

動的記述を取り入れたユーザインタフェース記述言語の開発

1 Q-1

宮尾 淳一†

坂上 雄一‡

† 広島大学総合科学部

‡ 広島大学大学院工学研究科

1 まえがき

最近、グラフィカルユーザインタフェース (GUI) の利用容易性と現実感を増すために、動きのある表示動作を取り入れることが試みられている。例えば、アイコンからウィンドウへ展開する場合やウィンドウ上のボタンを指示した場合などである。しかし、動きのある表示動作は実時間性を持つため、その記述や実現に多大な労力を要することが問題となる。

本稿では、このような GUI を構築するための記述言語について考察し、特に、その記述に動的な要素を取り入れること、および、その記述と表示を可能にする記述支援ツールについて説明する。

2 動的記述と GUI への応用

動きのある表示動作の代表的な例として、アイコンがウィンドウへ展開される場合がある。また、図1のウィンドウの各構成要素において、ボタンが押された場合の表示や、メニューの現われ方、スクロール動作など、ほとんどの要素が特定の動きと速度を持っている。また、このような動きと速度を適切に構成、制御することにより、利用容易性や現実感が増大すると考えられる。ただし、これまでの GUI では CPU の処理能力や OS の制約により十分な表現が行なわれていなかった。現在、前者の CPU の処理能力の問題に関しては、著しい性能向上により解消されつつある。

ところが、動きのある表示動作を従来の記述を静的に行なう GUI 構築ツールやプログラミング言語で行なうことは困難であり、具体的には、次のような問題点がある。

1. 記述の困難さ: 速度などを数値で記述するためにどのような数値が適当か求めるのが困難である。
2. 理解の困難さ: 記述されたものを見ても、実際に動作させなければその表示動作が分からない。
3. 表示の困難さ: 動きの速度を制御する直接的な方法が提供されていないため、実際に表示することも困難である。

筆者らは、3の問題点に関して既に基礎的な考察を行い、時間制御された描画技法を開発している⁽¹⁾。そこで、本研究

では動的な記述を導入することにより残りの問題点解消を目指す。上述の問題点は、本来動きのある動作の記述を静的に行なうために生じたものであり、動的な記述により、直接的、かつ、容易に記述できることが期待される。

さらに、本研究では動作速度が一定でない場合の記述も考慮する。例えば、ある動作の動作開始から終了までの加速度と速度が図2に示すようになったとする。この場合は、ゆっくりと動作が開始され、段々に速くなり、最後は速度を緩めて終了する。このような記述を可能にすることを考える。

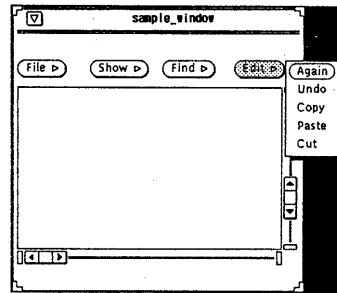


図1 ウィンドウとその構成部品

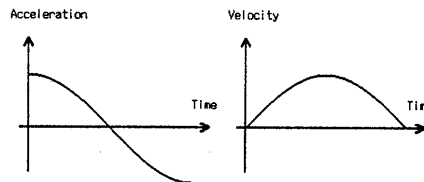


図2 処理開始から終了までの加速度と速度

3 動的記述を用いた GUI 記述言語の概要

現在、動的な記述も含めて GUI 全般を記述する視覚言語を文献(2)の FML 言語を基礎として、開発中である。この言語では、まず GUI で用いられるウィンドウ、メニューなどの各部品を定義し、次にそれらに対応するアイコンを節点としたグラフで結合して GUI の処理を定義する。この言語において動的な記述を必要とするのは主に部品の定義である。以下では、部品の定義について説明する。

Development of a Specification Language with Dynamic Specification for User Interfaces

Jun'ichi MIYAO and Yuichi SAKAUE

Hiroshima University

GUIの部品としては、OPEN LOOKTMスタイルガイド⁽³⁾においてウィンドウ、アイコン、ボタン、メニュー、フィールド、スライダ、ゲージ、などがある。これらの中で、特に動的記述が必要となるのはウィンドウであるため、ウィンドウの定義を用いて動的記述について説明する。

ウィンドウの記述には図3に示すようなウィンドウを利用する。このウィンドウの上半分がウィンドウパラメータを設定する静的記述部分であり、下半分が動的記述部分である。

1. 静的記述部分：ウィンドウの種類、ヘッダー、フッター、スクロールの有無などを記述。
2. 動的記述部分：アイコンからウィンドウへの展開動作 (Opening) とその逆動作 (Closing)、およびスクロール速度やアイコンの動作などを記述。

図3では、静的記述部分でデフォルトのウィンドウとコン

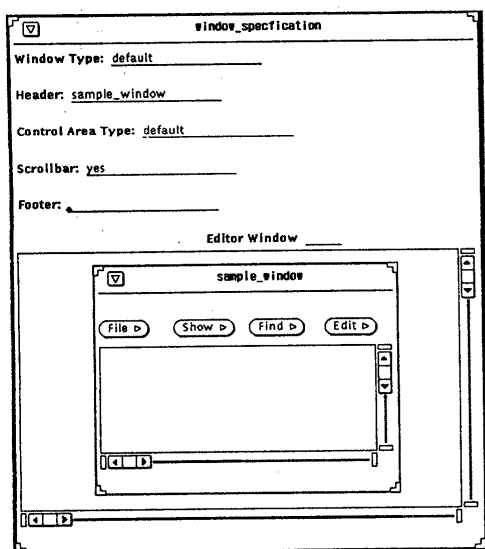


図3 ウィンドウの記述

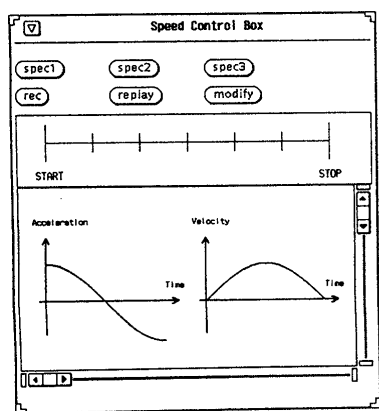


図4 マウスによる動作速度の記述

トロールエリアが選択されている。また、動的記述部分では定義中のウィンドウが表示されスクロールの速度やメニューの展開速度などを記述できる。

スクロールの速度を記述する場合は、図4に示す速度設定ウィンドウを用いてスクロール開始から終了までの一定でない速度の記述を動的に行なうことができる。具体的には、同図のSTART, STOP というスケールの間をマウスマウスが移動するときの各単位時間ごとの位置を測定して、加速度と速度を求める。

4 動的記述支援ツール

ここでは、前節で述べた記述を実現する動的記述支援ツールについて説明する。動的記述を実現するためには、設定した時間経過後に処理を開始する機能が必要である。この機能は基本的には設定時間が経過したとき必要なシグナルを発生させることにより、実現可能である。

このシグナルを用いた動的記述支援ツールの構成を図5に示す。同図において、GEが描画データを作成し、TEがMVMを用いて時間データの作成を行なう。また、TCの時間制御に従いVMが画面上への描画を行なう。これらのモジュールで、MVMとTCが前述のシグナルを用いるモジュールである。

本支援ツールはSun-4/IPX上でSunOS4.1.1Bと日本語Open Windows2.0.1Bを用いて試作中であり、応答速度などについて実験中である。ここで、SunOSはリアルタイムOSではないために、処理の中断あるいは多少の遅れが出ることがある。また、今後、X-Windowへの移植も計画している。

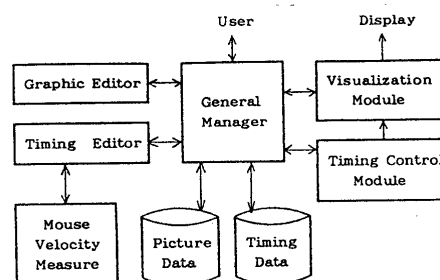


図5 動的記述支援ツールの構成

なお、本研究の一部は平成4年度文部省科学研究費補助金奨励研究(A)(課題番号04750344)による。

文 献

- (1) 宮尾, 坂上: “プログラム動作の視覚化表示機能をもつプロトタイピングシステムの開発”, 信学技報 OS91-63 (1992).
- (2) Miyao, J. et al.: “Visualized and modeless programming environment for form manipulation language”, Proc. IEEE Workshop on Visual Languages, pp.99-104 (1989).
- (3) Sun Microsystems, Inc.: “OPEN LOOK スタイルガイド”, Addison-Wesley (1990).