

制約伝播を使った 3 次元モデルの構成とストリームを使ったアニメーションの表現 — 「制約」 による新しいアニメーションの方法 —

1 背景及び目的

近年、3 次元コンピュータグラフィックスの技術は非常に高いレベルにまで達しており、映画などのメディアにおける映像表現の技法として、また現代アートの表現媒体として、従来のものにとって代わるほどのスタンダードなものとなった。

しかし他方、家庭用テレビゲームのようなユーザと対話的に動作するメディアに関しては、実行時の処理の難しさから、まだまだ貧弱な表現しか出来ていないと言わざるを得ない。

特に、ゲームキャラクターのアニメーションについては、デザイナーが制作したアニメーションシーケンスを、プログラム実行中にただ単調に繰り返し再生するだけであり、丁寧に作られた人間のキャラクターであっても、機械的な印象が強くなってしまふことが多い。

コンピュータグラフィックスを使って映像をつくり出す過程では、技術的にもアートのにもさまざまな技法が組み合わされるが、本プロジェクトではこれらのうち、3 次元モデルの形状のアニメーションと対話的な入出力の取り扱いにスポットを当て、従来とは違った視点でシステムを再構成し、技術者やアーティストにとってより使いやすいものを提案することを目的としている。

2 開発内容

本プロジェクトでは *MEPHISTO* と題して、以下に示すいくつかのコンセプトを、Scheme 処理系 *Gauche* のライブラリとして実装した。

2.1 ストリームによるアニメーション

時刻を表す整数 n と、画面に図形を描画する手続き $proc$ の対を使って、 $(n . proc)$ という形式で**フレーム**（アニメーションの 1 コマ）を表現する。そして、**アニメーション**をこのフレームの無限**ストリーム**として表現する。無限ストリームとは無限の長さを持つリストである。概念的には、次のような構造の無限リストとして扱うことができる。

$$((1 . p_1) (2 . p_2) (3 . p_3) \dots) \quad (1)$$

このようにしてストリームとして表現したアニメーションを、連続した映像として画面上に出力するのである。

2.2 制約システム

制約システムとは、いくつかの変数の間にある関係を関数や方程式の形で記述し、**制約解消系**によって実行時に自動的に必要な変数の値が計算されるような仕組みのことである。*MEPHISTO* では独自のシンプルな構造の制約解消系を用いて、3次元モデルの構成やアニメーションを実現している。

モデルの記述

人体のような関節を持つ3次元モデルを表現する際、一般的には、木構造を使ってモデルを表現することが多い。木構造は確かに合理的な表現方法ではあるが、ノード間の関係が一方向的であるために制限も多い。

そこで *MEPHISTO* では従来の木構造を捨て、全てのノードをフラットに扱い、ノード間の関係は制約システムを使って記述する。そのモデルを特徴づける各種の方向ベクトルや、その他の**モデル固有のパラメタ**も含ませることができる。

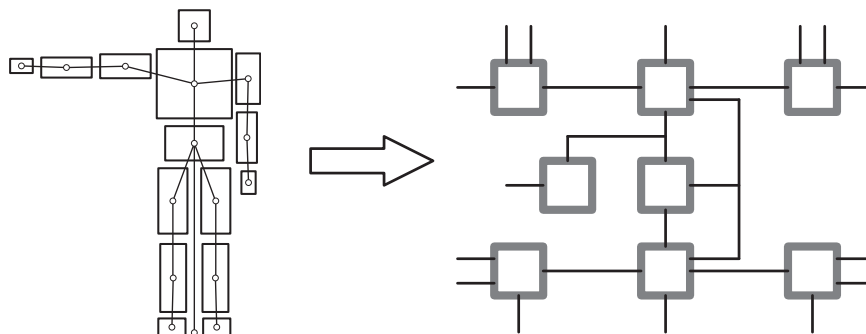


図1 木構造による表現と制約システムによる表現

木構造による表現と制約システムによる表現の概観を比較すると図1のようになる。

制約解消系

制約解消系は、図2のような構造をしており、図の中の矩形で表された**制約**と、線で表された**ワイヤ**からなる。ワイヤはアプリケーションで使われる変数に結びつけられ、制約によって値が設定されたり値を消去されたりする。

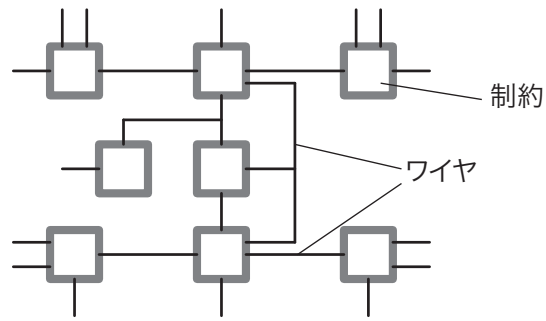


図2 MEPHISTOの制約解消系

制約としてのアニメーション

図3のようにモデルの状態を時刻と結びつける制約を設定すれば、簡単に宣言的にアニメーションが表現できる。

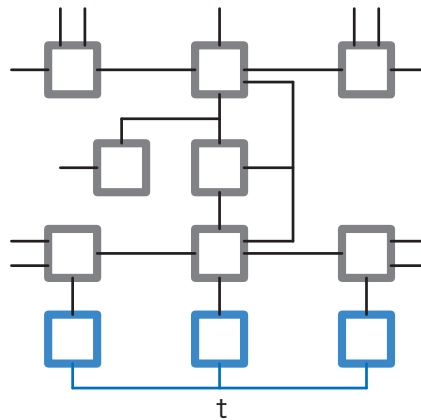


図3 アニメーションを表現する制約。時刻を表すワイヤ t に他の部分が結び付いている

3 従来の技術（または機能）との相違

制約システムとストリームを使用したユニークな構造

MEPHISTOでは、従来のゲームプログラミングで一般的に使われるデータ構造やデータフローのアルゴリズムを根本的に見直すところから始めた。

制約システムとストリームというユニークな組合せを使って、3次元モデルの表現や対話的なアニメーションを実現している。

汎用性と実行効率を重視した制約システム

MEPHISTOの制約システムは、3次元グラフィックスの計算に用途を限らない汎用的な構造をしており、扱えるデータ型やアルゴリズムにも制限はなく、アプリ

ケーション側で自由に実装することができる。

また、制約解消の際に、不要な探索や計算量が予め分からないような処理を避けるため、必ず一意な解が定まるような制約を与えるようアプリケーションに要求する。これにより、制約システムによる処理のオーバーヘッドを最小限にとどめ、実用的な実行効率を実現することを狙っている。

ストリームを使った統一的な状態遷移の扱い

アプリケーションを作る際、デバイスの状態やアプリケーションの内部状態など、さまざまな状態の遷移を扱う必要があるが、これらに対してストリームという統一的方法でアクセスすることができる。これにより、実行環境に依存する部分を隠蔽することができ、より可搬性の高い実装が可能になる。

4 期待される効果

本プロジェクトで提案するアニメーションの表現方法が確立すれば、おもに家庭用テレビゲームの開発において、これまで困難とされていたゲームキャラクターの自然な動作が可能になることが期待される。また、アーティストにとって表現できる範囲がひろがり、これまでにないタイプのゲームやメディア・アートの表現が可能になるので、新しいタイプのコンテンツの登場も期待できる。

5 普及（または活用）の見通し

MEPHISTO の直接的な利用者は、インタラクティブなコンテンツの制作においてプログラミングを行う技術者であるが、現状では Scheme による実装しかないため、製品レベルの制作に用いることは考えにくい。今後は Scheme という言語処理系の枠を越えて、PC 以外のプラットフォームにも対応させることを予定しているが、それが実現すれば新しい表現手法として、製品レベルのコンテンツの制作現場に届くこともあり得ると考えている。

6 開発者名（所属）

久井 亨（フリーランス）

プロジェクトページ <http://sourceforge.jp/projects/mephisto/>