TENSEのオブジェクト管理システムのカスタマイズ

4S−11

渡邊多恵子，西岡健自
横河電機株式会社

1．はじめに

TENSEとは，オブジェクト管理システムに基づいて統合したソフトウェア開発支援環境である．このオブジェクト管理システムの大きな特徴としてカスタマイズ機能がある．

TENSEではクラス情報から，格納・逆生成等のオブジェクト管理処理を自動生成する方式で，管理処理の応答性を落とさずにカスタマイズを実現している．ここで，クラス情報とは，情報の管理単位であるソフトウェア構成要素の種類に共通なメタ属性とデフォルト値である．従って，後に新たな形式のドキュメントを格納する必要が生じた時，ソフトウェアデータベース管理処理群を作り替えることで対応できる．また，クラス構成が全く異なるようなソフトウェアプロセスにも対応できる．

本論文では，管理処理の1つである逆生成メソドの機能と構成をとおして，TENSEデータベースのカスタマイズ機能について述べる．

2．逆生成メソドの概要

逆生成メソドとは，ソフトウェア構成要素をクラス毎に付随する手続きである．この手続きは，ソフトウェア構成要素を特定の書式のドキュメントとして出力する．この特定の書式は，ユーザが予め定義したもので，逆生成メソドに組み込んである．従って，逆生成メソドは書式の数だけ存在する．

逆生成メソドは，基本部とフォーマット依存部に分類することができる．基本部はインスタンスを参照し，フォーマット依存部をその値を引渡しながら制御する．ここで，制御とはプログラムの実行の順番を決定していくものである．フォーマット依存部は，所定の書式の中に，引き渡された値を埋め込んでいくような形式でドキュメントを出力をする．また，基本部はクラス／グループ毎に存在し，フォーマット依存部は，クラス／グループに対するドキュメントの種類毎に存在する．

実際の処理構成を示したのが図1である．指定したクラスのフォーマット依存部の中で，そのクラスのREF関係，BMO関係，SPC関係で結ばれているクラスの基本部を順次呼び出し，それぞれ，そのクラスのフォーマット依存部を制御する．従って，逆生成処理は，階層的に書式の情報を呼び出す形式をとっている．

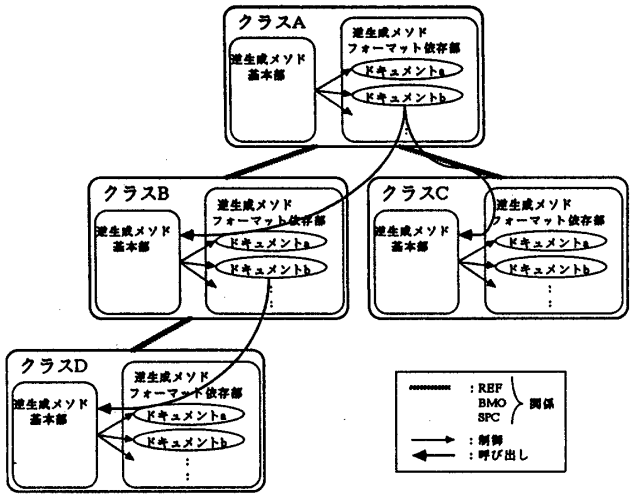

図1　逆生成処理の構成

3．逆生成メタメソドの特徴

逆生成メソドは，図2に示した形式のフォーマット情報定義とクラス情報定義を作成することより生成することができる．この自動生成方式をとることにより，インタープリティブにメタ属性を実行するのに比べて，処理速度を上げることができる．

以下に，処理速度をあげるための逆生成メタメソドの特徴を述べる．

・フォーマットトランスレータ
カラムコンパイル（図2の①）
　ドキュメント出力時のカラム位置制御を行うための処理を，トランスレータ内部で計算し自動生成する．
書式用関数の使用（図2の②）
　定義側での簡単な指定で，フォーマット依存部内の出力用関数を，トランスレータにより記述する．
マップ情報
　各スロットの書式内での使用状況を示す情報である．この情報により，メソド基本部の処理内容の振り分けを行う．

Customizing of the TENSE Object Management System
Taeko Watanabe, Kenji Nishioka
Yokogawa Electric Corporation
tae@kenkai4.yokogawa.co.jp

・インスタンス参照メソドの自動生成

逆生成メソドでの呼び出しルーチン

ソフトウェアデータベース管理処理群内部で使用するルーチンにもメタメソドによって自動生成するものがある．インスタンス参照メソドは，逆生成メソド内で呼び出すルーチンであり，データベースよりインスタンスをとってくるルーチンである．

上位継承

インスタンス参照メソドでは，ＡＫＯ上位からの継承によってデーモン処理を行うが，この継承処理自体は，メソド生成時に，メタメソドの中で部分的に予め行うため，メソドを起動する度に，継承を起こす処理を省いている．

4．ソフトウェアデータベース管理処理の構成

ソフトウェアデータベース管理処理群は，図３のように，６個のツールにより構築している．図中の円形が各ツールである．これらのツールが，各クラスのデータとしてメタ属性とデフォルト値を入力することで，格納メソド，逆生成メソド，ＢＭＯサーチ，インスタンス参照メソドを出力する．

5．おわりに

TENSEでは，データベース管理処理群を自動生成するカスタマイズ方式をとることにより，応答性を高めている．また，多様なソフトウェアプロセスやドキュメントの書式にも容易に対応できる．今後は，有用な上流ツールの組み込みを予定している．

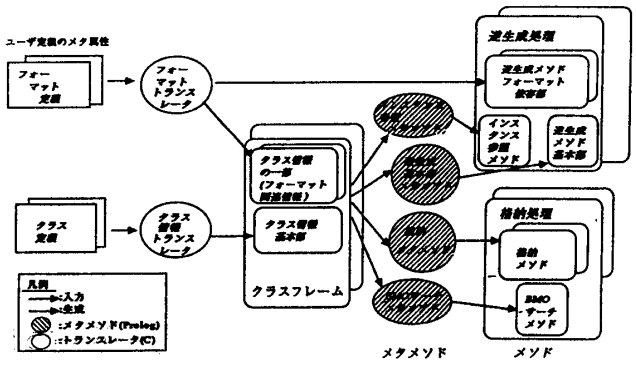

図３　メタメソドの構成

【参考文献】

1) 渡邊，西岡 "TENSE:ｿﾌﾄｳｪｱﾃﾞｰﾀﾍﾞｰｽでﾌﾟﾛｸﾞﾗﾐﾝｸﾞの整合性を保障するC言語向けCASE環境"，INAP'90「PROLOG産業　応用ｼﾝﾎﾟｼﾞｳﾑ」論文集，6-4 (1990)

2) 西岡，渡邊 "TENSEにおけるｵﾌﾞｼﾞｪｸﾄ管 理ｼｽﾃﾑの環境"，情報処理学会第42回全国大会予稿集 (1991)

ユーザ指定情報

フォーマット情報定義書式例

```
        fout(BMOtypedef).

^typedef(,,[0,[-$upper,$comment,#REFdatatype]]){
 title:{ "/* ***  データタイプ定義  *** */".
 }
 body:{ 'typedef ' typeout(#REFdatatype,#name) ':'-33'
        /* '  $comment ' */'.                    ①
 }                    ②
```

クラス情報定義書式例

```
^typedef:1,   2:   ユーザデータタイプ（グローバル）(r
                   /* グループ：1，スロット：2；
     AKO:^logical  /* 上位：論理情報，下位：なし
     BMO:^forModule /* 上位：仮想モジュール，下位：
      :1           /* ローカル：1  */
     (basic :2,基本   /* スロット：2，基本グループ
      storeReflect:  ^mSource;
      ($comment      :%slist        /* コメント
      )
```

自動生成

データベースに格納されているインスタンス情報

```
'data2 '('^typedef','asrc1 ', '通常型データタイプ','int ').

'PF1 '('^typedef','asrc1 ', '関数ポインタ ','int (*_)() ').
```

入力

逆生成メソド

フォーマット依存部

```
'^typedef'('@fout',BMOtypedef,srcEntity,_,[Name|_])

'^typedef'('@title','srcEntity',[U1,U2,U3|_]):-
write('/* ***  データタイプ定義  *** */'),nl,!.
'^typedef'('@body','srcEntity',Name,Upper,Comment,R
write('typedef '),
typeout(REFdatatype,Name,9,L1),
write(' '),(33>L1,L2 is 33,tab(33-L1);L2 is L1),
write('/* '),
sout(Comment,L2+4,L3),②
write(' */'),nl,!.
'^typedef'('@tail','srcEntity',[U1,U2,U3|_]).
```

基本部

```
'^typedef'('@bodyOut',Format,Name,[U1,U2,U3|_],[Uppe
        '^typedef'('@irefer',Name,Upper1,Comment,REF
        '^typedef'('@body',Format,Name,Upper1,Commen

'^typedef'('@bodyOut',Format,Name,_,_) :-
        syntheWarrning(Format,'^typedef',Name).


'^typedef'('@irefer',Name,Upper1,Comment,REFdatatype
        Xinst =..[Name,'^typedef',Upper1,Comment,REF
        call(Xinst)
        ; StateVar = ng.
```

出力

生成されたドキュメント

```
/* ***  データタイプ定義  *** */
typedef int data2;              /* 通常型データタイプ*/
typedef int (*PF1)() ;          /* 関数ポインタ */
```

図２　逆生成メソドの生成処理