

EWS上のプログラミングツールにおける部品化支援方式

4 G-5

溝渕 順子*, 建部 周二*, 宮本 喜久男**

*(株) 東芝 府中工場, **東芝FAシステムエンジニアリング(株)

1. はじめに

ソフトウェア生産一貫支援環境New-SWBでは、大規模リアルタイム・ソフトウェアを対象に、エンジニアリングワークステーション(EWS)の上のツールを用いた分散処理環境でソフトウェア生産の流し化を実現しようとしている[1]。この流し化では、ハードウェア生産と同様、ソフトウェアの生産ラインの構築が目標である。しかしながら、この対象システムでは、ソフトウェアがシステムの様々な新しい機能の実現を受持つため、人間の思考過程が様々な形で入込む。このため流し化の重要なポイントである部品化支援は簡単な図式では行なえない。[2]

ここでは、一番ベースとなるプログラミングレベルを中心に部品化の方式とその評価方法を検討する。

2. 部品化とは何か

様々な分野で部品化が行われているが、実際に成功した例と、失敗した例を文献と実施例をもとにまとめると、表1のようになる。

部品化に必要な議論の範囲は、下記のようによく用いられる5W1Hの形になる。

- (1) Why 部品化の目的は?
- (2) What 何を部品とするか?
- (3) Who 部品開発は誰か? 利用者は誰か?
- (4) When 開発の手順との関係は?
- (5) Where 組織は、データの保管・管理の形態は?
- (6) How どのようにして部品を作るのか、どのような形で使えるようにするのか?

ここでは、プログラミングレベルで共通なWhatとHowの枠組みに注目する。

3. プログラミング部品の体系(Whatの枠組み)

EWS上のCプログラミングを例に下記のように洗い出した。

スタティック部品 (切り貼りで持ってくる部品)

- ・プログラムフレーム部品
プログラム全体の骨格を決める。
- ・モジュール部品/データ部品
これには更にシステム提供部品とユーザ部品がある。

ダイナミック部品 (ルールで応用される部品)

- ・標準構文展開ルール
言語で標準で持つ構文パターンの展開
例えば if に対して
if () {} else {}
を展開する。
- ・組込構文展開ルール
データに基づく構文パターンの展開
例えば switch に対して
データ 'a' を指定すると 'a' の値に対応した case を展開する。
- ・ローカル変数辞書
ループや関数呼び出しに使われる変数の生成を行う。

表1 部品化の成功例と失敗例

成功した部品化の例	部品化の失敗の原因
モジュールやデータ名称を分類して検索できるようにした。 特定の分野に関して仕様、設計から自動合成できる形を実現した。 制御の複雑でない分野に関してデータフローのパターンを洗い出し部品とした。 自ら作るのが困難と思われる所が部品化された(グラフィックまわり等)。	部品化できないところを部品化しようとした。 部品の定義があいまいだった。 部品化の目的や効果があいまいだった。 対象の調査が不十分のまま進んだ。 ハードウェアの部品との近視眼的同一視の下で、やみくもに部品化した。 部品に所要の品質(機能・性能・信頼性)や納期が確保されなかった。 部品に魅力が無かった。 効果の測定が無かった。

Approach to Components-Oriented Production using Programming Tools on EWS.

Junko MIZOBUCHI¹, Shuji TATEBE¹, Kikuo MIYAMOTO²

1 TOSHIBA CORPORATION 2 TOSHIBA FACTORY AUTOMATION SYSTEMS ENGINEERING CORPORATION

実際EWSではプログラミングに必要な部品はシステムで用意されるシステム提供部品(モジュール部品)のみでも1200に達した。これらを使いこなすには十分な訓練が必要であるが機械化によりその過程を短縮することが期待できる。

4. プログラミング手順の検討(Howの枠組み)

実際のプログラミングの手順は下記ようになる。

プログラムフレーム部品の選択と取込み

```
for (; 気にいったところまで; ){
  switch ( ){
    case モジュール定義(分割)
      ユーザモジュール登録
      モジュールフレーム生成
      ローカル変数生成/参照
    case モジュール参照
      ローカル変数生成/参照
      データ部品参照
    case ロジック展開
      標準構文展開ルール適用
      組込構文展開ルール適用
  }
}
```

この手順は、データもしくはロジックを参照しながら詳細化を進めていくこととともとれる。しかし一方では、既存のモジュールもしくはデータ定義(これらがスタティック部品)を参照してあらたなモジュール、もしくはデータを生み出す(これがダイナミック部品)過程とも解釈できる。

この手順を見る限り、部品の参照/定義機能とそれらを組立てる場が、かなり融通ができる形で組合わさっている必要がある。

5. ツールの構成

プログラミング支援ツールであるEDTtoolsでは、従来言語構文の展開と関数部品の切り貼りを支援していた^[3]。

今回、部品の拡張検討に伴い支援ツールを下記のように検討した。

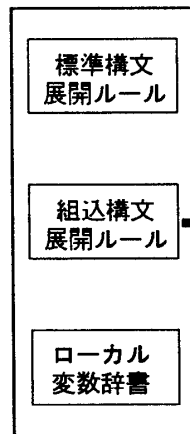
- (1) プログラム組立ての場としてエディタが用意される。
- (2) スタティック部品の検索

全てのモジュール部品及びデータ部品がブラウザから検索可能になっている。

ブラウザは、一般的に順次展開して操作出来る。これがスタティック部品の検索である。
- (3) ダイナミック部品への対応
 - (a) モジュール名称からデータ定義の候補を引いたりデータの組からモジュール定義の候補を引く
 - (b) 構文とデータ/関数定義を組合わせた展開

構文エリアでswitch-caseを指定すると一般的なswitch-case構文が展開されるだけであるがswitch-caseとデータまたは関数のリターン値を指定すると、そのデータもしくは関数のリターン値の取り得る値に対応した展開ができる。

ダイナミック部品



スタティック部品

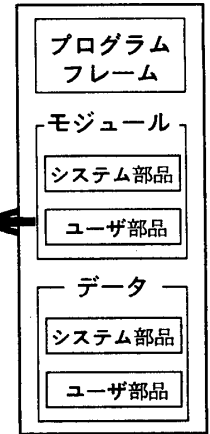
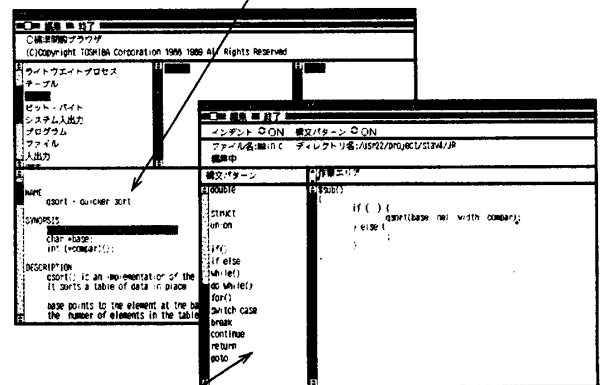


図1 ツールのシステム構成

現在、個々のエディタとブラウザの作成は完成しており、基本的な関数に対応したブラウザのデータは作成済みである。今後、ダイナミック部品の組込みのメカニズムを検討する。

スタティック部品検索ブラウザ



ダイナミック部品検索サブウィンドウ

図2 エディタとブラウザ

6. 今後の検討

今回、プログラミングを中心に部品化の方式について検討した。CASEツールが各種発表されているが、ソースプログラムが必要な分野では、どれだけソースプログラムが効率良くかつ品質良く生産できるかが決め手である。今後、プログラム設計やデータ設計ツールと組合わせた部品化について検討を続ける予定である。

参考文献

- [1] 小野他: New-SWB大規模リアルタイムソフトウェア開発環境, 情報処理学会第37回(昭和63年後期)全国大会 3M-4。
- [2] 田中他: 再利用支援システムSRED, 情報処理学会第39回(平成元年後期)全国大会 7S-1。
- [3] 建部他: New-SWBプログラミング支援ツール/EDTtools, 情報処理学会第37回(昭和63年後期)全国大会 3M-8。