

## データベースを利用したC言語プログラム開発支援システムCbaseの実現

## 5M-10

向井 雅樹 大久保 英嗣 津田 孝夫

(京都大学工学部)

## 1. はじめに

ソフトウェアの開発においては、設計、実現、テスト、および保守といったソフトウェアのライフサイクルの各段階で、様々な情報の授受が行われる。すなわち、ソースプログラム以外に、設計文書の記述、実現の際のソースプログラムの編集履歴、テスト結果、および保守の履歴といった情報である。従来のソフトウェア開発手法の多くは、これらの情報を単にそれぞれの段階でのみ使用しており、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。このような状況において、我々はライフサイクルの各段階で、種々の情報を抽出し、データベース・システムを用いてそれらを統合的に利用することによって、効率的なソフトウェア開発の支援を行うことを考えている。

本稿では、データベースを用いたソフトウェアの開発支援手法を提案し、我々が現在、開発を進めているデータベースを用いたC言語によるソフトウェア開発のための処理系Cbase(C program base)について述べる。

## 2. Cbaseの概要

Cbaseでは、ソフトウェア開発支援のため、以下に示す2つの考え方を採用している。

- (1) プログラムのモジュール化を支援するため、開発言語(C言語)にモジュール仕様を記述する部分(仕様部と呼ぶ)を付加し、それらの仕様から、モジュール相互間の資源の輸出入、あるいはモジュール階層等に関する各種の検証を行う。
- (2) ソフトウェア開発の過程で得られる種々の情報をデータベースに蓄積し、それを利用することにより、ソフトウェア・ライフサイクルの各段階の移行を円滑にする。

(1)の考え方は、従来のモジュール化の概念そのものであり、各モジュールが他のモジュールをアクセスする場合に制限を課すことを意味する。この考え方は、プログラムを構成する各部分の変更の他への影響を少なくするもので、大規模ソフトウェアの構築には不可欠である。さらに、完成したモジュールは、部品化によって再利用が可能となる。(2)の考え方は、従来のソフトウェア開発では記録されなかった種々の情報をデータベースに蓄え、利用しようというものである。ソフトウェアの開発過程では、最終的に目標とするプログラムのほかに、開

発者と開発システム間、開発者相互間、あるいはソースプログラム編集用エディタ、コンパイラといったソフトウェア開発ツール内で以下に示すような情報の授受が行われる。

- (a) 設計段階における、モジュール間の関係およびモジュールの内容が記述された設計文書
- (b) 実現段階における、プログラムの編集の履歴、あるいは、コンパイラによって生成され、利用されるシンボルテーブル、構文木等のデータ、およびコンパイル・エラーの記録
- (c) テスト段階における、テスト環境の設定、およびテスト結果
- (d) 保守段階における、プログラムの更新の履歴

従来のソフトウェア開発においては、以上のような情報は、開発されるプログラムとは独立した文書によって記録され、参照されるという手続きをとるものが多く、中には全く記録されずに捨てられるもの(例えば、コンパイラ内でのみ生成され、利用される情報)さえあった。我々のアプローチは、これら種々の情報のうち必要なものをデータベースに蓄積し、利用することによってソフトウェア開発を支援する環境を構築しようというものである。

我々は、以上の考え方をC言語によるソフトウェア開発に適用することを目的として、開発支援システムCbaseの開発を行っている。現在、設計、実現およびテストの段階について、データベースに情報を蓄積し利用する環境の構築を進めている。

## 3. 開発支援手法

我々が開発を進めているシステムCbaseは、ソフトウェア開発の各段階でデータベースを統合的に利用することを目標としているが、各段階で以下のような情報の抽出、データベース化、および情報利用の方法が考えられる。

## 3.1 設計段階

モジュール化プログラミングを支援するために、C言語に仕様部を追加し、設計者が各モジュール内の情報およびモジュール間の関係等を記述し、データベースに登録する機構を与える。これにより、モジュール間の資源の輸出入を制限し、プログラムのモジュール化あるいは抽象化を推進する。また、データベース中の情報を利用することにより、以下のことが可能となる。

(a) モジュール間の資源の輸出入等に関する整合性の検証を行う。

(b) 各モジュールのプログラマ間の意志疎通の手段を提供する。

さらに、仕様部の内容から設計文書作成の支援も行える。仕様部の内容はプリプロセッサにより抽出され、データベースに登録される。仕様部に記述される内容としては、モジュール名、作成者名、作成日といった当該モジュールに関する情報のほか、他モジュールからの資源の輸入、他モジュールへの資源の輸出、あるいはモジュール階層における当該モジュールの位置づけ等が記述される。資源の輸出入許可の宣言を仕様部で行うことにより、仕様部以外での資源の輸入(extern宣言)を禁止し、設計者に対してモジュール化を規律付けることができる。

### 3.2 実現・テスト段階

実現については、仕様部で得られた情報をもとに、ソースプログラムのコーディングを支援する。また、ここでは本段階以降で利用するために、ソースプログラムより各種の情報を抽出し、データベースに蓄積しなければならない。そのため、コンパイラに必要な情報を抽出するステートメントを埋め込む必要がある。これにより、前節で述べた仕様部の情報とあわせて、モジュール間の資源の輸出入等に関する整合性の検証、あるいはプログラムの一部変更による波及効果を受ける部分の検出、さらにはその部分の自動変更等も可能になるものと考えられる。また、得られた情報のインタプリタあるいはデバッガにおける利用も可能である。

また、データベースに蓄積された情報を用いることにより、モジュールごとの単体テスト、あるいはその一部のみで独立したテストを行うための環境が設定可能となり、迅速なテストが行える。

実現・テスト段階の支援としては、エディタを中心に据え、その中から各種のツールを起動するという方式を採用している。これにより、編集、コンパイル、テストが随時、容易に行える。

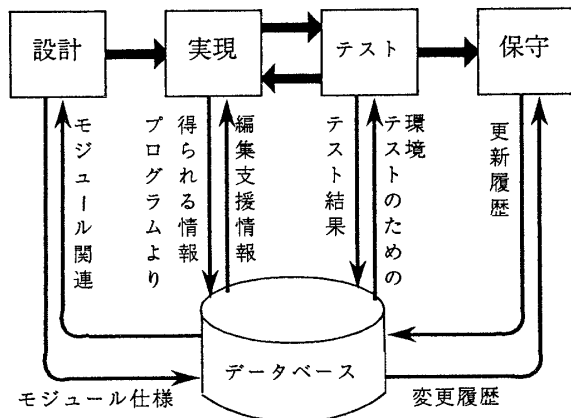


図1 開発過程で抽出および利用される情報

### 3.3 保守段階

ソースプログラムの変更による他のモジュール、あるいは文書への波及効果の検出あるいは自動修正等が行える。また、データベースを用いてバージョン管理を行うことにより、ソフトウェアシステムの一貫性の制御が行える。

各段階で抽出および利用される情報のデータベースとのやりとりを、図1に示す。

### 4. 開発支援システムCbaseの構成

以上で述べた開発支援システムCbaseはUNIX上で構築され、おもに、

- ・Cコンパイラ (目的プログラムの生成)
- ・データベース・システム (情報の蓄積・操作)
- ・ウィンドウ・システム (ユーザ・インタフェース)

の3つの部分から構成される。データベース・システムとしてUNIX上の関係データベース・システムINGRESを、ウィンドウ・システムとしてX-Windowを、それぞれ用いている。これらを用いることにより、環境の構築が容易になるとともに、優れたユーザ・インタフェースを提供することができる。Cbaseの構成の概略を図2に示す。

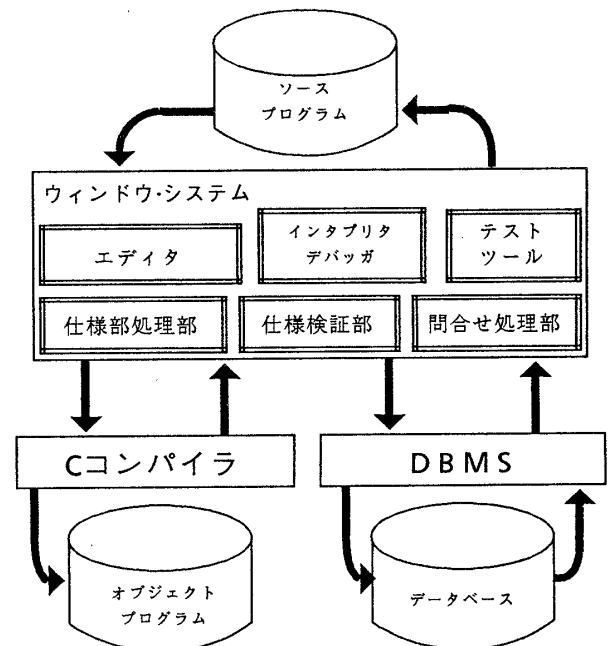


図2 開発支援システムCbaseの構成

### 5. おわりに

現在、おもにソフトウェアの設計、実現段階を対象として、エディタを中心としたツール群の開発を進めている。しかし、ソフトウェア開発のあらゆる段階で統合的にデータベースを利用することによる効果を調べるためには、他の段階を含めた種々の開発支援を行わねばならない。今後、このような観点からの開発支援環境の評価が必要である。