時相属性文法を用いたユーザインタフェース記述のX-Windowへの適用

1D－9

吉羽幹夫　山下義行　中田育男

筑波大学

1　はじめに

最近ではグラフィカルユーザインタフェース(GUI)の必要性が高まりつつある。しかしGUIを開発するにはかなりの労力を必要とすることが問題となっている。このような問題を解決する目的でユーザインタフェース管理システム(UIMS)の研究[1]が進められている。我々の研究室ではUIMSに属性文法の階層的記述を応用することを目指し、まず属性文法記述からGUIを生成するUIMS、wing[2]を開発した。さらにGUIの時間変化を容易に記述するため、時相属性文法(temporal attribute grammar)[3]を提案した。ところでワークステーションでのGUIの実現はX-Windowを用いるのが一般的である。そこで本研究では時相属性文法をX-Windowに適用する方法について述べる。

2　X-Toolkitと属性文法

X-Window上にGUIを構築する方法として、ここではそのライブラリであるX-Toolkitを使用する。X-Toolkitを利用したプログラミングではwidgetのクラスのインスタンスを生成し、それぞれを階層的に組み合わせ、外観などを決定する属性値(リソース)を与え、GUIを実現する。X-Toolkitを使用した簡単なGUIの例を図1に示す。このGUIはPress Me!と書かれたボタンをクリックするとHello, World!と書かれたボタンがポップアップする。

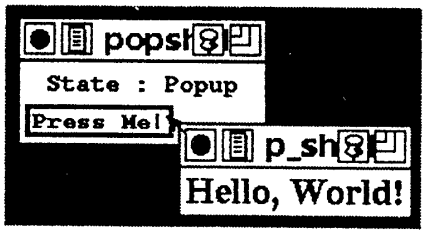

図1: 簡単なGUIの例

Application of an User Interface Description by Temporal Attribute Grammars to the X-Window System

Mikio YOSHIBA,Yoshiyuki YAMASHITA,Ikuo NAKATA
(Univ. of Tsukuba)

X-Toolkitを用いたプログラミングと属性文法の対応関係は次のようになる。これによって画面上の配置とその意味に添った属性文法記述が可能となる。

X-Toolkit	属性文法
widget	非終端記号
widgetの属性	非終端記号の属性
widgetの配置関係	構文規則
widgetの属性値の決め方	意味規則

3　時相属性文法を用いたGUI記述

従来の属性文法では木構造や属性値は処理系でコンパイルされた時点で静的に決定されるが、時相属性文法ではコンパイル後の実行時環境において木構造や属性値が変化する。つまり、構文規則や意味規則が再評価される。これによりGUIの動的な変化を記述することが可能となる。この節では時相属性文法における時間の考え方とGUI記述について簡単に述べる。

3.1　時区間

実行開始時点の時間をt_0とし、定義した外部イベントが発生した時間を$t_1,t_2,\cdots$とするとき、時間は時区間$T_0=[t_0,t_1)$, $T_1=[t_1,t_2),\cdots$に分割される。この時区間$T_0,T_1,\cdots$の列を時間軸と呼ぶ。時相属性文法では構文木全体に共通な時間軸は設定されず、部分木ごとに局所的な時間軸を設け、イベントの分散処理を行なう。そして、時区間が更新される(定義した外部イベントの発生)時に同じ時間軸を持つ部分木の構文規則と意味規則が再評価され、木構造と属性値が変化する。このとき、一つの時区間では属性値は一意でなければならない。

3.2　GUI記述

図2は図1のGUIを時相属性文法で記述したものである。

```
%nonterm toplevel : TopLevelShellWidget {};
%nonterm dialog   : DialogWidget
                    {syn int pstat;syn int pchange;};
%nonterm btn      : CommandWidget
                    {syn int pstat;syn char *dlabel;};
%nonterm p_shell  : ApplicationShellWidget {};
%nonterm p_btn    : CommandWidget {};
%time    dialog   : btn.<BtnUp>,p_btn.<BtnUp>;
%%
toplevel : dialog {};
dialog   : btn{
             (init)btn.label = "Press Me!";
             dialog.pstat = btn.pstat;
             dialog.pchange =
               (dialog.pstat != (prev)dialog.pstat);
             dialog.label = dialog.pchange ?
               btn.dlabel : (prev)dialog.label;
           };
btn      : p_shell WHEN(btn.BtnUp) {
             btn.pstat = 1;
             btn.dlabel = "State : Popup";
           };
         | /* no child */ WHEN(!btn.BtnUp) {
             btn.pstat = 0;
             btn.dlabel = "State : Popdown";
           } ;
p_shell  : p_btn {
             (init)p_btn.font = -*-times-*-r-*-*-24-*;
             (init)p_btn.label = "Hello, World!";
           };
p_btn    : /* no child */ {};
```

図 2:GUI 記述例

この記述は宣言部と構文、意味規則記述部に分けられる。1から8行目が宣言部である。ここで%nontermは使用するwidgetの名前、クラス、属性を宣言する。%timeは3.1節で説明した、局所的な時間軸を持つ部分木(実際には部分木の根となるwidgetを書く)とその時間軸を更新するためのイベントを宣言する。8行目は「dialogという名前のwidgetを根とする部分木が一つの時間軸を持ち、その時区間はbtnという名前のwidget上で<BtnUp>イベントまたはp_btnという名前のwidget上で<BtnUp>イベントが発生した時に更新される。」ことを意味する。

次に、10行目以降が構文、意味規則記述部になる。構文規則は左辺に親widget、右辺に子widgetを記述する。このとき、子を持たないwidgetは右辺にはwidgetを記述しない。{}で囲まれた部分が意味規則となる。意味規則には属性への代入式やC関数呼び出しなどを記述する。

3.2.1 属性値の変化

3.1節で述べたように時区間が更新される度に意味規則の再評価が起こり属性値が変化する。意味規則中では属性に(prev)という記号を付加することで、一つ前の時区間の属性値を参照できる。図2の16、17行目ではdialog.pchangeが偽の時は一つ前の時区間の属性値を参照している。ところで、過去の属性値は実行開始時点では不定である。そこで、(init)記号を用いて属性の初期値を設定することができる。(init)を付加した代入式は実行開始時のみに評価され、時区間更新による再評価時には評価されないので、実行開始時の過去の属性値の設定や再評価を必要としない代入式に使用できる。

3.2.2 木構造の変化

構文規則が複数存在する時は構文規則の右辺にガード(条件式)を付加する必要がある。図2ではWHEN()で囲まれた部分がガードとなる。実行開始時や時区間更新による再評価時にはガードが真となる構文規則が選択される。widgetの実体は実行開始時一斉に生成され、その後は選択された構文規則の右辺に現れるwidgetを表示し、右辺に現れないwidgetを非表示とする。X-toolkitを利用したプログラミングではポップアップがまさにこの動作である。この記述では構文規則の右辺にポップアップ用のwidget(popup shell)を置くことでポップアップの動作が記述できる。図2ではbtnはp_shellという名前のwidgetを子とするか、全く子を持たないとしている。これは「ガードの値によってp_shellがポップアップする」ことを意味している。

4 おわりに

ここでは、X-Window上にGUIを実現するための時相属性文法を用いたGUI記述を提案した。木構造の変化については、widgetを動的に生成／消滅をすることも考えられるが、実行効率などを考慮すると、表示／非表示の方が適していると思われる。

現在、処理系の開発中であり、今後、様々な例について実験を行なう予定である。

参考文献

[1] Pfaff, G , E.: User Interface Management Systems, Springer-Verlag(1985).

[2] 林謙一, 佐々政孝: 属性文法に基づく記述によるGUI生成系, 日本ソフトウェア科学会第8回大会論文集 (1991).

[3] 山下義行, 中田育男: 時相属性文法によるGUIの記述, 日本ソフトフェア科学会第9回大会論文集 (1992).