

電気系DAに於けるデータベースマネージャを対象とした 自動プログラミング技術の応用について

中村 琢八、 白石 洋一
群馬大学 工学部 情報工学科

ソフトウェアの開発に対し、部品化と再利用は生産性を向上させると言われている。しかし、対象となるアプリケーションが大規模化してくると、なかなか実用的なアプリケーションに対応できない。そこで、本稿では大規模なアプリケーションに対し、まず人間の知識がほとんど介在しない単純処理のプログラムを自動生成することが有効ではないかと考え、電気系DAに於けるデータベースマネージャを対象として、データベースのデータ仕様からそのデータを管理するデータベースマネージャを自動生成するツールを開発した。本ツールを使用して実際のデータベースマネージャを生成した結果、その設計開発工数を大幅に減らすことができた。

Program Synthesis Methods based on Software Component Re-use for a Database Manager in an EDA

Takuya NAKAMURA, Yoichi SHIRAIISHI
Department of Computer Science, Gunma University
E-mail: { takuya, siraisi }@keim.cs.gunma-u.ac.jp

The program synthesis based on the software component re-use is considered as one of the efficient ways to increase the software productivity. This synthesizer has made a certain amount of success for small-scale applications, however, it has much difficulty in coping with the real large-scale software systems. Therefore, an approach to develop a program synthesizer for large-scale software systems is adopted from the beginning and the methods for synthesizing their simple sub-programs requiring scarcely no human techniques are actually investigated. The database manager in the actual electronic design automation system is the first target of the newly suggested program synthesizer. Experimental results show the great reduction of the manpower in that database manager development.

1.はじめに

今日、ソフトウェアの生産性を向上させるために、ソースコードの部品化と再利用が重要視されている。そして、現在この分野で様々な研究がなされているが、対象となるシステムやアプリケーションが大規模化、複雑化してくるに従い、部品化されるソースコードが複雑になってきたり、部品の種類がかなり多くなってきたりして、以下のような問題に対して試行錯誤している¹⁾。

- どの部品を選んだら良いか。
- 選んだ部品をどのようにカスタマイズしたら良いか。

しかも、ある程度の小さなモデルに対しては対応できても、なかなか大規模で実用的なアプリケーションには対応できない。

そこで、大規模で実用的なアプリケーションに対して、まず、なるべく人の知識を必要としない部分のプログラム開発を自動化すると効果的ではないかと考えた。

特に、アプリケーションが大規模になると、それに伴ないデータ数がかなり多くなる。そして、多発するデータの修正、変更に対してデータの管理部分を人手で修正、変更するのは単純ではあるが、バグが発生しやすく多くの工数、期間を要する作業である。

一方、データを管理する側から見ると、あらかじめデータの仕様を決めておけば、決定した仕様通りにデータ管理プログラムを作成すれば良いので、ほとんど人の知識を必要としない。

要するに、大規模なアプリケーション

に対しても、データとデータの仕様を決定しておけば、部品を再利用することにより、データ管理プログラムを自動的に作成できるのではないかと考えた。

本研究では大規模化している電気系 D A (Design Automation)システムに対して、D A で扱うデータ(データベース、以下DB)とデータ仕様を与えることにより、そのDBを管理するデータベースマネージャ(以下DBM)を自動生成するツールを開発した。

2.D AシステムにおけるDBM

2.1.システム構成

V L S Iチップの設計工程の概要²⁾と、レイアウト設計の処理フローを図 1に示す。

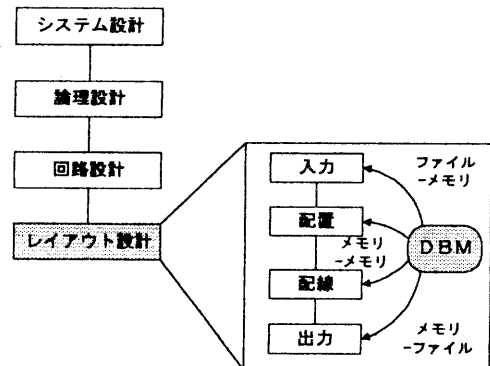


図 1：システム構成

まずV L S Iの仕様を決定する。システム設計では、アーキテクチャの設計、モジュール設計とその検証を行う。即ち、V L S I設計の仕様を満たすアーキテクチャの設計である。

次に論理設計を行う。アーキテクチャ設計の結果が論理要素に、即ち、ゲート、フリップフロップ、マルチプレクサ、等

の要素とその結線関係まで変換される。そして論理シミュレーションにより、論理設計の論理エラー、タイミングエラー、等を発見する。

回路設計において、論理設計結果はトランジスタレベルの回路に変換される。そして、回路シミュレータによって回路レベルで所望の論理が実現できているかどうかをチェックする。

レイアウト設計は、論理図と部品の情報を入力として、部品の配置と部品間の配線を行い、チップのレイアウトパターンを出力する。ここでは、チップ面積と遅延時間等の特性を最適化する。

レイアウト設計におけるDBMの役割は、入力部でデータをファイルからメモリにロードし、配置、配線処理に対してメモリ間のデータのやり取りを統一化するインタフェースを与え、出力部でメモリの内容をファイルにセーブすることである。

2.2.DBMの機能

DBMの機能の詳細を図2に示す。

ここで、意味的にまとまりのあるデータ群をテーブルと呼ぶ。そして、DBをメモリ上に読み込んだテーブル群は、配置、配線で共通に扱われるので共通テーブルと呼ぶ。それに対し、各配置、配線プログラムでローカルに用いる(メモリ上の)テーブルをワークテーブルと呼ぶ。

図2中の(1)、(2)、(3)に対応する機能を以下に説明する。

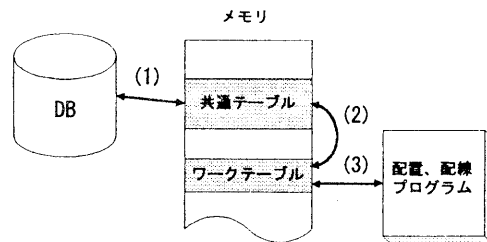


図2:DBMの機能

(1) DBとメモリとのファイル入出力

DBの全テーブルをメモリにロードしたり、共通テーブルをDBにセーブしたりする機能である。

(2) 共通テーブルとワークテーブルとの入出力

配置、配線プログラムは、基本的に直接共通テーブルにアクセスせず、共通テーブル内の必要なデータを一旦ワークテーブルにコピーし、ワークテーブルにアクセスして処理を行う。そして処理終了後、ワークテーブルの内容を共通テーブルへ書き込む。よって、共通テーブルとワークテーブルとの間でデータの読み書き等を行う機能である。

(3) ワークテーブルとプログラムとのインタフェース

配置、配線プログラム内でワークテーブルへの読み書き等を行う機能である。

2.3.DBの現状

電気系DAシステムのDBは巨大で、かつ複雑にデータ同士が相互参照している。現在、開発しているシステムのDBの規模は、テーブル数が90で、テーブル内の全てのデータ項目数は415である(表1参照)。

表 1：DBの規模

テーブル数	90
データ項目数	415

本研究で対象としているDBMは、各テーブルにデータの読み書き等の基本的な操作を行うメソッドを付随させたオブジェクト指向的なDBMであり、現段階では全てのテーブルに同種のメソッドを付随させている^[3]。この時、一つのテーブルに対するDBMプログラムの規模は、約0.8 Kstepとなり、90のテーブルに対するDBMを作成しようとする単純に計算して約72 Kstepとなる(表2参照)。

表 2：DBMの規模

1テーブル	約0.8 Kstep
全テーブル	約72 Kstep

このように大規模なDBMプログラムになるが、実際には各テーブルは類似した基本処理を持ち、更に各処理は非常に単純なものになる。

しかし、これを全て人手で作成しようとする、人の知識をほとんど必要としないが作業が複雑で、多大な設計工数を要する。また、システムの開発段階では、DBのテーブルやデータの種類の追加、削除、修正が頻繁に起こる。このような度毎にDBを管理するDBMプログラム

を人手で修正していると、バグ等が介入する可能性があり、多くの保守コストが必要になる。

2.4.自動生成による考え方

前節のDBの現状に対し、設計工数と保守コスト削減のため、「DBのデータ仕様のみから、DBMプログラムを自動生成する」問題を考える。

入力データとなるテーブル仕様は、構造データを表すのに比較的分かり易いC言語の構造体の形で与え、出力されるDBMプログラムは、オブジェクト指向的に作成できるようにC++言語とする。

つまり、C言語の構造体から、構造体や構造体内のデータにそれらの読み書き等を行うメソッドを付随させたC++言語のクラスと全テーブルのファイル入出力プログラムを自動生成する(図3参照)。

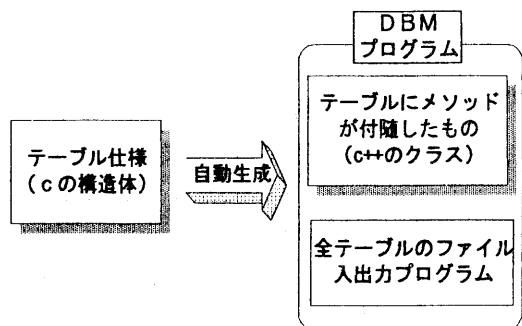


図 3：DBMの自動生成

3.DBMの自動生成手法

一般的に自動プログラミング技術は、プログラム変換による方法、論理による方法、帰納推論による方法、部品合成による方法、等に分類されるが、ここでは

手続き型やオブジェクト指向型言語による大規模なプログラムに対して優れている部品合成による方法を用いる¹⁴⁾。

3.1. 部品合成による方法

本研究における部品合成による方法を図 4 に示す。

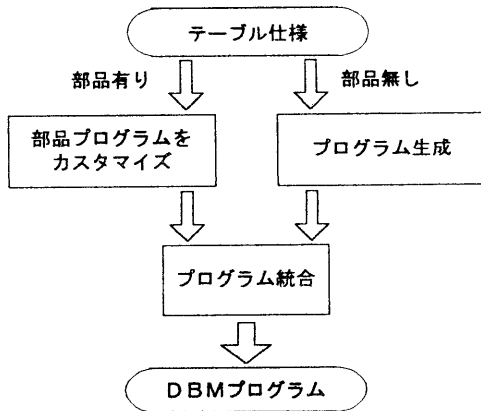


図 4：部品合成による方法

自動生成ツールは、テーブル仕様を入力として DBM プログラムを生成する際に、汎用的に部品化してあるプログラムについてはそのプログラムをカスタマイズし、部品化されていないものについてはツール自身がプログラムを生成する。ここでカスタマイズとは具体的に、部品内の汎用文字列変数に実際の変数を置き換えることである。そして、最後に両者から生成されたプログラムを統合して DBM プログラムが作成される。ここで部品化される汎用的なプログラムは、データ仕様に従って作成する。

この部品合成による方法が、生産性の面で大規模なプログラムに適している理由は、部品プログラムの一部を修正するだけで新しいプログラムを生成すること

ができたり、生成されたプログラム同士を組み合わせたりすることができることによる。

しかし、この方法によって生成されたソースコードの正当性を保証するのは困難である。従って、これらの部品合成で得られたプログラムが要求仕様を満たしていることを保証するには、与えられた要求仕様の全てのバリエーションに対してあらかじめデバッグしておき、部品合成の際に完全に自動でデバッグ済みのプログラムを得る方法が必要である¹⁵⁾。

今回、自動生成する DBM プログラムは、非常に単純な処理の集合なので、自動生成時に使用する部品プログラムのデバッグを完了しておくことにより DBM プログラムの正当性を保証している。

3.2. DBM 自動生成プログラムの開発

2.3 の 90 のテーブルの仕様中 60 のテーブル仕様に対して、DBM 自動生成のための実験プログラムを開発する。ここでは 1 テーブルに対し、20 の部品プログラムを用いる。

図 5 に示すように、テーブル群と部品群から DBM プログラム群を自動生成し、最後にそれらのファイルを統合、分類する。

この作業をテーブル仕様の数だけ繰り返すことにより、それらのテーブル仕様に対する DBM プログラム群が作成される。最後にシェルスクリプトと人手によりそれらの統合、分類作業を行う。

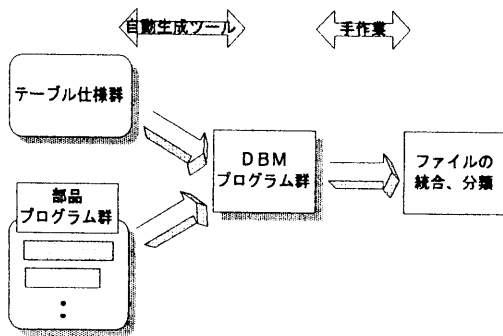


図 5：プログラム開発過程

3.3. DBM自動生成の具体例

自動生成プログラムの入力となるテーブル仕様の一例を図6に示す。

```

struct CompType {
char*      name;
int        spec;
Size       estimate;
Size       result;
SubRect*  subRect;
ExternalPin* externalPin;
:
};

```

図 6：テーブル仕様例

図6のテーブル仕様を入力とした時のDBM自動生成ツールによる自動生成過程の一部を図7に示す。DBMプログラム中の枠で囲まれた部分は、部品プログラムをカスタマイズして作成された箇所、枠で囲まれていない部分は、自動生成ツールにより生成された箇所である。

この場合部品のカスタマイズとは、部品プログラム群中の汎用文字列変数<Name>を実際のテーブル名のCompTypeという文字列に置き換える

ことである。本自動生成ツールは、全部で11種類の文字列変数を使用した。

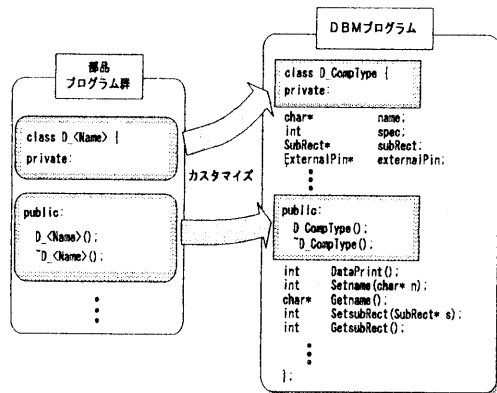


図 7：自動生成過程の具体例

4. DBMプログラムの評価

DBMプログラムを自動生成したときの効果を表3に示す。

表 3：自動生成による効果

DBM開発項目	ステップ数	設計開発工数	
		実験結果	手作成推定
部品作成	約0.8Kstep	1人月	12人月
自動生成による作成	約4.7Kstep	5分	
ファイルの統合、分類	—	2時間	

表3より、60のテーブル分のプログラムを約1人月で作成できることが分かる。しかし、これを人手で行うとすると約12人月かかると推定される。この12人月というのは、自動生成ツールを開発する前に人手でDBMプログラムを開発していたときの所要時間をもとにして計算した結果である。

自動生成ツールが出来てしまえば、60のテーブル分のプログラムは約5分で作成でき、さらに90の全テーブルに

対しても比較的傾きの小さい線形時間で、DBMを作成することができると思われる。

自動生成されたDBMプログラムに修正が必要なときは、部品プログラムを修正し、再度自動生成により作成することができる。生成されたプログラムは、オブジェクト指向言語C++であるので、差分プログラミング等によりプログラムを修正、拡張可能である。

また、本自動生成プログラムは電気系DAに限らず、このような特定のDBを用いるアプリケーションに拡張可能である。

5.まとめ

電気系DAシステムを対象として、DBのデータ仕様から、DBを管理するDBMプログラムを自動生成するツールを開発した。

自動生成には部品合成による方法を用い、入力となるテーブル仕様一つに対し、20の部品プログラムを再利用することにより、そのテーブル仕様に対するDBMプログラムを作成した。

このDBMプログラムに対して、人手による推定設計工数約12人月を自動生成ツールを使用することにより、約1人月と大幅に削減した。

今後の課題を以下に示す。

- 自動生成ツールに対し、生成されたプログラムの統合、分類作業を完全に自動化すること。
- 現段階では、DBMはデータの入出力操作を行う基本機能のみに制

限しているが、データの探索やポインタの辿り等を行う機能を追加すること。更に、入力の各テーブル毎にそのテーブル固有のメソッドを追加、修正できるようにすること。

参考文献

- [1] 大須賀 節男, 李 春野 : ソフトウェア設計自動化に向けて, 人工知能, Vol.6, No.2, pp159-166(1991)
- [2] Sadiq M.Sait, et al.: VLSI Physical Design Automation, IEEE Press, 1995
- [3] 宇田川 佳久 : オブジェクト指向データベースのCADへの応用, 情報処理, Vol.32, No.5, pp586-592(1991)
- [4] 古宮 誠一 : 高品質と高生産性を確保する方法, 人工知能, Vol.6, No.2, pp181-183(1991)
- [5] 古宮 誠一, 原田 実 : 部品合成による自動プログラミング, 情報処理, Vol.28, No.10, pp1329-1345(1987)