2H−9

オブジェクト指向データベース「沙羅」のデータベース言語

久保田和己　石丸知之　工藤礼子　陳思悦　日置真也　植村俊亮
東京農工大学

1．はじめに

オブジェクト指向データベースシステム「沙羅」[1]のデータベース言語について報告する。本言語は沙羅のデータモデルの特徴である「汎化関連」や「集約関連」を反映したデータベース構築機能と、応用プログラミング言語としての機能を備えている。

2．データベースの構造

沙羅では、データベース内のデータをすべてオブジェクトとして扱う。オブジェクトは静的な性質である属性(attribute)と動的な性質であるメソッド(method)をもっている。属性とメソッドが同じであるオブジェクトの集合をクラスと呼ぶ。クラスの間には、汎化関連と集約関連がある。

データベースを構築することは、クラスを定義することである。またデータベースにデータを追加する作業はクラスにインスタンスを追加する作業に当たる。

クラスの定義方法を人の友人関係の例を用いて説明する。図1はクラス person の定義の例である。

3．データベース言語の概要

本言語の処理単位は、クラスに定義されたメソッドの実行にもとづく。メソッドの実行は「メソッド実行式」で表し、一般に次のように記述する。

a.b(c)　オブジェクト a のメソッド b を実行する。引数として オブジェクト c をとる。

メソッドの実行は結果としてオブジェクトを返し、それが式の値となる。返されたオブジェクトに対してさらにメソッドを実行をしたり、式を引数にすることも可能である。

(1) 汎化関連

クラス間の汎化関連は、クラスの定義の一部として、superclass 句で記述する。クラス person のスーパークラスである S_class は、すべてのクラス階層のもっとも上位にあるクラスである。

下位のクラスは、上位のクラスの属性とメソッドを継承する。また沙羅では多重継承を許している。継承によって発生する名前の衝突は、自己のクラスのものを優先する。また、スーパークラスの宣言の順に継承の優先度を規定することで、多重継承での衝突を回避する。

継承はクラスを巡回しない。つまり、継承をたどっていっても自分に戻ってくることはない。この制約は、「スーパークラスとして宣言するクラスは、すでに定義されているクラスでなければならない」という規則と等価である。この規則は構文上の制約条件として保証されている。

```
defclass person
  superclass S_class;
  property
    name:string notnil;  /* 名前 */
    friend:set(person);  /* 友人 (person の集合)*/
  method
    /* クラス person のインスタンスを生成 */
    add_person:person(name:string)
      local p:person;
      proc
        /* 同じ名前の人はいないという制約 */
        if (person.getperson(name)) then return;
          else
            p=person.new;
            p.name=name;
            p.friend=set.new;
        endif
      endproc
    /* 名前でクラス person のインスタンスを検索 */
    get_person:person
      proc
        for p:person
          where (p.name == name) endwhere
            do
              return p;
            enddo
        endfor
      endproc
    /* 友人 f を追加する */
    add_friend:person(f:person)
      proc
        if (self.friend.exist(f)) then return;
          else
            self.friend.add(f); /* 友人を追加 */
            f.friend.add(self); /* 逆もまた保証 */
        endif
        return self;
      endproc
  endmethod
enddefclass
```

図1　クラス person の定義

A Database Language for the Object-Oriented Database System *Sarah*

Kazumi KUBOTA, Tomoyuki ISHIMARU, Reiko KUDO, Si-Yue CHEN, Shinya HIOKI and Syunsuke UEMURA
Faculty of Technology, Tokyo University of Agriculture and Technology

(2) 集約関連

クラス間の集約関連として、あるクラスの属性がどのクラスのインスタンスを値としてもつかを、属性の宣言に記述する。person の属性 name は、クラス string のインスタンスを値としてもつ。また属性 friend は クラス set のインスタンスを通して、クラス person のインスタンスを指すことで、集合を表現している。クラスをデータ型、インスタンスをデータの実現値と見ると、この宣言は属性の型を宣言していると考えられる。

また、集約関連によりオブジェクトを導出することができる。導出は、「属性参照式」で記述する。

```
p.name      クラス person のインスタンス p の
            属性 name の値であるオブジェクト
```

この属性参照式はメソッド実行式と同じ形をしている。

(3) 静的制約

データベースの一貫性を保証するための静的制約として、型と空値(nil value)の扱いを属性の宣言として記述する。クラス person の属性 name の宣言で、

```
name:string notnil
```

は、(2) で述べたように、属性 name の型を宣言している。属性の宣言にもとづき、実行時に型の整合を調べる。

notnil は、この属性の値として空値を認めないことを意味している。notnil 句がついていない属性は値として空値をもちうる。

(4) 動的制約

データベースの一貫性を保証するための動的制約はメソッドの一部として記述する。例えば、人 A が友人 B を持つ時は、人 B も友人 A を持つという制約条件は、図 1 のメソッド add_friend のなかで記述されている。

(5) 制御構造

制御構造は、条件分岐 (if 文) と、繰り返し (while 文) がある。いずれも条件として、論理値を返すメソッド実行式や変数を指定し、それに応じて処理の流れの制御を行う。

(6) オブジェクトの検索

オブジェクトの検索はクラスに対して行う。検索は、for 文を用いて記述する。クラス person に定義されたメソッド get_person は、名前からオブジェクトを検索するメソッドである。

また、検索はメソッドの定義の外で、会話的に記述することもできる。例えば、"久保田"と"石丸" の共通の友人を検索する手続きを 図 2 に示す。

```
k,i:person;
k=person.get_person("久保田");
i=person.get_person("石丸");
/* クラス person のインスタンスに対して検索*/
for p:person *
  where
    /* 条件は久保田と石丸の友人の集合に存在 */
    (k.friend.exist(p)) and (i.friend.exist(p))
  endwhere
  do
    /* 名前を表示する */
    p.name.display;
  enddo
endfor
```

図 2 共通の友人の検索

検索の対象とするクラスの指定で、クラス名の後に '*' をつけることで、検索の対象は指定したクラスのインスタンスに限定される。つけなければ、指定したクラスのインスタンスに加えて、そのすべてのサブクラスのインスタンスが、検索の対象となる。

検索の結果は、クラス set のオブジェクトとして変数に代入することが可能である。また、検索の結果を新しいクラスを作り、そのインスタンスとして得ることができる。このとき、新しくできたクラスは、指定がなければ、クラス階層の最上位のクラスである S_class のサブクラスとなる。

4. クラス parser

上で述べたクラス定義文や検索文などのすべての文は、クラス parser のメソッド parse に引数として渡される。メソッド parse では、これらの文を解析し、適当なオブジェクトへのメソッド実行文として実行する。このため、沙羅では、宣言的な問合せや手続き型の言語が、すべてメソッドの実行という形で統合されている。

5. おわりに

沙羅のデータベース言語を開発した。本言語ではデータベースの一貫性を保証するために、オブジェクト間の制約条件をクラスの定義の一部として記述する。特に動的な制約条件の表現能力の検証が今後の課題である。

また沙羅には、基本的な型や画像などの構造化されないオブジェクトを扱うための、システム定義クラスと呼ばれるクラスライブラリがある。今後は、本言語を用いてそれらのライブラリを充実させたい。

参考文献

[1] 石丸 他, オブジェクト指向データベース「沙羅」のデータモデル, 情報処理学会第 44 回全国大会論文集