例示によるプログラミングにおけるビジュアルルールの利用手法

1N－2

杉浦淳　古関義幸
NEC C&C研究所

1 はじめに

例示によるプログラミング (PBD: Programming By Demonstration)[2] は、実際のデータ上でのユーザの例示操作からプログラムを生成する手法である。PBD での主な課題の一つは例示操作の一般化である。しかし、ユーザが複雑な概念を意図した場合、システムがその意図を認識できないという問題点がある。

本稿ではグラフィカルエディタ (以下エディタ) を対象とした PBD システム VipDevil について述べる。VipDevil の特徴はビジュアルルールの手法を利用することである。これにより、ユーザは特定のオブジェクトパターンを提示して自分の意図を VipDevil に伝達でき、VipDevil は従来システムでは一般化できなかった複雑な概念を扱うことができる。

2 PBD における一般化の困難さ

PBD システムが一般化に失敗する典型的な状況は、ユーザが意図を説明するために複数のグラフィカルオブジェクトが必要であり、それらのオブジェクトに対する操作が一切行なわれない場合である。

例えば、図 1a のようにエディタ上に描かれた棒グラフを考える。ユーザの目標は、点線 *Obj_2* が移動された時のコールバックとして、売上 (proceeds) が点線を越える月の文字を反転させるプログラムを作成することであり、ユーザが次の操作をエディタ上で行ったとする。

1. テキストオブジェクト *Obj_S* をセレクト
2. *Obj_S* の文字の色を白に、背景色を黒にする
3. セレクションを解除

これだけの操作から売上を表す矩形 *Obj_1* と点線 *Obj_2* が交わっていることが重要であることを認識するのは困難である。

3 VipDevil

VipDevil はグラフィカルエディタ上でのユーザの例示操作を一般化し、プログラムを生成する。ただし、2 節の例のように例示だけからは一般化に十分な情報が与えられなかった場合、ユーザは例示の意図を説明するのに必要な (一般化で考慮すべき) オブジェクトパターンを提示した上で、再度 VipDevil に一般化を行わせることができる。VipDevil はそれらのオブジェクトの属性値 (座標値、色など) を参照し一般化を行い、提示されたオブジェクトパターンを含むプログラムを生成する。

こうして作成されたプログラムの実行時には、ユーザが提示したオブジェクトパターンを検索する必要がある。この検索には ChemTrains [1] での "pictorial rewrite rule"(ビジュアルルール) のグラフィカル検索の手法を用いている。この手法によりグラフィカルオブジェクトによって構成される絵の中から指定されたオブジェクトパターンを検索することができる。こうして実行時に特定のオブジェクトパターンを抽出できるため、ユーザは自分の意図を VipDevil に伝える手段としてオブジェクトパターンを使うことができるのである。

3.1 例題

第 2 節の棒グラフを例題を用いて VipDevil でのプログラミングについて説明する。先ず、ユーザは売上が点線を越えている月を表す *Obj_S* をセレクトする (図 1b)。ここで、VipDevil はこの操作に関する解釈をユーザに示す (図 1c) が、この解釈では点線の位置が変わった時にユーザの意図通りにプログラムが動作しない。そこでユーザは自分の意図を説明するために必要なオブジェクトパターン (*Obj_1*, *Obj_2*) を示し、VipDevil に再び推論させる (図 1d)。VipDevil は提示された矩形 *Obj_1* と図 1b でセレクトされた *Obj_S* の位置関係を考慮して、図 2 の解釈をユーザに提示する。図 2 の解釈はユーザの意図を反映するものである。

次にユーザは、文字の色および背景色を変更し、さらにセレクションを解除し (図 1e)、例示を終了する。

3.2 プログラム生成

VipDevil はそれぞれの例示操作を実行可能なコードに変換する。ただし、セレクト操作に対しては一般化を行ない、次の形式のコードを生成する。

Using Visual Rules in Programming By Demonstration
Atsushi SUGIURA, Yoshiyuki KOSEKI
C&C Research Laboratories, NEC Corporation

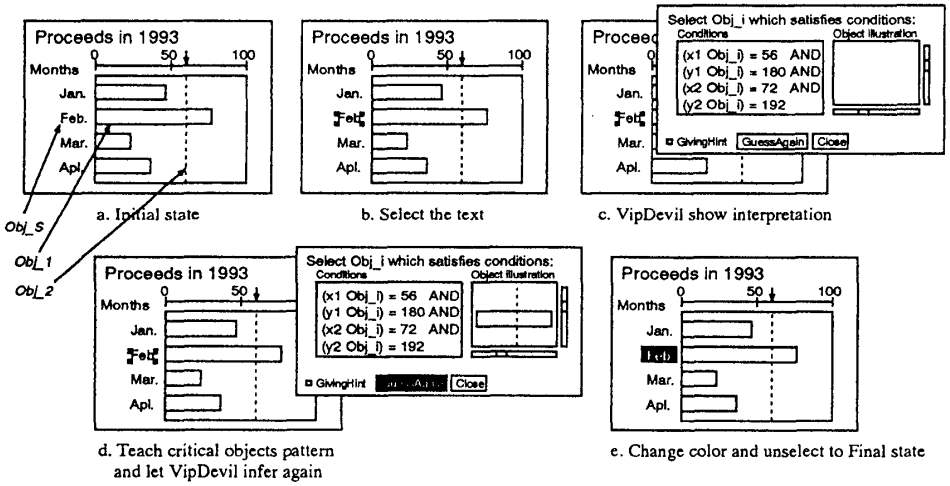

図 1: エディタに描かれた棒グラフ上での例示.

FOR *all objects* Obj_i *on Graphical Editor*
  IF Obj_i *satisfies condition* C
  THEN *select* Obj_i
  ENDIF
ENDFOR

条件 C は、セレクトされたオブジェクトとそれ以外のオブジェクトを完全に区別するものであり、図 2 に示すようにユーザによって指定されたオブジェクトとの位置関係、オブジェクトの属性値 (色、ペン幅など) などの特徴量の積和標準系から成る。条件 C は帰納学習手法を用いて生成される。

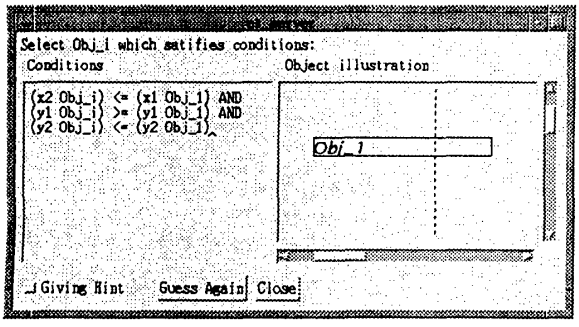

図 2: 一般化結果の提示ウィンドウ. ユーザは、"Giving Hint" のトグルボタンをクリックした後、自分の意図に関連するオブジェクトをエディタ上でクリックして指定することができ、指定されたオブジェクトは、"Object illustration" 欄に表示される。ここで、"Guess Again" ボタンを押すことにより、VipDevil に再度一般化させることができ、その結果が "Conditions" 欄に示される。"Object illustration" 欄の矩形に付けられたインデックス *Obj_1* は "Conditions" 欄の *Obj_1* に対応している。本図は、"Object illustration" 欄のように矩形と点線が交わったパターンがある場合には、その矩形と同じ行にあるオブジェクトをセレクトすることを意味する。

3.3 プログラム実行

例示操作に対応するコードを逐次的に実行する。ただし、図 2 のように、ユーザに指定されたオブジェクトがセレクトの条件に含まれる場合は、図 2 右側のオブジェクトパターンにマッチするオブジェクトの組を検索し、その上でセレクトすべきオブジェクトを決定する。

この検索は ChemTrains の手法を用いて行われる。すなわち、オブジェクト間のトポロジカルな関係 (包含、接続、交わり) に基づいている。例えば、図 2 のオブジェクトパターンの場合、矩形と点線が互いに交わってさえいれば矩形と点線の座標値に関わらず検索される。そのため、図 1 で点線が左方向に移動した場合、点線に交わる全ての矩形が検出され、それらの矩形と同じ行にあるオブジェクトがセレクト、色が反転される。

4 おわりに

様々なドメインにおいて、絵の意味付けは特定のグラフィカルオブジェクトのパターンやオブジェクトの相対的な位置関係でなされる場合が多い。そのため、ビジュアルルールの手法により特定のオブジェクトを抽出でき、そのオブジェクトとの位置関係を考慮して一般化を行う VipDevil の手法は有効であると考えられる。

参考文献

[1] Bell B. and Lewis C., "ChemTrains: A Language for Creating Behaving Pictures," *Proceedings of VL93*, 1993, pp.188-195.

[2] Cypher A., ed. *Watch What I Do: Programming by Demonstration*, MIT Press, 1993.