

イベント駆動型動画生成システムEASY

3E-6

花田恵太郎 宮本雅之 佐藤亮一 吉川耕平 千葉徹
シャープ株式会社 技術本部 コンピュータシステム研究所

1.はじめに

コンピュータを用いた動画を生成するシステム EASY (Event-driven Animation System)の記述モデルを紹介する。従来のキーフレーム法等のコンピュータアニメーション作成で一般的に用いられている手法によると、動画のストーリー展開が固定的でありストーリーへのユーザの参加や他のシステムとの同期、動きのシミュレーション等の表現が不可能であるという問題点がある。EASYは動きの記述から動画を生成することによりこの問題点を解消する。

2.系の時間発展の記述技術

動画の登場物や視点の動きと動作の指定を、複数登場物からなる系の時間発展の記述技術ととらえる。

2.1.時間発展の記述方式 I

時間発展の最も直接的な記述方法は、以下に示すように系の状態を決定する状態変数の値の時間発展を時間の関数として指定する方法である。

$$\mathbf{x}(t) = \mathbf{m}(t) \quad (1)$$

ここで時間 t は離散的な値(整数)を取る。キーフレーム法に基づく動画はこの範疇に入る。しかし、この方法では、

i) 登場物の内部状態の変化

ii) 登場物間の相互関係の変化

といったイベントを発生時刻に置き換えて記述する必要があり、記述の持つ意味が不明確になるだけでなく、ストーリーの局所的な変更が時間軸の大域に渡り影響してしまう。また、iii) 系の外部(ユーザや他のシステム)からの刺激に対応した動きの表現が不可能である。

2.2.時間発展の記述方法 II

上記i), ii), iii)を表現可能な記述方法を以下に示す。

$$\mathbf{x}(t+1) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{ex}) \quad (2)$$

$$\mathbf{x}_i(t+1) = \mathbf{f}_i(t, \mathbf{x}(t), \mathbf{ex}) \quad (3)$$

$$(i=1, 2, \dots, n)$$

\mathbf{ex} は、外部から入ってくる刺激を意味する。(2)式は状態変数毎に記述すると(3)式のようになるが、 \mathbf{x}_i の $t+1$ における値に $\mathbf{x}_j(j \neq i)$ の値が関係しており、記述を容易にしていない。記述の独立性を高め、時間発展の記述の容易性、再利用性を高めるため以下に示す記述方法を採用する。

$$\mathbf{x}_i(t+1) = \mathbf{f}_i^*(t, \mathbf{x}_i(t), \mathbf{ev}) \quad (4)$$

$$(i=1, 2, \dots, n)$$

\mathbf{ev} はイベント(\mathbf{ex} も含む)を意味する。 \mathbf{x}_i の $t+1$ における値は \mathbf{x}_i の t における値と t において発せられているイベント(上記i), ii), iii)のイベント)により定まる。言い換えると、動画を複数の登場物(\mathbf{x}_i)がイベント \mathbf{ev} により同期を取りつつ並行動作を行う系ととらえて記述する。

3.並行動作のモデル

ストーリーを持った動画を記述するには、メッセージ名から手続きが一意に定まり、メッセージの送信登場物は受信登場物の行う動作を知っていて、記述した通りの動きが実現する仕事要求通信モデル[1]が適する。

一方シミュレーション等、個別に行動様式を持った自律的な行動からなる動画を記述するには、メッセージ名から一意

に定まる手続きは存在せず、メッセージの判断を自分自身で行うデータ通信モデル[2]が適する。

本章では仕事要求通信、データ通信の双方を可能とする動画記述を行うための並行動作モデルを提案する。アクタと、通信を行うためのデマンド及びイベント、並行動作を実現するための時間の刻みティックが主要概念である。

3.1.アクタ

並行動作を記述する基本的な単位としてアクタという概念を用いる。アクタは複数の動き(プロセス)を同時に行うことができる。アクタはデマンド通信によって受け取ったメッセージの内容に従って後述のメソッドにより記述された動きをプロセスとして起動あるいは終了を行う。起動したプロセスは各アクタが持つプロセスリストに登録され、終了のメッセージを受け取るまで走行状態を保持し、各ティック毎に実行される。

3.2.ティック

ティックは大域的な時刻の最少単位であり、各アクタは1ティックに1動作を行う。同一ティックにおけるアクタの実行順序は結果に影響しない。1ティック中にアクタは以下の動作を行う。

- i) デマンドプール中に保持されているデマンドを取りだし、対応するプロセスを起動又は終了する。
- ii) プロセスリストに登録されているプロセスを実行する。

3.3.デマンドとイベント

デマンド及びイベントはアクタ間でメッセージ通信を行うための概念である。デマンドは前章で述べた仕事要求通信を行うもので、アクタへプロセスの起動及び終了を要求するメッセージ送信である。イベントはデータ通信を行うもので、他のアクタが参照可能なデータを出力する。

1) デマンド

デマンドの送信の形式を以下に示す。

i) プロセス起動要求

send(送信先アクタ名, メソッド名, パラメタ並び)

ii) プロセス終了要求

remove(送信先アクタ名, メソッド名)

デマンド送信は以下の特徴を持つ

- ・指定したアクタにのみ配達される1対1通信
- ・デマンドの到着は1ティック後
- ・プロセスの起動・終了要求

デマンドは送信された時点では送信先アクタがそれぞれ持っているデマンドプールに保持され、次のティックにおいてデマンドプールから取り出されて解釈、実行される。デマンド送信に用いられるメッセージ名は送信先アクタの持つメソッドに対応し、パラメタ並びはメソッドの引数に対応する。デマンドを受け取ったアクタは対応するプロセスの起動又は終了処理を行う。

2) イベント

イベント通信の形式を以下に示す。

i) イベント出力

out(名前, パラメタ並び)

ii) イベント入力

in(名前, パターン並び)

イベント通信は以下の特徴を持つ

- ・宛先を指定しない通信
- ・パターンマッチングによる読み込み

EASY : Event-driven Animation System
Keitaro HANADA, Masayuki MIYAMOTO,
Ryoichi SATO, Kohei YOSHIKAWA, Toru CHIBA
SHARP Corporation

- ・イベントの読み込みは1ティック後
- ・複数のアクタによって読み込み可能
- ・存在時間は1ティック

4. 動画の記述

前章で示した並行動作のモデルをもとに、動画記述のモデルを作成した。

4.1. アクタの分類

動画を生成するためには、場面の各登場物の動きの記述だけではなく、ストーリー展開を表現するために全体の制御を行う必要がある。このような機能を持つ特別なアクタとしてシナリオとウオッチャを導入する。

シナリオアクタのメソッドには動画のストーリーを表現する。すなわち、場面に登場するアクタに対して、どのようなタイミングで、どのような要求通信を行うかを記述する。

シナリオはイベントを受けて全体の動作を制御するが、各アクタの属性を見て、ある事象が起こったと判断してイベントを出力する機構が必要となる。この機能を持つ特殊なアクタがウオッチャアクタである。シナリオ、ウオッチャ以外のアクタ、すなわち画面に登場するアクタをキャラクターと呼ぶ。図1にキャラクター、シナリオ、ウオッチャ間の通信モデルを示す。

4.2. 記述

動画の記述は図2に示すようにキャラクター記述部、シナリオ記述部、ウオッチャ記述部の3つに分かれる。

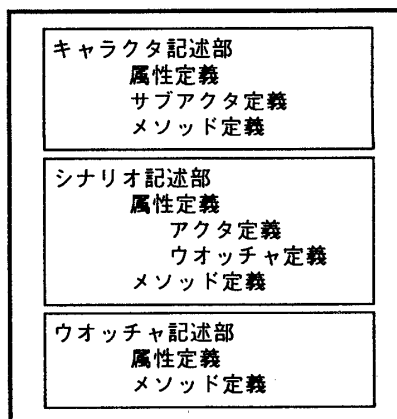


図2 動画記述

4.2.1. キャラクターの記述

キャラクターアクタの記述は属性、サブアクタ、メソッドの定義からなる。

・属性

キャラクターは動画を表示するのに必要な形状、空間における位置や色等の属性を持つ。表現する動画に応じて速度や大きさ等任意の属性をキャラクター毎に定義することができる。

・サブアクタ

キャラクターは他のアクタを部品として階層的に持つことができる。これらのアクタをサブアクタとして定義する。

・メソッド

メソッドは各アクタが実行可能なプロセスの記述である。逐次、繰り返し、条件分岐の制御構造を用いて1ティックにおける属性値の変化の仕方と、デマンド送信、イベント入力出力による他のアクタとの通信を記述する。

4.2.2. シナリオの記述

シナリオアクタの記述はキャラクターの記述と同様、属性とメソッドの定義からなる。シナリオアクタは動画に登場する全キャラクター、及びウオッチャを属性値として持つ。シナリオアクタはここで定義したキャラクター及びウオッチャに対してデマンド通信によりメッセージを送付することができる。

4.2.3. ウオッチャの記述

ウオッチャアクタは以下の特徴を持つ。

・他のアクタの属性値参照

ウオッチャは1つ以上のアクタに注目して、それらのアクタの属性値を与えられた条件式に当てはめ、条件を満たしたときにイベントを発する。従ってウオッチャのメソッドには注目すべきアクタをパラメータとして受け、そのアクタの属性値を直接参照する機能が必要となる。

・外部との同期の記述

外部と同期する動画を記述するには、外部交信専用のウオッチャを定義する。このウオッチャは、1ティック毎の動作で外部からの入力を解釈し、その結果をイベントとして出力、あるいは指定されたアクタへデマンドとして送信する。また、外部へ適当なメッセージ出力も行う。これにより動画記述における外部との交信部分を局所化できる。

5. まとめ

動画を登場物の並行動作系ととらえ、その記述を行うモデルを提案した。このモデルは次の特徴を持つ。

- 1) 他に従属せずに動作する独立性の高いアクタと、登場アクタのストーリーを表現するシナリオにより動画を記述する。
- 2) アクタの動きの同期のためにデマンドとイベントという2種類の通信機能を提供する。デマンドは確定的な動きの表現に適し、イベントはアクタの自律的な動きの表現に適する。これらの特徴により、
 - 1) アクタの部品化による、動きの再利用
 - 2) 記述時には発生時間を決定できない動きの表現が可能となる。

本研究は、通商産業省工業技術院大型プロジェクトの一環として(財)情報処理相互運用技術協会(INTAP)が新エネルギー産業技術総合開発機構(NEDO)からの委託をうけ、シャープ株式会社がINTAPからの再委託研究として実施したものである。

参考文献

- [1] Reynolds, C. W.: Computer Animation with Scripts and Actors, Computer Graphics, Vol. 16, no. 3, pp. 289-296 (1982).
- [2] 内木, 丸一, 所: 行動シミュレーションに基づいたアニメーションシステムParadise, コンピュータソフトウェア, Vol. 4, No.2, pp. 24-38 (1987).

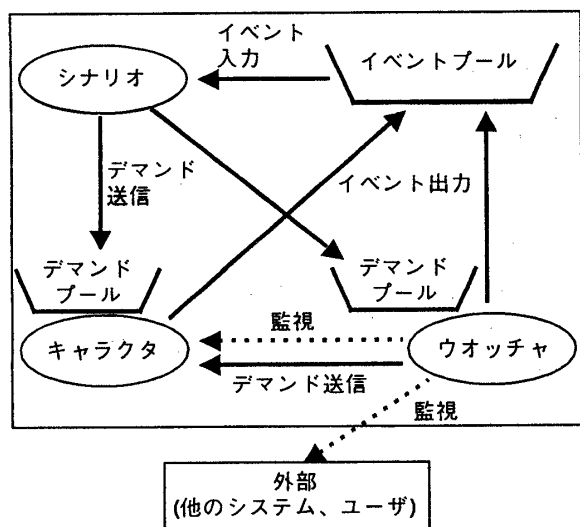


図1 各アクタ間の通信モデル