オブジェクト指向によるプログラム自動生成システムの構想

2S－1

川村敏和　光武雄一　飯村和茂

(株)東芝　重電技術研究所

1．まえがき

ソフトウェア開発において生産性の向上は重要な課題である。そこで我々は下記のテーマでプログラム自動生成システムの研究を進めている。

①記述言語の開発
②データ宣言部の自動生成
③ソフト部品の再利用によるプログラム合成

現在、プロセス制御システムのソフトウェアをモデルにして開発を行っている。本稿では、まず現在までの研究経過をまとめ、さらにオブジェクト指向技術を用いたシステムの構想を述べる。

2．ソフトウェア開発の特徴

ソフトウェア開発と一口に言っても、その対象分野や開発組織は多様である。開発方法や抱えている問題点にもそれぞれ特徴がある。

我々がモデルとしているプロセス制御システムのソフトウェア開発は図1の手順で行われている。仕様レベルでは機能仕様書と入出力データ仕様書が作成される点が特徴である。後者は入出力と処理の対応を記述した一覧表である。また設計以降の作業は次の2つの流れに分かれている。

①入出力データ仕様書をテーブルに分割設計し対応するデータ構造をコーディングする。
②テーブル操作ごとに適当なプログラム部品を検索し、修正して再利用する。

つまり、データ部はシステムごとに新規に作成し、処理部は既存部品の再利用を人手で行っているのである。

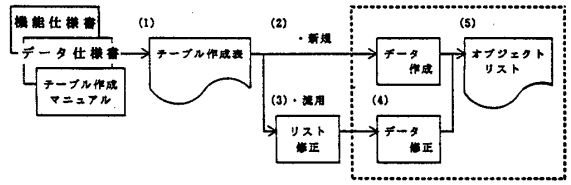

図1　ソフトウェア開発の手順

プログラム自動生成システムの実現方法は様々であるが、我々は上記の特徴から、システムをデータの自動生成部とソフト部品の再利用によるプログラム合成部の2機能に分け、それぞれに最適なデータベースを用いることにした。

3．関係ＤＢによるデータの自動生成

入出力データ仕様書は表形式のデータであり、関係データベースで扱うことができる。従って入出力データ仕様書から各テーブルを作成するために抽出するデータ項目が決まれば、必要なテーブルの生成とテーブルからソースへの変換が自動的に行える。

そこで関係データベースUNIFY を使用してデータベース上の入出力データ仕様書からアセンブリ言語のデータ宣言を生成する機能を試作した。ビットパターンやポインタで関連づけられた情報なども扱え、Ｃ言語のソース生成も可能である。

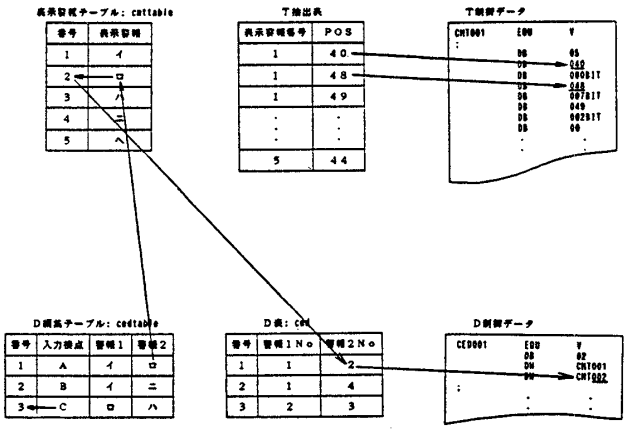

図2テーブルからデータ宣言への変換

4．オブジェクト指向による再利用システム

部品の利用には大きく分けて、

①既存の部品をそのまま使用する
②既存の部品を修正して使用する
③新しく部品を作成して使用する

の3形態がある。部品の修正に関しては、データ

Automatic Program Generation using Object-Oriented Database

Toshikazu Kawamura , Yuuichi Mitutake , kazushige Iimura

TOSHIBA CORPORATION

構造の変更に対応して計算機で自動的に修正できる部分と、計算機のサポートを受けながら人手で修正する部分に分かれる。ところがデータの加工や伝送が処理の中心である場合には、アルゴリズムはデータ構造に強く依存する。しかもデータ構造がシステムごとに大きく異なる場合には、「②＋人手による修正」の部分が多くなり、自動化が進めにくい大きな要因になっている。

部品の検索と修正を効率化するためには、必要な情報を簡単にかつ即座に得られる環境が必要である。情報としては、機能仕様書や入出力データ仕様書、処理部やデータ宣言部のソースなどがあり、これらの情報は互いに関連がある。

上記を整理すると下記機能が必要である。

・部品の登録・検索が容易にできること

・登録した部品の修正が容易にできること

・マルチメディアに対応できること

これらをオブジェクト指向データベースを用いたシステムとして実現する。システムでは、仕様書やデータ構造、ソフトウェア部品などすべてをオブジェクトとして扱う。ひとつの手続きは、仕様書とソースとデータ構造のオブジェクトで構成され、クラスとして登録される（図３）。ある機能単位は複数の手続きがリンクで結ばれたスーパークラスである。さらに、ひとつのアプリケーションシステムは、複数の機能単位で構成される。

再利用においては、まず必要なクラスを検索してそのインスタンスを生成し、修正を施して使用する。部品化したい場合にはサブクラスとして登録する。

既にＨｙｐｅｒＣａｒｄを用いて、評価用のモデルを作成し、上記の概念でシステムが実現できることを確認した。

５．機能仕様記述言語

記述言語は、従来のようなプログラム記述レベルではなく、オブジェクトを使ってシステムを記述するためのシステム記述言語として位置づける。図４は機能仕様記述からプログラムへの変換の概念を示したものである。

６．あとがき

今後は、実際の複雑なシステムがこの枠組みの中で表現できるかを検討する。同時に既存の部品はリニアテキストであり、膨大なリニアテキストをハイパーテキストに変換する方法の検討も必要である。

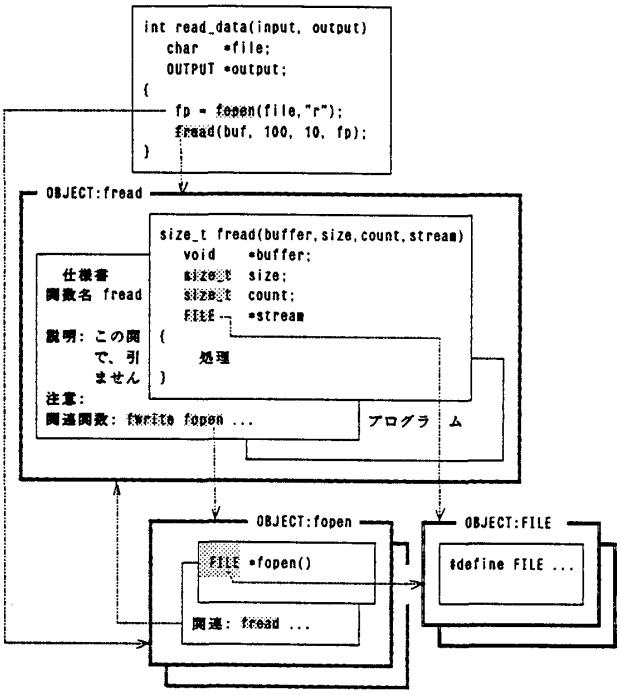

図３　ソフトウェア部品の管理形態

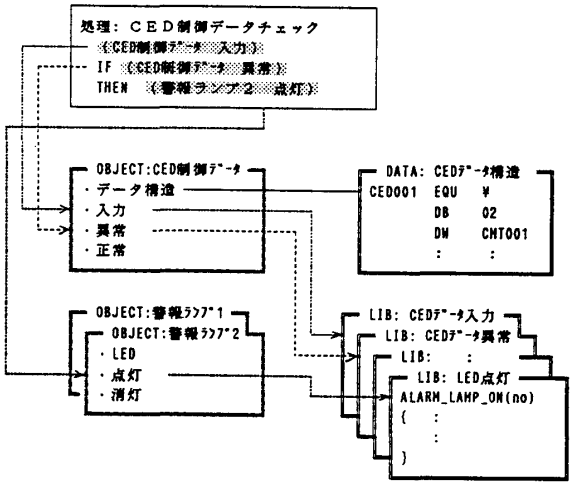

図４　仕様記述からプログラムへの変換概念