

スクリプト自動生成に基づくビジュアルな behavior 作成手法

1 A A - 3

中川 滋雄 石田 博文
 NEC ヒューマンメディア研究所

1.はじめに

インターネットにおける対話的な 3D コンテンツの記述フォーマットとして VRML2.0 [1]が登場し、教育/シミュレーション/ゲームや各種のサービスのユーザインタフェースへの利用などが期待されている。VRML では 3D オブジェクトの動作(behavior)を Java(JavaScript)で記述するが、このためコンテンツ作成者が新たにプログラミングを習得しなければならないという問題がある。筆者らはこうした負担を軽減するために、behavior をビジュアルプログラミングにより作成する手法を開発しすでに報告した[2][3]。提案した手法を含めて、アイコン的なビジュアルプログラムの実行には一般に専用の実行環境を用いている。今回、筆者らは VRML ブラウザ上でビジュアルプログラムとして作成したコンテンツを実行させるためのフォーマットの変換手法を開発したので報告する。

2. 動作単位リスト形式の behavior 記述

筆者らの提案する behavior 記述手法では、3D オブジェクトの「一動作」を一個のアイコン(動作単位)に対応させ、その動作の「開始条件」も併せて記述する。動作の種類としてオブジェクトの移動軌跡やスケーリングの時間変化などを設定し、開始条件として特定のイベントの発生(マウスクリック、オブジェクトの特定動作の終了など)を設定する。アイコンは動作の「状態」に対応し、その遷移は動作の実行順に接続したアイコンのリストで記述する(動作単位リスト)。複数の 3D オブジェクト間の動作の同期はメッセージ通信により設定する。動作単位をアイコンとして編集対象とすることで、ビジュアルなオーサリング

ツールを構築できる(図1)。このツールではコンテンツ作成者がオブジェクトの動作の遷移を把握しやすいという利点があり、特に実行中に対応するアイコンを表示することが可能なため、テストやデバッグが容易となる。

3. 動作単位リスト形式 behavior の VRML への変換

ビジュアル behavior オーサリングツールで作成した動作単位リスト形式の対話型 3D コンテンツを VRML 形式に変換する方式について述べる。まず、VRML の提供する behavior モデルについて概説し、VRML ファイルおよびスクリプトファイルの生成の方式を述べる。

3.1. VRML の behavior モデル

VRML では behavior を記述するために、(1)ユーザのマウス操作や設定時間の経過などのイベントを検出する Sensor ノード、(2)外部の behavior プログラム(スクリプト)を指定する Script ノード、(3)ノード間のイベント配送経路を宣言する ROUTE、の各機構が新たに導入された。例として TouchSensor ノードがユーザのマウスクリックイベントを検出して Script ノードに配送し、これが behavior を記述したスクリプトである Java バイトコードに配送され、スクリプト内部で 3D オブジェクトの位置の計算を行うモデルを示す(図2)。このモデルでは計算結果をフレームタイムごとに 3D オブジェクトの translation フィールドに配送することでアニメーション表示を行っている。

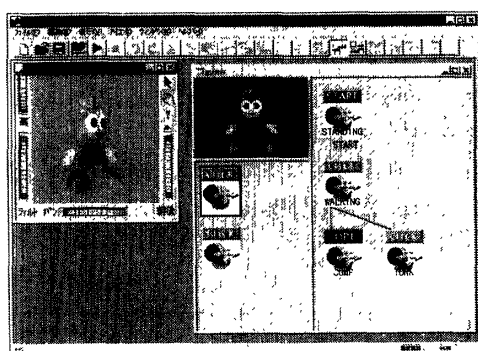


図1: ビジュアル behavior オーサリングツール

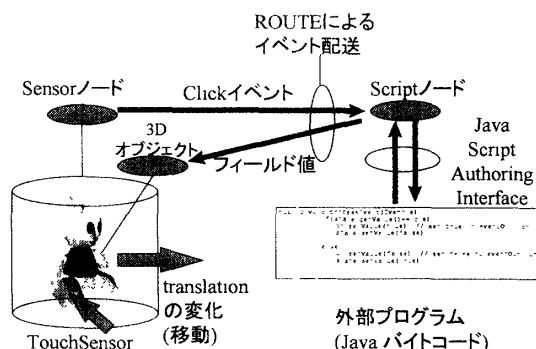


図2: VRMLのbehaviorモデル

Visual Behavior Authoring Method based on Automatic Script Code Generation

Shigeo Nakagawa and Hirofumi Ishida

Human Media Research Laboratories, NEC Corporation

3.2. VRML ファイル(3D シーン+ROUTE)の生成

ビジュアルオーサリングツールで記述したコンテンツから以下の記述を含む VRML ファイル(*.wrl)を生成する。

3D シーン記述: 3D シーンにはオブジェクトの形状、配置、光源・カメラ情報が含まれる。オブジェクトごとにノードを対応して生成し、各ノードごとに TouchSensor ノードと Script ノードを付加する。

ROUTE 記述: TouchSensor と動作終了イベント[2]に関して配送経路を検索して生成する。動作終了イベントは Script ノード間で通信を行うイベントであり、新規に定義して使用する。

3.3. Java スクリプトプログラムの生成

スクリプトプログラム(*.java)として以下を含むテキストファイルを生成し、JDK の javac コマンドで byte code(*.class)にコンパイルする。

動作データ: オブジェクトの回転・平行移動・スケールリングなどの動作をあらわす行列データを配列定数として生成する。

動作単位リスト: 動作単位を記述するクラス(配列)を生成する。動作単位間のリストの接続関係は配列のインデックスで指定する。

イベントハンドラ: 動作単位クラスを1個ずつ読み出し、イベントの発生ごとに解釈と動作の実行を行う処理を行うコードを生成する。イベントハンドリングには VRML の Java バインディングで規定されている processEvent()メソッドを使用し、イベント種別に応じて対応するメソッドを実行する(リスト1)。

4. 実装と評価

上記方式に基づき、動作単位リスト形式のコンテンツを VRML 形式に変換するツールを開発した。変換したコンテンツは市販の VRML ブラウザ[4]で動作を確認した。

いくつかのコンテンツで動作を比較した結果、Script ノード数が多くなるほどほどブラウザの反応速度が低下する傾向があった。

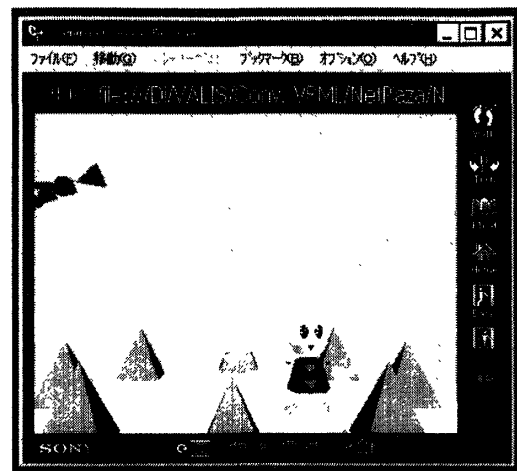


図3: 変換コンテンツの VRML ブラウザでの実行例

このことから VRML ブラウザにとって Java の実行処理はかなりの計算時間を必要とする処理であると予想できる。このため、スクリプトプログラムをより最適化すれば、全体の処理速度がさらに向上することが期待できる。

5. おわりに

behavior を「動作」の状態遷移として記述した対話型 3D コンテンツを VRML ブラウザ上で実行するためのデータ変換手法の開発を行った。コンテンツ作成者はビジュアルなツールを利用して動作の遷移として 3D オブジェクトの behavior を作成し、その結果をテキストプログラミングを行わうことなく VRML コンテンツとして生成して利用できる。今後は Java ソースコードの最適化を図る予定である。

参考文献

- [1] Virtual Reality Modeling Language,
<http://vag.vrml.org/>
- [2] 中川、石田「対話型アニメーションにおけるアクタ動作のビジュアル記述方式」、52 回情報処全国大会、分冊 2、pp.361-362、1996
- [3] 石田、中川「3次元インタラクティブアニメーションのビジュアル作成手法」、NICOGRAPH96 論文集、1996
- [4] CommunityPlace(TM) Browser, SONY,
<http://vs.sony.co.jp/>

```
public void processEvent(Event e){//Script ノード経由でイベントが渡される
String name = e.getName();
if(name.equals("actorEventIn")){//動作終了イベントに対応するメソッド
actorEventIn((ConstSFVec3f )e.getValue(),e.getTimestamp());}
if(name.equals("clicked")){//TouchSensor イベントに対応するメソッド
clicked((ConstSFBool )e.getValue(),e.getTimestamp());}
if(name.equals("move")){//一定時間経過ごとの位置座標の計算と出力
move((ConstSFTime )e.getValue(),e.getTimestamp());}
}
```

リスト1: スクリプトプログラム イベントハンドラ部 (抜粋)