

VRML 2.0 におけるイベント処理

3 S-5

谷口 雅昭

日本アイ・ビー・エム東京基礎研究所

1. はじめに

VRML(Virtual Reality Modeling Language) 2.0以降の仕様では仮想世界の状態を動的に変化させるための機構としてルートとイベントを提供している。概念的にイベントはルート上を瞬間的に伝搬されることとされている。しかし、これを実装することはたやすいことではない。本論文では VRML で規定されたイベント実行モデルの実装上の問題点をまとめる。なお、本論文で行う議論は 1997 年の 4 月に公開された VRML97 で規定された仕様に基づいているが、同年 12 月に改定された仕様にも適応される。

本研究は情報処理振興協会「創造的ソフトウェア育成事業」の一環として行われているものである。

2. VRML 2.0 におけるイベント処理

VRML 2.0 ではオブジェクトや、概念を表現する単位としてノードを定義している。各ノードはそのノードの属性や状態を表すためにフィールドをもつ。ノードの状態の変化はそのノードのもつフィールドを他のノードのフィールドにルートによって接続することにより伝搬される。ここで伝搬されるメッセージを VRML ではイベントと呼んでいる。ノード間のルート接続は VRML ファイル内に次のような記述をすることにより定義される:

```
ROUTE A.out TO B.in
```

これはノード A の out 出力フィールドから、ノード B の in 入力フィールドへのルート接続を意味する。

イベントはセンサ・ノードやスクリプト・ノードにおいて生成され、それがルートにより他のノードに伝えられる。イベントを受けとったノードは入力されたメッセージを評価し、必要に応じて別のイベン

トの出力する。このようなことを繰り返すことにより、次々にイベントがノードに伝搬されていく。

3. イベント処理実現上の考慮点

我々が開発中の Reactive Virtual Environment (RVE) システム[2] においてイベント処理を実現するにあたり、いくつかの実装上の問題点が生じた。これらについてここにまとめる。

3.1. ノードによるイベント生成の未確定性

ノードにおいてどのようなイベントが生成されるかは一般的には前もって知ることはできない。出力されるイベントはノードのタイプ、ノードの状態(フィールドの値)、入力イベントの値などに依存している。たとえば Script ノードはその動作が Java などのスクリプト言語で記述されており、通常はその振る舞いをスクリプトの実行以外の方法で知ることは難しい。従って、実装においてはノードにイベントを入力し、それを評価するまでは出力イベントを決定することができないと考える必要がある。

3.2. 複数イベントの同時入力

VRML のイベントモデルでは概念上イベントの伝達は瞬間的であると規定されている。実装では瞬間的にイベントの伝搬を行うことはできないので、ノードの評価を順序化する必要がある。

ルートの接続によってはノードに同時に複数イベントを伝搬する必要がある場合があるので、評価の順序はルート接続のトポロジーを考慮して決定する必要がある。例えば、図 1 で示されるようなルート接続が定義されていた場合、ノード C の評価をノード B の評価の前に行うとノード C の評価にノード B の出力を用いることができなくなってしまう。

Event Processing in VRML 2.0

Masaaki Taniguchi(taniguti@trl.ibm.co.jp),

Tokyo Research Laboratory, IBM Japan Ltd.,

1623-14 Shimotsuruma, Yamato-shi, Kanagawa

242, Japan

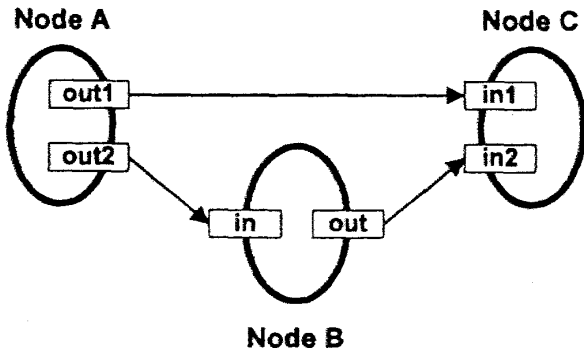


図 1:複数のイベントが同時にノードに入力される可能性のあるルート接続例

3.3. イベントループの防止

VRML ではルート接続の定義には自由度が高く、ループを含むような接続も定義することができる。例えば図 2に示すルート接続において、イベントがフィールドを

A.out → B.in → B.out → C.in → C.out → B.in

の順で伝わる場合にはこの接続にはループが存在している。しかしながらノード B がイベントの発生源になる場合のように B.in 以外の入力からイベントが生成された場合、ループは生じない。従ってループ検出はイベントの伝搬処理時にイベント伝搬経路に関する履歴を調べるによりおこなわれる必要がある。

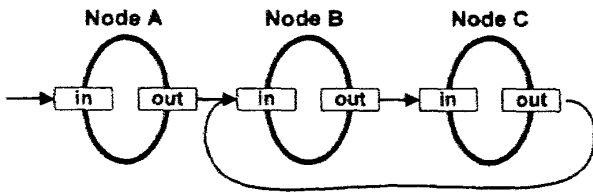


図 2:ループとなる可能性のあるルート接続例

4. ノードの相互依存

ノードが相互的にイベントの生成に関して依存するようなルートの接続も定義することができる。例えば図 3に示す例ではイベントが

A.in1 → A.out2 → B.in → B.out

→ A.in2 → A.out1 → C.in → C.out

といった順番で伝達された場合にはノード A のイベント出力のためにノード B の出力が、ノード B

の出力のためにノード A の出力が必要であり、イベント生成に関して相互的に依存している。この場合ノード A では複数の入力イベントがそろわ前にノード A の評価を行わなくてはならない。また、ノード A は A.out1 の出力のために 2 度評価する必要がある。

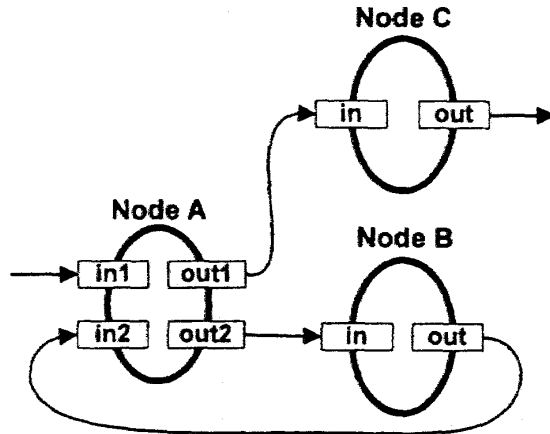


図 3:ノードがイベント生成に関して相互依存する可能性のあるルート接続

5. まとめ

VRML 2.0 において定義されたイベントモデルを実装する際に生じた問題点をまとめた。以上に述べた以外にもルート接続以外の方法でイベントが伝達される場合には例外的な処理が要求される。ダイレクト・アウトプットはスクリプトが直接フィールドにアクセスし、イベントを伝搬させる。また、プロトタイプタイプのフィールドに対するイベント入力は、IS によって受信フィールドを参照しているすべてのノードに対してイベントを伝搬させる。

参考文献

- [1] "The Virtual Reality Modeling Language Specification," ISO/IEC DIS 14772-1 April 4, 1997. Available as <http://www.vrml.org/Specifications/VRML97>.
- [2] "Reactive Virtual Environment (RVE) システムの概要," 情報処理学会第 52 回全国大会予稿集(2) p305-306, 1996.