

G-BASE用ビットマップ・ユーザインタフェースの試作

2R-2

平岡優希子, 稲葉清高

(株)リコー ソフトウェア研究所

1. はじめに

UNIXワークステーション上のウィンドウシステムとしては、Xウィンドウが標準となりつつあり、そのうえで実際にユーザインタフェースを開発するためのライブラリ(Toolkitという)の研究が盛んに行なわれている。[1] また、当研究所において開発した拡張リレーショナルデータベースG-BASEには、Xウィンドウの一部の機能を備えたDM-Lispがある。

そこで我々は今回、G-BASE上にDM-Lispでユーザインタフェースを開発するためのライブラリ(legと呼ぶ)を作成し、それを用いてオリジナルな画面を通してデータベースの操作を行うためのツール(wigと呼ぶ)の試作を行った。

本稿では、その概要について述べる。

2. 概要

我々が今回開発したwig(Window Interface for G-BASE)は、ビットマップディスプレイ上でオリジナルな画面を通してデータベースの操作を行うためのツールである。wigを用いることにより、ユーザが設計した画面を通して、さまざまなデータベースを検索、更新することが可能になる。

特徴としては、以下の4つがあげられる。

- ・ビットマップディスプレイを対象としている。
- ・ユーザが自由に画面を設計できる。
- ・leg(Lisp Environment for G-BASE)というライブラリを使っている(legもDM-Lispで開発した)。
- ・Xウィンドウ(Version 10.4)を用いている。

wigを使うためには、まず画面を定義するファイルを作らなければならない。このファイルのことをform-file(フォームファイル)と呼ぶ。また、定義した画面のことをform(フォーム)と言う。定義方法については、3章で述べる。

ユーザが画面に表示するものとして定義できるのは、1. ボタン, 2. 見出し, 3. データ表示領域の3つで、それぞれ出力したい位置、表示する文字列、フォントなどを自由に変えることができる。

3. form-fileの書き方

図1にform-fileの例を、図2にそれを実行した時の画面を示す。

form-fileは複数のS式を含み、それぞれS式中の最初のatomにdatabase, commandなどの識別子、2番目以降はname, xなどの属性名とその値を並べて書く。

Prototype of bitmap user interface
for G-BASE
Yukiko HIRAOKA, Kiyotaka INABA
Software Research Center
Ricoh Co., Ltd.

```
(font name small fn "7x14kana" fk "14x14kron")
(font name big fn "12x24kana" fk "24x24kron")
(database name "~/sp/demo")
(path name (// 部下 社員 社員1) order (:a 氏名 社員))
(alias name (alias 社員1 社員))
(button name "次へ" x 20 y 20 func next-data font big)
(button name "前へ" x 100 y 20 func previous-data font big)
(button name "検索" x 180 y 20 func search-db font big)
(button name "終了" x 300 y 20 func exit font big)
(label name "上司" x 20 y 80 font big)
(label name "名前" x 100 y 100)
(label name "生年月日" x 120 y 120)
(label name "血液型" x 120 y 140)
(label name "部下" x 20 y 180 font big)
(label name "名前" x 100 y 180)
(label name "生年月日" x 120 y 200)
(label name "社員番号" x 130 y 220)
(data length 180 x 200 y 100 field (: 氏名 社員))
(data length 180 x 260 y 120 field (: 生年月日 社員))
(data length 100 x 200 y 140 field (: 血液型 社員))
(data length 180 x 200 y 180 field (: 氏名 社員1))
(data length 180 x 260 y 200 field (: 生年月日 社員1))
(data length 100 x 200 y 220 field (: 社員番号 社員1))
```

図1 form-fileの例

現在、定義できる識別子及び属性名は、以下の通りである。

識別子	属性名
database	name
path	x
font	y
alias	length
button	func
label	font
data	fn
	fk
	field
	order

定義する順序は特に定めないが、database及びpathは必ず指定しなければならない。識別子の種類によって定義できる属性名が異なる。また、識別子より後の属性名の順序も特に定めない。簡単にそれぞれを説明する。

database
データベース名を指定する。
属性名nameは、データベース名を"で囲んだものを値とする。

path
検索する際に用いるパス[2]と、結果を表示する順序を指定する。属性名nameは、値としてパスをとり、DM-Lispにおけるデータベース検索文の形式[3](lrt表現)で書く。
orderは値としてlrt表現の順序指定を書く。
例のように、(順序 フィールド名 レコード名)と指定する。
順序は、:a ----- 昇順
 :d ----- 降順
である。

font

フォントを指定する。
 フォントは定義できる項目ごとに指定できるので、
 属性名nameの値としてユーザが任意にフォントに
 名前をつけて、その名前を後でフォントとして指
 定することになる。
 属性名fkは値として漢字フォントを、fnはANKフォ
 ントを書く。
 両方のフォントを指定しなければならない。
 フォント指定を行わないと、fkが"16x16krom",
 fnが"8x16kana"になる。

alias

再帰リンクの場合等に用いる異称宣言文を指定す
 る。
 属性名nameの値として、lrt表現のalias宣言文を
 書く。

button

コマンドボタンを指定する。
 属性名nameの値として、ボタンの部分に表示す
 べきストリングを書く。
 x, yには、pixel単位の座標を指定する。左上端
 を(0,0)とする。
 funcには、ボタンを押した時に起動すべき関数名
 を書く。現在のところ関数名は、以下にあげる物
 の中から指定する。

next-data ----- 次のデータを表示する。
 previous-data -- 前のデータを表示する。
 top-data ----- 先頭のデータを表示する。
 bottom-data ---- 最終のデータを表示する。
 retrieve-data -- 検索する。
 update-data ---- 更新する。
 insert-data ---- 挿入する。
 delete-data ---- 削除する。
 exit ----- 終了する。

label

文字列を表示する際に用いる。
 属性名nameの値として、表示したいストリングを
 書く。xとyは、buttonと同じ。

data

データベース中のデータを表示するための領域を
 指定する。
 属性名lengthの値として、pixel単位の長さを書く。
 lengthを指定しないと、データ辞書の中から長さ
 を取り出してくる。
 fieldには、lrt表現のフィールド指定を書く。
 (: フィールド名 レコード名)
 のように表現する。
 xとyは、buttonと同じ。

図2 図1のform

4. 効果

今までに、ビットマップディスプレイ上に実現され
 たパソコン用RDBソフトとしては、Macintoshで使える
 dBASE Mac, 4TH Dimension, Reflex(Plus)などがある。

今回は、ライブラリと共にそのアプリケーションも
 並行して作成したわけだが、ライブラリへのニーズが
 即座に反映され、効率が向上した。

また、legを用いたことにより、イベントのハンド
 リングを行わなくて済んだので、wigと同様の機能
 を持つデモプログラムと比べて、ソースの行数が約1/3
 になった。

5. おわりに

DM-LispでXウィンドウシステム上にToolkitを作成
 し、それを用いてデータベースを操作できるツールを
 試作した。

今後の課題としては、以下のものがある。

- ・tableを表示できるようにして、レポートライ
 タとして応用する。
- ・現在のform-fileでは座標指定が非常に面倒なので、
 フォーム定義用のツールを作る。
- ・ボタンが押された時に起動される関数を、ユーザが
 自由に定義できるようにする。
- ・表示するデータを動かすために、スクロールバーを
 使えるようにする。

今回は、このような点を考慮して開発したい。

【参考文献】

- [1] Digital Equipment Corp., Hewlett-Packard Co.
 and M.I.T. Project Athena: X Toolkit
 A Proposal Architecture Preliminary, Draft
- [2] (株)リコー編: システムマニュアル
- [3] (株)リコー編: lrtマニュアル