

## 物語概念表現から映像表現への変換における動作概念記述ツール

Tools for Defining Action Concepts in the Transformation from Narrative Conceptual Representation into the Animated Movie

高橋 昇<sup>\*1</sup>

TAKAHASHI Noboru

小方 孝<sup>\*2</sup>

OGATA Takashi

<sup>\*1</sup> 永山コンピューターサービス(株)

Nagayama Computer Service Co.,Ltd

<sup>\*2</sup> 岩手県立大学ソフトウェア情報学部

Faculty of Software and Information Science, Iwate Prefectural University

Our narrative generation system is composed of conceptual representation generation part and surface representation (language, animated movie, music and so on) generation part. In the research on animated movie, we have developed a tool for supporting the process of the conversion from narrative conceptual representation to the corresponding animated movie with TVML 3G animation tool. In this process, especially difficult problem is how we prepare the information about narrative action and movement. In this paper, after we introduce the overview of above tool, we show a basic idea on a hierarchical system of actions and movements that is broke down from the level of Propp's function to the level of a sequence of movements as TVML script. And we show a tool supporting the description process of TVML script in the above tool.

## 1. まえがき

物語の概念表現生成から映像表現への変換が自動的に行われるシステムへの準備としてこの変換過程を支援するツールを開発した。このツールは、物語の概念表現を入力とし、これを映像表現のための記号記述である TVML スクリプトに自動変換する。TVML とは 3DCG による映像番組制作が可能なテキストベース言語である(但し工夫すれば番組風でなく映画風のコンテンツも作れる)[林 2005]。筆者らの物語生成システムは、概念表現生成の部分と映像の外言語や音楽等を含む表層表現生成の部分に分かれ、上記ツールの入力概念表現の生成部の出力となる。概念表現中に記述されたオブジェクトや行為・動作が具体的な映像イメージに変換されそれらが適切に配置される必要があるが、このツールでは、オブジェクトや行為・動作の映像データはユーザが予め用意しなければならない。また配置もユーザが調整しなければならない。

映像への変換において特に行為・動作の映像表現を用意することが困難な問題である。概念表現における行為・動作の記述は直接動作を指定しない抽象的な情報に留まっている場合も多いが、これを具体的なイメージに直さなければならず、そこに一貫して効率的な方法を導入するのは容易ではないと思われる。何らかの形で行為・動作の体系的なオントロジーを用意することが求められるが、本稿はまだそのような本質的な方法を提案するものではない。ここでは、上記のユーザとのやり取りを通じて物語の概念表現を TVML スクリプトに変換するツールの概要を示し、行為・動作の定義に関する基本的な考え方を示した上で、上記ツールにおける行為・動作を登録するための一ツールを紹介する。結論としてこのツールは使用の便利さや有効性において不満な部分が多いため、新たな方法を考案中[真部 2007]であるが、使用可能性はあるのでここに紹介する。なお物語における映像表現の基本的アイデアや上記変換ツール、動作支援の方法(その他のものを含む)等について[小方 2007]はより詳細な記述を提供する。

## 2. 物語の概念表現から映像表現への変換支援システム

XML で記述された物語の概念表現を入力とし、ユーザインタフェースによりユーザが登場人物や道具、動作等の映像情報を選択

もしくは作成・登録することにより、TVML スクリプトを作成するツールを作成した。この TVML スクリプトは TVML プレイヤーによって自動で映像化される。このツールへの入力となる物語概念表現の形式は次のように決めている。

```
<doc>
...
<event id="event_6">
  <action adverb="ある日">散歩に出る
  <agent id="2">第一皇女</agent>
  <agent id="3">第二皇女</agent>
  <agent id="4">第三皇女</agent>
  <counter-agent></counter-agent>
  <counter-agent></counter-agent>
  <counter-agent></counter-agent>
  <instrument></instrument>
  <object></object>
  <location>庭</location>
  <goal></goal>
  <from></from>
  <next-location></next-location>
  <narration></narration>
  <caption></caption>
</action>
<action>眺める
  <agent id="2">第一皇女</agent>
...
</doc>
```

これは、「...ある日、皇女達はお庭へ散歩に出て、花を眺めていらっやいました...」という意味を表している。それぞれのタグは次のような意味である doc:文章全体を表す。 event:一文を表す。 action:一つの動作を表す。 agent:動作主を表す。 counter-agent:動作の対象(人)を表す。 instrument:動作中に登場する道具を表す。 object:動作の対象(物)を表す。 location:動作を行っている場所を表す。 goal:動作の目的位置を表す。 from:動作の開始位置を表す。 next-location:動作の移動先(場所)を表す。 narration:動作のない文章を表す。 caption:動作のない文章を表す。

ツールを通すことで、これが次の図 1 のような TVML スクリプトに自動変換される。

連絡先:小方孝, t-ogata@iwate-pu.ac.jp

```

skripscript(switch=on)
//カメラのセットアップ
camera: assign( name=CamA )
camera: movement( name=CamA, x=0.0, y=1.0, z=3.0, tilt=5.0, vangle=70 )
camera: switch( name=CamA )
//照明のセットアップ
light: assign( name=light1 )
light: model( name=light1, type=flat, x=1.0, y=1.0, z=1.0, r=1.0, g=1.0,
b=1.0 )
light: assign( name=light2 )
light: model( name=light2, type=flat, x=-1.0, y=1.0, z=1.0, r=1.0, g=1.0,
b=1.0 )
//スタジオのセットアップ
set: assign( name=location )
set: openmodel( name=location, filename="desert/desert.obj" )
set: change( name=location )
//next-location のセットアップ
set: assign( name=nextlocation )
set: openmodel( name=nextlocation, filename="desert/desert.obj" )
//小道具のセットアップ
//キャラクターのセットアップ
character: casting( name=agent1 )
character: openmodel( modelname=baby, filename="baby/baby.bm" )
character: bindmodel( name=agent1, modelname=baby )
character: setvoice( name=agent1, voicetype="male01" )
character: position( name=agent1, x=0, y=0, z=0, d=0, posture=standing )
character: casting( name=agent2 )
character: openmodel( modelname=baby, filename="baby/baby.bm" )
character: bindmodel( name=agent2, modelname=baby )
character: setvoice( name=agent2, voicetype="male01" )
character: position( name=agent2, x=1, y=0, z=0, d=0, posture=standing )
character: casting( name=agent3 )
character: openmodel( modelname=baby, filename="baby/baby.bm" )
character: bindmodel( name=agent3, modelname=baby )
character: setvoice( name=agent3, voicetype="male01" )
character: position( name=agent3, x=-1, y=0, z=0, d=0, posture=standing )
//ナレーションのセットアップ
narration: narratorvoice( voicetype="male02" )
skripscript(switch=off)
//ナレーションの実行
video: switcher( source=black )
narration: talk( who=narrator, text="", caption="" )
set: change(name=location)
video: switcher( source=studio )
character: walk(name=agent1, z=1, wait=no)
character: walk(name=agent2, z=1, x=0.5, wait=no)
character: walk(name=agent3, z=1, x=-0.5,)
reset()

```

図 1: XML の物語概念表現から変換された TVML スクリプト

ユーザは以下に示すようなインタフェース画面を通じて操作を行う。まず図 2 の画面から入力となる XML 形式の物語概念表現を読み込む。次に図 3 の画面により、一つの action に含まれた内容(登場人物の CG、道具、場所、動作等)の編集に移る。それらは予めデータベースに登録されている CG 映像のストックの中から選択する。図 4 は登場人物(agent, counter-agent)を選択する画面、図 5 は道具(instrument)を選択する画面、図 6 は場所(location, next-location)を選択する画面である。これらを選択した後、その配置等詳細な情報も別の画面で設定することが出来る。

必要な全ての情報の選択や設定が済むと、図 7 に示す画面上に自动生成された TVML スクリプトが表示される(右上のウィンドウ)。この画面ではまた動作自体のスクリプト(action の値に相当するもの)を追加・修正・登録することが出来る。既存の動作 CG 映像が存在すればそれを使用出来る。本稿のテーマはこの映像データを登録するための方法を検討することに当る。

### 3. 動作の具体的展開

ここでは上記ツールにおいて行為・動作の映像情報の登録に関わる部分に着目する。物語生成システムにおいては、物語の概念

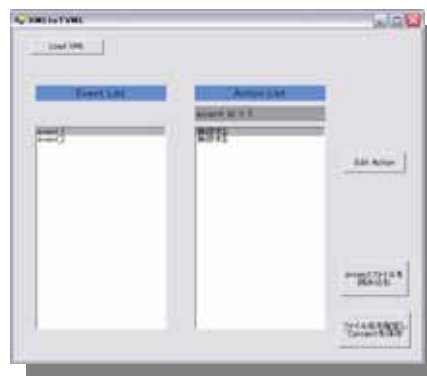


図 2: 物語概念表現の入力画面

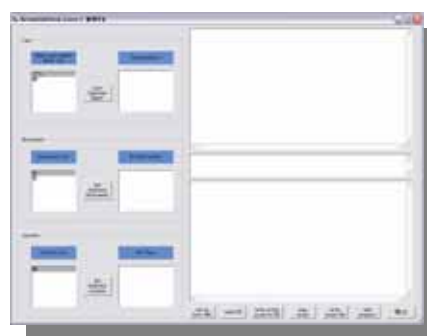


図 3: action の内容の編集のためのメイン画面



図 4: action 中の登場人物の選択画面



図 5: action 中の道具の選択画面

表現では action の中で必ずしも動作系列として具体化された情報が使用されるとは限らない。例えば「誘拐する」という行為は、それ自体が具体的な動作そのものを示すものではない。複数の動作の組が一定の順序で配列されることでその具体的なイメージが得られる。映像化に当ってはこの動作の組とその配列の情報を利用しなければならない。なおここで動作と言っているのは主体の身体の動きとして具現化されるレベルであり、行為とはこれに対して特定の動作を一定の順序で束ねた集合の意味論的な定義を意味する。動作のプリミティブな集合とそれを組み合わせた動作の階層的体

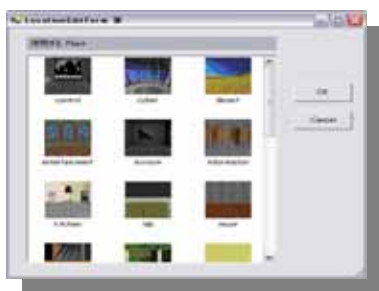


図 6: action 中の場所の選択画面



図 7: TVML スクリプトを書き出し動作を追加・修正・登録する画面

系が存在し、一方で行為についても抽象度の度合に基づく階層的体系が存在し、特定の動作の集合・配列が特定の行為と対応付けられるようなオントロジー構築が目標であり、これについては[真部 2007]から基本的検討を始めている。ここでは原理的な問題には入らず、基本的な発想を示すのみとする。

前述のように映像表現は TVML を用いて行う。TVML ではキャラクターの動作を制御する主なコマンドとして、表 1 に示すような基本的動作が予め用意されている。

表 1: 動作を記述する TVML コマンド

character: talk()	キャラクターを喋らせる
character: talkfile()	音声データファイルでキャラクターを喋らせる
character: gesture()	喋りのジェスチャー
character: walk()	歩く
character: step()	一歩で目的座標に移動
character: stop()	歩いているキャラクターを止める
character: sit()	座る
character: stand()	座っているキャラクターを立ち上がらせる
character: turn()	CGキャラクターの向きを変える
character: bow()	お辞儀する
character: look()	指定した対象を見る
character: gaze()	指定した方向を見る
character: jump()	指定位置までジャンプする
character: tumble()	倒れる
character: byebye()	手を振ってバイバイする
character: action()	各種アクションをする
character: shake()	震える
character: blink()	瞬きをする
character: openmouth()	口を開ける
character: expression()	顔の表情を変える
character: swing()	体の自然な揺れ
character: changemodel()	モデルのチェンジ
character: pose()	ポーズの実行
character: definepose()	ポーズの定義
character: deletepose()	ポーズ定義を解除
character: sequencepose()	連続ポーズの実行
character: definesequencepose()	連続ポーズの定義

character: deletepose()	連続ポーズ定義の解除
character: openkeyframe()	モーションキャプチャーデータファイルを開く
character: closekeyframe()	モーションキャプチャーデータを閉じる
character: keyframe()	モーションキャプチャーデータの実行

これらをプリミティブと決めて映像表現を行う場合、概念表現中のより抽象的な行為や動作を特定のプリミティブの系列として記述することが考えられる(TVML における動作定義に関してはユーザがキャラクターの動作を定義出来る definepose コマンドが用意されておりそれによってこれを増やせる可能性もあるので、これらがプリミティブと言うのはあくまで暫定的なものである)。抽象的な行為の記述から映像化可能な程度に具体的な動作の記述との間には階層性が存在すると思われる。ここではこの階層的体系に関する素案を、筆者らが物語生成における物語内容(ストーリー)の部分の一方法として再構成して利用しているプロップのモデル[Propp 1969]を援用して示す。[Propp 1969]はストーリーの大局的構造を規定する 31 の機能のそれぞれについてその具体的な実現のための様々な例を挙げており、筆者らはそれをもとにストーリーの構造に関する階層的体系のモデルを組み立てた([佐久間 2006]等)。例えば、「加害」の機能の下位階層にはその実現方法を示す約 19 個の「副機能」が存在し、そのうち「敵対者が人間を誘拐する」という副機能は、例えばさらに具体的に「不意にその蛇が現れ、皇女達を火の翼に乗せて、凌ぎ行つてしまいました。」という風に展開される。そしてこの展開を、「[動作主格]が現れる」、「[動作主格]が[対象格]を[道具]に乗せる」、「[動作主格]が[対象格]を[格フレーム]のように格フレーム風の記述で表す。これがさらに TVML における上述の動作のプリミティブ的な記述の系列として具体化されることによって、初めて映像化が可能になる。以上のような機能から TVML の動作コマンドの系列に至る階層的な詳細化を図 8 に示す。31 個の機能のそれぞれについてこのような例を作成した([松田 2005]の付録に記載されている)。物語概念表現における行為の記述がこの階層の上位レベルで記述されれば、TVML を使った自動的な映像化が可能になる。

#### 4. 動作記述支援ツール

2 節のツールとの連動により、過去に書いた action 用のスクリプトを再利用出来る仕組みを使って登場人物の動作を含む TVML スクリプトのユーザによる記述の補助を行う支援ツールを開発した。本システムは、スクリプト記述の意味内容は一切考慮せず、既存のスクリプトを逐次的に蓄積し、それを記述候補として提示し、ユーザがその中の適切なものを選択して行くことで目的とする記述を得ようとするものである。携帯電話の電子メールシステムで良く使われているユーザの使用履歴に基づく仮名漢字変換候補の提示と類似の発想である。開発は Visual C++6 で行った。詳細な説明は[高橋 2006][小方 2007]にある。

候補の提示方法について説明する。それ以前に記述され蓄積されているスクリプトと現在記述中のスクリプトを比較し、次に来ると思われるイベントもしくはコマンドを予測する、という方法を取る。図 9 にその概要を示す。n=1 から始め、現在入力中の行の n 行前までを取り出し、今までに記述したスクリプトと比較して行く。同じパターンの並びを見つけた場合、その並びの次の行のスクリプトのスコアを 2<sup>n-1</sup> 上げる。一度でも同じパターンの並びが見つかった場合、n に 1 を増やし、見つからなくなるまで同様のことを繰り返す。スコアが高い順に表示され、ユーザはそこから適切と思われるものを選択し、新しい TVML スクリプトを記述して行く。これらの処理は図 10 のようなインタフェース画面を通じて行うことが出来る。



機能	加害
副機能	敵対者が人間を誘拐する
イベント例 実例	不意にその蛇が現れ、皇女達を火の翼に乗せて、凌って行ってしまいました。
イベントフ レーム	[動作主格]が現れる。/[動作主格]が[対象格]を[道具]に乗せる。/[動作主格]が[対象格]を浚う。
動作のテ ンプレート とその概 要	<pre>// [動作主]が中央に向かって歩く 01 &gt; character: walk(name=[AGENT], x=0) // [乗り物]を用意する 02 &gt; character: definepose(name=[AGENT], pose=Call,     joint=LeftUpperArm, rotx=-135, roty=-45) 03 &gt; character: definepose(name=[AGENT], pose=Call,     joint=RightUpperArm, rotx=-135, roty=45) 04 &gt; character: definepose(name=[AGENT], pose=Call,     joint=Head, rotx=-45) 05 &gt; character: pose(name=[AGENT], pose=Call) 06 &gt; character: shake(name=[AGENT]) 07 &gt; wait(time=1) 08 &gt; character: shake(name=[AGENT], state=off) 09 &gt; character: pose(name=[AGENT], pose=default) // [主体]が[対象]を[乗り物]に乗せる 10 &gt; prop: position(name=[INSTRUMENT], x=-0.5,     scale=0.5) 11 &gt; prop: visible(name=[INSTRUMENT], switch=on) 12 &gt; character: walk(name=[AGENT], x=-0.5) // [対象]が姿を消す 13 &gt; character: visible(name=[OBJECT], switch=off) // [乗り物]が飛んでいく 14 &gt; skipscript(switch=on) 15 &gt; prop: position(name=[INSTRUMENT], x=0, y=-0.7,     yaw=90) 16 &gt; prop: attach(name=[INSTRUMENT],     charactername=[AGENT], joint=Head, switch=on) 17 &gt; skipscript(switch=off) 18 &gt; character: shake(name=[AGENT]) 19 &gt; wait(time=3) 20 &gt; character: shake(name=[AGENT], state=off) 21 &gt; skipscript(switch=on) 22 &gt; prop: attach(name=[INSTRUMENT],     charactername=[AGENT], joint=Head, switch=off) 23 &gt; prop: position(name=[INSTRUMENT], x=0.0, y=-     0.75) 24 &gt; skipscript(switch=off) 25 &gt; character: visible(name=[AGENT], switch=off) 26 &gt; prop: transform(name=[INSTRUMENT], ty=3.0,     tframe=40)</pre>

図 8: 行為・動作の階層的構造の例

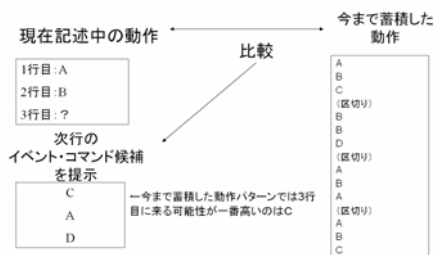


図 9: 候補選択の概要



図 10: 記述支援のインタフェース画面例

## 5. むすび

以上、物語の概念表現を映像用スクリプトに半自動的に変換するツールを説明し、その中で行為・動作の概念を映像化するための基本的な考えを示した。この時難しい問題は、意味論的な体系である行為を物理的な体系である動作に変換することである。上記ツールでは予め特定のラベルを付せられた行為と動作との対応付けと行為の具体化としての動作の定義がなされていなければならない。そのためにここでは過去の履歴からパターンを検出してそれを候補としてユーザに提示し、ユーザがその中から適切なものを選択することによって TVML スクリプト記述の支援を行うツールを試みた。これにより一編の短編物語全てを映像化する作業が可能になり、予備評価では登録に要する時間がツール未使用の場合と比べて短縮されていることが確認されているが[高橋 2006]、本稿では、評価以前に以下のような基本的な問題があるので、評価については述べない。まずこのツールは適切なものを選択する作業を伴うので TVML スクリプトの形式をある程度理解した上でなければ使えない。仮名漢字変換に見るように自然言語では便利な処理であるが、プログラム言語ではそれ程容易ではない。また蓄積されたデータがある程度増えないと有効に機能しない。これらの問題が容易に浮ぶためこれ自体としての評価作業はまだ行っていない。より有効な処理として、動作の映像を直接ユーザが操作すると、それを自動的に TVML スクリプトとして記述する新たな支援ツールの開発を行っている[真部 2007]。またより根本的な考察として、動作と行為それぞれの体系及びその関連性の検討と行為・動作のオントロジー開発を予定している。その基礎的発想は本稿の 3 節に示した。本稿における動作記述支援ツールは、これらの関連研究との関連において再考する予定である。

## 参考文献

- [林 2005] 林正樹:めざせ!テレビ番組クリエイター パソコンと番組記述言語 TVML で実現!! ,技術評論者,2005 .
- [真部 2007] 真部雄介・小方孝:物語生成における映像からの動作概念記述のボトムアップアプローチ 動作から行為への階層性に基づいて ,人工知能学会第 21 回全国大会,1F1-7,2007 .
- [松田 2005] 松田亜矢子:物語生成システムにおけるマルチメディア表現に向けて 物語の概念構造と TVML の結合 ,山梨大学工学部コンピュータ・メディア学科卒業論文,2005 .
- [小方 2007] 小方孝・松田亜矢子・内藤祐介・真部雄介・高橋昇・中嶋美由紀・吉尾貴史・沼田真克:物語生成システムにおける映像表現,人工知能学会第二種研究会 ことば工学研究会(第 25 回)資料,19-59,2007 .
- [Propp 1969] Propp, V. (Пропп, В. Я.): Морфология сказки, Изд. 2е. Москва:Наука, 1969. (北岡誠司・福田美智代訳:昔話の形態学,白馬書房,1987.)
- [佐久間 2006] 佐久間友子・小方孝:物語自動生成によるストーリー作成支援の検討,人工知能学会第二種研究会 ことば工学研究会(第 24 回)資料,83-88,2006 .
- [高橋 2006] 高橋昇:動作記述の支援 物語生成システムにおける映像表現のための一機構として ,岩手県立大学ソフトウェア情報学部卒業論文,2006 .