

## プログラム開発支援システム C++base の構成

2N-5

鈴木 竜生 久保田 耕司 中村 素典 大久保 英嗣 大野 豊  
立命館大学理工学部情報学科

## 1 はじめに

ソフトウェア開発のライフサイクルにおける各フェーズで生成される情報は、単にそれぞれのフェーズでのみ使用されており、フェーズ間での各種情報の効果的な利用が行われていないのが現状である。また、再利用の単位としてはオブジェクトコードレベルのものが多く、変更や拡張をユーザが行うことは困難である。

本稿では、我々が研究・開発を行っているプログラム開発支援システム C++base の構成について述べる。C++base はソフトウェアライフサイクル内の、コーディング、テスト、デバッグ、運用、保守、再利用の支援を目的としている。C++base では、C++ で記述されたソースプログラムから各種情報を抽出し、それらの情報をデータベースに格納する。現在、データベース管理システム (DBMS) としてオブジェクト指向 DBMS である VERSANT を利用している。ユーザは C++base に問い合わせでデータベースに格納された情報を利用することにより、効果的なソフトウェア開発を行うことができる。

## 2 C++base の構成

C++base は、図 1 に示すようにユーザインタフェース、パーザ及びデータベースインタフェースからなる。これらはそれぞれ、ユーザとのやりとり、ソースプログラムからの情報抽出、データベースへのアクセスを主に行う。

## 2.1 ユーザインタフェース

ユーザは、本インタフェースを介してデータベースに格納された情報を利用することができる。ユーザインタフェースは OSF/Motif を用いて構築されており、以下の 3 つから構成される。

A Structure of Program Development Support System  
C++base  
Tatsuo SUZUKI, Kouji KUBOTA, Motonori NAKAMURA, Eiji OKUBO, Yutaka OHNO  
Department of Computer Science, Faculty of Science and Engineering, Ritsumeikan University

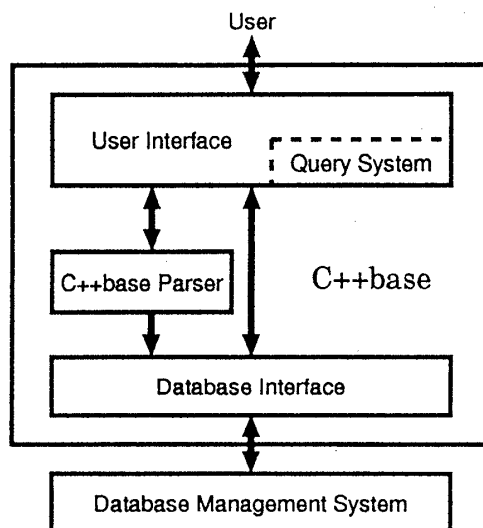


図 1 C++base の構成

- (1) ユーザインタフェースマネージャ  
データベースに格納された情報の問い合わせ要求を受け付け、処理の結果を作成する。
- (2) ウィンドウマネージャ  
ウィンドウの描画や消去などの機能を提供する。ユーザインタフェースマネージャはウィンドウマネージャに依頼を行い、処理の結果を表示する。
- (3) 問い合わせ処理部  
各種情報の問い合わせはユーザが記述しなければならない。問い合わせ処理部では、ユーザの記述をデータベースインタフェースの形式に変換する。なお、基本的な問い合わせに関しては C++base 側で標準にサポートしている。ユーザの記述により、宣言・定義といったシンボル情報、プログラム内の制御の流れや変数間のデータの流れといったフロー情報、その他、継承関係や関数間の関係などが得られる。

## 2.2 データベースインタフェース

C++base は DBMS に依存しない形態をとっているため、データベースインタフェースで C++base

の形式から DBMS の形式へ変換を行っている。データベースインタフェースは、トランザクション管理部、オブジェクト管理部、バージョン管理部の3つから構成される。

(1) トランザクション管理部

トランザクションの開始・終了、データベースの接続・切り離し、コミット・ロールバックなどを行う。

(2) オブジェクト管理部

複数のオブジェクトに対してロック、チェックイン・チェックアウト、演算などを行う。また1つのオブジェクトに対して属性の変更や参照などを行う。

(3) バージョン管理部

オブジェクトの親子関係を管理する。バージョンの生成・消去、オブジェクトの履歴などを管理する。

## 2.3 パーザ

パーザはユーザが記述したソースプログラムから情報を抽出する。C++baseでは、言語文法をBNFで置き換え、各終端・非終端記号を情報抽出の単位としている。パーザ自体も yacc を利用して、BNF にそって記述される。抽出を行う部分は、yacc の記述の中で各終端・非終端記号に対して操作を行うアクション部を利用している。

## 3 C++base におけるスキーマ

我々が使用している DBMS である VERSANT は C++ の class を用いてスキーマを記述することができる。例として、図 2(a) のような言語文法の一部を考える。図 2(a) は (b) に置き換えることができる。

```

statement : block
          | ID '=' expression
          | if_statement else_statement      (a)
          ;

statement : statement1
          | statement2
          | statement3
          ;                                  (b)

statement1 : block ;
statement2 : ID '=' expression ;
statement3 : if_statement else_statement ;

```

図 2 BNF の例

したがって、statement は statement1, statement2,

statement3 のいずれかからなる (is-a の関係) といえ、statement3 は if\_statement と else\_statement という非終端記号を包含する (has-a の関係) といえる。BNF の記述からスキーマを得るために、次のような操作を行う。

(1) 左辺の非終端記号に対して右辺が1つだけの場合は、左辺のクラスは右辺の構成要素を属性として持たせる。右辺が2つ以上ならば、各右辺を1つのクラスとしてまとめる。

(2) 右辺は左辺のクラスを継承したクラスとする。

図 2(b) から生成したスキーマを図 3 に示す。

```

class statement {
    ...
};

class statement1 : public statement {
    class block *block_ptr ;
    ...
};

class statement2 : public statement {
    class id *id_ptr ;
    class expression *expression_ptr ;
    ...
};

class statement3 : public statement {
    class if_statement *if_ptr ;
    class else_statement *else_ptr ;
    ...
};

```

図 3 スキーマの例

パーザは各非終端記号ごとに図 3 の形式でインスタンスを生成し、データベースに格納する。

## 4 おわりに

本稿では、C++ のプログラム開発支援システムである C++base の構成について述べた。現在、パーザ及びデータベースインタフェースを中心に実装を行っている。今後は問い合わせ処理部の実装を行いながら、各種問い合わせ機能の詳細を検討していきたい。

## 参考文献

- [1] 白井謙良: オブジェクト指向ソフトウェアリポジトリ C++base の設計, 立命館大学大学院理工学研究科情報工学専攻修士論文 (1993).