

オーサリングシステム Extempore — スクリプト言語

3S-4

五味 弘* 山田 雅彦* 水崎 拓* 長坂 篤**

*(株) 沖テクノシステムズ ラボラトリ

**沖電気工業 (株)

1 はじめに

我々が開発しているオーサリングシステム Extempore^[2] のスクリプト言語について述べる。

オーサリングシステムを用いてマルチメディア・アプリケーションを作成するとき、図形や動画などのメディアオブジェクトを自由に定義でき、その動作や複数のオブジェクトの関連を記述できる言語が必要である。

このときに求められる言語機能として、(1) インタプリタ実行 (2) オブジェクト指向機能 (3) コンパクト (4) 可搬性 (5) イベント処理機能などがある。

さらにオブジェクトが他のオブジェクトとともに関連を持って自律的な動作記述を行なえるようにエージェント指向機能を実現した。これにより、すべてのオブジェクトの関係を記述するのではなく、注目するオブジェクト間の関係のみを記述することにより、大規模なマルチメディア・アプリケーションの記述が容易になった。

本稿ではスクリプト言語の言語仕様および処理系の実現について述べる。

2 特徴

オーサリングシステム Extempore のスクリプト言語の特徴を以下に示す。

1. Lisp 系言語

核部分は DylanTM [1] に準じた言語仕様であり、インタプリタとコンパイラを作成した。

2. イベント処理機能 (割り込み処理機能)

タイマー、キー、マウスなどのデバイスイベントやユーザイベントを処理するための割り込み機能を持っている。

3. メディア処理機能

マルチメディア・オブジェクトを操作するため

の豊富な抽象化ライブラリを持っている。

4. オブジェクト指向機能

総称関数をベースにした多重継承、メソッド結合を持つオブジェクト指向機能を作成した。

5. エージェント指向機能

一つ一つのオブジェクトが自律的に動作する機能を持っている。

6. コンパクト、可搬性

マルチメディア・アプリケーションを単独で動作させるときのために、スクリプト言語は実行時では機能を切り離してスクリプト処理系を小さくできる機能を持ち、異機種でも動作するように設計している。

スクリプト言語のエディタの GUI を図 2 に示す。

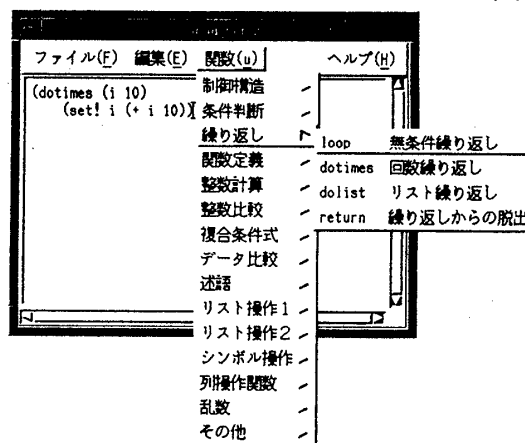


図 1 スクリプトの GUI

3 イベント処理機能

イベントには (1) デバイスイベントと (2) ユーザイベントがある。デバイスイベントには、タイマー、マウス移動、マウスボタンのクリック、キーの押下などがある。

ユーザイベントには、エージェント記述の「関係」^[3]がある。関係は、特定または不特定のオブジェクト間の関係とオブジェクトとその集まりである環境との間の関係がある。この関係をユーザ定義のイベントとして定義する。

Authoring System "Extempore" — Script Language
Hiroshi GOMI*, Masahiko YAMADA*,
Hiroshi MIZUSAKI* and Atsushi NAGASAKA**

*Oki Technosystems Laboratory, Inc.

**Oki Electric Industry Co., Ltd.

これらのイベントが発生したときに、スクリプト言語処理系は割り込みを行ない、それぞれの割り込み処理を行なう。

4 メディア処理機能

マルチメディアオーサリングシステムのスクリプト言語であるため、各種のメディア処理機能を持っている。メディアには、テキスト、図形、動画、音声がある。

例えば、図形の回転、縮小、拡大、半透明などの各種効果や MPEG などの諸機能呼び出す抽象化されたライブラリを持っている。

これらの機能をクラスライブラリとしてユーザに提供する。

5 オブジェクト指向機能

各メディアを容易に扱うためにオブジェクト指向機能が必要であり、これを實現した。

本処理系のオブジェクト指向機能は、(1) 多重継承 (2) メソッド結合 (3) 動的変更 の機能を持っている。また高速化のためにフリーズ機能を実現した。

フリーズ機能により動的機能を制限して高速化を計っている。以下のものを用意している。

1. freeze-methods

メソッドの変更を許さず、固定にする。

freeze-methods は DylanTM とほぼ同じ。

2. fix-accessers

引数として与えられたクラスのアクセッサを含む総称関数の変更を許さず、固定にする。

3. fix-generic-function (make-read-only)

総称関数にメソッドの追加、変更を許さず、固定にする。

DylanTM の make-read-only と同じ。

その他に freeze-accessers, freeze-constructors を用意している。

6 エージェント指向機能

本スクリプト言語ではエージェント指向機能を提供している。

エージェント単位でインタプリタ、キュー、メソッドを持ち、メッセージは特定または不特定のエージェント間の関係または環境との関係で表現される。不特定のエージェントに対する関係は、間接通信である。

関係はデバイスイベントと同様に扱える。以下に関係の例を示す。

(is-near agentA Candidates)

特定のエージェント agentA と不特定のエージェント Candidates が関係 is-near にあるかどうかをパターンマッチして調べる。Candidates はエージェントの集まり、即ち環境を示す。

関係には、プライオリティとアクションを記述する。アクションには関数または関係を記述する。

インタプリタでは、複数の関係が成立した時の処理や、アクション実行中の関係成立の処理を記述する。特に記述がないときはプライオリティ優先の先着順で動作する。関係やアクションのプライオリティ制御の記述だけでなく、キューに対する操作関数を用意して細かな制御も可能にしている。

以下に関係エージェント・インタプリタの例を示す。

```
(definterpreter small-fish
```

```
  (relation-description
```

```
    (variables: ((Big (candidates <big-fish>)))
```

```
      relation: (is-near self Big)
```

```
      rel-priority: 2
```

```
      action: (run-away self Big)
```

```
      action-priority: 2)
```

```
  (relation: #f
```

```
    action: (random-swim self)))
```

7 おわりに

オーサリングシステム *Extempore* のスクリプト言語について報告した。

各メディアオブジェクトをエージェントとして定義し、クラスライブラリを提供した。動作として各種のイベントドリブンのプログラムが記述でき、またエージェント間の関係をイベントとして記述することにより、マルチメディア・アプリケーションの記述が容易になる。

現在、永続的オブジェクトの實現を行なっており、他のマシンへのエージェントの移出が容易にできるようにする予定である。

参考文献

- [1] Apple Computer: "DylanTM An object-oriented dynamic language", 1992
- [2] 井田、新谷、長坂、五味: "オーサリングシステム Extempore — 概要", 情報処理学会第 43 回全国大会, 3S-1
- [3] 大島、小島、井上、五味、長坂: "オーサリングシステム Extempore — エージェント記述", 情報処理学会第 43 回全国大会, 3S-3