

視覚的ソフト開発環境 vic

4S-6

(2)アニメーションマニュアル概要

山本里枝子, 小川知也, 中山裕子, 市川 至

(株)富士通研究所

1.はじめに

ワークステーションの発展に伴い、ソフトウェア開発の効率化を目的として図形情報を活用する試みが広く行なわれている。我々は図形情報、文字情報、それらを用いたアニメーションにより実現される動きの持つ情報などを総括して視覚的情報と呼び、視覚的情報をソフトウェア開発の全工程に有効利用するための枠組を提案し、これに基づくソフトウェア開発環境をvicと呼んだ[1]。

vicの適用例として、アニメーションによるマニュアルを想定した。マニュアルは、仕様書やプログラムと比較して入力情報がより表示的であり、意味を深く考慮する必要が少ない。また、マニュアルでは用意された情報をユーザがどのように利用するかという、「視覚情報の参照」が重要であるのに対し、仕様記述やプログラム記述ではその記述自体が持つ意味をどのように視覚化するかという、「視覚情報の入力」が重要となる点が異なる。我々は、まず視覚情報を利用する立場から視覚情報の意味を考慮しない表示レベルで検討し、後に意味を扱うレベルへの拡張を行なうアプローチを取る。

本稿では、開発したアニメーションマニュアルのプロトタイプに関して、システムの概要を報告する。

2.アニメーションマニュアルシステムの概要

アニメーションマニュアルシステムは、ワークステーションのマルチウィンドウ上で文字情報や図形情報を含むアニメーションを行なう。システムの基本構成の概要を図1に、出力画面例を図2に示す。

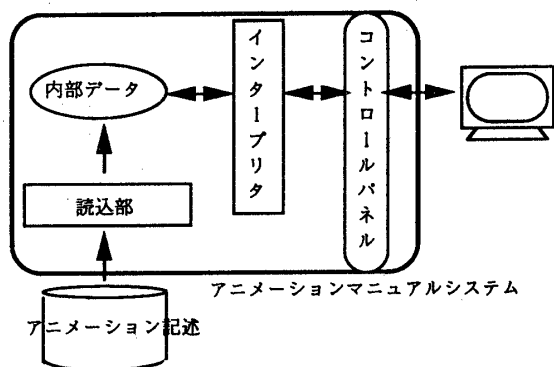


図1. システムの基本構成

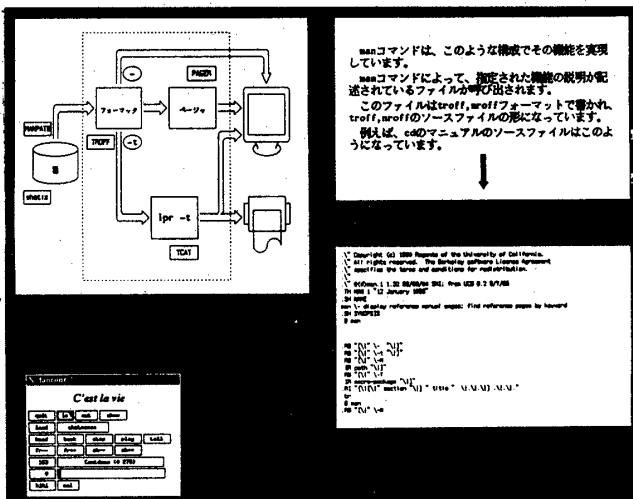


図2. システムの出力画面例

システムはテキストで記述されたアニメーション情報を与えられ、それを読み込み部によって内部データに変換し、ユーザからのアニメーション操作によりディスプレイ上にアニメーションを行なう。

プロトタイプには、視覚的ソフト開発環境 vic における、表示解析機構で内部的に保持される仮想的なフィルムイメージから表示画面レベルまでの、記述および機構が実現されている。内部記述層と意味記述層、およびそれらに関する機構は存在しない。

3.アニメーション記述

アニメーション記述は、映画フィルムのワンシーンに相当する「ショット」と、映画のシナリオに相当する「コンテ」の二つの枠組により実現されている。

3.1 ショット

ショットはアニメーションの記述単位であり、「図形」、図形に対する「操作」、およびそのショットの画面を制御する「カメラワーク」に関する情報が記述されている。

図形や操作はオブジェクト指向風にクラス・インスタンスの概念に基づいて管理される。例えば、ある長方形を動かす場合は、長方形を意味する図形クラス **box** からインスタンス **box1** を生成し、移動を意味する操作クラス **move** のインスタンスを **box1** に適用する。図形には全て名前を定義し名前により管理する。操作は名前を持たず、その操作の開始・終了時間により図形のデータへの適用を管理する。

```

box box1 90 220 100 20
move box1 120 128 0 20
    
```

Visual Software Development Environment vic (2):

Conspectus of Animation Manual

Rieko YAMAMOTO, Tomoya OGAWA, Yuko NAKAYAMA,  
Itaru ICHIKAWA

Fujitsu Laboratories LTD.

図形は、あらかじめいくつかの基本図形が用意されており、図形の種類と必要な属性パラメータとを記述する。基本図形には、幾何図形（直線、矢印、リンク、長方形、円、楕円弧、凸多角形）、文字列、ビットマップがある。これら基本図形を階層的に組み合わせることで合成図形を作ることができる。

図形の属性には、全図形共通の共通属性とそれ以外の各図形固有の固有属性がある。共通属性には、各図形の名前、図形的位置、拡大縮小率、可視性、線幅、線種、塗りつぶしパターン、重なり順がある。固有属性は例えば長方形の場合には、長方形の幅と高さである。

操作は、基本操作として、移動、拡大縮小、可視性／線幅／線種／塗りつぶしパターン／重なり順の変更が用意されている。操作も図形と同様に共通属性と固有属性を持ち、共有属性には、操作対象の図形の名前、操作の開始時間と終了時間があり、固有属性には、例えば移動の場合にはx軸y軸各方向の単位時間あたりの移動量を指定する。

カメラワークは、CAMERA という予約された名前の長方形の大きさや位置を指定／変更することで実現している。

### 3.2 コンテ

コンテは、ショットを組み合わせることでアニメーションを構築するものであり、ショットの時間的空間的關係を記述する。

コンテでは、「ショット」、ショットを結び付ける「グルー」、ショットを表示する「ウィンドウ」を用いる。現在、ショット、グルー、ウィンドウの關係を図3のようにとらえており、これは視覚的ソフト開発環境 **vie** のコンテの仮想的なフィルムイメージでの一モデルである。グルーにはショット間をつなぐ道具として様々な機能が考えられるが、ここでは、一つのウィンドウ上のショットの順序を決定するグルー、複数ウィンドウのショット間の同期をとるグルーにより、マルチウィンドウ上のアニメーションを一本のフィルムのように扱っている。

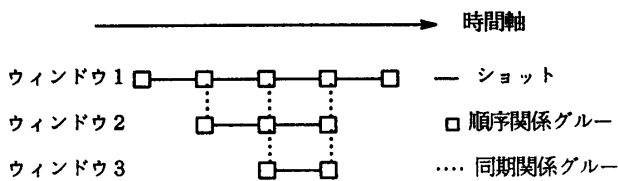


図3. ショット、グルー、ウィンドウの關係

コンテには、以下のように、ウィンドウの位置と大きさ、各ウィンドウ毎のショットの順序、複数ウィンドウ間のショットの同期、ショット表示時のベル音の指定を記述する。

```
glueOpen stage1 640 400 100 100 ; ウィンドウの指定
glue kai1 stage1 kaisetsu1 ; ショットの順序関係
glue kai2 stage1 kaisetsu2 ; ショットの順序関係
glueSync kai2 ex1 ; ショット同期関係
bell ex1 ; ベル音の指定
```

表示するショット群の有無によりウィンドウの開閉を自動的に制御する。ショットはひとつのテキストファイルであり、ウィンドウ毎の順序関係指定時に固有の名前を与え、この名前前で管理することで、アニメーションで一つのショットを自由に再利用することを可能にしている。

マルチウィンドウを利用する場合、どのウィンドウに注目すべきか、また、いつどこに新しいウィンドウが現れるかをユーザに示唆する機構が求められるが、指定ショット表示時にベル音をならすことで現在は対処している。

### 4. アニメーション操作

アニメーション操作は画面上のコントロールパネルのボタンをクリックして行なう。コントロールパネルを図4に示す。アニメーションの再生／巻き戻し(play / back)、停止(stop)、先頭／末尾への移動(head / tail)、コマ送り(fr++ / fr--)、ショット間の前向き／後向きの移動(sh++ / sh--)などの、基本的な表示制御機能を提供している。

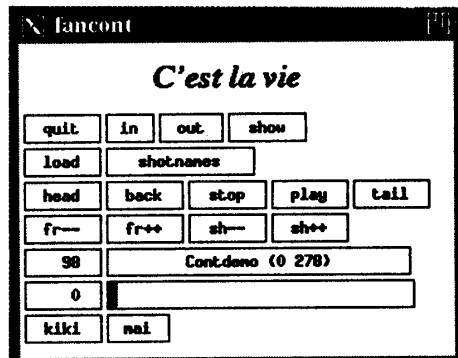


図4. コントロールパネル

### 5. まとめ

アニメーションマニュアルのプロトタイプの概要を述べた。プロトタイプは、視覚的情報を表示レベルで記述し、一本の仮想的フィルムイメージを作成してアニメーション表示する。アニメーションマニュアルとしてでなく、プレゼンテーションシステムとしても使用できる。マルチウィンドウ上にアニメーションを組み立てる枠組として、「ショット」と「コンテ」を用意した。本システムを用いていくつかのアニメーションマニュアルを実現している[2]。

現在は最小限度の表示操作を実現した段階であり、例えば、ユーザが画面上の図形をクリックし他の場面に遷移するというような、柔軟な制御を順時拡張していく。また、視覚的情報の内部記述と意味記述およびそれらに関する機構へ拡張することが将来の課題である。

#### [参考文献]

- [1] 小川, 山本, 中山, 市川: 「視覚的ソフト開発環境 vie (1) 構想」, 本大会予稿, 1991.
- [2] 中山, 小川, 山本, 市川: 「視覚的ソフト開発環境 vie (3) アニメーションマニュアル適用例」, 本大会予稿, 1991