

## SEA/Iにおけるプログラム木の 意味単位サイズについての一考察

5F-2

早野 晴、北野 宏明、中田 修二

日本電気(株)ソフトウェア生産技術研究所

1.はじめに

SEA/I [1] では、プログラムの設計支援のためにプログラム構造設計ツール (PRGDES) [2] が提供されている。PRGDES はプログラムのアルゴリズムを木構造でモデル化し、木構造 (プログラム木 (図 1)) をプログラムの設計者が容易に作成し理解できるように支援する。

現在、より高度な設計支援のための、マンマシンインターフェース (MMI) や、プログラム木作成支援機能などの検討を進めている。そのためには、PRGDES を使用して設計したプログラム木がどのような特徴を持つのかを分析することが必要となる。

我々は、プログラム木を構成している部分木をプログラムの意味単位として捉え、その大きさ (意味単位サイズ) を部分木の子ノード数として測定した。本稿では、意味単位サイズの測定方法と測定結果についての考察を述べる。

2. プログラム木

プログラム木のノード (図 1 の各行のケイ線で囲まれた部分) には、順次処理、条件分岐、繰り返しなど計 16 種類がある。それらのノードを組み合わせてプログラム木を画面上に作成することによってアルゴリズムを設計する。

設計の手順は、まず大まかな処理の順序を決定したプログラム木を描き (図 1 A)、そして次に各処理を詳細化していく (図 1 B)。詳細化した内容は 1 レベル下に新しい木として描かれる。このような過程を繰り返すことにより、プログラム木は成長し設計が進む。

3. プログラム木の意味単位・意味単位としての部分木

プログラム木をレベル差 1 の部分木に分解 (図 2) すると、その部分木は、プログラムの処理を定義している意味の単位と考えられる。つまり、プログラム木のルートノード (図 2 の A) を含んだ部分木の処理内容を理

(A) 処理概略の決定

```
PROGRAM:INPUTTAMA 開始段階入力
1. CALL :SCH030 開始段階化
2. CODING:OPEN I-O SHASTER 開始マスター・ファイルのオープン
3. ITERATE:IMP-RECEIPT SIMPLE:開始段階入力終了
4. CODING:CLOSE SHASTER 開始マスター・ファイルのクローズ
```

(B) 各処理内容の決定

```
PROGRAM:INPUTTAMA 開始段階入力
1. CALL :SCH030 開始段階化
2. CODING:OPEN I-O SHASTER 開始マスター・ファイルのオープン
3. ITERATE:IMP-RECEIPT SIMPLE:開始段階入力終了
1. PARTS :IMP-JCD ENTRY=JCD-RTN EXIT=JCD-EXIT
2. EXIT :IF END=STATUS "+" 開始段階コード入力
3. PARTS :INPUTL ENTRY=DTL-RTN EXIT=DTL-EXIT
4. PARTS :UPDATE ENTRY=UPDATE-RTN EXIT=UPDATE-EXIT
4. CODING:CLOSE SHASTER 開始マスター・ファイルのクローズ
```

図 1 プログラム木の設計

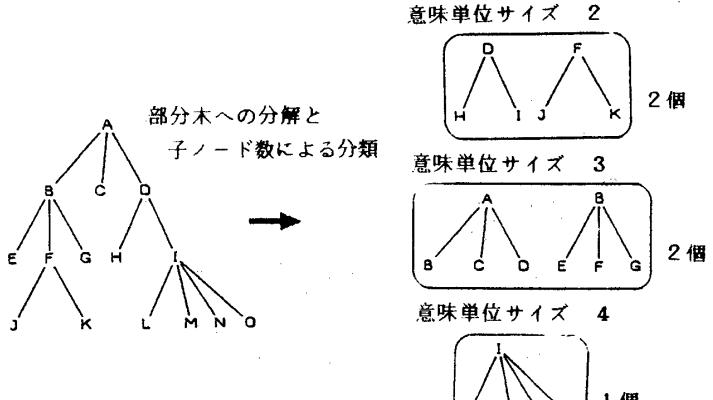


図 2 プログラム木の意味単位

解するとプログラム全体の概略が理解でき、さらにその部分木の子ノード (図 2 の B, D) をルートノードとした部分木を順次理解していくことにより詳細が理解されていく。従って、部分木はプログラムを理解するための 1 つの意味単位となっている。

・意味単位サイズ

意味単位のサイズを部分木が持つ子ノードの数とした。意味単位サイズが意味単位を理解する容易さの1つの指標となる。

#### ・測定手順

以下に示す手順でプログラム木の意味単位サイズを測定した(図2)。

- ①プログラム木をレベル差1の部分木に分解する。
- ②部分木を部分木の持つ子ノードの数で分類する。
- ③同じ分類に入った部分木の数を集計する。

#### 4. 測定結果

測定結果を図3～図5に示す。

測定結果は開発担当者とプログラムの種別(内部処理とMMI処理の2種類)で分類した。

#### 5. 考察

##### ・意味単位サイズの分布

人間の認知特性として一時に処理し理解できる項目数(プログラム木ではノード数)に限界があるため、設

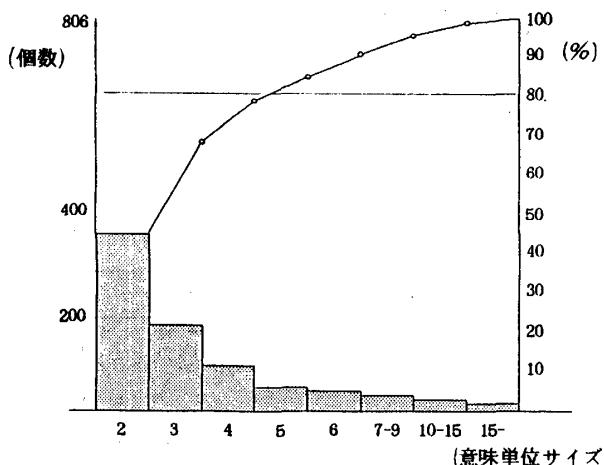


図3 意味単位サイズの分布

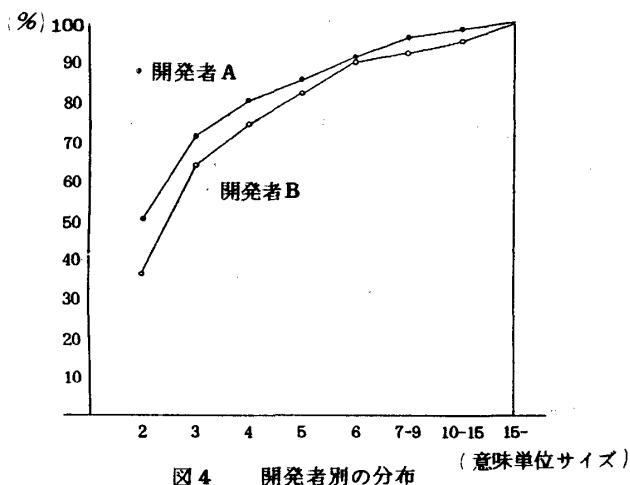


図4 開発者別の分布

計時の詳細化の個数にも、ある限界数があると予想される。実際、図3を見ると、意味単位サイズが、2、3、4、5の範囲で全体の約80%を占めており、7を越すと極端に割合が減少していることがわかる。また、図4に見るように、担当者やプログラムの種別の違いによっても、この傾向に差は見られなかった。

##### ・意味単位サイズが大きい箇所

意味単位サイズが大きい箇所はさらに構造化することにより、意味単位サイズを小さくし理解しやすく出来る可能性があり、図3の結果から意味単位サイズが7以上の箇所を要チェック箇所として検出できる。

#### 6. おわりに

プログラム木の意味単位サイズを測定した結果、その分布は、2～5に集中していることが分かった。

今後、このデータを、

- ・プログラム木の設計画面の大きさの推定
  - ・意味単位を効果的に処理するための編集機能
- などに役立てていく。

また、今回は意味単位のサイズにのみ注目して測定したが、ノードの種類を考慮した分析などにも今後取り組む予定である。

#### 参考文献

- [1] 中田、「日本電気のSEA/I(ソフトウェア・エンジニアリング・アーキテクチャ)」、事務管理、第24巻、第11号、pp71～75
- [2] 北村他、「SEA/Iにおけるプログラム構造設計ツール(P R G D E S)」、情処全大29回、1983年10月、pp591～592

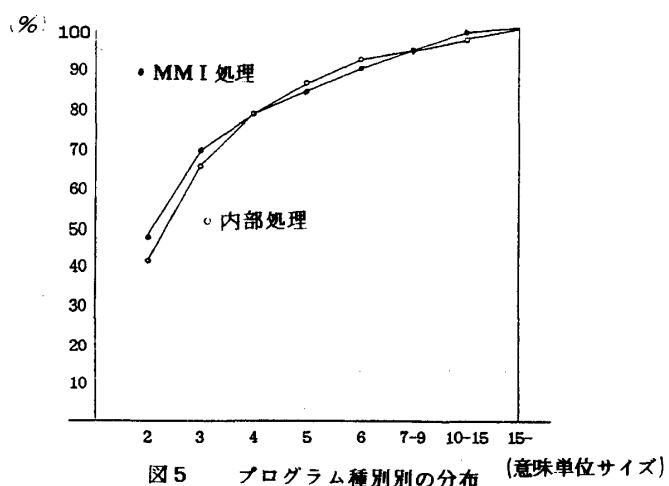


図5 プログラム種別別の分布