

モジュール部品を用いたソースコード自動合成システム

1F-6

小尾 俊之 松村 一夫

(株式会社 東芝 システム・ソフトウェア技術推進部)

1. はじめに

ソフトウェア生産の工業化を目指すIMAPシステムの開発を行なっている[1]。そのサブシステムとして、ソフトウェアのモジュール設計・製造・組立てを支援する50 SM支援システムがあり[1,2]、ここでは設計記述法TFFで設計した設計仕様書から、モジュール部品等を用いてプログラムを自動合成することを最終目標としている。

部品を用いて合成する支援システムの前段階として、TFFで設計された仕様書にTFFコード(詳しくは後述)を入力し、それ以降、モジュールのソースコード生成、コンパイル、リンクを行ない、ロードモジュールまで自動生成するシステムTCGEN-I(TFF to Code GENerator-I)は既に開発されている[3]。

本稿では、特に部品を用いてプログラムを生成していく際の部品の組込み方法について問題点を整理し、その問題点を解決するため、①部品とのインターフェースを自動的にとる、②部品の組込みによって生じる設計の変更指針を提示する、などの特徴を持った知的な部品組込み・合成支援システムTCGEN-IIについて述べる。

2. TCGENの位置づけ

TCGENはTFFで設計された設計仕様書からモジュール部品を利用して、自動プログラミング(プログラムの作成過程で、人間が通常行なう仕事の一部または多くの部分をコンピュータが肩がわりすること(Balzer))することを目的としたツールである。

TCGENの構成(図1)

これまでTCGEN-Iとして次の機能を開発してきた。

(1) TFFコードの編集(TFF設計書と対応した構造の中

の処理記述及び条件記述をプログラミング言語(

記述したもの(TFFコード)の編集)(図1の①②)

(2) 設計情報からの自動合成(モジュールインターフェース定義のソースコード(図1の③)、制御構造から制御ステートメント(図1の④)、データ宣言・定義ステートメント(図1の⑤⑥))

(3) 設計情報からモジュールのリンクの自動化(共通データ部と処理部の各ソースコードのマージ、分割コンパイル、リンク)(図1の⑦)

さらに今後TCGEN-IIとして、次の機能がある。

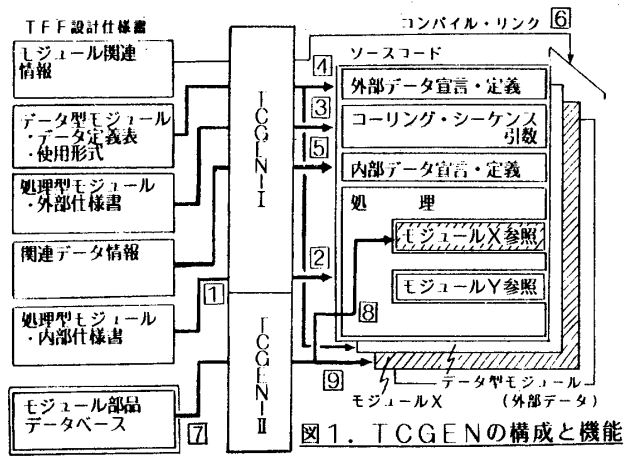
(4) モジュール部品の組込み・合成の自動化

組込み・合成の定義: 組込み・合成とは、プログラムの作成(設計、コーディング)の過程で、事前に準備されたモジュール部品(特定のプログラムのためだけでなく、他の類似プログラムでも使用することを目的として作られた再利用のためのモジュール。部品作成部門が作成し、品質保証する)から、使えそうなものを見つけ出し(図1の⑦)、自分のプログラムの一部として、自分が作成した他の部分と組合せて(図1の⑧⑨)、目的の機能・処理で動かせるように作り上げることである。

なお、モジュール部品を組込み合成する際、部品の内部を修正せず使用する方法(Black Box部品。以下B.B.部品と略す)と、内部を修正して使用する方法(White Box部品。以下W.B.部品と略す)とがあるが、TCGEN-IIの範囲では、W.B.部品は扱わない(別途、報告する)。また、部品の検索機能については文献[4]にゆずるとし、使えそうな部品を見つけた後の自動化支援とする。

3. TCGEN-Iの効果

TCGEN-Iではモジュール毎に、TFFの設計に従って処理や条件の部分をTFFコードで記述すると(図2参照)、あとはロードモジュールまで自動的に作成するので、今まで手作業で行なっていたのに比べ生産性が向上している。具体的には、ソースコードの制御構造に関するエラーが大幅に減少した、モジュールの分割コンパイルやリンクのためのJCLやUNIX makeファイル作りに相当する作業が自動化された、設計仕様書とソースコードが完全に対応づけて編集管理でき、保守、再利用し易い、等があげられる。しかし、モジュールから他のモジュールを参照する(callする)様な場合はそのインターフェースはプログラマが確認し、TFFコードで記述しているため



効率が悪い。TCGEN-IIでは、この問題に注目しモジュール部品の組込みの部分の自動化をはかる。

4. TCGEN-II

4.1 モジュール部品の組込み・合成の課題

標準モジュール部品が管理されているライブラリからモジュール部品を使用する際に問題となる点を整理すると次の2つに大別できる。

- (1)部品とのインターフェースの不一致
 - a.モジュールネームの重複
 - b.引数の型、順序、数の不一致
 - c.入力情報(引数、共通データ)の値の範囲不一致
 - d.出力情報(引数、共通データ、戻り値)の値の意味と扱い方の不一致
 - ・エラーとして扱うか、正常値として扱うか
 - ・値のレンジ変換
 - e.共通データ名の重複
 - f.下位モジュール(使用ライブラリ等)の暗黙使用
- (2)部品の世界の約束と自分のプログラムの世界の約束の相違
 - g.モジュール名、変数名などネーミング規則
 - h.意味体系(例えば、EOFを-1と扱うか、0と扱うか)
 - i.OS、言語の制約(対象とするOSや言語が異なる場合、等)
 - j.機能優先、性能優先(メモリサイズ、特殊なアルゴリズムの使用、等)
 - k.作り方(コメント、引数の数や順番)の規則

g、hは(1)のインターフェースの問題に帰着できるものが多い。iの解決には言語変換や異言語とのインターフェース修正が必要であり、jでも部品内部の修正が必要となり、これらはW.B.部品の課題といえる。

TCGEN-IIでは、このうちa、bとdの一部、fについ

て支援する。特に次節でa、bのモジュール名、引数名、dの戻り値の一致をとる方法と、部品を組込んだ時の設計の変更について述べる。

4.2 引数、戻り値の統一

モジュール部品の組込みの際の引数の一致について例で示す。図3aの様な内部仕様書がある場合、ここにgetword というモジュール部品を組込みたい。getwordの外部仕様書の引数の情報は図3bの様になっている。一方getwordを参照する側(main)の変数については図3cの様な情報がある。これらの型から対応する変数を限定し、続いて表題及び説明に記述されている内容を解析してキーワードを抽出し対応をとる。対応がとれた所でシステムから図3dの様な確認の画面が表示され、ユーザはそれを見て確認、あるいは必要な場合は直接そこから修正する。この様にして全ての引数、戻り値について統一をとる。

4.3 設計の変更

モジュール部品を組込む際、その部品の前後で行なわれる標準的な処理がある場合は、部品はこれらの処理パターン情報や追加すべき位置などの知識を持っており、組込まれた時その処理も設計仕様書に追加される。例えば、部品の前後で初期処理やエラーチェック処理が必要な場合などはその追加する必要があることを明示し、ユーザからの指示があった場合にはそれを仕様書上に追加してソースコードにおける対応をとる。

5. おわりに

TCGEN-IIは、現在プロトタイプを開発中であり、モジュール部品の組込みに関しては、システムとの対話によって若干の情報の不足分を補うことが必要となっている。

今後は、このプロトタイプを完成させるとともに、システムとの対話に代わる知識をシステム内部に持たせ完全自動化へ近づけていく予定である。

参考文献

- [1] 大澤他「IMAPシステム(1)~(10)」全10冊 情報処理31回大会
- [2] 松村他「ソフトウェア設計記述法」東芝レビュー 8月号、1986
- [3] 伊豆野他「コード生成ツール(TCGEN)の開発-設計記述法1+1からソースコードへの変換-」情報処理32回大会
- [4] 石川他「ソフトウェア部品検索システムの開発」情報処理32回大会

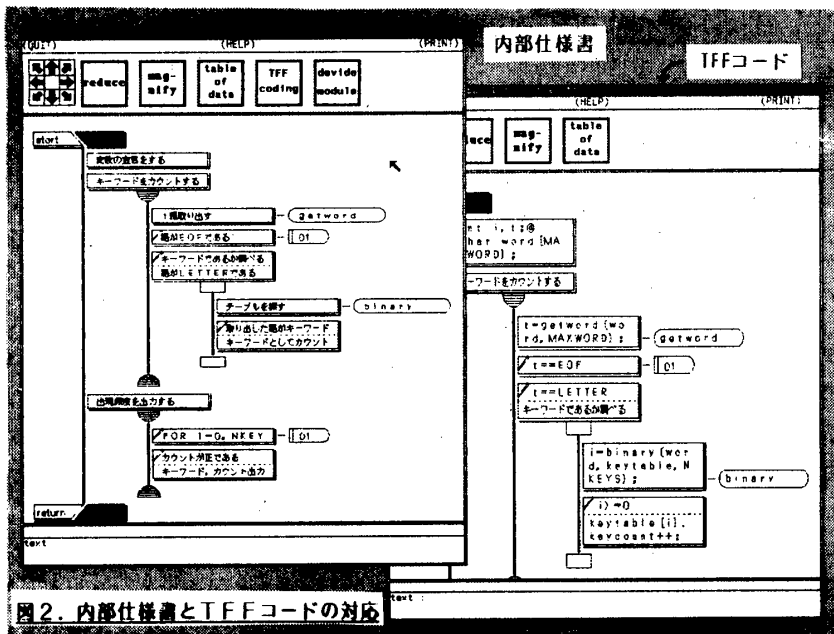


図2. 内部仕様書とTFFコードの対応

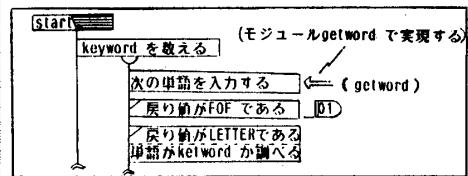


図3a. mainモジュールの内部仕様書(TFF)

図3b. 引数の情報

第1引数 名称: w 型: char *	変数名: word 型: char *
10R: output 表題: 読み込んだ単語...	種類: 内部データ
表題: 単語へのポイント...	表題: 読み込んだ単語...
説明: 入力した単語を返す...	説明: 読み込んだ単語...
	変数名: lim 型: int

図3c. プログラムの変数の情報

図3d. 引数、戻り値の確認画面(例)

プログラム	部品の情報
fd-getword(word, lim)	getword(w, l)
word:	w:
lim:	l:

図3d. 引数、戻り値の確認画面(例)