

7R-3 データフローモデルに基づく知的プログラム支援環境

香月和宏 廣田豊彦 矢鳴虎夫
九州工業大学

1 はじめに

ソフトウェアの開発・保守において、経験による知識を機械的に使用できる環境を実現することは重要である。そこで我々は、プランをベースとした知的プログラミング支援環境IPS(図1)の構築を進めている。IPSはソースプログラムを抽象化したデータフローモデルと、プログラムの処理内容に関する知識を表すプラン[1]を中心として、1)プランに基づく既存プログラムの意味理解、2)プランを用いたプログラムの生成、の2つの処理を行うシステムである。今回はそのうちで、IPSのデータフローモデル及び意味理解のためのプラン具体化系について報告する。

2 データフローモデル

データフローモデルとは、プランと呼ばれるプログラムの処理内容に関する知識とのマッチングに適したプログラムの構造表現である。従来のプログラム解析では、コンパイル時のパーサーから出される構文解析木によるプランとのマッチングが行われてきたが、マッチングレベルが低く、本当に必要な抽象化レベルの高いプランとの適合は困難であった。ここに示すデータフロー・ブロック・ダイアグラム(DBD)は、構文解析木およびデータフロー解析結果から得られる情報を基にして作ったデータフローモデルである。

DBDは、あるプログラム単位に対応するブロックの木構造で表される。DBDの木構造は、プログラムの制御構造を表しており、各ブロックはプログラムの文や文の集まりを表している。ブロックは、木構造の終端であるかどうかによって2種類に分類される。

ターミナルブロック(図2) DBDの終端ノードで、代入や条件判定など、それ以上分割できない処理を表す。

構造化ブロック(図3) DBDの非終端ノードを示し、if文、while文や複合文など、構造化された処理を表す。

各ブロックには、そのブロックの属性の他にブロックの入出力変数が記述される。この入出力変数がプランマッチングに際して重要な役割を果たす。

ターミナルブロックは、プログラムの実際の一文を表す最も抽象化レベルの低いブロックであるため、ここにはプログラムの一文の内容が記述される。つまり、ターミナルブロックの流れは、元のプログラムの流れを写像したものである。このターミナルブロックも性質によって3種類に分かれる。

定義ブロック そのブロックで一つの変数が定義(出力)されている。

条件ブロック 制御文の条件部を示す。

通過ブロック 他のブロックに影響を与えないブロック(出力命令など)。

この3種類で表されるターミナルブロックが幾つか集まって構造的に一つのまとまったブロックで表される時、それを構造化ブロックで表す。構造化ブロックは、構文解析でいう非終端ノードに対応

して作られる。従って、各構造化ブロックは、そのブロックの内容を示す属性が記される。つまり、if文であるとか、while文であるといった条件文や、while-do文とか、then文といった条件文の中の実行文などを示す。こういった属性は、そのブロック自体の性質を示すだけでなく、その下位に並ぶブロックの役割も明示することになる。属性の具体的種類は、上記のとおり条件付き実行を表すもの{if, while, for, swich, do-while}と、順次的な実行を表すもの{comp}となる。属性compをもつ構造化ブロックは、プログラム全体を表すブロックだったり、条件付き実行を表す構造化ブロックの条件部と実行部に分かれるうちの実行部を表す下位ブロックなどに現れる。

3 プラン

プログラムの処理内容に係わる知識をプランというが、ここで扱うプランは、プログラム断片の機能や、使用される変数の意味などの知識である。ブロック単位といっても、様々な抽象化レベルのブロックがあり、それとマッチングするプランも様々なレベルになる。最初から抽象化の高いプランとのマッチングは無理であるので上記で説明したDBDの下位層であるターミナルブロックとそれに対応した抽象化の低いプラン(ターミナルプラン:図4)とのマッチングからはじめる。ターミナルプランは、DBDのブロック群を抽象化の高いプラン(抽象化プラン:図6)とマッチングさせる土台となるものでターミナルブロックをプランの表現形式に変換するものである。よって、ターミナルプランは、抽象化プランと異なり、必ずターミナルブロックとのマッチングが行わなければならない。ターミナルプランは、図4のように{定義変数の数、使用変数の数、種類(定義、条件、通過)、内容}が示されていて、これによってターミナルブロックとのマッチングが行われ、具体化プラン(図5)が作られる。具体化プランとは、ブロックがプランとのマッチングに成功した後、次のプランマッチングに使用される実際に具体的情報の入ったプランのことである。

図4で示される抽象化テキストと書かれた部分は、そのターミナルプランおよびブロックのプログラムの一文の内容が示されており、リスト構造で表す。

プランには、そのプランを認識するための名前(認識番号)がついており、そのプランの抽象的説明は、その認識番号に対応したテンプレート上に記載する。また、具体化プランにおいて{使用変数のリスト、定義変数のリスト}は、実際マッチングしたブロックの変数が、指定された順序ではいる。このことは、あとの抽象化プランのマッチングの時に必要になる。

抽象化プランは、今回目的とする知識を扱う。抽象化プランは、先に述べたDBDの制御構造の下位から、部分マッチングを行ってボトムアップにマッチング解析を行うもので、従って、色々な階層構造をもつ。抽象化プランは、基本的には複数の具体化プランに対応したプランブロックと構造化ブロックに対応したプランブロックを持った構造化プランである。

この様な形で表される抽象化プランは、DBDの下位ブロックが、一つ一つ具体化プランで埋め付くされた後、上位の構造化ブロックとその下位の具体化プランとでマッチングをはかり、マッチングが成功したら、その抽象化プランは一つの具体化プランとなる。

Intelligent Programming Support Environments
Based on Data-Flow Model

Kazuhiro KATSUKI, Toyohiko HIROTA, Torao YANARU
Kyushu Institute of Technology

4 プランの具体化と説明生成

本システムは保守者や再利用者に対して既存プログラムの意味理解をサポートする。つまり、プログラム内のあらゆるモジュールに対応した説明を生成する。DBDによってプログラムの構造をブロックで表現し、それを利用者（保守者や再利用者）に示すことで、どのブロックの内容が知りたいのかシステム側が利用者に質問する。質問に答えて、あるブロックが指定されたら、そのブロックの説明にマッチングするプランを下位階層からボトムアップすることで見つけ、具体化プランを作る。目的のブロックに対応する具体化プランができたら、その具体化プランの認識番号から説明プレート呼び出して、実際の数値・変数を代入して説明文を生成し、それを利用者に表示する。

図7のようにして、説明生成は行われるが、指定されたブロックとプランのマッチングおよびそのとき生成される具体化プランは図8のような手続きをとる。この時、プランとのマッチングをする前に、ブロック構造の最適化を計る。手法としては、

- 記号代入： 変数 a が変数 b に代入されているとき、次の2つの条件を満たすならば変数 b を変数 a に置き換え、このプランを削除する。
 1. 途中で変数 b が再定義されていない。
 2. 置換が行われる箇所よりも前の変数 a が再定義されていない。
- プラン縮退： Aomp 構造化ブロックの子ブロックにあるプランが適用されているとき、次の2つの条件を満たすならば、Comp 構造化ブロックを下位のプランに置き換える。
 1. 当該プランの使用変数が Comp 構造化ブロックの使用変数に含まれており、しかも当該プランより前でそれらの変数が再定義されていない。
 2. 当該プランの定義変数が Comp 構造化ブロックの定義変数に含まれており、しかも当該プランより後でそれらの変数が再定義されていない。

を用いて最適と思われるブロック構造に変換した後、プランマッチングをすることによって、余分なプラン蓄積の増大を防げる。

5 まとめ

本報告では、IPS 環境におけるデータフローモデルの具体的な形を示し、プラン具体化のアルゴリズムを記した。この一連のマッチング・ツールは現在開発中であるが、今後は、抽象化プランの拡大を検討する。

参考文献

[1] Mehdi T. Harandi and Jim Q. Ning, "Knowledge-Based Program Analysis," *IEEE Software.*, January 1990, pp.74-81.

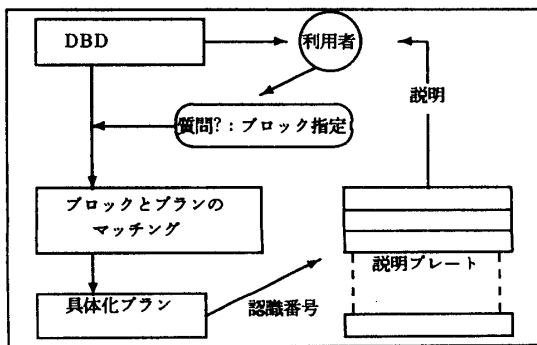


図7 説明生成の概要

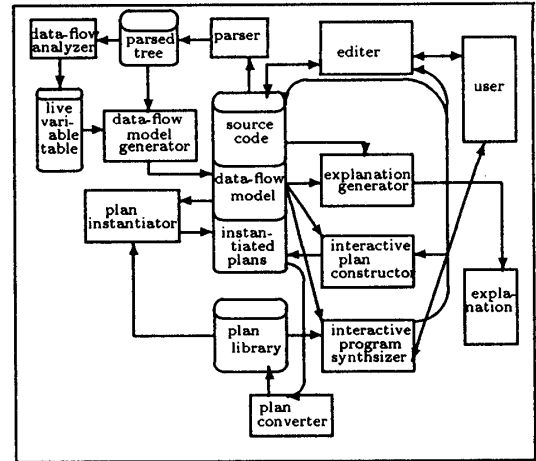


図1 IPS の概要

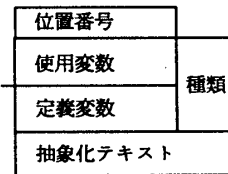


図2 ターミナルブロック

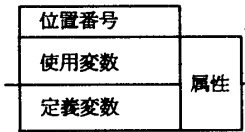


図3 構造化ブロック

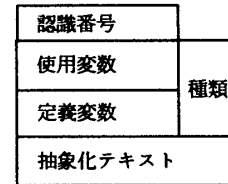


図4 ターミナルプラン

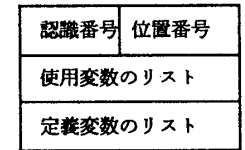


図5 具体化プラン

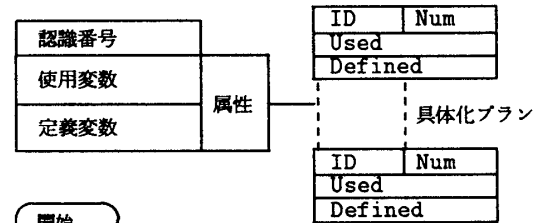


図6 抽象化プラン

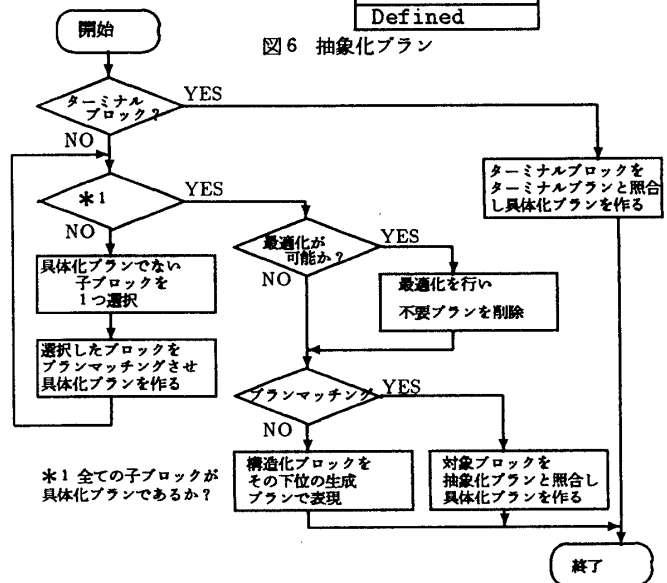


図8 指定されたブロックを具体化プランで置換する手続き (プラン具体化系)