

Xツールキットを利用した図形部品の評価

2 T-3

塩沢 正三、二宮 寿朗、漆原 広

日立エンジニアリング株式会社 システム研究所

1.はじめに

ワークステーションの国際標準GUIの一つとして、OSF/Motifが利用できるようになった。しかし、各分野別のアプリケーション・プログラム(AP)を開発しようとした場合、Motifの提供するWidget(メニュー、リスト、ラベル、各種ボタンなどのGUI部品)だけでは作成できない。例えば、OA用途のAPを作成することを考えると、Motifで作成した対話画面上に表や折れ線グラフ、棒グラフ、円グラフなどの図形を表示することが必然的に要求されるであろう。その時は、X-Windowシステムのライブラリ(Xlib)を使って対話画面プログラムの中に図形表示プログラムを作成する必要がある。作成するには、300種類以上もあるライブラリの中から適切な関数の選定や使用方法を習得する必要があるなど、AP開発者にはかなりの負担と高度な技術力が要求される。

そこで、我々は上述したような図形表示に関するAP開発上の課題点を解決することを目的に、図形部品を作成し評価したので報告する。

2.図形部品の作成方式

図形部品を作成するにあたり、以下の点に着目してXツールキットのWidget作成方法を利用することにした。

- ・現在、Motifの普及率が高いためMotifとの併用が可能であること
- ・Motifで作られたAPの有効利用を考え、そのプログラムに容易に追加して拡張できること
- ・AP開発者の負担軽減を考えて、図形部品の使用法はMotifと同様であり、そのための学習はほとんど必要でないこと
- ・図形部品の信頼性と再利用性のためオブジェクト指向の概念が導入されていること

(1) 図形部品(図形Widget)の位置付け  
Xツールキットを利用して図形Widgetを作成する

Evaluation of Graphical Objects with X-Toolkit  
Masami SHIOZAWA, Toshiro NINOMIYA  
and Hiroshi URUSHIHARA  
Hitachi Engineering Co.,Ltd.

場合、Xツールキットがすでに持っている基本的なWidget(例えば、Core Widget)の機能を拡張して作成する。そのため、図形Widgetの位置付けは図1ようになる。(図の太枠がXツールキット部)

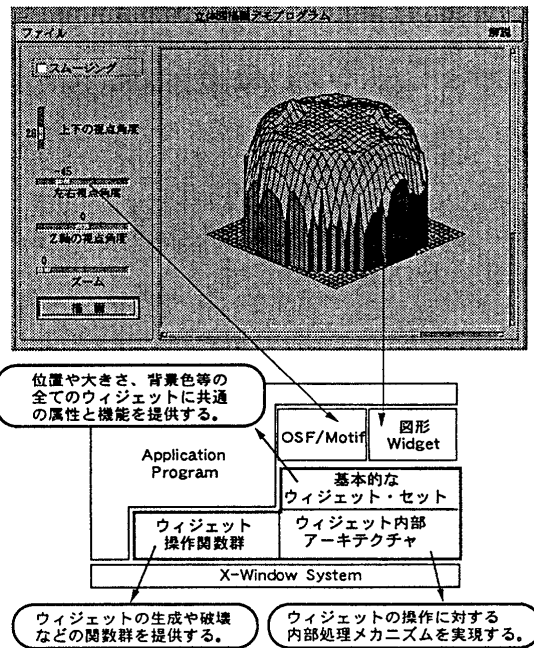


図1 図形Widgetの位置付け

(2) 図形Widgetの作成手順

Xツールキットで図形Widgetを作成するには、以下の手順で実現できる。

[手順1] クラス階層の検討

新規にWidgetを作成する場合、作成しようとするWidgetが属するクラス(WidgetClass)を、木構造をなすクラス階層中のどこに位置させるか検討する。

[手順2] Widgetの属性の設計と定義

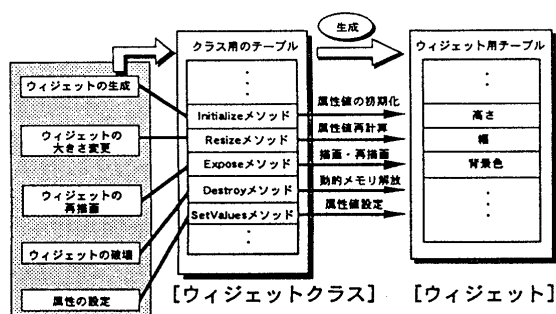
手順1のクラス階層を基に、新規属性を設計・定義する。

[手順3] メソッドの定義

属性を操作するメソッドを定義する。

[手順4] WidgetClassのライブラリ化

図2は、このようにして作成したWidgetをAPから利用したときの動作を示している。



【APからの  
ウィジェット操作】

図2 APからWidgetを利用したときの動作

(3) 図形Widgetの一例

XツールキットのWidget作成方法により、MeterWidgetおよび、その機能を拡張したMessageMeterWidgetを作成した。[1]

MeterWidgetは、自動車のスピード・メータを想定している。MeterWidgetに数値データを渡すと、針でそのデータ値を指し示すことができる。MessageMeterWidgetは、XツールキットのCoreWidgetのサブ・クラスとして位置付けた。

また、MessageMeterWidgetは、MeterWidgetに危険領域を持たせ、針がその中にあるときは、Widgetに設定されたメッセージを表示させるようにした。上述したように、本Widgetは、MeterWidgetの機能を拡張したサブ・クラスとして作成した。

図3にMeterWidget、図4にはMessageMeterWidgetのそれぞれの外観とWidget固有の属性を示す。各図とも、各WidgetをMotifのFrameWidgetに配置して、ScaleWidgetで数値データを与えて針を動かしているところである。

[針属性]

- ・針色 (黒)
- ・針位置 (0)

[目盛属性]

- ・目盛数 (10)
- ・目盛長 (5)
- ・前景色 (黒)
- ・目盛最小値 (0)
- ・目盛最大値 (100)

( ) 内はデフォルト値

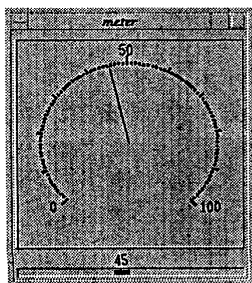


図3 MeterWidgetの外観と固有の属性

[危険領域属性]

- ・領域最小値 (0)
- ・領域最大値 (100)
- ・領域色 (赤)

[メッセージ領域属性]

- ・表示可否 (可)
- ・メッセージ (NULL)
- ・配置 (中央)
- ・フォント (asc08)

( ) 内はデフォルト値

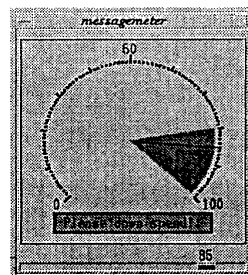


図4 MessageMeterWidgetの外観と固有の属性

3. 図形部品群開発に関する評価

今回、MeterおよびMessageMeterWidgetを、図形関数として作成した場合とXツールキットを利用して図形Widgetとして作成した場合の開発工数とプログラミング・ステップ数の比較表を表1に示す。開発工数に着目すると、前者の方がMeterを作成するには少ないが、Meterを拡張したMessageMeterを作成する場合は、逆に後者の方が少なくなる。これは、後者で図形部品を作成すると、オブジェクト指向の継承の概念が実現され、差分プログラミングを利用して作成できることに起因する。そのため、図形部品を図形Widgetとして作成した方が、部品の機能拡張性に優れているといえる。

表1 開発工数とプログラミング・ステップ数

図形	図形関数		図形Widgetクラス	
	開発工数	プログラムステップ数	開発工数	プログラムステップ数
メータ	48hr	650	80hr	427
メッセージメータ	64hr	948	16hr	275
合計	112hr	1,598	96hr	702

4. おわりに

今回の評価により、基本となるWidgetの作成には開発工数が若干増加するが、そのWidgetの汎用性や再利用性は優れていることが確認できた。また、図形Widgetの使用方法はMotifのWidgetと同様なため、AP開発者の負担軽減に寄与することが分かった。今後は、汎用的な分野別の図形部品の品揃えが必要になってくる。

参考文献

- [1]D.A.ヤング:XToolkitプログラミング:トッパン 1990
- [2]森本寛:XToolkitプログラミングテクニック:日刊工業新聞:1992