インターフェイスの街角 (62) — Flash によるサーバー通信

増井俊之

いろいろな Web ブラウザで、Macromedia[1] の Flash ムービーのプラグインが利用できるようになってきました。Windows や Macintosh だけでなく、Linux でも標準的に使えますし、最近は CLIE や PocketPC などの携帯端末でも Flash ムービーの再生が可能なものが増えつつあります。

Flash では、時間軸に沿ったアニメーションを表示するだけでなく、**ActionScript** というスクリプト言語を組み合わせた実行制御や対話的な処理もおこなえます。Flash のバージョンアップとともに ActionScript も進化を続けており、現在は通常のスクリプト言語と遜色ないほどになっています。

Flash は、見栄えのよい Web ページを作るための技術として、現在はおもにデザイナーによく使われているようです。しかし、アニメーション・システムとしてだけではなく、データの視覚化ツールとしても利用すれば、応用範囲が大きくひろがるはずです。たとえば、検索サーバーの入出力として HTML の代わりに Flash を使えば、比較的簡単に美しい出力結果が得られるでしょう。

Flash の開発環境

Flash ムービーの開発には、Macromedia のソフトウェアを使うのが一般的で、そのための解説書もたくさん出ています。

しかし、Macromedia のソフトウェアは高価ですし、統合開発環境のような GUI を使用するため、ActionScript などのテキスト・プログラムを記述する場合には、かならずしも便利とはいえません。また、人間が対話的にムービーを作成することが前提になっているため、ムービーの自動作成ができないという問題もあります。

一方、いろいろな情報を自動的に Flash に変換できるのなら、各種の動的なデータを Web ブラウザ上で視覚化する際に便利でしょう。

Flash の仕様は 1998 年 4 月に公開されたため、現在では Macromedia 以外のベンダーや個人が作成したさまざまなツールで Flash ムービーが作れるようになりました。たとえば、数値データを自動的に Flash ムービーのグラフに変換するフリー・ソフトウェアや市販のパッケージなどもあります。

Flash ムービーを自動的に生成するシステムの 1 つとして、Dave Hayden 氏が開発した Ming[2] というライブラリがあります。Ming ライブラリは C、PHP、Perl、Ruby、Python などの言語から呼び出せるので、これらの言語で書いたプログラムで Flash を容易に生成することができます。これならば、通常のテキストエディタや Makefile を使って Flash ムービーを作成できますし、CGI や PHP などで自動的にムービーを生成して表示させるといったことも可能です。

Ming

Ming については、2002 年 8 月号の「横着プログラミング」に解説がありますが、今回は Ming の Ruby バインディングである Ming/Ruby[3] を使って説明します。

1 http://www.macromedia.com/

2 http://ming.sourceforge.net/

3 Ming の配布パッケージにも Ruby バインディングが含まれていますが、Ming/Ruby は池上大介氏が独自に開発したものです。
http://www.aist-nara.ac.jp/~daisu-ik/ruby/ming/

図 1　移動する正方形を生成する Ming/Ruby プログラム

```
# moverect.rb
require 'ming/ming'
include Ming

# ムービー生成
movie = SWFMovie.new
movie.set_dimension(400,400)
movie.set_rate(96)

# 正方形を生成
shape = SWFShape.new
fill = shape.add_fill(0x00,0x00,0xff)
shape.set_right_fill(fill)
shape.move_pen_to(10,10)
shape.draw_line_to(10,-10)
shape.draw_line_to(-10,-10)
shape.draw_line_to(-10,10)
shape.draw_line_to(10,10)

# ムービーに図形を追加
item = movie.add(shape)

# 図形を移動させつつ動画フレームを生成
(1..100).each { |i|
  item.move_to(i*2,50)
  movie.next_frame
}

(1..100).each { |i|
  item.move_to(200-i*2,50)
  movie.next_frame
}

movie.save("moverect.swf")
```

Ming による簡単なアニメーション生成

Ming/Ruby の使用例は、片山俊明氏のページ[4]にたくさん紹介されています。簡単な例として、青い正方形が左右に移動する Flash ムービーを生成する Ming/Ruby プログラム moverect.rb を図 1 に示します。

Ruby でこのプログラムを実行すると、moverect.swf という Flash ムービーのファイルが生成されます。これを Flash Player で再生すると、図 2 のように左右に動く正方形が表示されます。

Ming による ActionScript 記述

Flash には、あるフレームが表示されたりユーザーが特定の操作を行なった時点で、ActionScript で記述したプログラムを実行できる仕組みがあります。

4 http://nite.kuicr.kyoto-u.ac.jp/~katayama/ming/

図 2　Flash Player で moverect.swf を再生

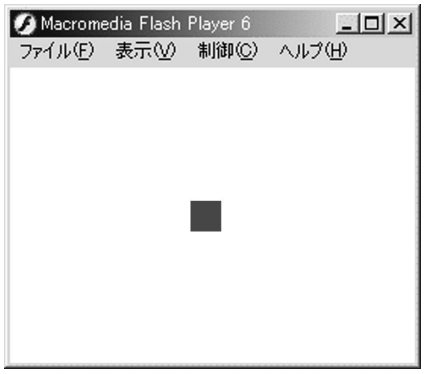

図 1 のようなプログラムでは、せいぜい紙芝居的なアニメーションの作成くらいしかできませんが、ActionScript を使えば、ユーザーの操作に応じて動きを変えたり、色やサイズを自動的に変えたりするといったことも可能になります。

図 3 は、図 1 のプログラムにすこし手を加えたものです。ActionScript を追加することで、ランダムな位置に正方形を表示する Flash ムービーを生成します。

ActionScript のプログラムは、SWFAction.new の引数として指定されており、ムービーの最初と 2 番目のフレームに割り当てられています。

また、図 3 の action1 への代入を図 4 のように変更すると、正方形がマウスとともに移動するようになります。

Flash とサーバーの連携

Ming や ActionScript の機能を使うと、各種のサーバーと Flash を連繋させて使うことができます。

Flash の自動生成

Ming をサーバーで実行させれば、ユーザーからの要求に従って動的に SWF ファイルを生成し、視覚化することができます。

たとえば複雑なシミュレーションや大規模な計算をおこなったりするサーバーにおいて、ユーザーからの要求に従って計算した結果を Ming で処理して Flash ムービーとして返すようにすれば、普通の Web ブラウザでも大量の情報を動的に表示できるようになります。

図 3　ランダムな位置に正方形を表示する Ming/Ruby プログラム

```
# moverect.rb
require 'ming/ming'
include Ming

# ムービー生成
movie = SWFMovie.new
movie.set_dimension(400,400)
movie.set_rate(10)

# 正方形を生成
shape = SWFShape.new
fill = shape.add_fill(0x00,0x00,0xff)
shape.set_right_fill(fill)
shape.move_pen_to(10,10)
shape.draw_line_to(10,-10)
shape.draw_line_to(-10,-10)
shape.draw_line_to(-10,10)
shape.draw_line_to(10,10)

# 移動のためのムービークリップ生成
movie_clip = SWFMovieClip.new
movie_clip.add(shape)
movie_clip.next_frame
clip_item = movie.add(movie_clip)
clip_item.move_to(200,200)
clip_item.set_name("rect")

# 正方形をランダムな場所に表示する
# ActionScriptを生成して最初のフレームに追加
action1 = SWFAction.new("
  rect._y = Math.floor(Math.random() * 200);
  rect._x = Math.floor(Math.random() * 200);
")
movie.add(action1)
movie.next_frame

# 最初のフレームに戻るActionScriptを
# 生成して二番目のフレームに追加
action2 = SWFAction.new("
  prevFrame();
  play();
")
movie.add(action2)
movie.next_frame

movie.save("moverect2.swf")
```

図 4　マウスに正方形を追随させるスクリプト

```
action1 = SWFAction.new("
  rect._y = _ymouse;
  rect._x = _xmouse;
")
```

loadVariablesによるActionScriptからのCGI 呼び出し

ActionSctipt の loadVariables() 関数を使うと、ローカルファイルの内容を読み取ったり、リモートサーバーの CGI を起動して結果を取得し、ActionScript の変数の値として代入することができます。

サーバーが返す情報の量があまり多くなければ、サーバー側の情報をこの方法で取り出して ActionScript で処理し、ムービーとして視覚化することができます。この場合はサーバーで Ming を実行する必要がありませんし、異なるデータに対しても同じ Flash ムービーが使えます。

図 5 のプログラムでは、ActionScript の loadVariables() 関数を用いてサーバーからデータを取得しています。ここでは CGI を呼び出す URL を指定していますが、その下のコメント行にもあるように、この部分でローカル・ファイルシステム上のテキストファイル名を指定すれば、そのファイルを代わりに使うことができます。

CGI の出力結果やテキストファイルとしては、〝変数名=文字列〟というテキストを〝&〟でつないだ形式のものを使います。

図 6 のような CGI を呼び出すと、

```
text=text_from_server
```

という文字列が返りますが、ActionScript からこの CGI を呼び出すと ActionScript の text という変数に〝text_from_server〟という値が入ります。

この Flash ムービーを実行すると、図 7 のような結果が得られます。

サーバーとの TCP 通信

loadVariables() を用いたサーバーから情報を取得する方法は、新しいデータが必要になるたびに CGI を呼び出すことになるのであまり効率的とはいえません。

Flash には、XML を解釈する Web サービスと通信するための XMLSocket という機能が用意されています。これを利用すると、サーバーとのあいだで XML データを TCP によってリアルタイムにやりとりできます。

XMLSocket 機能を使用すれば、Flash ムービーとサーバーのあいだを TCP で結び、リアルタイムに XML データをやりとりして表示を更新することもできます。

ActionScript では、以下のようにして XML サーバーと通信をおこないます。XML データを用意して XML-

図 5　サーバーの CGI を呼び出す ActionScript

```
require 'ming/ming'
include Ming

movie = SWFMovie.new
movie.set_dimension(200,40)
movie.set_rate(12)

text = SWFTextField.new(0)
font = SWFFont.new("Techno.fdb")
text.set_font(font)
text.add_string("initial string")
text.set_height(20)
text.set_name("textfield")

movie.add(text)

action = SWFAction.new(<<EOF
  url = "http://www.csl.sony.co.jp/⇒
         person/masui/data.cgi";
  text = "";
  this.loadVariables(url);
  // ローカルファイルを読み出す場合は以下のように変更
  // this.loadVariables("data.txt");
EOF
)
movie.add(action)

movie.next_frame
movie.next_frame

action = SWFAction.new(<<EOF
  if(text == ""){
    gotoAndPlay(1);
  }
  else {
    textfield = text;
    stop();
  }
EOF
)
movie.add(action)

movie.save('loadvar.swf')
```

図 6　サーバーの CGI プログラム (data.cgi)

```
#!/usr/bin/env ruby
require 'cgi'
cgi = CGI.new("html3")
cgi.out { 'text=text_from_server' }
```

図 7　Flash Player で表示したところ

Socket オブジェクトの send メソッドでサーバーに送出すると、結果が返されたときに onXML で定義された関数が呼び出されます。

```
function func(data) {
//  …… XMLデータ受信時に呼び出される関数
}
Host = "www.example.com";  # サーバーホスト
Port = 12345;              # ポート番号
# XMLSocketオブジェクト生成
socket = new XMLSocket();
socket.onXML = func;
socket.connect(Host,Port);
// ……
socket.send(xmldata); # サーバーにXMLデータ送信
```

将来、Web を利用する各種サーバーの多くは XML 形式でデータをやりとりするようになると思われますが、この方法では既存の各種のサーバーが使えません。XMLSocket() の onXML の代わりに onData にコールバック関数を定義すれば、単純なテキストを扱う従来型のサーバーにも対応できます。

たとえば、図 8 のプログラムでは、図 9 のような単語サーバーと TCP で通信する Flash ムービーを生成することができます。Flash ムービーは send メソッドでマウスの XY 座標をサーバーに送信し、サーバーから返された単語を onData で受け取ってマウスの位置に表示しています。

この Flash ムービーを実行すると、図 10 のような結果が得られます。マウスを動かすと、サーバーから返された単語がその場所に表示されます。

インターフェイス作成支援ツール

現在、C や Java などのプログラミング言語で GUI アプリケーションを作成するには、イベントドリブンの GUI ツールキットを使う手法が一般的なようです。一方、

図 8　単語サーバーと通信する Flash ムービーの生成

```
require 'ming/ming'
include Ming

movie = SWFMovie.new
movie.set_dimension(400,400);
movie.set_rate(96)

text = SWFTextField.new(0)
font = SWFFont.new("Techno.fdb")
text.set_font(font)
text.set_line_spacing(4)
text.add_string("                    ")
text.set_height(20)
text.set_name("text")

text_sprite = SWFSprite.new
text_sprite.add(text)
text_sprite.next_frame
text_clip = movie.add(text_sprite)
text_clip.move_to(200,200)
text_clip.set_name("textclip")

initaction = SWFAction.new(<<EOF
Host = "localhost"; // server host name
Port = 5678;        // server port

prevmouse = 0;

function serverevent(src)
{
  textclip.text = src;
}

socket = new XMLSocket(); // Socket object
socket.onData = serverevent;
socket.connect(Host,Port);

// On receive data function

socket.send(string(int(_xmouse)) + ' ' + ⇒
            string(int(_ymouse)) + "\n");

EOF
)

movie.add(initaction)
movie.next_frame

action1 = SWFAction.new(<<EOF
  textclip._y = _ymouse;
  textclip._x = _xmouse;

  if(_ymouse != prevmouse){
    socket.send(string(int(_xmouse)) + ' ' + ⇒
                string(int(_ymouse)) + "\n");
    prevmouse = _ymouse;
  }
EOF
)
movie.add(action1)
movie.next_frame

action2 = SWFAction.new("
  prevFrame();
  play();
")
movie.add(action2)
movie.next_frame

movie.save('wordsearch.swf')
```

図 9　マウス位置を受け取って単語を返すサーバープログラム

```
# server.rb

require "socket"

gs = TCPserver.open(5678)
addr = gs.addr
addr.shift
printf("server is on %s\n", addr.join(":"))

words = []
File.open("/usr/lib/words","r"){ |f|
  words = f.readlines
}

while true
  # save to dynamic variable
  Thread.start(gs.accept) do |s|
    print(s, " is accepted\n")
    while str = s.gets
      md = /(\d+)\s+(\d+)/.match(str)
      x = md[0].to_i
      y = md[1].to_i
      ind = (x / 5).to_i * 400 + y
      word = words[ind]
      word = "" unless word
      s.write(word + "\000")
    end
    print(s, " is gone\n")
    s.close
  end
end
```

図 10　図 9 の実行結果

動画や音声を多用するオーサリング・システムを使う場合は、動画や音声のデータに ActionScript のようなスクリプトを付加するという考え方が主流になっているようです。

これまで、インタラクティブなシステムを簡単に記述するための手法について、いろいろな研究がおこなわれてきました。システムの状態遷移をベースとしてインタラクションを記述する手法が数多く考案されてきましたが、Director や Flash の場合は、ムービーとして表現される連続的な状態遷移に対話的なプログラム要素を加えるという方法でこれを実現していることになります。

イベントドリブンのツールキットを使っていても、アニメーションや状態遷移のような機能を追加したくなることがあります。これらのアプローチには一長一短がありますし、どちらかを利用したとしても、最終的にはあらゆる機能をもつ汎用言語とムービー操作ライブラリの組合せになってしまうため、システム全体が巨大化するおそれがあります。状況に応じて必要な機能をうまく組み合わせて使えるようなインターフェイス作成支援ツールの登場が望まれますが、私自身は 2002 年 10 月号で紹介したような並列処理の記述を応用すればいいのではないかと考えています。

おわりに

今回はごく単純な例しか紹介しませんでしたが、Ming や ActionScript をうまく組み合わせて活用すれば、比較的簡単にいろいろなサーバーのフロントエンドとして使えることがお分かりいただけたのではないでしょうか。

現在の Web 上の検索サーバーのほとんどは、動的検索 (Dynamic Query) をサポートしていないので、検索ボタンなどを押すことによってはじめて検索が実行され、結果が表示されます。しかし、さきほど例に挙げた単語表示ムービーのような手法を導入すれば、Flash ムービーと検索サーバーのあいだで TCP による通信をおこないつつ、動的に検索結果を表示させることが可能になるため、Web 上での対話的検索システムの構築に有用でしょう。

最近は、CLIE などの携帯端末でも Flash や ActionScript が軽快に動きます。Flash で作成したアプリケーションは、原理的には Windows や Linux、携帯端末などで動かせるはずです。適当なアプリケーション・サーバーを用意すれば、ワープロやメーラー、表計算などの機能をすべて Flash で実行することも不可能ではないでしょう。

現在も、Flash を使った Web 上の掲示板はあるようです。しかし、掲示板などは CGI を利用する場合と機能的にあまり違いがなく、Flash の利点を十分に活かしているとはいえません。一方、動的に検索結果を表示する検索システムであれば、魅力的なコンテンツになると思われます。

今後、各種の Web サービスのフロントエンドとして、魅力的な画面を容易に作れる Flash の利用が進むかもしれません。

（ますい・としゆき　ソニー CSL）