

C++ 言語向け母体理解支援ツールの開発（2）

7N-7

- メッセージフロー図逆生成ツールの開発と評価 -

佃軍治* 渡辺知宏** 館山正一*** 築地純子* 梶原茂喜*

* (株) 日立製作所 ** (株) 日立情報ネットワーク *** 日立東北ソフトウェア (株)

1 はじめに

近年オブジェクト指向言語で記述されたソフトウェアを母体とする開発が増加している。この開発に必要な母体ソフトウェアの理解を支援する母体理解支援ツールを開発した[1]。本ツールは、C++のソースコードを解析して、クラス設計など比較的上流工程で利用する図式や、指定したクラス・メンバ関数の変更影響範囲を表示する。

本稿で述べるメッセージフロー図逆生成ツールは、母体開発支援ツールを構成する1ツールであり、クラス間のメッセージ送信関係を示すメッセージフロー図（図1、OMTの事象フロー図に相当）を自動生成する。

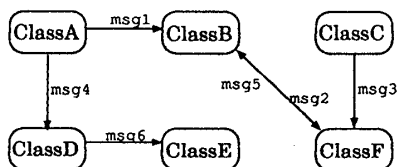


図1: メッセージフロー図の例

2 メッセージフロー図逆生成ツールの概要

メッセージフロー図逆生成ツールは、以下の2つの部分より構成する（図2）。

- ソース解析部

C++ソースコードを解析し、各クラスで使用し

Development of Software Basic Structure Analysis Tool for C++ Language(2) - Development and Estimation of Reverse Engineering Tool for Message Flow Diagram -
Gunji TSUKUDA, Tomohiro WATANABE, Shoiti TATEYAMA, Junko TUIJI, Shigeki KAJIWARA

* Hitachi, Ltd.

**HITACHI Information Network, Ltd.

***HITACHI Tohoku Software, Ltd.

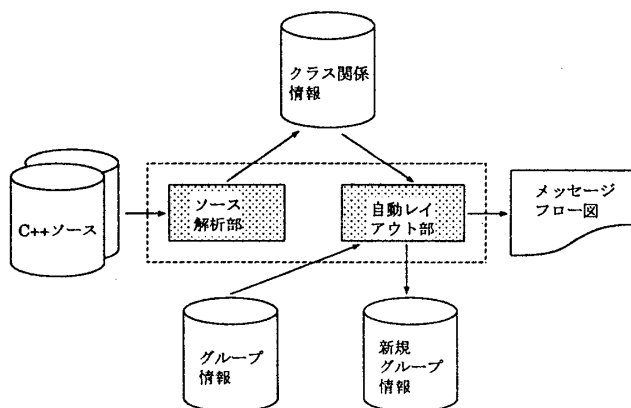


図2: 構成図

ている他クラスのインスタンスとメソッドを抽出し、クラス間のメッセージ送信関係とメッセージ名（すなわちメソッド名）により構成されるクラス関係情報を生成する。

- 自動レイアウト部（3章参照）

生成されたクラス関係情報とグループ情報（後述）から、クラス間の結合度に基づくクラスの自動配置とクラス間のメッセージの自動配線を行い、配置・配線結果を出力する。

グループ情報には、（1）クラスを要素とするグループ、（2）クラスまたはグループの表示位置、を指定する。

特定のクラス群に対しグループ指定を行なうことにより、そのクラス群を連続した領域に配置することができる。

また、本ツールは自動レイアウト部によって配置された位置情報を含むグループ情報を出力する。出力されたグループ情報を修正して、本ツールを再実行することにより、一度出力したメッセージフロー図のクラス配置を簡単に修正することができる。

また、特定のクラスを表示しないように指定することができるので、文字列クラスなど多数のクラスとメッセージ送信を行なう共通クラスの出力を抑制することが可能である。

以上のような機能により、理解が容易なメッセージフロー図を作成することができる。

3 自動レイアウト方式

3.1 配置処理

理解が容易なメッセージフロー図を作成するためには、指定されたグループ内のクラスを近傍に配置すると共に、無駄な領域をなるべく少なくし、図をコンパクトにする必要がある。そのため、クラスの多いグループから配置を行うようにした。配置するクラスの選択には、既配置要素の集合と未配置要素の集合の間で結合度が最大のペアを選択するペアリンク法 [2][3] を適用した。

3.2 配線処理

クラス間のメッセージ送信関係を示す線は重ならない方が望ましい。しかし、配線するクラス間の距離が遠くなればなるほど、重ならない経路を見つけるために、多くのバックトラックが生じ、処理時間が増大する。よって、障害物を避けながら目標点に向かって猪突猛進する、比較的高速な LSI の配線手法である高速迷路法 [2][3] を適用した。条件を満たす経路が発見できない場合は、線の重なりを許すか、または両方のクラスを示すボックスにコネクタを付加することによりメッセージ送信関係を示すようにした。ただし、対象とするソフトウェアの規模が大きい場合、線の重なりやコネクタの使用を必要としない経路を可能な限り探索させようとするれば、高速迷路法を採用しても、処理時間が爆発し、実用にならない。そこで、クラス間の距離が一定以上の場合には、重なりやコネクタの付加を行なうようにした。

高速迷路法では、線の折れ曲がる数に制約がない。しかし、理解しやすいメッセージフロー図を作成するためには、線の折れ曲がる数なるべく少ない方がよい。そこで、高速迷路法による配線後、折れ曲がる数が少なくなるように配線を修正するようにした。

4 性能評価

実用的な性能の範囲内で、メッセージの重なりやコネクタの付加をできるだけ避けるため、重なりを許す配線処理やコネクタによる配線処理を開始するクラス間の距離をパラメータとして、性能測定を行なった。

表 1 に、クラス数 94、メッセージ数 212 のソースコードを対象とした時のメッセージフロー図自動生成の処理時間を弊社の WS である 3050R (76MIPS) で測定した結果を示す。例えば、表 1 の太文字部分は、クラス間の距離が 1 のときは重なりを許さない配線を行ない、距離が 2~5 のときは重なりを許す配線を行ない、距離が 6 以上のときはコネクタによる配線を行った場合には、3 分 39 秒の処理時間を要したことを示している。

表 1: 測定結果 (分:秒)

	2	4	6	8
1	0:38	0:55	2:34	60 超
2	2:02	2:50	3:39	60 超
3	-	15:26	16:13	60 超

縦軸: 重なりを許す配線処理を開始するクラス間の距離

横軸: コネクタによる配線処理を開始するクラス間の距離

複数のソースコードを対象とした測定結果により、クラス数が 100 個程度のソースコードの場合、実用的性能を得るために、クラス間の距離が 2 以上のときは重なりを許す配線処理を行ない、距離が 5 以上のときはコネクタによる配線処理を行なうことが望ましいと考える。

5 おわりに

ソースコードからメッセージフロー図を自動生成するツールを開発し、性能評価を行なった。クラス数が 100 個程度までのソースコードであれば、パラメータを適切に設定することにより、実用に耐えることを確認した。

参考文献

- [1] 坂本 他, C++ 言語向け開発母体理解支援ツールの開発 (1) - ツール概要 -, 情報処理学会第 49 回全国大会予稿集, 1994
- [2] 藤城 編, CAD/CAM, 丸善, 1990
- [3] 渡辺 他, VLSI の設計回路とレイアウト, 岩波書店, 1985