重ねあわせを用いたシェルスクリプトプログラミング

Visual Shellscript Programming Using “Overlapping.”

奥村 穂高[†] 田中 二郎[††]
Hotaka OKUMURA Jiro TANAKA

†筑波大学 大学院修士課程 理工学研究科
Master’s program in Science and Engineering, Univ. of Tsukuba
††筑波大学 電子・情報工学系
Institute of Information Sciences and Electronics, Univ. of Tsukuba

概 要

データフローに「重ねあわせ」を制御フローに「有向グラフ」を利用した２つのフロー構造を同時に表記する手法を再検討した。また、本方式を Shellscript を対象とする直接操作可能な統合プログラミング環境に応用しシステムを実装した。本論文では、特に多対多の重ねあわせを行なえるようにするための「だんごイメージ」による表記法について述べ、また、本システムが内部で行なっている図形プログラムの解析について述べる。

1 重ねあわせを使った表記

ビジュアルプログラミングとは、プログラムの構造を図形を用いて表現するものである。そのシステム例として Pict [1]などがある。これらのシステムでは、プログラムを表現するために有向グラフが主に使われている。一方、HI–VISUAL [2]では、プログラムを表記するために図形の重ねあわせを重視している。ここでの「重ねあわせ」とは、２つのノードを重ねることによって、その間に意味を持たせるものである。

重ねあわせは、有向グラフとは全く異なった記述法であるのでこの 2 つの表記法は併用することができる。そこで、我々はこの利点に注目し、データフローを重ねあわせで制御フローを有向グラフで表現する表記法を提案した [3]。

例えば、OMT [4]にある機能モデルではデータフロー図の中に例外的扱いで制御フローを記述をしている。我々の提案した手法を用いればこの例外記述の必要がなくなり、有向グラフの簡略化が行なえるようになる。また、わざわざ制御フロー、データフローの 2 つのグラフを見比べるなどの必要がなくなるため、利用者のグラフ全体の把握や部分的なパターン認識の効率を向上することにつながる。

2 関連研究との比較

アイコンプログラミングとして代表的な物に、先に示した Pict が挙げられる。これは制御フローをエッジで表現する方法であり、ジョイスティックのみで BASIC や簡単な Pascal 程度のプログラミングが可能となっている。アイコンを使うことによってテキストを完全に排除するなど、アイコンプログラミングシステムの先駆けとなるものである。

また、アイコンプログラミングによって Shellscript を生成するシステムとして dish [5] がある。キーボードを基本的に使わず、「串刺し」という技術を使って文字列を生成するなど、ペン入力などのポインティングデバイスのための GUI Shell 環境を目指している。しかし、スクリプト自体はフローチャート形式のままであり、また、コマンドの内容は別途アイコン窓で記述するなど Shell にかなり特化した

形を取っている。このため、スクリプト言語自体に着目した本研究とはアプローチの方向が異なる。

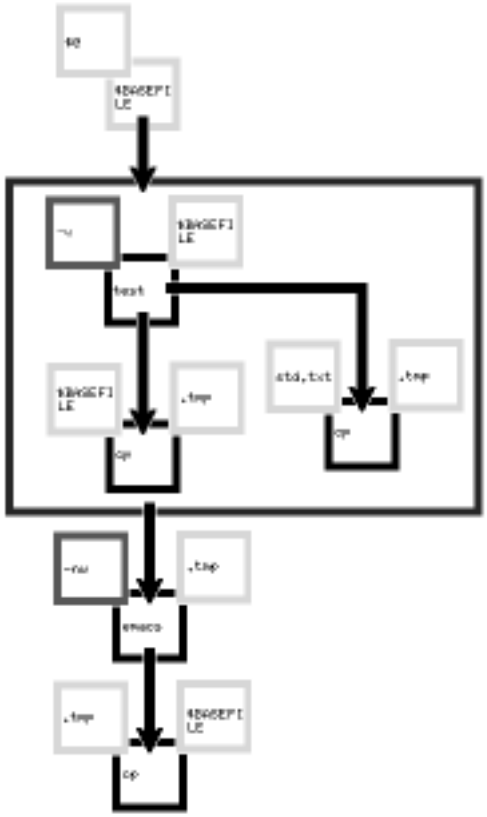

図 1　図形によって記述されたプログラム

```
BASEFILE=$@

if test -w $BASEFILE
then
  cp $BASEFILE .tmp
else
  cp std.txt .tmp
fi

emacs -nw .tmp
cp .tmp $BASEFILE
```

図 2　図 1から得られる Shellscript コード

3　Shellscript への応用

スクリプト言語は他の標準的なプログラミング言語よりより簡略な記述で表現することが可能である。したがって、ビジュアルプログラミングの対象として有効である。我々は、スクリプト言語の中でも特に UNIX の Shellscript に注目し、[3]で提案した表記法を用いたシステムを試作した。Shell とは UNIX 上で使われるコマンドインタプリタであり、Shellscript とはそこで使われるコマンドとその手順を記述した実行可能なプログラムのことである [6]。

今回、我々は [3]で提案をしたシステムの改良や新たな機能の実装を行なった。本システムは実装言語として、Java（JDK 1.1）を使用しており、次のような方針に基づき実装を行なった。

- コマンド、ファイル、引数などをノードで表す。
- マウスによる直接操作を用いる。

本システムでは、次のような手順で図形プログラムを入力する。

1. ノードの作成
   選択ウインドからノードの種類を選択し、制作側のウインドウに配置する。
2. ノードの重ねあわせ
   ノードを動かし、重ねあわせをおこなう。
3. 制御フローを記述する
   選択側のウインドウにより、エッジを選択し、分岐の作成、制御フローの指定などを行う。
4. グループ化を行う
   必要によってプレートをもちいてグループ化を行う
5. １～４の手順を必要に応じて繰り返す。

図 1は、我々が試作したシステムで作成したプログラムの例である。ノードの枠の色はそのノードの種類を示している。このプログラムは、emacs テキストエディタを利用する際に使うプログラムである。引数のファイルが新規のものならばテンプレートを、既存のものならばそのファイルを.tmp にコピーし、emacs で加工した上で目的のファイルとして.tmp をコピーする。

このようにユーザによって入力された図形プログラムをリアルタイムに解析し、図 2の Shellscript と等価なコードを出力する。

3.1　制御フローについて

制御フローは有向グラフによって表記される。重ねあわせを行った１つの塊が基本的に Shellscript の 1 行を構成する。その間をエッジが結び制御フローを作成する。また、フロー中の条件分岐は同一ノードを始点としたエッジの設定順序で決める。最初に作成したエッジが真、2 つ目が偽となり、以後、追加されるエッジは真と偽の無い方を補完する形となる。

3.2　プレートについて

プログラム中でループや分岐を表現する際はグループ化によっておこなう。このグループかを行な

う部品をプレートと呼ぶ。プレートに対するエッジの扱いは他のノードと同様である。ただし、分岐を示す際、ノードで偽を示すエッジがプレートではdefaultを示すことになる。

4　だんごイメージによるノードの表現

[3]で提案した重ねあわせ手法では、重ねあわせたノードをそのまま表示していた。しかし、このままでは多数のノードを重ねあわせるのは困難である（図3左）。この問題の解決方法として、ノードを拡大、縮小して対処する方法も考えられるが、それでは、必要以上に画面を占有したり、内容把握や直接操作がしにくくなる。これを解決するため、我々は「だんごイメージ」（図3右）を提案する。

だんごイメージとは重ねあわされたノード群を省略して表示する手法である。基本的にノード本体とセレクタから成る。図3は等価な重ねあわせを示している。

セレクタは串だんごのような形をしている。串に刺さっているだんご状のノードの１つ１つは、重ねあわされているノードを省略したものである。これをだんごノードという。kushi0左のノード１～３は図3右では省略されており、セレクタの付け根から１、２、３の順で配置されている。

セレクタ内のノードには隠蔽マークと呼ばれるものが付加される場合がある。これは省略しているノード上に、さらに重ねあわせたノードが１つ以上存在していることを示している。ノード２、３を示すだんごノードには、隠蔽マークがついている。これは、ノード２、３のそれぞれがノード４、５を持っているためである。

本論文では、入力を意味する上方からのノードの重ねあわさせに対して適応しているが、同様により下の方向へも適応することによって、出力ノードにも利用できる。

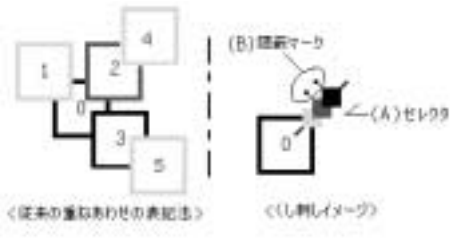

図3　今までの重ねあわせ（左）とだんごイメージ（右）

4.1　だんごイメージによるノードの切替え

図4はノードの選択について示している。だんご状のノードの一つをクリックすることによって、選択されたノードが省略されていない形として再び表示される。

例えば、左図のセレクタから真中にあるノードを選択（クリック）することで、右図のように選択されたノードとしてノード２が表示され、また、ノード２を示すだんごノードは白抜きとなる。この時、ノード２を示すだんごノードには隠蔽マークがある。これは、ノード２の上にさらに重ねあわされていたノードであるノード３が存在していることを示している。ノード３を操作するには、ノード２のセレクタによりノード３を選択することによって表示される。また、ノード０をドラッグした場合、表示されていたノード２は隠蔽され、図4左の状態に戻る。

選択されたノードが表示されている場合でも、それに対応するだんごノードは削除されず残る。これは、だんごノードが重ねあわせの順序を示しているためである。最もくしの付け根に近いノードが第１引数となり、外に向かうにしたがって第２、第３引数となる。

図4　表示ノード群の切替

4.2　だんごイメージへの重ねあわせ

だんごイメージで表されているノードに重ねあわせを行なう場合は次のようになる。

重ねあわされているノードがすべて隠蔽されている時に、他のノードを重ね併せた場合、だんごノードの最後の順序（最後の引数）として挿入される。しかし、常に最後に挿入するとは限らない。そこで次のような操作によって、途中挿入を可能にする。

ノード２が選択されているノード０へ、ノード４をドラッグ＆ドロップによって重ねあわせる。すると、セレクタ上の現在表示されているノード２より１つ手前のところへノード４が登録され、ノード４に表示が切替えられる。逆にノード０からノード４を外すとノード４がセレクタから外され、１つ外側

に登録されているノード２が再び表示される。

このように、表示するノードを１つに制限することによって、複数個の重ねあわせを可能にし、また、ドラッグ＆ドロップが行ない易くなる。

5 図形プログラムの解析

5.1 解析の流れ

本システムではインタラクティブにプログラムを作成するため、図形プログラムの解析をノードを配置した時に行なっている。具体的には、次のような流れになる。

1. ノードの下にあるプレートの有無を調べる
2. 重ねあわされるノードの選択・決定する
3. ノードに重ねられている引数に順序をつける
4. 重ねあわせを作るグループごとのコードを生成する
5. エッジからの情報で制御フローの関係を構築する
6. プレートもしくはキャンバスのコードを生成する

この過程を繰り返すことによって、図形プログラムが記述されるのと同時に目的の Shellscript を生成して行く。この流れの中において、重ねあわせは重要な位置を締めている。以降、この重ねあわせの解析について述べる。

5.2 重ねあわせにおけるノード

我々が提案した表記法では、データフローを重ね合わせで表現する際、データの流れは重ねあわされた手前のノードから奥のノードへ流れるとみなしている [3]。この制約に従って作成された図形プログラムを解析する際、次のような状態が考えられる。

重ねあわせる先に複数個のノードがあった場合、たとえば、図5のように、ノード０の下にノード１～３が図のように存在する場合を考える。このうち、ノード３に着目すると、データの流れとしてノード１から直接くるルートと、ノード２を経由するルートの２つの経路が考えられる。この場合、ノード３はノード０からのデータとノード２を経由したデータの２つを持つこととなる。

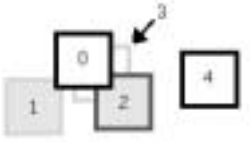

図５　重ねあわせの有効範囲

5.3 引数の順序について

重ね合わせを用いる際、引数などの複数のノードが１つのノードに重ね合わされることがある。この時、ノード解析の順番を決める。同一のノード上に重ねあわされた場合、ノードの優先順位は左上から時計回りに引数等の順序を割り当てている。また、だんごイメージの場合はセレクタ内のより付け根に近いノードを優先する。

プレート内部の解析は他の物より先に行ない、解析後は１つのノードとして扱うことになる。

6 結論と今後の課題

本論文では、データフローに重ねあわせを、制御フローに有向グラフを用いた表記法で記述された図的プログラムの構造を解析し Shellscript を生成するシステムを作成した。この際に発生した、多対多の重ねあわせを行なえるようにするため「だんごイメージ」という表記法を提案した。さらに、アイコンを用いた同様なシステムにおいて、本研究との比較を行った。

今後、このシステムを応用して他のスクリプト言語に適用したり、どのような分野に実用として利用できるかについて、調査研究する必要がある。

最後に、 本論文について多くの貴重なコメントをいただいた 小川徹氏 に感謝する。

参考文献

[1] Glinert, E., Tanimoto, S. : PICT: An Interactive Graphical Programming Environment, *IEEE Computer* , Vol.17, No.11, pp 7–25 (1984)

[2] Hirakawa, M., Tanaka, M. and Ichikawa, T. : An Iconic Programming System, HI–VISUAL, *IEEE Transaction on Software Engineering*,Vol.16, No.10, pp.1178–1184(1990)

[3] 奥村穂高, 田中二郎: 重ねあわせを用いたビジュアルプログラミングの表記法, 第 15 回ソフトウエア科学会大会論文集,pp 121–124 (1998)

[4] J. ランボー, M. ブラハ, W. プレメニラ, F. エディ, W. ローレンセン, 監訳：羽生田栄一: オブジェクト指向方法論 OMT, トッパン (1992)

[5] 早野浩生: ドラッグ＆ドロップでスクリプトの表記が可能なシェル, 第 15 回ソフトウエア科学会大会論文集,pp 169–172 (1998)

[6] UNIX System V プログラマ・リファレンス・マニュアル 第 2 版リリース 3.0, 共立出版 (1986)