4V-7

# 設計言語Ｄｅｌｐｈｉの実行方式

小笠原秀人* 石橋勝典* 石川 孝** 菅野 文友*
* 東京理科大学 工学研究科
** ぺんてる株式会社中央研究所

## 1．はじめに

設計支援システム構築用プログラミング言語として設計した，設計言語Ｄｅｌｐｈｉ［1］の実行方式について述べる．

Ｄｅｌｐｈｉコンパイラは，Ｄｅｌｐｈｉで記述された設計知識を，内部表現としてのＳ式に変換するパーザ（構文解析部）と，変換されたＳ式に基づいて変数の評価を行なう，エバリュエータ（意味解析・実行部）とからなるシステムである（図1）．

ここで，どの変数を評価するかは，質問によって指示される．

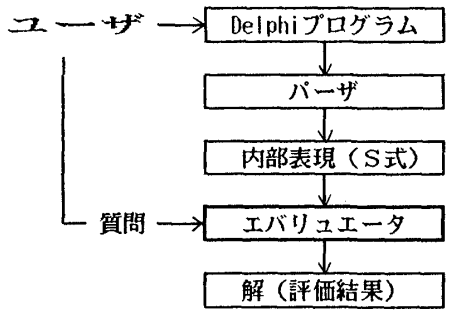


図1 Delphiコンパイラの処理の流れ

Ｄｅｌｐｈｉコンパイラでは，内部表現と評価結果を知識ベースに保管することによって，再評価における計算を効率化している．したがって，本システムは知識ベースコンパイラとしての機能を持つ．

本報告では，ノウハウ的設計知識の宣言的記述である制約の評価を行なう，エバリュエータの，評価方法と実行結果を示す．

## 2．エバリュエータの評価方法

本システムは，パーザによって変換されたＳ式を用いて，以下の規則にしたがって，Ｄｅｌｐｈｉプログラムを評価する．

① 変数評価演算子（@＜変数名＞）は，
(1) 値が代入されていれば（すなわちｎｉｌでなければ）その値．
(2) 値がｎｉｌなら代入式を評価した値．
とする（変数評価演算子@の評価規則）．

② 選択演算子（＜計算式＞ // ＜条件文＞）は，次のように評価する．
(1) 計算式を評価して，結果の１つを選択し，条件文を評価する．
(2) その結果が真だったら，計算式の評価結果を値とする．
(3) 偽だったら，別の結果を選択し，条件文を再評価する．
(4) さらに別の結果がなければ，nil とする．
計算式または条件文を評価する際に，さらに選択演算子を評価する場合は，最後に選択した結果を優先して再選択する．すなわち，縦形探索を行なう．

③ 条件文は，左側の条件式から順に評価する（論理演算子の評価規則）．

④ 条件式は，左側の項から評価する（比較演算子の評価規則）．

⑤ 項は，算術演算子の優先順位にしたがって，左側から評価する（算術演算子の評価規則）．

## 3．実行結果

設計言語Ｄｅｌｐｈｉで，力変換器の設計問題を記述し，パーザによって内部表現（Ｓ式）に変換した結果をを，図2に示す．

この内部表現をいかに評価するのかについて，以下に示す．

［質問の記述と出力結果］

力変換器の設計問題を解くには，Ｄｅｌｐｈエバリュエータに対して，図3の形式で質問をする．

この質問は，「力変換器において，最大測定力が１００であるもののストレインゲージを求めよ．」ということを意味する．

```
(力変換器//最大測定力=100) @ストレインゲージ
```

図3 質問の記述

---

An Evaluation Method of Design Language "Delphi"
Hideto OGASAWARA* , Katsunori ISHIBASHI** , Takashi ISHIKAWA* , Ayatomo KANNO*
*Science University of Tokyo , **PENTEL CO.,LTD.

```
((力変換器 (super nil) (variable
 (ビーム  (//'(AND ( < (@ 'WT (@ 'ビーム self))
          (@ 'MWT self))( < (@ 'LD self)
          (@ 'MLD self))) 'ビーム) nil)
 (ストレインゲージ  (//'( < (@ 'STRAIN self)
   (@ 'MAXSTRAIN (@ 'ストレインゲージ self)))
    'ストレインゲージ) nil)
 (最大測定力 nil F) (最大重量 nil MWT)
 (最大横変位 nil MLD) (最大長さ nil ML)
 (横変位 (* (* (* (/ (@ 'F self)
          (@ 'L (@ 'ビーム self)))
          (@ 'L (@ 'ビーム self)))(@ 'E self))
          (@ 'I (@ '断面 (@ 'ビーム self)))) LD)
 (ひずみ (* (/ (* (@ 'F self)(@ 'SGA self))
          (@ 'S self))(@ 'E self)) STRAIN)
 (SGA (@ '長さ (@ 'ストレインゲージ self)) nil)
 (S (@ 'S (@ '断面 (@ 'ビーム self))) nil)
 (E (@ 'E (@ '材質 (@ 'ビーム self))) nil) )
((力変換器 (nil nil 100 40 0.6 50 nil nil))
 (力変換器 (nil nil 150 50 1 100 nil nil))) )

 (ビーム (super nil) (variable
 (材質 '材質 nil) (長さ nil L)
 (重量 (* (@ 'L self)(* (@ 'A (@ '断面 self))
       (@ 'D (@ '材質 self)))) WT)
 (断面 '断面 nil) )
((ビーム (nil 10 nil nil))
 (ビーム (nil 20 nil nil))) )

 (材質 (super nil) (variable
 (材質名 nil nil) (弾性係数 nil E) (密度 nil D) )
((材質 (炭素鋼 2000 1) (材質 (低合金鋼 2100 2))) )

 (断面 (super nil) (variable
 (弾性モーメント nil I) (断面係数 nil S)
 (@ '断面積 nil A) )
((断面 (0.1 50 3)) (断面 (0.2 100 5))) )

 (ストレインゲージ (super nil) (variable
 (許容ひずみ nil MAXSTRAIN) (長さ nil nil) )
((ストレインゲージ (10000 1))
 (ストレインゲージ (15000 2))) ) )
```

図2　設計言語Ｄｅｌｐｈｉの内部表現（Ｓ式）

図3の質問をすると，結果として，図4に示す力変換器のインスタンスデータが中間結果として生成され，最終結果として，<ストレインゲージ>のインスタンスデータが返される．

```
[力変換器のｲﾝｽﾀﾝｽﾃﾞｰﾀ] =
( < ﾋﾞｰﾑ > < ｽﾄﾚｲﾝｹﾞｰｼﾞ > < F > < MWT >
  < MLD > < ML > < LD > < STRAIN > )
```

図4　中間結果出力形式

[質問の処理]

上述の質問が入力されると，本システムでは，次のような処理を行なう．

① 設計条件［最大測定力＝１００］を満たす，力変換器のインスタンスデータを選択する．

② このインスタンスについて，変数ストレインゲージの値を読むとｎｉｌなので，ストレインゲージの代入文を評価する．ここで，この評価のためには，変数ビームの値が必要となるので，まず，変数ビームについて，　　選択演算子を評価する．

③ ビームの候補をリストアップする．

④ 次に候補を１つ選択して，//の右側の条件文を評価する．

⑤ 条件文が満たされれば，そのときの候補のデータを返す．

⑥ 条件文が満たされなければ，次の候補を選択して③～⑤の処理を繰り返す．

⑦ 条件文の評価において，変数を評価する際に//があれば，③～⑥の処理を繰り返す．

図3の質問によって，変数ビームとストレインゲージの　　選択演算子が評価される．

本システムで，図3の質問を実行した結果，制約条件を満たす力変換器のインスタンスデータが中間結果として生成され（図5），ストレインゲージのインスタンスデータ（10000 1）が得られる．

```
(((炭素鋼 2000 1) 10 30 (0.1 50 3)) (10000 1)
                    100  20  0.5  50  0.5 4000)
```

図5　生成された中間結果

## 5．おわりに

本システムは，設計言語Ｄｅｌｐｈｉで宣言的に記述された設計問題を，変数評価演算子@と，　　選択演算子//によって，評価することを可能とした．

今後，さらに有効なシステムとするためには，オブジェクト指向データベースプログラミングとの融合（特に，設計条件に基づく知識ベースの管理）と，ユーザインタフェースの整備が必要であろう．

[参考／引用文献]


[1] 石橋勝典：“オブジェクト／制約指向パラダイムによる設計言語Ｄｅｌｐｈｉ”，情処第36回全大，(1989-03)．

[2] 石川孝：“CADｼｽﾃﾑ構築のﾌﾟﾛｸﾞﾗﾐﾝｸﾞ･ﾊﾟﾗﾀﾞｲﾑ”，CAD機能評価分科会ﾒﾓ，(1988)．

[3] 中島震：“COOLによる制約プログラミング”，情処技報 Vol.87,No.38,87-SW-54-1,(1987)．