6ZB-9

3次元グラフィックス記述フレームワーク Sardonyx の提案と実装

片山 修治 † 松原 俊一 †† Martin J. Dürst ††

† 青山学院大学大学院理工学研究科理工学専攻
†† 青山学院大学理工学部情報テクノロジー学科

1 背景と目的

3 次元コンピュータグラフィックス (3DCG) の記述量は表現が複雑であるほど多量となる傾向がある．著者らの研究室では SVG 作成フレームワーク SVuGy [2] をプログラミング言語 Ruby による内部 DSL として実装した．本研究では 3DCG 記述フレームワーク Sardonyx を提案・実装し，SVuGy の原理の 3DCG 記述への適性を検証する．簡潔な記述により利用者の 3DCG 生成の利便性を高めることを目的とする．出力対象は Web 用 3DCG の形式である X3D [1] とする．

2 Sardonyx

2.1 特長

Sardonyx は Ruby の内部 DSL であり，X3D 記述フレームワークである．Ruby を用いることで，宣言的な記述と共に手続き的な処理ができる．そのため繰り返し処理などを含む複雑な X3D の生成も容易である．

2.2 記述方法

Sardonyx では宣言的な記述のようにメソッドを利用でき，ブロックの入れ子が木構造を表す．簡潔な記述で X3D を表現できる．要点を得やすく，全体像を俯瞰して捉えることで直観的に図形の作成が可能である．

立体の特性を指定した X3D の例は次のようになる．

```
<X3D>
  <Scene>
    <Transform translation='x y z'>
      <Shape>
        <Box size='width height depth'/>
        <Appearance>
          <Material diffuseColor='r g b'/>
        </Appearance>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

Sardonyx を用いることで次のように要素と属性値をメソッドと引数で記述して上記の X3D の作成が可能である．

```
transform(x, y, z) {
  box(width, height, depth) {
    material r g b
  }
}
```

2.3 X3D と SVG の差異による実装方針

表 1 に X3D と SVG の差異を示し，実装方針を述べる．X3D では要素名に大文字を含むものもあるが，

表 1: SVG と X3D の比較

項目	X3D	SVG
要素名	大文字を含む	小文字のみ
ルート直下	Scene 要素	各図形要素
図形の属性	サイズ	サイズ・座標・色

Ruby ではメソッド名は小文字で始まる．そのため，出力の際変更処理を行う．Scene 要素や Shape 要素，Appearance 要素などの一部要素に対しては，記述の省略を可能にした．X3D において色合いや座標などの特性は図形要素の属性としてではなく，別の要素として定義されている．そのため，SVG と比較し多くの要素を実装した．

2.4 特性の指定

立体の特性の指定に用いるのは，列挙された引数のみ，もしくは diffuseColor のように属性と同名のメソッドである．X3D では図形の作成に必要となる属性値は少数である．一方で色や座標といった特性は別要素の属性として扱う．

Proposal and Implementation of Sardonyx - A Framework to Create Three-Dimensional Graphics
Shuji Katayama †, Shunichi Matsubara ††
and Martin J. Dürst ††
†Department of Integrated Information Technology, College of Science and Engineering, Aoyama Gakuin University
††Graduate School of Science and Engineering, Aoyama Gakuin University
katas@sw.it.aoyama.ac.jp,
{matsubara, duerst}@it.aoyama.ac.jp

Copyright ©2013 Information Processing Society of Japan.
All Rights Reserved.

要素に対して属性値が少ない場合，直接メソッドの引数にする．例としてある大きさの直方体を X3D で次のように記述する．

```
<Box size='width height depth'/>
```

それに対応する Sardonyx の記述は次のようになる．

```
box width, height, depth
```

逆に，ある要素に必要な属性が多い場合，前述の方法では数値を連続して羅列するため，属性ごとにメソッドを用意する．例として X3D の Transform 要素を使う．

```
<Transform translation='tx ty tz'
           rotation='rx ry rz r'
           scale='sx sy sz'>
```

Sardonyx では以下のように属性をメソッドで指定できるようにした．

```
transform.translation(tx, ty, tz)
         .rotation(rx, ry, rz, r)
         .scale(sx, sy, sz)
```

2.5 必須要素の簡略化

X3D ではルート要素である X3D 要素の子として，実際にグラフィックスを生成する要素の他に形式的に記述する要素がある．情景を指す Scene 要素，各図形要素の親要素となる Shape 要素が該当する．Scene 要素は一つの X3D ファイル内に唯一つ存在し，Shape 要素は生成する図形ごとに一つ存在する．

Sardonyx ではこれらの要素の簡略化が可能である．Scene 要素は X3D 要素の生成時に生成する．図形要素の生成時に親要素を調べ，Shape 要素が親でなければ親要素を作成する．

3 Sardonyx の応用例

3.1 情報の可視化

Sardonyx を用いることで 3 種のパラメータを持つデータに対して 3DCG として可視化させることができる．図 1 の右は Sardonyx を用いて Bidi アルゴリズム[1]の制限を調査したデータの可視化を行った一例である．

3.2 再帰的生成

分子構造や幾何学で用いられる立体では多くの繰り返しが見受けられる．これらを 3 次元化させる際に多量な記述となることが予想されるが，Sardonyx を用いてより簡潔に記述させることが可能となる．図 1 の左は Sardonyx の効果が顕著に表れるフラクタル図形の再帰的生成の一例である．自己と全体が相似となる立体を再帰処理により作成し，通常のソースコードに比べ非常に簡潔に記述できる．

[1]Mark Davis. Unicode Standard Annex #9 The Unicode Bidirectional Algorithm. http://www.unicode.org/reports/tr9/, 2012

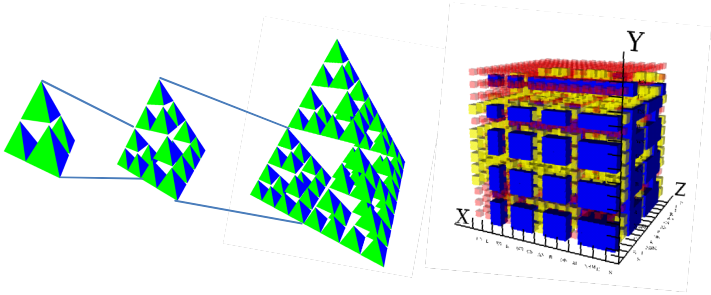

図 1: 応用例：再帰的生成 (左) と情報の可視化 (右)

4 記述量における比較

情報の可視化において用いたプログラムを比較対象として調査を行った．純粋な Ruby のみで X3D を生成するものと，Sardonyx を用いて生成するものを比較した．その際データファイルの読み込み，およびそれに対する繰り返し処理などのイテレータは変更しない．本フレームワークを用いた場合，XML 文書の型宣言やタグの繰り返し記述などにより，コード量・ファイルサイズにおいて半減できた．

5 まとめ

本研究では 3DCG 記述フレームワーク Sardonyx を提案し，実装した．著者らの研究室で開発された SVG 作成フレームワーク SVuGy の原理を 3DCG への適応に成功した．Sardonyx は簡潔な記法により，XML におけるタグ構造の冗長さをなくし，常用する要素の省略ができる．さらに Ruby プログラムを用いた複雑な図形の作成が可能となった．今後，Sardonyx ならびに SVuGy の機能を共有化することで他用途へ拡張の幅が広がると考えられる．

参考文献

[1] Web3D Consortium. Extensible 3D (X3D) ISO/IEC 19775:2004. http://www.web3d.org/files/specifications/19775-1/V3.0/index.html, 2004.

[2] Martin J. Dürst, Makoto Fujimori, Takeshi Maemura, Tohru Koga, and Kazunari Ito. SVuGy - Exploring the Space between Procedural and Declarative Graphics. In *Proceedings of the 5th International Conference on Scalable Vector Graphics (SVG Open 2007)*, 2007.

Copyright ©2013 Information Processing Society of Japan.
All Rights Reserved.