

自動再生ツールの実現

4H-1.1

馬場恵子* 水崎拓* 本多英明*

*(株) 沖テクノシステムズ ラボラトリ

1 はじめに

プログラムの品質向上のためには、プログラム試験、システム試験などの試験工程を十分に行うことは重要なことである。

しかし何度も同じ操作を繰り返して試験をする場合において、人間の手のみで行った場合、無駄な時間が多くなってしまふ。

そこで我々はマウス操作やキー入力などのイベントを記録し、自動で再生するツールを、Windows3.1を持つマシン上で開発した。

本稿では記録・再生機能の実現方法、そのほかの付属機能について述べる。

2 自動再生ツールの概要

本ツールは、以下の機能を持つ。

- 記録機能
マウスのクリック、ドラッグ、キー入力などのイベントを記録し、ファイルに書き込む。
- 再生機能
記録ファイルから情報を読み込み、再生を行う。
- そのほかの機能
指定した画像をウィンドウ単位で取り込み、ビットマップに保存する画像取り込み機能や、2つの画像を比較し、差分を表示・保存する画像差分検出機能。

3 記録機能

3.1 実現方法

マウスカーソルの移動やマウスのクリック、キー入力を行った場合、WM_MOUSEMOVE や WM_LBUTTONDOWN などのあらゆるメッセージがアプリケーションに渡されるが、そのメッセージ渡しは、ドライバから Windows にイベントとして渡されることが原因となっている。

本ツールでは、ドライバからのイベントをフックすることにより、記録機能を実現した。

フックをセットする Windows 関数で WH_JOURNALRECORD のフックを使用してイベントを取り出す。そのイベントを文字列に変換してファイルに記録するようにしている。

3.2 記録ファイル

フックしたイベントを、1 イベント 1 行で文字列に変換し、ファイルに出力する。

記録ファイルの例

```
;コメント  
(jump 558 174 0) ;マウスカーソルが(558,174)に移動  
(press button1 500) ;マウスをクリック  
(release button1 0)  
(jump 742 334 5)  
(press "a" 3) ;キー a を入力  
(release "a" 0)  
(jump 742 334 11)
```

再生するとき内部で Lisp 処理系を用いるので、文字列は Lisp の形式で記録する。

イベントの実行時間は、直前のイベントからの待ち時間を記すことで示した。単位は 1/100 秒である。

例) (press button1 500) のイベントは、直前のイベント (jump 558 174 0) の 5 秒後におきる。

4 再生機能

4.1 実現方法

Lisp 処理系が、記録したファイルを 1 イベントずつ読んで解釈し、フックで処理することにより、再生機能を実現した。

フックをセットする Windows 関数で WH_JOURNALPLAYBACK のフックを使用し、フィルタ関数で記録したイベントを供給し、再生を行う。

4.2 ファイルの編集

ファイルを編集することにより、繰り返しなどの制御ができる。

次に示す例は、前出の記録ファイルの内容を 3 回繰り返すように編集したものである。

Implementation of automatic trace tool
Keiko BABA*, Hiroshi MIZUSAKI*
Hideaki HONDA*

*Oki Technosystems Laboratory, Inc.

編集した記録ファイルの例

```
(dotimes (i 3)
  (jump 558 174 0)
  (press button1 500)
  (release button1 0)
  (jump 742 334 5)
  (press "a" 3)
  (release "a" 0)
  (jump 742 334 11))
```

また、出力コマンドをファイルに記入することにより、デバッグ文などを出力することができる。

例えば、princ という出力コマンドを使用し、(princ "Error") という行を記入した場合、"Error" という文字列を表示用ダイアログボックスに表示する。

4.3 トレース機能

トレース用ダイアログボックスに再生中のファイルを表示し、実行中の行を反転表示にする。この機能を用いると、現在実行中のイベントを知ることができる。

4.4 再生の応用

Lisp 処理系がイベントを読み込んで解釈する間の動作を操作することにより、通常の再生以外に、以下に示すような再生を行うこともできる。

- 一時停止

再生中に一時停止キーや一時停止のショートカットキーを入力すると、その後のイベントを読み込まずに実行を中断する。これにより、任意の位置で再生を一時停止することができる。

- 再生速度変更

イベントの実行時間を示す値を調整することにより再生速度を変更することができる。

例) 最速実行は、上記の時間がどんな値でも強制的に 0 にすることにより実現している。

- 疑似カーソルでの再生

再生中は、マウスカーソルが再生に使用されているので、通常は使用することができない。そこで、マウスの移動(ドラッグ操作は除く)操作をアプリケーションが使用することは少ないので、マウス移動のイベントを実際に起こさず、その位置に偽のカーソルを表示することにより、本物のマウスカーソルの操作を可能にする。

5 そのほかの付属機能

記録・再生のほかに、画像を取り込む機能、画像の差分を表示・保存する機能がある。

画像取り込み機能は、記録中や再生中にも用いることができる。

記録時と再生時の画像を取り込んでおけば、2つの画像の差分を画像差分機能で検出することができるので、試験時のデバッグに役立てることができる。

6 レコーダーとの比較

Windows には、レコーダーというアプリケーションが付属でついている。

一連の操作を記録・再生するという点では、本ツールとレコーダーの機能は類似している。

レコーダーはマウスマクロ的な手法により、記録・再生を行う。

表 1 に、本ツールとレコーダーとの比較を示す。

表 1. レコーダーとの比較

	本ツール	レコーダー
記録ファイルの形式	テキスト形式	バイナリ形式
記録ファイルの制御	高級言語 Lisp を使用	不可能
記録ファイルの可視性	内容を確認できる	不可能
記録ファイルの編集	エディタで編集できる	不可能
繰り返し機構	最小の操作単位で設定可能	一連の作業単位のみ繰り返し可能
不必要操作の削除	相当部分を削除可能	不可能
複数のアプリケーションへの対応	複数のウィンドウを扱える	単一のアプリケーションのみ
隠れたウィンドウでの再生	自動的にウィンドウを前面に移動	再生不可能
画面イメージの保存	ウィンドウまたは、全画面を保存	不可能
画面イメージの比較	ウィンドウまたは、全画面で可能	不可能
デバッグ機構	再生時にメッセージを表示可能	なし
イベント(メッセージ)の構造化	関数定義やモジュール化が可能	一連の操作の組み合わせのみ可能
再生速度の変更	標準、高速度、低速で再生可能	標準と高速度のみ

7 おわりに

今後の課題として、操作の記録を Windows が扱うオブジェクト(ボタンなど)単位にし、ファイルを見たらどんな動作を記録したかがわかるような記録・再生方式にすることなどが挙げられる。

参考文献

- [1] Microsoft WindowsTM Version 3.1 ソフトウェア開発キット プログラマーズリファレンス, Microsoft Corp., 1993.