

4T-2

対話型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

- プレゼンテーション・プラットフォームの開発(2) -

高橋 亨 柳 邦宏 里山 元章 米澤 恵

㈱日立製作所 システム開発研究所

1. はじめに

スライド、OHP等の従来型プレゼンテーション手段に取って代わるものとして、パソコンやワークステーションの画面そのものを媒体として用いるDTPR (Desk Top Presentation) が注目を集めている¹⁾。しかし、そのアプリケーション開発を個別に行ったのでは多量の重複開発が発生し、また操作性の統一やデータの交換も困難になる。そこで、各種のDTPRアプリケーションを構築するための共通基盤を提供することを目的として、「対話型プレゼンテーションプラットフォーム」の開発を行っている²⁾。本稿では、このプラットフォームの構成要素のうち、プレゼンテーション動作の記述に用いるシナリオ記述言語 Conteman について述べる。

2. シナリオ記述言語の提供する機能

プレゼンテーションのシナリオを記述する上では、各種ユーザインタフェース部品やメディアデータの性質の違いにかかわらず、それらを統一的に取り扱えることが望ましい。また、図1に例を示すように、マルチメディアを駆使して効果的なプレゼンテーションを実現するには、複数動作の並列実行や同期、細かい時間管理等を行う機能が必要になる。

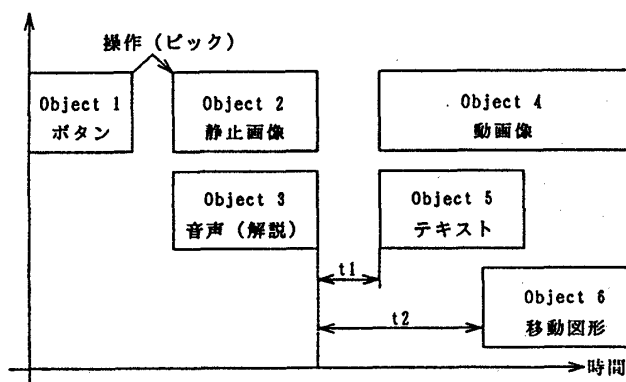


図1 プレゼンテーションシナリオ例

このため、Conteman では各種の操作対象をすべてオブジェクトとみなし、ユーザ操作や外部事象に伴うイベントはこれらのオブジェクトに対するメッセージとして扱うオブジェクト指向の枠組みを採用している。また、(1) 複数のオブジェクトを並列に動作させ、またそれらの間での同期を可能にする並列・同期制御機能、(2) オブジェクトの動作タイミングを制御する時間制御機能、(3) オブジェクトの生成や破棄に際して、明示的なメモリの割当てや解放を不要にする自動メモリ管理機能、等を提供している。

3. 言語仕様の概略

Conteman では、すべてのオブジェクトは何らかのクラスのインスタンスとして生成される。また、各クラスは既存のクラスのサブクラスとして定義され、上位のクラスから全てのプライベート変数(インスタンスごとに固有の変数)およびメソッドの定義を継承する。

アプリケーションの作成を容易にするため、基本的なユーザインタフェース部品や各種のメディアデータに対応するクラスはシステムクラスとしてプラットフォーム内に組み込まれている。このため、簡単なアプリケーションであれば、これらシステムクラスのインスタンスを生成して組み合わせることにより、容易に作成することができる。システムクラス群の内容については別稿³⁾にて報告する。

図2はユーザクラス定義の例である。このように、適当なシステムクラスのサブクラスとしてユーザクラスを定義し、プライベート変数やメソッドを追加/変更することにより、目的に応じてカスタマイズしたクラスを得ることができる。

ただし、Conteman はすべてのデータをオブジェクトとして扱うわけではない。数値、文字列、ベクタ等の基本データ型は、組込みの演算子や関数を用

いて直接処理を行う。また、クラスやインスタンスも言語仕様上はデータ型的一种として扱われる。

4. メッセージ送信と並列・同期制御

オブジェクトにメッセージを送るには、

```
send レシーバ メッセージ名(引数, ...)
      または
post  レシーバ メッセージ名(引数, ...)
```

という形のメッセージ送信式を用いる。

send と post との違いは、メッセージの送り手（センダ）と受け手（レシーバ）との間での同期の有無にある。send を用いた場合、センダはレシーバが対応するメソッドを実行し終るまで待たされる。これに対して post を用いた場合、センダはレシーバ側でのメソッドの終了を待たずに次の動作の実行に移る。この結果、二つのオブジェクトは並列に動作することになる。このように、メッセージの送受を契機としてオブジェクトを単位とした並列・同期制御を行うことができる。

なお、メソッド内で wait 文を実行することにより、待ち時間または再開時刻を指定して残りの動作の実行を待たせることができる。このような時間制御もオブジェクトごとに独立に行われる。

5. オブジェクトの動作モデル

図3は Conteman におけるオブジェクトの動作モデルを示している。オブジェクトに到着したメッセージはいったんリクエストキューに蓄えられ、ここから一つずつ取り出されて処理される。処理中に条件待ちが必要になった場合、メッセージは中断され、中断メッセージリストに退避される。一時にはただ

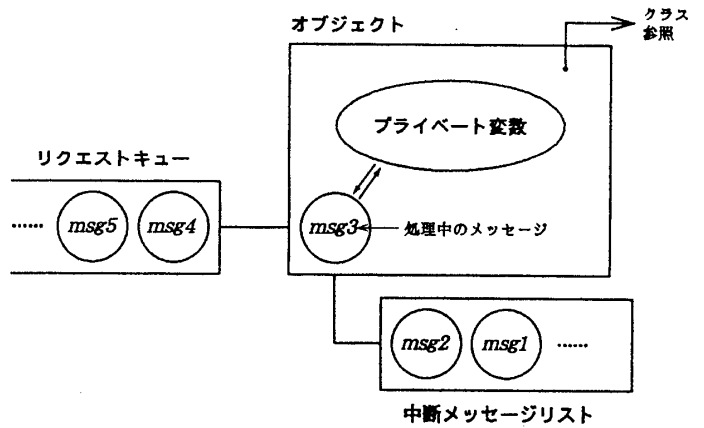


図3 オブジェクトの動作モデル

一つのメッセージだけが処理中となることができ、これによりプライベート変数への排他アクセスが実現されている。待ち条件が成立して再開可能となったメッセージは、リクエストキュー内の未処理メッセージより優先的に処理される。

6. 実装方式

本プラットフォームでは、プレゼンテーション動作全体を制御する実行制御部として Conteman 言語のインタプリタを実装した。読み込まれたシナリオ（クラス定義や動作スクリプト）はある仮想機械の機械語の形に変換された後、解釈実行される。上記のような並列・同期制御や時間制御の機能は、インタプリタ内にスケジューラを内蔵し、オブジェクト単位のスケジューリングを行うことで実現している。

7. おわりに

対話型プレゼンテーションプラットフォームにおいて、プレゼンテーションの手順（シナリオ）を記述するための言語 Conteman と、その処理系を開発した。並列・同期制御や時間制御の機能、豊富なシステムクラス等を備えることにより、DTPRアプリケーションの作成を容易にしている。

参考文献

- 1) 大谷：DTP/DTPR：日経バイト，1989.9
- 2) 柳，他：対話型プレゼンテーション・プラットフォームの全体構想：情処44回全国大会（1992.3）
- 3) 里山，他：対話型プレゼンテーション・プラットフォームにおけるユーザインターフェース管理方式：情処44回全国大会（1992.3）

```
# クラス定義
#
class SampleSheetClass {
  superClass SheetClass
  private button
}

# シート
# ボタン

# メソッド定義
#
method SampleSheetClass init(color)
{
  button := new PushButtonClass
  send button linkObject(self, 'clear, color)
  send self child(button) # ボタンを配置
  send self map() # シートを表示
}
```

図2 ユーザクラス定義の例