SVuGyを例にしたXML用入出力指定DSLの開発

地主 龍一, 松本 章代, Martin J. Dürst
青山学院大学理工学部情報テクノロジー学科

1 研究背景

ベクターグラフィックスの代表的なフォーマットにSVGがある．これはXMLフォーマットであり，テキストエディタでの編集，DOMやCSSとの併用，ウェブとの親和性の高さなど多くの利点が挙げられる．SVGをより手軽に作成するため，本研究室では2007年からフレームワークSVuGy [1]の開発に取り組んでいる．本研究ではSVuGyの入出力に関する部分を新たに構築し，XML形式の入出力指定DSLの開発を目指す．

2 SVuGy

2.1 概要

SVuGyはスクリプト言語Rubyによって実装された内部DSLである．Rubyではメソッド呼び出しの際のかっこを省略できるので，プログラムを宣言的に表現できる．DSLでは変数の指定など宣言的な表記も多く，そのような外見を提供できるのはきわめて重要である [2]．SVGは属性値に変数が利用できず，座標の再利用に制限がある．SVuGyでは変数や繰り返し処理を用いた記述が行えるので，より手軽にSVGを作成することができる．SVuGyはRubyの母音部とSVGを組み合わせたもので,「スヴィージー」と発音する．

2.2 記法

図1で示す図形をSVuGyで作成する手順について以下に示す．SVuGyが記述されたRubyファイルを実行するとSVGファイルが出力される．メタプログラミングによって図形の部品のidとして指定された識別子，をそのまま部品を示す変数として使える．図2の例では，`arrow re1.mr, re2.ml`で以前に定義した二つの長方形（`rect`）を矢印で簡単に結ぶことができる．その他にも変数や繰り返し処理など，Rubyの機能をSVG作成に応用することができる．

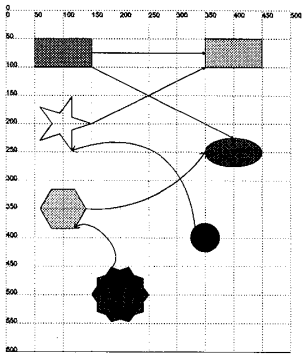

図1: SVuGyで作成したサンプル

```
g { # 図形作成の指定
  stroke 'black'
  rect :re1, 50, 50, 100, 50
  rect :re2, 350, 50, 100, 50
  ellipse :el, 400, 250, 50, 25
}
g { # 矢印の追加
  stroke_width 1
  arrow re1.mr, re2.ml
  arrow re1.br, el.tc
}
re1.fill "red"
```

図2: 図1を作成するSVuGyプログラムの一部

3 XML用入出力指定DSL

3.1 入出力DSLの概要

これまで，出力に関するメソッドはSVuGyの内部に手続的に記述されていた．これを宣言的なDSLとして独立させることで，SVuGy自体を簡略化する．また，その機能を他のXMLツールからも利用することが期待できる．

3.2 XML形式への変換

入出力DSLは，プログラム内のオブジェクト構造とXMLの構造が類似していることを前提としている．クラスは要素に，クラス名は要素名に，インスタンス変

Development of an Input and Output Specification DSL for XML using SVuGy as an Example
Ryuichi Jinushi, Akiyo Matsumoto and Martin J. Dürst
Department of Integrated Information Technology, College of Science and Engineering, Aoyama Gakuin University
5-10-1 Fuchinobe, Sagamihara, Kanagawa 229-8558, Japan
ryuj@sw.it.aoyama.ac.jp, {akiyo, duerst}@it.aoyama.ac.jp

数は属性にそれぞれ対応する．この原則を元に自動で出力が行われる．しかし，例外に対してはDSLによる指定が必要になる．変数名に利用できないハイフンなどの記号は，アンダーバーなど別のものと置き換えて利用している．

3.3 属性表示の指定

属性名と属性値はインスタンス変数名とその値が対応する．ただしSVuGyには出力しないインスタンス変数も存在しており，出力する属性を指定する必要がある．図3の`output_attrs`は出力する属性を指定するもので，変数値が`nil`でないかぎり図形要素の属性として出力を行う．

一部の属性の場合には属性値によって出力の必要がないものも存在する．たとえばSVGの場合，座標の初期値は0になっている．`rect`要素の属性に`x="0"`という記述がなくても出力する図形に違いはない．それらの対応をSVuGy側で行う場合，インスタンス変数の値をnilに置き換える操作が必要になってしまう．そのような例外に対しても，図3の`non0_output_attrs`のように指定を行うことができる．これは属性値によっては出力の必要がない属性を指定する．

```
class Rect < Shape
  output_attrs :width, :height, :rx, :ry
  non0_output_attrs :x, :y
  ～～～ # Rect クラス用の定義
end
```

図 3: `Rect` クラスにおける属性出力の指定

出力する属性には共通のものが多い．それらを効率よく指定するには，クラスの継承機能を利用する．たとえば`id`属性はどの要素でも利用が可能なので，各図形の上位クラスである`Shape`クラスで定義している．

3.4 有効桁数の指定

データが数値を含む場合，その用途によって要求される精度は異なる．たとえばSVGで図形の拡大表示が行われる場合，正確に表示するためには高い精度が必要となる．データを頻繁に送信する場合，通信コストなどの面から見るとデータはコンパクトなほうが良い．精度を低くすることで数値を短くし，ファイルサイズを圧縮することができる．このような用途に対応するため，入出力DSLでは出力する数値の有効桁数をあらかじめ指定する．これはファイル全体に対して指定するだけでなく，インスタンスのみを対象とすることもできる．

3.5 インデントの指定

インデントとは字下げを意味し，XMLの階層構造を視覚的に示す際に用いられる．インデントが不規則でもエラーは発生しない．しかし，ソースコードの可読性という点では大きな問題がある．インデントを整えることは可読性の向上につながる．また，サーバに保存するなどあまり人の目に触れない場合，インデントを0にすることでファイルサイズを圧縮することもできる．用途に適した出力を行うため，入出力DSLではインデントに空白文字を何文字分用いるのかを指定する．インデントはファイル全体で共通のものとするため，この指定は出力の際に一括で行う．以下にインデント数2，有効桁数3の例を示す．

```
to_xml(:indent => 2, :digits => 3)
```

3.6 XML ファイルの入力

RubyのXMLパーサであるREXMLを用いて既存のXMLを解析し，オブジェクトとして取り込んで編集することができる．ファイル全体を読み込むものと，`id`を指定してファイルの一部分を抜きだすものの2種類の方法が可能である．

3.7 SVuGy 以外での利用

入出力DSLを利用する際には，Rubyの`include`機能を用いる．これは入出力に関するメソッドをまとめたものを，外部から読み込んで利用している．このメソッドは同名のメソッドを再定義することで上書き可能なので，用途に合わせて仕様を変更し，効率よく利用することが期待できる．

4 まとめ

本研究では，XML形式での入出力指定内部DSLの開発を行った．要素やインデントなどを含めた階層構造の表示だけでなく，コメント文やCDATAセクションなどの特殊な構文にも対応させた．それに加え，有効桁数やインデントの文字数などをユーザが独自に指定できるよう実装した．入出力DSLはSVuGy以外からも利用されることを想定しているため，今後も様々な用途に対応できるよう改良を重ねる．具体的な改良案として，DTDなどスキーマ言語の定義に従ってXMLファイルを作成することが考えられる．

参考文献

[1] Martin J Dürst, Makoto Fujimori, Takeshi Maemura, Tohru Koga, and Kazunari Ito. SVuGy - Exploring the Space between Procedural and Declarative Graphics. `http://www.svgopen.org/2007/papers/SVuGyProceduralDeclarative/`, 2007.

[2] まつもとゆきひろ．まつもと直伝　プログラミングのオキテ　：ITpro. `http://itpro.nikkeibp.co.jp/article/COLUMN/20070604/273453/`, 2007.