

対話型プレゼンテーション・プラットフォームにおける 4T-3 ユーザインタフェース管理方式 ープレゼンテーション・プラットフォームの開発(3)ー

里山 元章、柳 邦宏、高橋 亨、米澤 恵

(株)日立製作所 システム開発研究所

1. はじめに

従来、プレゼンテーションシステムはダイナミックに切り替わる電子的なカードやOHPなど、限定された形式のプレゼンテーションを提供しているにすぎない。我々はユーザが自由にプレゼンテーションシステムを作成できる環境をプラットフォームとして提供するプレゼンテーションプラットフォームを提案¹⁾し、画面やシナリオを作成するプラットフォームを試作した。本研究において、シナリオ記述言語やシナリオ作成用のビジュアルエディタなどが合わせて試作されているが、本報告では、プラットフォームの標準装備UI (User Interface) 部品について述べる。

2. プレゼンテーションプラットフォームの概要

プレゼンテーションプラットフォームはシナリオインタプリタ、システムクラス、シナリオエディタの3つの機能階層に分類される。

シナリオインタプリタはプレゼンテーションのシナリオを進める並列オブジェクト指向型言語インタプリタである。シナリオインタプリタにはクラス継承、メモリ自動回収、オブジェクト間の並行動作と同期、時間軸制御などの機能を備えた²⁾。

また、OSF/Motifに準拠したユーザインタフェース部品やマルチメディアの制御、ハイパーメディアを実現するリンク、別プロセスと画面を介して複合するなどの機能はシナリオインタプリタにあらかじめ組み込まれているシステムクラスとして実現した。

シナリオエディタは、シナリオを対話的に作成するためのエディタである。

3. システムクラスの階層

UI 部品はシナリオインタプリタにあらかじめ定義してあるシステムクラスとして実装した。UI 部品をオブジェクト指向でいうオブジェクトとしてとらえ、必要な機能はメソッドとして呼び出す。ユーザが必要なUI 部品を作成する場合、システムクラスとの差分をユーザクラスとして記述すればよい。

図1はシステムクラスの階層構造図の一部である。

以下に、代表的な部品について述べる。

(1) ステージ

1つのプレゼンテーション画面には、最低1個のステージがある。プレゼンテーション画面はステージをひとつまとまりとして管理する。

(2) シート

プレゼンテーションに使う画像や図形の表示編集メソッドを備えたウィンドウである。アウトラインフォントによる文字描画も可能である。

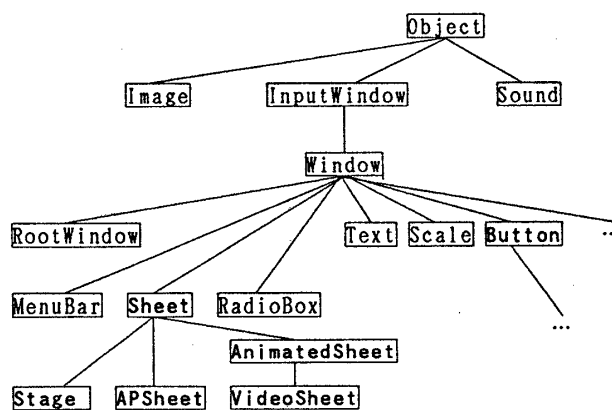


図1 システムクラスのクラス階層(抜粋)

(3) アニメーションシート

シートのサブクラスで複数のセル画像を登録しておき、順次表示することでアニメーションを表示するメソッドを備えている。

(4) ビデオシート

レーザーディスクなどのビデオ画像をシートの位置サイズに合わせて拡大縮小表示できる。また、レーザーディスクの制御用メソッドも備えている。

(5) APシート

データベースやエキスパートシステムなどの出力を使ってプレゼンテーションを行う場合、これらのプロセスと分散協調する機能が必要である。

APシートと呼ぶUI 部品にはネットワーク上の他のワークステーション上で動作するプロセスとのメッセージ通信機能と、他のプロセスのウィンドウをAPシート上に動的に重ね合わせる機能を備えた。これにより、他のプロセスの画面をそのままプレゼンテーションに利用できる。また、他のワークステーションから動的にシナリオを実行するリモートローディング機能や他プロセスを起動する分散実行機能も備えている。

(6) ボタン

シートを切り替える場合などに使う。

(7) メニュー

複数のボタンをまとめてメニュー(プルダウン、ポップアップなど)を構成する。

4. ユーザインタフェースの制御構造

本プラットフォームではマウスやキーなどのイベントが発生すると抽象化されたメッセージに変換される。こ

のとき、1つのイベントが複数のメッセージに変換される場合と複数のイベントが1個のメッセージ変換される場合がある。これらの操作と画面と処理の関係を明確に記述するため、イベントをメッセージに変換する規則をシナリオとは別にイベント変換表に定義できるようにした。このイベント変換表はクラス単位で定義でき、該当するクラスのすべてのインスタンスに有効になるようにした。イベント変換表はクラスに固有のビヘイビアを規定することになる。

図2はユーザインタフェースの制御構造を図式化したものである。図2において、イベント入力部は読み込んだイベントをイベント変換部へ送る。イベント変換部はイベント変換表に従ってイベントをメッセージや表示変更命令に変換する。表示変更命令は、例えばプッシュボタンをマウスで選択すると凹むように、その部品があらかじめ備えている(シナリオで記述する必要がない)特有のビヘイビアを処理する場合だけ呼ばれる。メッセージはメソッド呼出し部(シナリオインタプリタ)に送られ、該当するメソッドが検索され実行される。もし、表示状態を変更する必要があるならシステムクラスのメソッドを呼び出す。このとき、イベント変換によって生じたメッセージとシナリオのsend文によって生じたメッセージは等価である。

以下にイベント変換表の記述例を示す。この例では、MyStageClassのインスタンス上でマウスを押下するか、コントロールキー+AキーでmapVSheetメソッドが起動される。なお、イベント部分の記述はX Toolkitのトランスレーションと同じ表記にした³⁾。

```
TranslationTable MyStageClass
{
  <Btn1Up>:      SendMessage(mapVSheet)
  Ctrl<Key>A:    SendMessage(mapVSheet)
}
```

次に上記イベント変換表を使ったシナリオの例を示す。ここで定義される MyStageClass は内部にビデオ画像を表示するシートを持ち mapVSheetメッセージによってビデオの表示を開始する(newObjectメソッドはインスタンス生成時にシステムが呼び出す)。

イベント変換表を使うことによって、操作と処理の対応関係をシナリオと別に記述でき、容易に操作と処理対応を変更修正できる。また、イベントでなく処理内容を反映したメッセージでシナリオが記述できるため、シナリオの可読性が高まった。

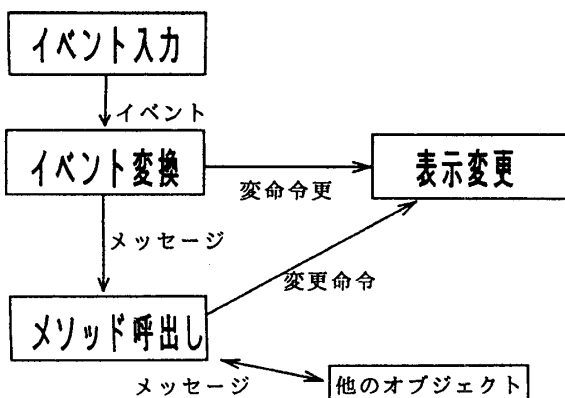


図2 制御構造

```
class MyStageClass {
  superClass StageClass
  private vsheet
}
method MyStageClass newObject() {
  vsheet := new VideoSheetClass
  send vsheet openPlayer("LaserDisk")
  send self child(vsheet)
}
method MyStageClass mapVSheet(){
  send vsheet map(), play()
}
stage := new MyStageClass
send stage map()
```

5. リンク機構

ボタンが選択されたときにシートを表示するようなシナリオを作るだけでクラスを定義するのは煩わしい。そこで、インスタンス間の関係を簡便に記述できるようにリンク機能を用意した。

リンクの実体は手続きであるメッセージをデータとして記憶したものである。リンクをたどるには記憶しておいたメッセージを実行するメソッドを呼出す。このとき、オブジェクトにおいて選択操作のイベントが発生すると、リンクをたどるメッセージを発生させるようにイベント変換を定義してある。よって、リンクをたどる操作はイベント変換で容易に変更できる。以下にリンクを使ったシナリオ例を示す。

```
stage := new StageClass
button := new PushButtonClass
vsheet := new VideoSheetClass
send button linkObject(vsheet,'map')
send stage map()
```

6. おわりに

本報告ではプレゼンテーションを対象としたユーザインタフェース管理用部品の提供方式について述べた。本プラットフォームは、すぐに使える部品が豊富に準備されており、既存部品の組合せだけである程度のシステムが構築でき、少ないプログラミングで自分向きのシステムが組める。また、オブジェクト指向に基づき、組み込みのシステムクラスとして実現されるので、差分を記述するだけで自分好みのシステムを、容易に効率よく作成できる。

<参考文献>

- 1) 柳他：プレゼンテーション・プラットフォームの全体構想、情報処理学会第44回全国大会講演論文集(1992.3)
- 2) 高橋他：プレゼンテーション・プラットフォームにおけるシナリオ記述言語、情報処理学会第44回全国大会講演論文集(1992.3)
- 3) 日立クリエイティブワークステーション2050 Xツールキットイントロシックス文法 文法書、日立製作所