

リフレクション機構を導入した3次元空間記述言語の提案と実装*

工藤 祐介[†] 大胡 里志[‡] 武田 正之[§]
 東京理科大学大学院 理工学研究科 情報科学専攻[¶]

1 はじめに

3D空間を記述する言語には、VRMLやX3D、XVLなどが存在し、現在ではX3Dを中心に、表示画質の向上やデータ圧縮技術の導入による転送データ量の削減が試みられている。しかし、そうした言語による3D世界の記述は、シンプルな物体を表示する場合でも煩雑になり、記述性や保守性といった面においては、オーサリングツール等の開発環境に頼らざるを得ない。加えて、物体の繰り返し表示のような簡単な計算によってシステム側で容易に生成できるオブジェクトも、製作者側で、スクリプト言語を用いることによって実現しなければならない。

本稿では、こうした問題点に対して、リフレクション機構の概念を導入した3D記述言語の提案、及び物体の振る舞いを記述したファイルの動的ローディング機能を持つブラウザの実装とライブラリの実装について示す。これにより、既存の3D記述言語よりも記述や保守が容易になる。

本システムの利点は、複雑な3D世界をテキスト形式の3Dデータフォーマットにて記述する際に生じる深いネスト構造を回避し、また、物体の記述と物体の制御を分離することで、可読性・保守性の向上が可能になることに加え、振る舞いの動的ローディングの手法を用いて、計算システムをカプセル化することにより、3D世界の記述の簡素化と、テキストデータの容量の削減を図っている点にある。

2 言語の概要

3Dを記述するための基本となる言語としてVRMLを用い、VRMLにリフレクション機構を導入することで、改良を図った言語RVRMLを提案する。

2.1 リフレクション

リフレクションは、自分自身の構成や計算過程をモデル化した表現を内部に持ち、その表現を参照や変更して、自分自身の構成や計算過程に反映させることである。具体的には、システムが結果を導くための最低限のコードであるベースレベルと、システムの構成や計算を制御するメタレベルから成る。RVRMLにおいてコードは、「球である」「立方体である」という制御対象の実体の情報を表すベースレベルと、「位置制御」「衝突制御」「関節制御」などの計算によって物体を制御するメタレベルに分類される。ここで、ある制御に対して別の制御が必要な場合は、メタのメタとして記述することができる(図1)。

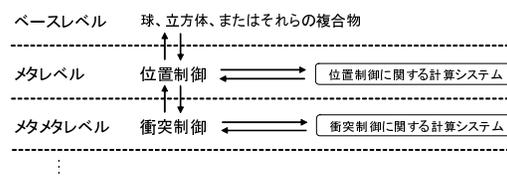


図1: リフレクション

2.2 言語仕様

RVRMLでは、「ノード」と呼ばれる、振る舞いの種類を表す識別子が記述の基本単位として扱われる。ノードは、そのノードの記述の仕方を定めた仕様書ファイルと、振る舞いを記述したスクリプトファイルによって定義される。ベースレベルにおける記述は、データ量や関係性によって、表現方法を選択可能にするため、VRMLのように完全なネスト構造になる記述方式に加

*An Implementation of Reflective 3D World Modeling Language

[†]Yusuke Kudoh[‡]Satoshi Daigo[§]Masayuki Takeda[¶]Dept. of Information Sciences, Graduate School of Sciences and Technology, Tokyo University of Science

え、フラットな記述の仕方でもできるようにした。メタレベルにおいては、リフレクションの概念上フラットな記述方式のみの採用とした。以下の記述例では、半径1の球が Shape ノードによって実体化され、位置を制御する Transform ノードによって中心座標が(1,1,1)に設定されることを示している。ここで、MAIN という識別子は、描画される木構造の最上層を表すものである。

```
#ベースレベル
SPHERE Sphere { radius 1 } # 半径1の球
#メタレベル
BALL Shape { geometry SPHERE } # 物体の生成
MAIN Transform { # 位置制御
  translation 1 1 1
  children BALL
}
```

3 ブラウザの構成

次に、RVRML が動作するブラウザの構成について述べる。ブラウザの実装は Java+Java3D API を用いて行なった。読み込まれた RVRML ファイルは、RVRML パーサへと渡され、パーサ側では、ファイルを解析しながら、木構造を構築していく。解析途中にノードを表す文字列が現れると、対応した仕様書ファイルと、スクリプトファイルが読み込まれる。ファイルを解析し終えて木構造が完成すると、次にトランスレータへと木構造が渡され、個々のスクリプトファイル内で構築された Java3D オブジェクトが次々に結合され、Java3D シーングラフが作成される。ここで、仕様書ファイルとスクリプトファイルは、実行時に動的にローディングされるため、自分の環境に必要なファイルだけをロードするということが可能である。

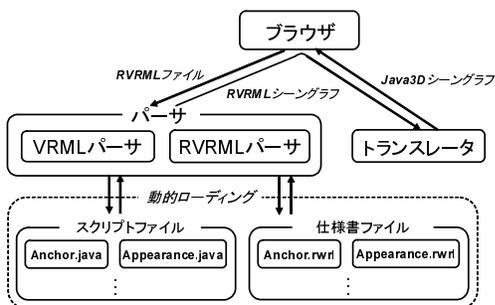


図 2: ブラウザの構成

4 拡張ライブラリ

ここでは、いくつかのライブラリの中で、位置情報などに数的規則性を持った複数の物体の生成を計算システムを用いて行なうことで、記述性の改善とデータ量の削減を支援するための Clone ノードについて示す。ライブラリは、RVRML 標準のノードと同様、仕様書ファイルとスクリプトファイルを用いて作成される。

Clone ノードでは、主に、繰り返し処理を適用する対象ノード名、変更フィールド名、及び、繰り返し処理時に使用される変位や生成個数の設定を行なう。スクリプトファイル側では、これらのパラメータを元に、対象ノード名で指定されたノードを複製後、生成されたノードの指定されたフィールドに対して変位を加算し、それを指定された回数繰り返すことによって、物体の生成を行なう。従来の 3D 記述言語では、振る舞いの異なるノードに関する情報は全て記述しなければならなかったため、物体の数が増えれば増えるほど、Clone ノードによるデータ量削減の効果は大きくなる。

種類	振る舞い
複製制御	物体の複製
衝突制御	物体同士の衝突の検出
関節制御	関節による移動制限

表 1: 主な拡張ライブラリ一覧

5 おわりに

本稿では、3D 世界のデータ記述における記述性と保守性の向上を目指し、言語仕様の提案とブラウザの実装、記述性の改善やデータ量の削減を支援するライブラリを作成した。今後は、言語仕様の改良に加え、ライブラリを充実させることにより、更なる記述の簡素化や、データ量の削減を目指すことなどが考えられる。

参考文献

- [1] 春日達也、武田正之： リフレクション機構を導入した仮想三次元空間モデリング言語 RVRML の研究、 AIST-TR-99-006, IMC, SUT. (1999)
- [2] 小谷野修宏、宇佐美健司、武田正之： An Implementation of Lightweight VRML Browser for Web3D, 情報処理学会第 65 回全国大会, 3N-5 (2003)