
      and i2.number=f2.number
```

この場合、2 つの動画は同じ 1 つの時間軸上にあり、同時に再生、コマ送りなどが行われる。また、装飾リストは、原画像に対して幅・高さを 50% に縮小することを意味している。

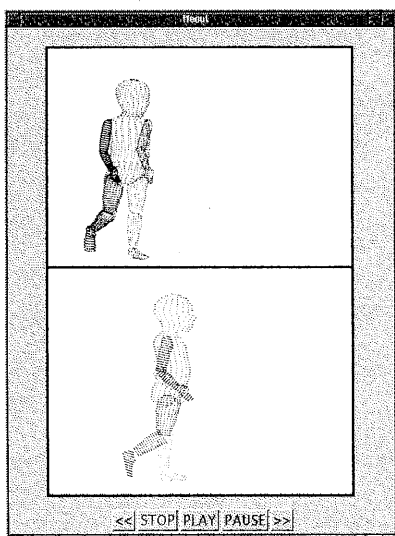


図 7: 2 動画 1 画面再生

この問い合わせを行なうと、関係データベースからの出力結果はファイル名に関して直積をとったものが得られるが、システムではその結果から各カラムにある重複しているバリューを削除して、各動画のフレームのファイル名を取得するようになっている。

4.3 間引き

ある動画が N 枚の画像からできている場合、その画像を水平、垂直方向に連結して出力すると、画面内に収まらなかつたり、たとえ縮小をして画面内に収まったとしても、隣り合う画像の差は小さく、かえって見づらいものとなる。

そこで、動画を水平、垂直方向に連結して出力するような場合には、装飾リストの中の interval で指定した秒数の画像だけを出力するようにする。しかし、(フレーム数) \times (秒間隔)が整数でない場合、指定した秒数のフレームが存在しないことになる。このような時は、その前後の近いフレームで代用することにする。つまり、動画が n ($n = 1, 2, \dots$) 個、各動画のフレーム数を fr_n とし、表示する動画の間隔を i

とすると、

$$D(n, c) = \text{int}(fr_n \times i \times c + 0.5) \quad c = 0, 1, \dots$$

となる $D(n, c)$ 番目のフレームを表示すればよい。次の図 8 が間引きを行なった例である。上半分がフレーム数 10、下半分がフレーム数 15 の動画であり、interval は 0.6 秒である。

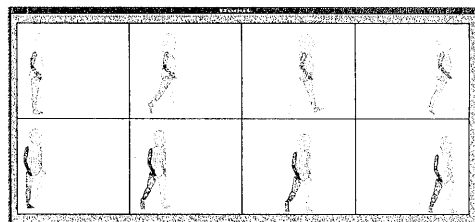


図 8: 間引き

5 結論

TFE の時間 (第四次元) 方向の反復子を用いることで、気象衛星からの画像のような静止画の集合を動画として再生させることが可能になった。TFE を用いると、動画 1 つを単純に再生させるだけでなく、複数の動画を自由にレイアウトして再生させることができ、また、動画を水平、垂直方向に連結して表示させる場合には、間引きが行なわれてより見やすい (自然な) 出力を得ることができる。このように TFE を用いることでユーザーは関係データベース中にある画像の情報を自由に、かつ簡単に利用することができるようになった。また、本研究で TFE が時間軸を含めたレイアウトに対しても有効であることが確認できた。

参考文献

- [1] 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 Database, World Scientific Pub. Co., pp139~153(1992)
- [2] 遠山 元道 他: 『レイアウト式 TFE の拡張』, 情報処理学会データベースシステム研究会資料, P217-224, 1995