

RVEシステムでの仮想環境記述言語

1H-9

吉田 亮、小出 昭夫

日本アイ・ビー・エム(株) 東京基礎研究所

1 はじめに

前出の報告にあるように、今回我々が開発中の Reactive Virtual Environment (RVE) システムでは仮想世界中における時間経過の概念を重視し、描画時のフレームレート・画質をコントロールしながら、VR環境中に物体間、あるいは、物体・仮想世界体験者間での相互作用やその結果の動きといった要素を提供することを目的としている。本報告では、RVEシステムにおいてこれらを表現するファイルフォーマット、仮想環境記述言語について述べる。

2 言語概要

仮想世界の記述に関しては、これまでも世界記述言語 vef [1] を通して検討を行ってきた。が、ユーザがこの種の言語を利用して新たなアプリケーションを開発するとした時、その言語がそれまでに存在していたどの言語/表現形態とも全く異なるのであれば、利用者は一からそれを学ぶしかなく、学習の結果得られる言語の有用度や生産性、他への応用度や一般性、その言語の将来での存続度から見て、よい言語仕様ではないだろうとの結論に至っている。一方、現在では Virtual Reality Modeling Language (VRML) [2] や、そのベースであるシリコングラフィックス社の Open Inventor ファイルフォーマット [3] が広く知られるところとなっている。

そこで、我々のRVEシステム開発にあたっては上記を参考にしつつ、これらのファイルフォーマットを知るユーザであれば、容易に記述できるような言語仕様とすることにした。当然のことながら、ここで述べる内容はあくまでファイルフォー

マットであり、RVEシステムの内部メカニズムを直接表したのではないことを明記しておく。

RVEシステムではVRML Ver.1.0に見られる静的な世界に加え、リアクションのような動きも取り込まれることから、いわば「VRMLの拡張」といった形態でファイルフォーマットを実現している。(動作に関しては今後のVRMLへの導入も検討されているため、我々も同時にその動向[4]を見ている。)

シーングラフ、ノードとフィールド

RVEファイルフォーマットにおいては、ノードとシーングラフの概念を利用する。形状や属性を表すノード、ノード間の階層構造であるシーングラフ共に既にVRMLなどで用いられており、ほぼそれと同一であると見なしてよいものとなる。従って、次に述べるRVE拡張部分を含めない場合、そのファイル内容はVRMLとほぼ同一(図1参照)となる。各種パラメータを表すフィールドのタイプも、基本的にVRMLなどに規定される値、つまり、SFColor、SFRotation、MFVec2fなどを用いている。

3 RVE拡張

3.1 動作の記述

RVEシステムにおける動作の記述としては、我々が scheduled field と呼ぶフィールドを含んだ動作データ提供オブジェクト Motion Engine を主に扱うことを考えている。体験者によるインタラクション、物体の衝突により引き起こされる動作の要素を盛り込みつつ、リアルタイムでの処理に重きを置くRVEシステムでは、前後の値変化が予測可能、つまり、補間が容易なフィールドとして scheduled field を位置づけている。この Motion Engine の内容としては、1) 各時刻とその時点での

```

#RVE V1.0 utf8

DEF Scene1 Separator {
  Separator {
    Translation {
      translation 0 1 0 = MyEngine.op
    }
    Separator {
      Material {
        emissiveColor 0.1 0.3 0.3 }
      Cube {
        width 0.1
        height 0.1
        depth 4
      }
    }
    Rotation {
      rotation 0 1 0 1.57079 }
    DEF Door Separator {
      # draw door here
      ....
    }
  }
}
SpotLight {
  location 0 4 0
  direction 0 -1 0
}
OrthographicCamera {
  position 0 0 1 }
}

```

図 1: サンプルファイルの一部

データ（接続先に対応したデータ型）の組、すなわち、scheduled field、2) その組数、3) 動作開始時刻、4) 動作再生速度を決めるスケール、5) 動作をループするか、6) 動作の補間をするか、7) データ列を含んだ外部ファイル名などの各フィールドがあり、これらを用いて、時間内の各時点での速度、加速度、位置といったデータが記述される。この Motion Engine の出力が、例えば、Translation ノードのフィールドに接続されることで、時間の経過に伴う動作が実現されるのである。

リアクションを記述する場合には新しく CLASS キーワードを用い、オブジェクト形状と共にコールバックを宣言するように、1) イベントタイプ、2) 現在の状況、3) イベントを引き起こしたオブジェクト、4) 引き起こされるオブジェクト、5) 動作内容などをイベント別に記述する。

3.2 衝突、その他の記述

衝突判定に用いる形状として、RVE システムでは代表形状と外郭形状の2種類を用いている。衝突することが前提とされるオブジェクトについては、システムが用意する外郭形状以外に、1) 線・面などの判定用代表形状、2) 衝突対象（識別名）を宣言した上で、3) オブジェクト形状を記述する。衝突されることが前提となる側については、同様の外郭形状だけが用いられるため、衝突判定用識別名が加わる程度のほぼ従来通りの記述となる。

一方、ネットワークを経由しての使用も考慮し、シーングラフの連結／共有のための記述を追加している。これにより別々に作成されたシーングラフが連結ノードを介して連結可能となる。

4 おわりに

仮想環境構築システム RVE では、単に街のウォークスルーだけではない VR アプリケーションをも開発できるツールキットを目的としている。なお、本報告では RVE システムの目的などについて触れておらず、詳細については別報「RVE システムの概要 (1H-8)」、「モーションの補間と拘束 (1H-10)」を参照されたい。最後に、本言語仕様をまとめるにあたって同研究所 宮澤 達夫氏、村尾 高秋氏、谷口 雅昭氏の助言に感謝したい。

参考文献

- [1] 宮澤, 大淵, 青野, 「仮想環境システム構築ツールキット mtv の概要と世界記述言語 vef」, 情処第 5 1 回全国大会, 3S-5, 1995.
- [2] G. Bell, et al., "The Virtual Reality Modeling Language - Version 1.0 Specification," <http://vrml.wired.com/vrml.tech/vrml10-3.html>.
- [3] J. Wernecke, *The Inventor Mentor*, Addison-Wesley, 1994.
- [4] G. Bell, et al., "The Virtual Reality Modeling Language - Version 1.1 Draft," <http://vag.vrml.org/vrml-1.1.html>.