オブジェクト指向言語Koola

-グラフィックスクラスライブラリー

4G-3

岡崎 薫[1] 渡守武 和記[2] 西山 保[2]
1松下電器産業(株) 映像音響研究センター 2半導体研究センター

1. はじめに

今日、高度なグラフィックスユーザインタフェースに対する要求の高まりに対して、それを構築するために多大な労力と時間を費やさねばならない、という問題がある。

この問題を解決するために、今回われわれは、オブジェクト指向言語Koola[1]を用いてグラフィックスを容易に構築できるグラフィックス構築用クラスライブラリを開発した。

2. 概 要

2.1 構 成

今回開発したグラフィックスクラスライブラリは、オブジェクト指向言語Koolaのグラフィックス環境をサポートするものである。本ライブラリは、ウィンドウシステムに対し独立である。また、様々な描画方法が用意されていて、イベントも容易に扱える。

図1に本グラフィックスクラスライブラリの構成を示す。このライブラリは、抽象ウィンドウシステムと呼ぶ仮想的な中間ウィンドウシステムを基にして構築されている。

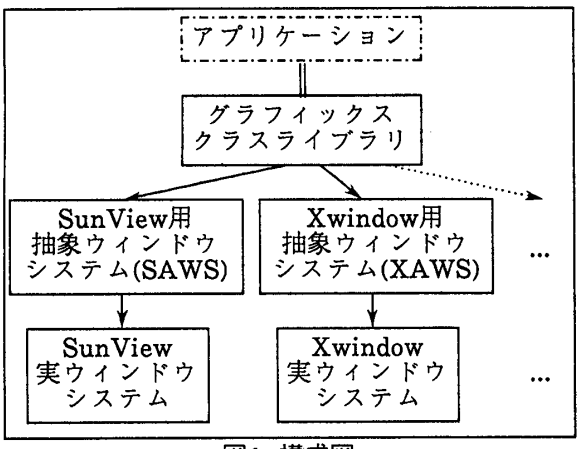

図1 構成図

2.2 特 徴

①ウィンドウシステムに独立

本グラフィックスクラスライブラリは、実ウィンドウシステムの抽象化を行なう抽象ウィンドウシステムを利用して開発されており、ライブラリは実ウィンドウシステムに対し独立である。

② 様々な描画方法のサポート

本ライブラリには、イメージを描画する方法を種々用意しており、利用者はアプリケーションに応じて最適な方法で描画できる。また各ウィンドウの状態管理や座標管理、イベント管理は自動的に行われ、利用者は特にこれらについて配慮する必要がない。

③ 容易なイベントハンドリング

本ライブラリには、イベントハンドリングを容易化するためにイベント管理クラスを設けている。イベント管理クラスは、イベントが発生すると、そのイベントに対応するメッセージを対象のオブジェクトに送る。対象のオブジェクトには、イベント管理クラスから送られたメッセージに対する処理を定義しておくだけでよい。イベントを解析する手続きを記述する必要がないので、イベントハンドリングが非常に容易である。現在、イベント管理クラスがオブジェクトに送るメッセージの種類には、up、down、moveの他に、clickとdragがある。

3. 機 能

3.1抽象ウィンドウシステム

図1において、抽象ウィンドウシステムは、実ウィンドウシステムに依存する部分を吸収し、ウィンドウシステムに依存しない抽象化された機能を提供する。これは、Koolaのタイプクラスで記述されているため効率がよく、更にこの上に作るグラフィックスクラスライブラリを完全なオブジェクト指向型記述で作成できる。また、簡単なものはマクロ定義することで高速化を図っている。例えばウィンドウの生成は、タイプクラスAbstructWindowを用いて、[AbstructWindow new]で記述できる。このAbstructWindowクラスのnewメソッドは、SAWSでは、SunViewのwindow_create()関数でウィンドウを生成するように定義されており、XAWSでは、XToolkitのXtCreateWidget()関数でウィンドウを生成するように定義されている。

3.2 描画方法

イメージを作成するために次の3つの方法をサポートしている。

1)図形プリミティブによる方法
2)ペンを用いる方法
3)切り貼りによる方法

Koola: An Object Oriented Language and its Graphics Classes Library
Kaoru OKAZAKI, Kazunori TOMOTAKE, Tamotsu NISHIYAMA
Matsushita Electric Industrial Co., Ltd.

1)は、直線や円の様な基本的な図形を組み合わせてイメージを作成する。
2)は、選択したペンの動作を記述することでイメージを作成する。
3)は、予め用意した「シート(Sheet)」を、任意の大きさの「ピース(Piece)」に切り出し、貼り付けることによりイメージを作成する。シートの大きさは有限であるが、タイリングにより任意の大きさのピースを切り出せる。

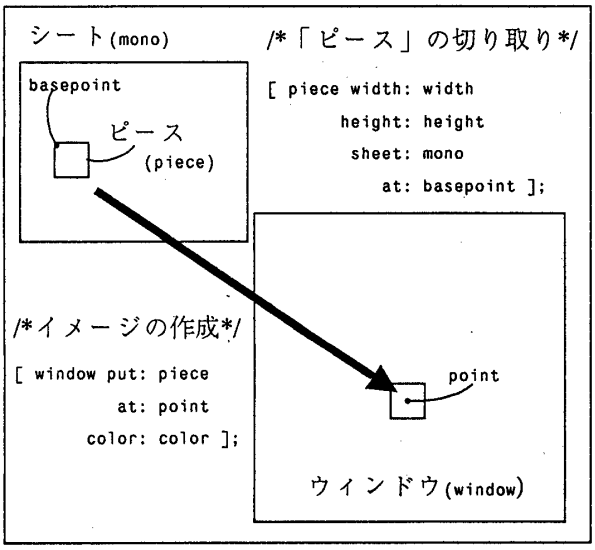

図2. 切り貼りによる描画

3.3 座標管理

グラフィックスの位置関係を管理する座標管理クラスは、各ウィンドウ内の点の位置を、ディスプレイ座標、スーパー座標、セルフ座標の3つの座標で管理しており、処理内容に応じて最適な座標を利用できる。ディスプレイ座標はディスプレイ面のもつ固定された座標で、座標点を画素単位による絶対座標で表す。スーパー座標は座標点を直属の親ウィンドウの座標で表す。セルフ座標は座標点をウィンドウ内の座標で表す。スーパー座標とセルフ座標は、画素単位による座標系であっても、任意に設定した座標軸に基づいた座標系でもよい。

3.4 イベント管理

イベント管理クラスは、イベントの識別と、対象オブジェクトの識別を行なう。イベントの識別は、入力デバイスの様々な状態や動作を区別することで行ない、対象オブジェクトの識別は、カーソルの位置を監視することで行なう。入力デバイスが操作されると、イベント管理クラスは該当するイベントと対象オブジェクトを識別し、そのイベントに対応するメッセージを対象オブジェクトに送る。イベントの処理は各々対応するメソッドを定義するだけでよい。例えば、マウスをdrag(ボタンを押したまま移動)した時に実行する処理の定義は、図3の様になる。このメソッドが定義されているウィンドウで、マウスの左ボタンをdragすると、マウスの動きに追随して四角形のラバーバンドが表示される。

```
InstanceMethods

   Instance  [ leftButtonDrag ]
   {
      [ RubberBand rectangle ];
      return self;
   }
```

図3. dragのメソッド定義例

4. 適用例

図4は、本ライブラリを適用して作成しているLSI設計用CADの表示例である。回路図Schematic_1 中の回路部品 oscilloscope_1 をセレクトし、その出力波形を表示している。

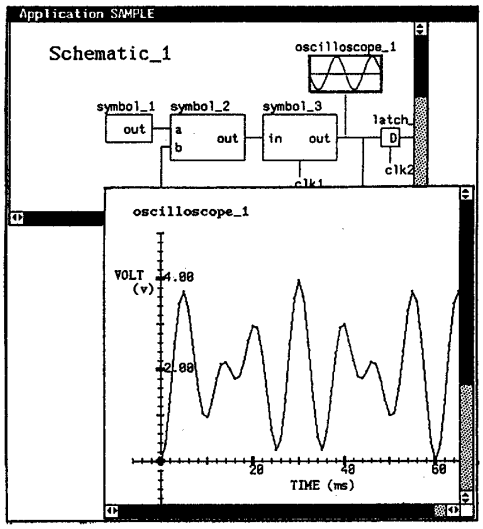

図4. LSI設計用CADへの適用例

5. おわりに

オブジェクト指向言語Koolaのグラフィックスライブラリについて説明した。本ライブラは、抽象ウィンドウシステムを利用することで実ウィンドウシステムに依存しない汎用的なものになっている。また、様々な描画方法と扱い易いイベントで、変化に富んだグラフィックスインタフェースを容易に構築することができる。

今後は、このライブラリの充実を図ると共に、現在SunView用抽象ウィンドウシステムの開発のみを行なっているので、他のウィンドウシステムへの対応を検討してゆく。

参考文献

1)渡守武他: オブジェクト指向言語Koola -新しい概念と機能- 情報処理学会第40回全国大会
2)B.J.コックス: オブジェクト指向のプログラミング
3)Adele Goldberg and David Robson:
Smalltalk-80 The Language and its Implementation
4)Stepstone社: ICpak 201 User Reference Manual
5)SunMicrosystem社: SunView Programmer's Guide