

マルチメディアスクリプティング環境におけるアイコンプログラミング

5N-2

今関幸一 池田浩彰 芝野耕司
東京国際大学 商学部 経営情報学科

1. はじめに

AV機器制御を含めたマルチメディアスクリプティング環境下でのアイコンを中心としたビジュアルプログラミング言語の提案を行う。このプログラミング環境では、AV機器を利用したプレゼンテーション、講義などでの利用を想定し、シナリオベースの実行だけでなく、状況に応じて、柔軟にプログラムの変更が行えることを目的とする。

また、ドラッグアンドドロップを用いることによって、直接操作によるプログラムの作成を可能にする。

2. アイコンプログラミングの利点

AV機器等のマシン制御などにおいては、プログラムの実行を予め用意された内容を順にこなして行くだけではなく、実行の経過状況や周りの動きに対しての柔軟な対応を迫られる。

そのためには次にどんな作業が行われるのかといったプログラムの流れを容易に把握できなくてはならない。そこでプログラム言語の主要構成要素をアイコンで表現し、それらをビジュアル化された制御構造で繋げていくことで流れを表す。そうすることで、これからの作業の移り変わりを一目で理解できるようにする。

また、アイコンプログラムの利点として、制御の分かりやすさだけでなくその実行形態にも見ることができ。アイコンプログラムでは各オブジェクトがアイコンであるからそれら各々の実行は勿論であるし、プログラム実行中のアイコンの入れ替え等によるプログラムの直接編集が可能となっている。

つまり、まずマウスによる簡略化された操作を可能としたスクリプティング環境の用意。アイコン及びビジュアル化された制御構造を中心とした、プログラム流れの理解しやすさ。そしてまた周りの状況に則した対話型の実行を実現させるのである。

3. 制御構造のビジュアル化

プログラムの基本構成要素である制御構造は順次処理、分岐、繰り返しである。これらのビジュアル化はスクリプトボード上に流れを線によって表示する。これらの生成や設定など、その詳細を以下にそれぞれ説明する。

3.1. 順次処理

この処理は、このシステムを起動したときにスクリプトボード上に用意される。左から右へ一本の線として流れている基本ラインである。この基本ライン上に主要要素のアイコンを並べていくことによってプログラムを組み立て、そのラインの流れに沿って左から右にアイコンが実行される。

3.2. 分岐

ツールの分岐ボタンを選択し分岐を行いたい部分の基本ラインをクリックすると、そこでの条件分岐が行える。分岐点には論理式をドロップする。また上段の基本ラインが真、下段の枝別れしたラインが偽の流れを示すため、分岐点から次の流れ合流点までにそれぞれ真偽の処理をアイコンを並べ表わす。

このときの手順は、基本ライン上に論理式をドロップしただけでも、ツールの分岐を選択したのと同じ状態になる。

3.3. 繰り返し

繰り返しも分岐と同様に、ツールの繰り返しボタンを選択し、繰り返し部分を指定することで行う。繰り返しの始点と終点にはそれぞれループ端が表示され、内部の基本ラインはプログラムを見やすいよう、一段低く階層的に表示される。繰り返し範囲の変更はループ端をマウスで引張ることによって可能である。繰り返しは次の三つに分けられる。

a) 前条件ループ

始点のループ端へ、判別式、論理式をドロップ。

b) 後条件ループ

終点のループ端へ、判別式、論理式をドロップ。

c) 指定回数ループ

始点、終点いずれかのループ端へ式、あるいは数値を持つ変数をドロップ。

4. 主要要素のアイコン化

コマンド、手続き、変数などといった言語主要構成要素をアイコンで表現する。

アイコンには、処理内容を表した絵、処理タイトルといった既存の概要の他に、アイコンの中身を表示させるためのオープンスイッチをその右上部に付随させた。これによりアイコンのダブルクリックによる実行だけでなく、オープンスイッチのクリックによる、各アイコンに応じた内容あるいは操作パネルといった各種情報の表示がされるのである。

以下ではアイコンとなる主要要素の各々について説明する。

4.1. 処理のアイコン化

基本となる処理アイコンはコマンド及び手続きに該当する。このそれぞれ特定の処理を行うアイコンをスク립トボードの基本ライン上にドラッグ&ドロップすることで、先に述べたビジュアル化された制御構造と連結させプログラムを組んで行く。

処理アイコン群には予め用意された基本コマンドとしてのアイコン群の他に、ユーザ定義によるアイコン群がある。このアイコンの作成方法は大きく次の三つに分けることができる。

a) アイコン群+制御構造のグルーピング

アイコンをスク립トボード上に並べることでプログラムを表現することは幾度も触れたが、この時、一つの処理を表すアイコンが複数集まることで大きな処理、つまり手続きになる。これらの複数のアイコンのグルーピングによって、一つのアイコンとしてまとめることを可能とする。

b) テキストスク립トエディタによる生成

基本コマンドのアイコン群及び上記グルーピングをすれば確かにプログラミングは可能なのだが、これは、ある程度の大きさをもつ仕事のアイコン化に有効であり、まだ一定の水準を超えていない小さい処理の段階から利用するには、かえって作業を難解なものにしてしまう。そのため、ビジュアルベースよりも、むしろテキストベースでの記述の方が理解、作業効率の面で適している小さな処理は、テキストスク립トエディタを用意し、その結果、処理がある程度の仕事として確立した上でアイコンを生成させる。

c) 特定処理のための簡易エディタによる生成

ある特定の決まった処理をさせるようなアイコンを生成する場合、例えばビデオの1シーンを部分再生する等のアイコンを生成させる場合、それらアイコン生成をマウス操作だけでできる支援ツールにあたる簡易エディタを用意し、それを使用することで、スク립トを組むといった煩わしさから解放される。

4.2. 代入文の生成

代入文は、処理結果を格納する変数と、変数及び処理を行う計算式や関数のアイコンで表現する。

代入文は一行だけの手続きであるからテキストエディタでの生成も可能ではある。が、簡略化のために代入用のテンプレートを用意し、ここに変数アイコンや式をドロップする事で行う。このテンプレートは矢印でアイコンの相互関係を表している。

4.3. 計算式・論理式の生成

同様に式を表現するための、それぞれの型に合わせた演算子をテンプレートとして用意する。このテンプレートにそれぞれ、変数アイコンや式をドロップする。テンプレートを用意する基本演算子は以下の通り。

- a) 数値 +、-、*、/
- b) 文字 +
- c) 判別 <、>、=
- d) 論理 and、or、not

4.4. 変数のアイコン化

変数アイコンはオープンスイッチのクリックによって変数オブジェクトのための操作パネルが開かれる。変数操作パネルは、アイコンの表示で出てくる変数名や値情報の他に、変数の型や初期値といった情報を操作するものである。

変数アイコンの生成は生成用のボタンをクリック時に表示される変数操作パネルで指定すればデスクトップ上に変数アイコン生成される。勿論、他のアイコン同様にコピーし、必要なパラメータを書き換えるだけでも可能である。

また、変数の有効範囲は変数アイコンのディスプレイ上の格納場所をビジュアル的に示すことでわかりやすいものとする。つまりはオブジェクトごと階層的に敷かれたプレートの上に変数アイコンを乗せ、配置することで、その変数がどの範囲のスコープを持つのかを表現し、かつプレートごとの移動やコピーを可能とすることで他のオブジェクトでの利用等の指定も簡単に行えるようになる。

5. 対話型の実行

周りの状況に即した機械制御などの処理を考えたとき、対話型の実行が必要になる。予め想定した作業の流れだけでなく、実際のプログラム実行中の直接編集を行わねばならない。

このアイコンプログラムの場合、アイコン単体での、つまり、作業単体での実行が可能なのはもちろんである。それだけでなく、プログラム実行中の、アイコンの入れ替えや、追加、削除といったことも可能とする。アイコンの追加は追加したい箇所にドラッグ&ドロップすることで簡単に再編成される。またプログラムの直接編集だけでなくマウス指定によるプログラムの実行箇所を変更する制御移行も可能とすることで、より対話形式の実行を行いやすいものとする。

6. 終わりに

プログラム言語の主要構成要素をアイコンで表現し、それらをビジュアル化された制御構造で繋げていくことで流れを表す、アイコンプログラミングの提案を行ってきた。今後、ビジュアル言語としてのさらなる深化を検討していく予定である。

【参考文献】

- [1] Apple Computer Inc. , INSIDE MACINTOSH QuickTime Components (日本語版), アジソン・ウェスレイ, 1994.
- [2] 赤松正行, 漢字Talk7プログラミング マルチメディアプログラミング BNN, 1993.
- [3] 池田浩彰: 今関幸一: 湯浅聖記: 高橋幸信: 堀口雅美: 芝野耕司, アイコン化とスク립トエディタを備えた教育用AV制御システム, 第50回情報処理学会全国大会, 1995.