

# TFE を用いた木構造における 関係データベースから オブジェクト指向データベースへのインスタンス変換

鶴戸口 志郎 † 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

## 1 はじめに

オブジェクト指向技術が今日のアプリケーション開発の主流となりつつある。オブジェクトアプリケーションに相性が高いのは当然オブジェクト指向データベースであるが、現状では大部分のデータ資産が関係データベースに格納されている。そこで本研究ではフラットな表型データに構造を持たせる能力を持つ TFE 処理形を用いて関係データベースに格納されているデータ資産をオブジェクト指向データ型に変換してオブジェクト指向データベースに移し変える処理系を試作した。なお、今回使用したオブジェクト指向データベースは O<sub>2</sub> Technology 社の O<sub>2</sub> システムである。

## 2 TFE によるオブジェクトの生成

本処理系でのオブジェクト指定にはこれまで開発されてきた TFE のシンタックスの一部もしくはそれを拡張したもので表現される。

### 2.1 クラスの表現

クラスの表現は TFE の水平連結子、装飾演算子を用いて表現される。例えば、

```
GENERATE o2c {S.id,S.name,S.major}
           @{class=STUDENT}
FROM STUDENT = S
```

The Translation from Relational Data to Object Oriented Data in the tree structure  
SHIRO Udoguchi, TOYAMA Motomichi  
Department of Administration Engineering, Faculty of Science and Technology, Keio University.

### 2.2 オブジェクト間の集約関係の表現

親オブジェクトの中に子オブジェクトを埋め込むことで表現する。例えば、

```
GENERATE o2c {U.id,U.name,
             {A.id,A.name}@{class=Address}
             }@{class=University}
FROM      University = U,Address = A
WHERE     U.adress=A.id
```

### 2.3 グループ化

親オブジェクトのメンバとして子オブジェクトを複数持っている場合 (Set, Bag 等のコレクションライブラリを用いるような場合) 反復演算子 [, ] を用いてこれを表現する。例えば、

```
GENERATE o2c [{s.number,s.city,
             [d.number,d.floor,d.name,
             [{i.number,i.name}@{class=ITEM}]!,
             [{e1.number,e1.name}@{class=EMPLOYEE}]!
             ]@{class=DEPT}]!
             ]@{class=STORE}]!
FROM store = s, dept = d, item = i, employee = e
WHERE s.number=d.store
AND d.number=i.dept
AND d.manager=e.number
```

とすれば STORE オブジェクトが複数の DEPT オブジェクトを所有し、DEPT オブジェクトは複数の EMPLOYEE オブジェクトと複数の ITEM オブジェクトを所有していることが表現できる。図 1 にシステム実行のイメージを示した。

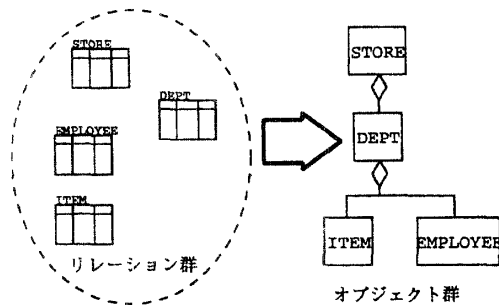


図 1: TFE を実行して得られるオブジェクト

## 2.4 実装上の工夫点

### 2.4.1 オブジェクトの識別

あるオブジェクトの子オブジェクトが他のオブジェクトにも共有されている場合がある。例えば次の図2のようにC君がAサークルとBサークルに掛持ちで所属している場合である。

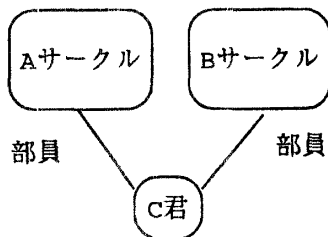


図 2: 子オブジェクトが親に共有されている場合

このような場合はAサークルに所属しているC君とBサークルに所属しているC君を同じオブジェクトとして認識しなくてはならなくなる。よってAサークルオブジェクトを生成する際にC君オブジェクトを生成し、Bサークルオブジェクトを生成する際にC君オブジェクトをもう一度生成してしまうとメンバが全て同じであるオブジェクトが2つ生成されてしまうことになる。(このときOIDは別である。)

そこで本処理系ではオブジェクトを生成する際にクラスメンバにIDを含めることを強制している(これはプログラマに対して可視的でありOIDではない)。このIDは関係データベースの主キーをもとに自動的に生成されるものである。このIDをもとに本処理系ではクラスインスタンス(オブジェクト)を識別している。

表 1: ID テーブル

クラス名	ID
UNIVERSITY	"8-63-120-113" "7-67-21-117"
ADDRESS	"54-23-103" "70-118-120"
STUDENT	"116" "115" "114" "109"

本処理形ではオブジェクトを生成する際には表1のように生成されたIDをクラスごとにしきられたIDテーブルに格納している。オブジェクトが生成される際にはこのテーブルを確認し、同じIDを持つオブジェクトが重複して生成されないように管理している。

## 3 評価・検討

本処理系では既存の関係データベースからオブジェクト指向データベースへのインスタンス変換を試みた。この際、上で示したTFEを用いてオブジェクトの構造、オブジェクト間の関係を直観的に指定することが可能となった。

## 4 課題

現在では本処理系で実現できるオブジェクト間の関係は集約関係のみである。よって集約関係以外の関係も自然な形で指定して生成できるようにしたいと考えている。また、オブジェクトメンバの可視属性も指定したいと考えている。

## 参考文献

- [1] 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の概視の実現」、信学技報、Vol.93, No.152、PP.79-88、電子情報通信学会、1993
- [2] 「オブジェクト指向データベース-基礎知識から製品の評価まで-」、日経インテリジェントシステム別冊
- [3] Persistence Software Inc. Persistent software: <http://www.persistence.com/>