

属性グラフ文法に基づく視覚的プログラミング環境の開発

6 P-2

- C 対応 Hichart 統合システム - *

今木 孝哲[†], 安達 由洋[†], 土田 賢省[†], 夜久 竹夫[‡][†]東洋大学 [‡]日本大学

1 はじめに

人間はテキストだけからでは読取れない複雑な情報を図からだと非常に迅速に理解できる場合がある。プログラムの開発や教育においても、図を用いた可視化はプログラムの構造や仕様を把握するのに有効な手段である。

本研究では、企業や教育でよく扱われるプログラミング言語 C を対象言語とした視覚的プログラム開発支援システム (C 対応 Hichart 統合システムと呼ぶ) の実現を目指す。本システムは、Hichart[1] を用いた図的プログラミングとテキストベースプログラミングを融合したものである。構文指向プログラム図エディタにより初心者は C 言語のデータ構造や制御構造の構成法を視覚的・直感的に学習することができる。また、このエディタで作成したプログラム図から対応する C プログラムを自動生成できる。一方、熟練者は既存の C プログラムを可視化し、その情報をプログラムの再利用や編集のために有効に利用できる。

2 C 対応属性グラフ文法

プログラム図を数学的对象として取扱うために、形式的なモデルとして ANSI 規格準拠の C に対応した属性グラフ文法を定義した。この文法は、プログラム図の生成を形式的に定義するプロダクションとレイアウト情報を属性として計算する意味規則から成り、全部で 110 の規則がある。この規則の一つである if_then_else 文を Fig.1 に示す。

3 C 対応 Hichart 統合システム

以下の機能を有する C 対応 Hichart 統合システムを実現する。

3.1 システムの機能

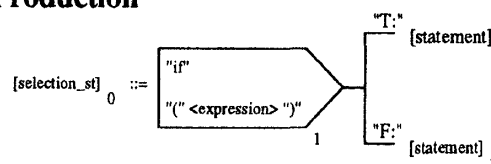
- (1) 既存の C プログラムを再利用またはデバッグするとき、ユーザにそのプログラムの全体構造が把握し易いように視覚化する。
- (2) 初心者がプログラミングする際は、扱い易い構文指向プログラム図エディタを提供する。

*Visual Programming Environment for C based on Attribute Graph Grammar

[†]Takanori IMAKI, Yoshihiro ADACHI, Kensei TSUCHIDA, Faculty of Engineering, Toyo University

[‡]Takeo YAKU, College of Humanities and Sciences, Nihon University

Production



Semantic Rules

```

top(2)=top(0)
top(3)=bottom(2)+GapY
x(1)=x(0)
x(2)=x(1)+w(1)+GapX
x(3)=x(2)
y(0)=y(1)
y(1)=(y(2)+y(3))/2
w(1)=MinW
h(1)=get_height(["if","(<expression>)" ])
cell(1)="exclusive_selection"
string(1)=get_str(["if","(<expression>)" ])
lines(1)=get_line(1,[2,3])

bottom(0)=max(bottom(1),bottom(3))
bottom(1)=y(1)+h(1)
nc(0)=nc(2)+nc(3)+1
id(1)=id(0)
id(2)=id(1)+1
id(3)=id(2)+nc(2)
cl(2)="T:"
cl(3)="F:"

```

Fig. 1. 生成規則と意味規則

- (3) プログラム図エディタとテキストで書かれた C プログラムを相互に変換可能にする。

これにより以下のプログラム開発支援機能が利用できるようになる。

- テキストとプログラム図エディタの双方向リンクにより、リアルタイムにプログラミングをテキストから図、図からテキストに反映する。
- ユーザが任意に選択した範囲をドラッグ & ドロップで、プログラム図エディタまたはテキストに表示する。
- 視覚化の対象をモジュール、プログラム、変数に変更し表示する。
- コンパイル時に、生じたエラー箇所をテキストまたはプログラム図エディタにハイライトする。

これらの特徴をもつ Hichart 統合システムの構成を Fig.2 に示す。システムの核となるトランスレータと構文指向型 Hichart エディタを連携させることで次の動作が実現される。

- プログラムの視覚化
エディタとブラウザの両方で既存の Hichart 図データや C プログラムを Hichart 図表示する。
- 図的プログラミング
新規にあるいは既存の Hichart 図データや C プログラムを読み込んで図的プログラミングする。

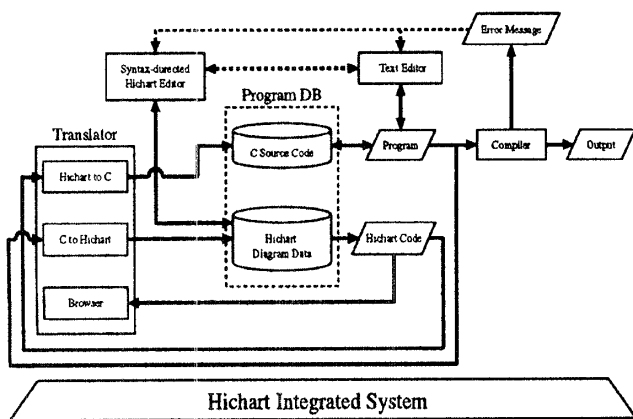


Fig. 2. Hichart 統合システム構成図
(破線部は開発中であり、実線部は実現済みである。)

- プログラムの実行
プログラムを C コンパイラを用いて実行する。
- Hichart 図データと C プログラムの相互変換
必要とする Hichart 図データや C プログラムがなければ、トランスレータを通して Hichart 図データまたは C プログラムを生成する。

3.2 インタフェースプログラム

ここで、トランスレータと図的エディタを統合することで生じる問題点について述べる。トランスレータを通して生成されるプログラム図は、終端ラベルのノードのみで構成されている。ところが、構文指向型 Hichart エディタではプログラム図の生成プロセスを表示するため、非終端ラベルのノードがある。このため、非終端ラベルのノード情報がない Hichart 図データを構文指向型 Hichart エディタで図的に編集することができない。そこで、非終端ラベルのノードだけを生成または削除してプログラム図の座標を再計算するフィルタを実現した。このフィルタにより、統合された個々の機能に対して連携動作が可能となった。

4 システムの機能部品

4.1 C 対応トランスレータ

本トランスレータの主な機能を次に示す。

(1) C と Hichart 間のデータ変換機能

- 変換部 C プログラムに対して字句解析と属性グラフ文法に基づいて構文解析し、属性評価を行い Hichart 図データを生成する。
- 逆変換部 Hichart 図データを属性グラフ文法に基づいて構文解析し、C プログラムを生成する。

- (2) 表示機能 データ変換機能で得られた Hichart 図データまたは既存の Hichart 図データからプログラム図を表示する。

4.2 C 対応構文指向型 Hichart エディタ

本エディタは、定義した属性グラフ文法に基づいており、レイアウト情報を含む属性をインクリメンタルに評価して Fig.3 のようにプログラム図を編集できる。このエディタは編集過程で構文エラーを起こさない特徴がある。また、意味的エラーを検出する機能をもつ。

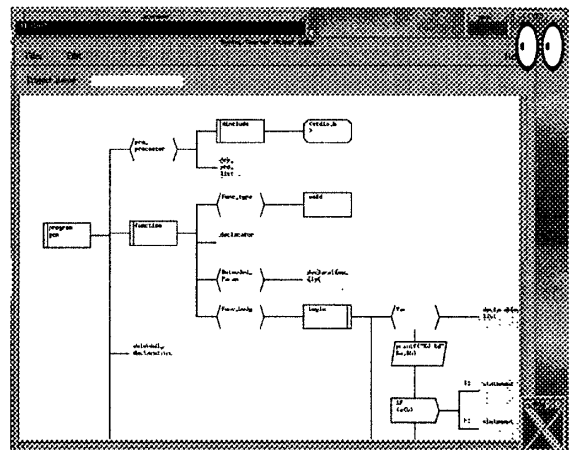


Fig. 3. プログラム図の編集

5 おわりに

プログラム図の形式的なモデルとして ANSI 規格準拠の C に対応した属性グラフ文法を定義した。次に、C と Hichart 間のトランスレータおよび C 対応構文指向型 Hichart エディタを定義した文法に基づいて実現し、これらを統合した。なお、Pascal に対応した属性グラフ文法、トランスレータ [2] および構文指向型 Hichart エディタ [3] については既に報告している。

今後は未完成の機能を実現し、さらに仕様書システムやプログラムデータベースとの統合により高度なプログラム開発支援環境を目指す。

参考文献

- [1] T.YAKU, K.FUTATSUGI, et al.: "HICHART - A Hierarchical Flowchart Description Language", Proc. IEEE 11th COMPSAC, pp.157-163(1987).
- [2] 大井 裕一, 安達 由洋, 夜久 竹夫: "Pascal から Hichart へのトランスレータの属性グラフ文法による記述と Prolog による実現", 電子情報通信学会技術報告, Vol.94, No.525, pp.89-96(1995).
- [3] Y.Adachi, K.Anzai, et al.: "Hierarchical Program Diagram Editor Based on Attribute Graph Grammar", COMPSAC96, IEEE Computer Society Press, pp.205-213(1996).