

# オブジェクト指向データモデル による非正規形レポート生成方式

2R-3

鶴岡 邦敏

(日本電気株式会社 C & C システム研究所)

## 1. はじめに

レポート生成ツールは、データベースの利用者にとって、見やすい報告書を少ないプログラミングコストで作成できるという点で重要である。非定型のレポート生成に関しては、関係データベース上の問合せ言語が知られているが、一般に単純な形式の表しか生成できず、オフィス等で頻りに用いられる非正規形のレポート(図1参照)は作成できない。

一方、近年、非正規関係モデルの研究が進み、実験システムも開発されつつある。しかし非正規形レポート(図1上)は、外部ビューとしての非正規関係(図1下)の上に構築される提示ビューであり、より複雑な構造を持つ。現状ではこうした複雑な提示形式の生成は考慮されておらず、利用者が相当量のプログラムを作成せねばならない。

本資料では、複雑な非正規形レポートを生成するために、オブジェクト指向データモデルを提示形式にまで拡張し、「動的に参照しあうセル群」を実行時に組み合わせて出力する方式を提案する。

## 2. 非正規形レポートの構造

以下では、外部ビュー、提示ビューの個々の関係/属性に関する記述をそれぞれ外部クラス、提示クラスと呼び、それらの実現値(またはその集合)をインスタンスと呼ぶ。図2のように、一つの外部クラスに対して種々の形式の提示クラスが考えられる。これらは、用紙の制約や報告書の見やすさを考慮して選択される。提示クラスは繰り返し項目の数に依存した可変の構造を持ち、その形式の多くの部分が実行時に決定される。本方式では、提示インスタンスをセル(矩形領域)の集合としてとらえる。各セルの各座標は、他のセルのある座標に依存して実行時に決まる。

## 3. 外部クラスの定義

外部クラスは、非正規関係の各部分関係ごとに定義される。インスタンス変数が各属性に対応し、それが下位のオブジェクトを指す。図3に例を示す(図2のRに対応)。また、各属性(図2のA、D等)に対応したクラスも定義され、これを基本クラスと呼ぶ。外部クラスのインスタンスの例を、図4に示す。

## 4. 提示クラスの定義

提示クラスは、各基本クラスに対応して定義さ

レポート (提示ビュー)

部別発表論文一覧表 (63年)				ハジ: 001	
論文番号	発表先	論文名	年月	著者名	社員番号
0826	情処全 国大会	フォーム管理システム	63/09	鈴木一男	123456
				山田次郎	987654
				佐藤三郎	345678
0839	信学ソ シテム	電子文書作成システム	63/10	渡辺花子	765432
				伊藤太郎	567890
				山田次郎	987654
0847	情処DB S研	知識データベース管理方式	63/11	鈴木一男	123456
				山田次郎	987654

非正規関係 (外部ビュー)

部別発表論文一覧表 (63年)					
論文番号	発表先	論文名	年月	著者名	社員番号
0826	情処全 国大会	フォーム管理シ ステム	63/ 09	鈴木一男	123456
				山田次郎	987654
				佐藤三郎	345678
0839	信学ソ シテム	電子文書作成シ ステム	63/ 10	渡辺花子	765432
				伊藤太郎	567890
				山田次郎	987654
0847	情処DB S研	知識データベー ス管理方式	63/ 11	鈴木一男	123456
				山田次郎	987654

図1. レポートと非正規関係

外部クラスのインスタンス

R	A	B	C	D	E
a 1	b 1	c 1	d 1	e 1	
			d 2	e 2	
			d 3	e 3	

提示クラスのインスタンス

(a)

	c 1	
a 1	d 1 e 1	b 1
	d 2 e 2	
	d 3 e 3	
a 2	c 2	b 2
	d 4 e 4	
	d 5 e 5	

(b)

	d 1 e 1	
a 1	d 2 e 2	
	d 3 e 3	
	c 1	
	b 1	
a 2	d 4 e 4	
	d 5 e 5	
	c 2	
	b 2	

(c)

	b 1	b 2
c 1	d 1 d 2 d 3	d 4 d 5
	e 1 e 2 e 3	e 4 e 5
	a 1	a 2

図2. 外部クラス上の種々の提示クラス

れる。図5に例を示す(図2(a)に対応)。各提示インスタンス(セル)は、その4頂点の座標をインスタンス変数として保持する(図5のHF~VL)。また各変数の値を返すためのメソッドを保有する。メソッドは、自セルが依存する他のセルにメッセージを送って相手の座標を受取り、自己の座標を決定する。

提示クラスPresBに対して、垂直方向の下部座標を決定するためのメソッドVLを例に説明する。セルBの座標VLは、セルDの座標VLによって決定される(図2(a)参照)。従ってセルDからそのVL座標を得るためのメッセージ式(PresD# VL)を記述する。また自分のVLが既に設定されていれば、その値を返す。ここでセルDは最初のセルと2番目以降のセルとで参照座標が異なるため、PresDのサブクラスとしてPresDNが定義される(図5右)。セルBは最後のインスタンスセルDに依存するので、PresBはPresDNの方を参照する。

次に、PresD#は、以下のような実際のインスタンスを参照するメッセージ式に書換えられる。  
 (((B ownerB) lastG) D) PresD  
 これは、実行時のPresBのインスタンスから見て、参照すべきPresDNのインスタンスを決定するための式である。例えば図2(a)のb1からd3を参照するためには、図4のようなインスタンス群の中での相対的なパス指定が必要であり、上式はこれを示している(ownerB、lastG等は各クラスに定義されたメソッドである)。

```
(defineclass R
  (classvariables ... )
  (classmethods ... )
  (instancevariables
    (A instance A)
    (B instance B)
    (C instance C))
  (G multipleinstances G))
  (instancemethods
    (nextG ((Y instance G) ... )
    (A (if (null A) (lowerobject A) A)
    ... ))
```

図3. 外部クラス定義

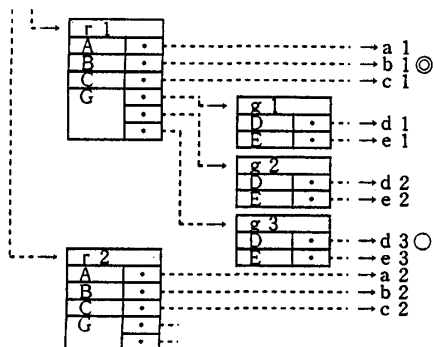
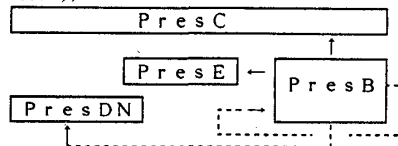


図4. インスタンス間の関連

```
(defineclass PresB
  (classvariables
    (C_B class B)
    (Hsize instance Integer)
    (Vsize instance Integer))
  (instancevariables
    (B instance B)
    (HF instance Integer)
    (HL instance Integer)
    (VF instance Integer)
    (VL instance Integer))
  (instancemethods
    (HF (if (null HF) (PresE# HL) HF))
    (HL (if (null HL)
      (+ (self HF) Hsize) HL))
    (VF (if (null VF) (PresC# VL) VF))
    (VL (if (null VL) (PresD# VL) VL))
    (display ... )
    ... ))
```



```
(defineclass PresD...
  (instancemethods ...
    (VF (if (null VF) (PresC# VL) VF))
    ... ))
  (defineclass PresDN
    (superclass PresD)
    (instancemethods ...
      (VF (if (null VF) (PresD# VL) VF))
      ... ))
```

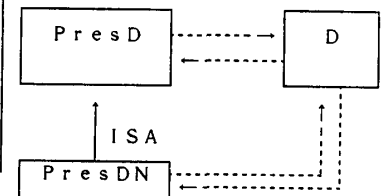


図5. 提示クラスの定義

5. 対話型設計—提示クラス記述の生成

- 提示クラス記述は、以下のように生成される。
- ①利用者が画面上に矩形を描き、外部クラスとの対応関係をつけると、提示クラスが作られる。
- ②システムが隣接セル間の参照関係を判定し、メソッド群を生成する。

6. 提示インスタンスの出力

- レポートの出力時には以下の処理が実行される。
  - ①出力すべき外部クラスのインスタンスに対応して、提示インスタンスが生成される。
  - ②各提示インスタンスは、その座標がすべて決定済みなら自身の内容を出力する。
  - ③未定の座標があれば、参照先のインスタンスにメッセージを送り、座標値を得る。
- このように、各オブジェクトは独立に動作する。

7. おわりに

- 本資料で提案した方式は、以下の特長を持つ。
  - 非正規関係を更に拡張した複雑なレポートを生成可能である。
  - 実行時にレポートの形式を動的に修正することが可能である。
  - レポート生成の記述量が削減され、かつ利用者がプログラムを記述する必要がない。
- なお本方式は、FORMATION[1](オフィスフォーム自動生成システム)において、一部実現されている。

参考文献

[1] 齊藤、栗本、鶴岡「FORMATION(3)-フォーマット編集方式」 情報処理学会第36回(昭和63年前期)全国大会、3S-9