

Python を用いたオブジェクト指向 TVML の提案

Proposal of Object-oriented TVML Using Python

浜口 斉周[†] 道家 守[‡] 金子 浩之[†] 井上 誠喜[†]
Narichika Hamaguchi[†] Mamoru Doke[‡] Hiroyuki Kaneko[†] Seiki Inoue[†]

1. まえがき

我々はテレビ番組など映像コンテンツを記述するコンピュータ言語として、TVML(TV program Making Language)^[1]とそのTVMLをプラットフォームとしたさまざまなメディアの研究^[2]を行ってきた。

TVMLはプロトタイプベースのオブジェクト指向言語の一種であるが、いわゆるクラス概念はなく、メソッド定義なども言語としてはサポートしないため、むしろスクリプト言語と呼ぶにふさわしい。制御構造がなく、1行ずつ実行するシンプルなインタープリタ動作であるため、プログラミングの経験がないユーザでも台本を書くような感覚で、比較的簡単に映像コンテンツを記述・制作することができる。

一方で、TVMLはカプセル化をサポートしていないことから、部品化や再利用性の面で課題があった。その課題を克服するため、我々はAPE(Automatic Production Engine)^[3]と呼ぶ、演出をカプセル化したスタイルシートを考案し、XML記述の台本と組み合わせることによって、TVMLを出力する仕組みを開発した。

しかしAPE、XML台本、TVMLはそれぞれ別々の言語体系を持つ。従って、すべてをハンドリングするためには、ユーザは3つの言語体系をマスターしなければならず、メンテナンス性に限界があった。そこで本稿ではTVML/APEに、統一されたクラスベースのオブジェクト指向の言語体系を導入し、TVMLをPython^[4]で再構築することを提案する。

2. 既存TVML/APEの特長と問題点

新しいTVML/APEの言語体系を導入するに当たり、既存のTVML/APEの特長を踏襲しつつ、問題点を解決できる言語体系であることが望ましい。本章では既存の仕組みの特長と問題点を挙げてみる。

2.1 特長

特長1: TVMLは行区切りは改行だけである。また、引数名と値を明示する書式のため、可読性が高く、見通しがよい

特長2: TVMLの引数は必須パラメータ以外は省略が可能でデフォルト値をサポートする。従って、必要な引数のみを記述すればいいので、記述が容易である

特長3: APEはメタデータを持ち、APE単独で流通可能で、検索や演出の交換にも対応できる

特長4: XML台本はネスト構造を用いず、1階層のみの記述のため、TVクリエイターという台本編集ツールを用

いることによって、エンドユーザでも簡単に記述・編集ができる

特長5: インタープリタ動作なので、XML台本やTVMLを外部プログラムで動的に生成して、リアルタイムに動作させることができる

2.2 問題点

一方、問題点としては

問題点1: TVMLはクラス/メソッド定義などを備えていないため、カプセル化が難しい。従って、部品化、再利用性の面で困難が伴う

問題点2: TVMLとAPE+XML台本という2階層のみであるため、ユーザの選択肢が限られる

問題点3: 台本はXMLで記述され、APEはXSLT、あるいはバイナリプログラム(C++など)で作られ、出力はTVMLであるため、それぞれのフェーズに応じて3種類の別々の言語体系を用いなければならない

2.3 Pythonを選択した理由

前述したような特長を包含しながら、問題点を解決できる言語としてPythonを選択した。Pythonには、前述した特長と問題点に対して、以下のような特長がある。

- (1) 記述がシンプルで可読性が高い。引数名の明示/省略の両方が可能である。行の区切りはTVMLと同じく改行を用いる
- (2) TVMLと同様にメソッドにおける引数の省略が可能で、デフォルト値が使える
- (3) コードのまま流通が可能である。また、ドキュメンテーション機能を用いることにより、コード内部にメタデータを埋め込むことができる
- (4) 動的プログラミング言語であるため、あるコードを実行しながら、他のコードを動的に生成・実行したり、あるコードが他のコードを読み取って、それに応じて動作を動的に変化させたりするような、いわゆるリフレクティブプログラミングが可能である
- (5) 複雑な処理の記述も可能であるが、シンプルな処理は記述もシンプルになるため、台本記述に用いることができる
- (6) TVML、APE、台本すべてをPythonというひとつの言語体系で記述することができる

以上のようなPythonの言語そのものの特長のほか、実装においては、マルチプラットフォームであり、他の言語やシステムを結びつけるグルー(糊)言語として、CGソフトのMayaなど、さまざまなシステムで多用されているという点も重要である。

[†]日本放送協会 放送技術研究所, Japan Broadcasting Corporation, Science & Technical Research Laboratories.

[‡](財)NHK エンジニアリングサービス, NHK Engineering Services, Inc.

3. Python ベース TVML/APE の試作

3.1 TVML Wrapper の開発

新しい TVML 言語をテストするため、図 1 の概略図のように、既存 TVML 言語と TVML Player の外部制御 API(TvIF)^[5]に対する Wrapper を試作した。Python から既存 TVML への変換は文字列フォーマットを用い、TvIF の Wrapper は SWIG^[6]という Wrapper Interface 生成ツールを用いた。図 2 に既存 TVML の書式と Python の書式との比較を示す。図にあるように、既存 TVML とよく似た書式でよりシンプルに記述することができる。

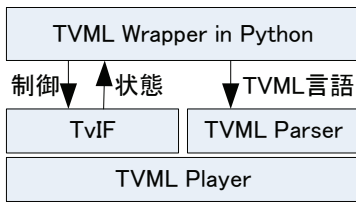


図 1 TVML Wrapper

Python TVML	既存 TVML
<pre>import tvml bob=tvml.character('bob/bob.bm') camA=tvml.camera() bob.locate(1, 0, 2) camA.closeup('bob', wait='no') bob.talk("What' re you doin' ?")</pre>	<pre>characer: bindmodel(name=bob, model="bob/bob.bm") character: locate(name=bob, x=1, y=0, z=2) camera: closeup(what=bob, wait=no) character: talk("What' re you doin' ?")</pre>

図 2 Python TVML と既存 TVML の比較

3.2 Python による APE 試作と台本記述

次に、演出スタイルシート APE と台本記述を Python を用いて試作した。図 3 に Python で記述した APE と XSLT で記述した APE を示す。図にあるように、Python を用いると同じ内容でも非常にシンプルに記述することができる。また、XSLT は変数を保持することができないため、その点でも Python を用いた APE の方が格段に記述性に優れる。

Python APE	XSLT APE
<pre>... def title(self, value): ttl=tvml.drawing('TITLE') ttl.settext(value) def subimage(self, value) img=tvml.prop('PLATE') img.openimageplate(value) def super(self, value) sp=tvml.super() sp.onText(value) def text(self, value) act=tvml.character('A') actr.talk(value) ...</pre>	<pre>... <xsltemplate match="title"> drawing: settext(name=TITLE, text="<xslvalue-of select="."/>") </xsltemplate> <xsltemplate match="subimage"> prop: openimageplate(name=PLATE, filename="<xslvalue-of select="."/>") </xsltemplate> <xsltemplate match="super"> super: on(type=text, text="<xslvalue-of select="."/>") </xsltemplate> <xsltemplate match="text"> character: talk(name=A, text="<xslvalue-of select="."/>") </xsltemplate> ...</pre>

図 3 Python APE と XSLT APE の比較

図 4 に、Python を用いた台本記述と、XML 記述の台本を比較する。APE は Python のモジュールファイルとして製作されているため、台本記述ではその APE モジュールを import して使用する。import された APE は台本を基に既存 TVML に変換し Wrap された TvIF を通して TVML Player をコントロールし、再生する。図の比較例のように XML 台本よりシンプルな台本記述になっていることがわかる。図 5 に Python シェルから TVML/APE を制御している例を示す。

Python台本	XML台本
<pre>import news ape=news.midnightnews() ape.title("Today's News") ape.subimage('newspic.jpg') ape.super('Snow in Spring') ape.text('We had snow today...')</pre>	<pre><ape file="news_midnightnews.ape"> <title>Today's News</title> <subimage>newspic.jpg</subimage> <super>Snow in Spring</super> <text>We had snow today...</text> ...</pre>

図 4 Python 記述台本と XML 台本の比較



図 5 Python シェルからの TVML/APE 制御の例

4. むすび

TVML システムに本格的なオブジェクト指向を導入するため、Python を用いた新しい TVML/APE を試作した。例示したように TVML/APE のシンプルさを損なうことなく、オブジェクト指向を導入できることがわかった。

今後は、この新しい TVML を使ったアプリケーションを開発しながら、新言語の有効性を検証していく予定である。

参考文献

- [1] <http://www.nhk.or.jp/strl/tvml/>
- [2] 浜口, 道家, 林, 八木, "演出スタイルシートを用いたブログ型テレビ番組制作・公開・視聴システム," 信学論 D vol.J89-D no.10 pp.2194-2205(2006)
- [3] Hayashi, Doke, Hamaguchi, "Automatic TV program production with APEs," 2nd International Conference on Creating, Connecting and Collaborating through Computing(C5 2004), pp.20-25(2004)
- [4] <http://www.python.org/>
- [5] 道家, 金子, "私の研究開発ツール(第 8 回) TvIF," 映情学誌, vol. 62, no. 2, pp. 193-196(2008)
- [6] <http://www.swig.org/>