

構造に着目した物語 音楽連携システムにおける循環的物語生成

- 音楽変奏機構と音楽から物語への変換機構の実装を中心に -

Cyclic Narrative Generation in the Narrative/Music transformation System based on the Structures
Implementations of Music Variation Mechanism and Transformation Mechanism from Music to
Narrative藤田 洋佑^{*1}
Yosuke Fujita小方 孝^{*2}
Takashi Ogata^{*1} 日本システムウェア (株)
Nippon Sytemware, Co.,Ltd.^{*2} 岩手県立大学ソフトウェア情報学部
Faculty of Software Information, Iwate Prefectural University

ABSTRACT: In our research in which combines narrative generation process and music variation process based on the structures, in this paper, we mainly propose concrete implementation in the many parts of the system framework. In the section 2, we show a piece of process of music making from a narrative representation, music variation based on narrative discourse techniques and the transformation from the changed music to a different narrative representation. In the section 3, we especially explain the last part, the transformation from music to narrative.

1. はじめに - システムの枠組みと基本思想 -

それぞれの構造に着目することで、物語と音楽とを相互変換し、音楽からの物語生成を可能にするシステムのプロトタイプについて提案して来た[Kobayashi 2004][小方 2004]。これまで試作システムの大枠を作り部分的に詳細な実装を行ったが、未設計・未実装の部分も多く残っていた。今回それらの部分の多くを実装し、同時に従来の方法の幾つかを改訂したので、その結果を示す。但し紙数の都合で実装の詳細は示せないのので、物語と音楽の相互変換、つまり物語と音楽との循環的な変奏サイクルについて簡単な実行例を使って提示することにする。詳細な記述は別稿に譲りたい。

システムの枠組みは従来の研究を踏襲する(詳細は[小方 2004]を参照されたい)。システムは原曲構成機構、音楽変奏機構、物語変換機構に分かれる。Common Lisp で記述された物語概念表現を入力として原曲構成機構がそれを曲に構成してMax/MSP による原曲情報として出力する。音楽変奏機構はこの原曲情報をもとに物語言語論と対応付けられた音楽変奏技法によって原曲を変奏し、同じくMax/MSP による変奏情報を出力する。物語変換機構はこの変奏情報をもとに再度変奏された物語の概念表現を出力する。この物語概念表現をそのままの形式で再度原曲構成機構に渡すことが出来るので、原理的に無限の変換過程が実行出来ることになる。

本研究では、どのような物語・音楽を目指すかといったイメージ・目標・意図のような概念は一切排し、構造的な対応関係だけを使って変換を行う事を基本思想としている(音楽構造とテキスト構造との対応関係についての考察は[Bod 2002]、音楽構造と物語構造との対応に基づく音楽分析の試みは[Tarasti 1994]がある)。また物語の理論を音楽の変奏技法に流用しているが、逆に音楽の理論を物語の生成や変奏に使うことも狙っており、両者の方法論的な浸潤関係の実現を意図している。さらにコンピュータによる物語生成の方法と文学理論や物語論との融合を目指す「拡張文学理論」[小方 2003]の一環として、本研究でもプロットの物語内容研究[Propp 1969]やジュネットの物語言語論研究[Genette 1972]の成果を換骨奪胎して流用している。

2. 実行例 物語と音楽の循環の概要

システムの実行例を入出力を中心に示す。以下は入力となる物語概念表現の一部省略した記述である(これが原曲構成機構への入力となる)。

```
(Event1
 (sign A-1)
 (concept (($s (敵対者)) ($v 誘拐する) ($do (被害者))))
 (((ActionData (id 1) (action (action ((adverb 不意に)) ...)))
 (Event2
 (sign B-1)
 (concept (<evt-++> (($s (派遣者 名前 国王)) ($v 叫びをあげる for (物 名前 助け))))
 (((ActionData (id 1) (action (action (agent nil) ...))))
 (Event3
 (sign C-0)
 (concept (/=> (($s (主人公 行動型 探索者型))(&v 対抗開始する)) ...))
 (((ActionData (id 1) (action (action 選抜される) ...))))
```

この表現はプロットの機能論の応用としての物語表現に基づくもので、event の中に機能(抽象的な動作)に対応する concept と具体的な動作系列に対応する action が含まれる。実際の物語の表現に相当するのは action の方で、以下に、上の例の action を手作業でテキストに変換したものを示す。

「(イベント1)不意にその蛇が現れ、皇女達を火の翼に乗せて、渡って行ってしまいました。(イベント2)皇帝はいつまでたっても皇女達がお部屋に帰って来ないので、お側の者をやって、お庭を訪ねさせましたが、影も形も見えません。翌朝になると、皇帝は一層心配になりました。そこで侍臣一同をお集めになって、こう仰せられました。物が皇女達を探し出したものには、望みどおりの褒美を取らせる！」(イベント3)三人の者が選抜されました。大酒呑みの兵士と、隠者のフールカ、及びエリイマ

^{*1} 連絡先: 藤田 洋佑, 住所: 東京都渋谷区桜丘町 31-11
Mail: fujirik o59@yahoo.co.jp

の三人でした。三人の者は皇帝と打ち合わせをしてから、皇女を探しに出掛けました。」

なおこの例は、プロットが分析に取り上げた物語の中的一篇であり、TVML による物語生成からの映像表現の研究で使用している例で、その研究と将来的に連携を取ることを意図して、ここで用いている。

図 1 の楽譜は構成された原曲で、右側のデータは変奏機構に渡される原曲情報である。原曲情報における*1 から*9 までの情報が一つの event に対応するもので、そのうちの最初の三つはそこに登場する人物を意味する。この例は予め人手で用意した曲の断片を組み合わせ原曲を作っている。なお[小方 2004]でも述べたように原曲構成の元歌作成の方法は一つでなく複数あり、自動作曲システム[Kobayashi 2002]を利用しても可能である。原曲情報におけるその他の番号は以下のような情報に対応している。*4 - 時間順序,*5 - 頻度,*6 - 持続,*7 - 距離,*8 - パースペクティブ,*9 - 変調。持続は曲の速さに対応し、1024 をデフォルトとしている。

原曲情報

- 11, teki;
- 12, higaisya;
- 13, kagai;
- 14, 0;
- 15, 0;
- 16, 1024;
- 17, 0;
- 18, 0;
- 19, 0;
- 21, hakensya;
- 22, syujinkou;
- 23, tyukai;
- 24, 0;
- 25, 0;
- 26, 1024;
- 27, 0;
- 28, 0;
- 29, 0;
- 31, syujinkou;
- 32, higaisya;
- 33, taikoukaisi;
- 34, 0;
- 35, 0;
- 36, 1024;
- 37, 0;
- 38, 0;
- 39, 0;

図 1 原曲の楽譜と対応する原曲情報

図 2 は変奏された曲の楽譜と変奏情報を示す。これはイベント1に休止法、イベント2に距離 大の変奏を行った変奏曲である。

変奏情報

- 11, teki;
- 12, higaisya;
- 13, kagai;
- 14, 0;
- 15, 0;
- 16, 100;
- 17, 0;
- 18, 0;
- 19, 0;
- 21, hakensya;
- 22, syujinkou;
- 23, tyukai;
- 24, 0;
- 25, 0;
- 26, 1024;
- 27, 4;
- 28, 0;
- 29, 0;
- 31, syujinkou;
- 32, higaisya;
- 33, taikoukaisi;
- 34, 0;
- 35, 0;
- 36, 1024;
- 37, 0;
- 38, 0;
- 39, 0;

図 2 変奏された曲の楽譜と対応する変奏情報

物語変換機構がこの変奏情報に基づいて Lisp によるフレーム表現としての物語を構成する。次はその action の部分を手作業で日本語に変換したものである。ここでは event 自体の個数は変化していないが、その細部が少し変化している。もし反復の技法や括復法の技法が使用されれば、よりマクロなレベルでも返還が行われ、event の個数も変る。

「(イベント1)不意にその蛇が現れ、皇女達を火の翼に乗せました。蛇はどこか集団にも所属はしていません。蛇は皇女達を浚って行ってしまいました。(イベント2)皇女達は影も形も見えませんが、皇帝というのは非常に大変な立場です。皇帝の名前は国中で知らない者はいないでしょう。侍臣は王国に仕えています。皇帝は侍臣達にこう仰せられた。「わが皇女達を探し出したものには望みどおりの褒美を取らせる！」皇帝は男です。(イベント3)三人の者が選抜されました - 大酒呑みの兵士と、隠者のフローカ、及びエリョマの三人でした。三人の者は皇帝と打ち合わせをしてから、皇女を探しに出掛けました。」

3. 変奏された音楽から物語への変換機構

音楽から物語への変換機構(物語変換機構)は、物語概念表現と変奏情報を入力とし、最初の入力と同じ形式の物語概念表現を返す。変奏情報から変奏パラメータを抜き出し、音楽変奏機構において音楽に対して行われた変奏方法に対応した変換方法により、入力された物語概念表現を变形し、物語概念表現とその部分を成す action の系列を、それぞれテキストファイルとして出力する。ここで、action 系列を書き込んだテキストファイルは、システム内の処理のための情報のような余分な情報がない、純粋な物語の表現、ユーザに物語を提示するためのものである。

処理の流れを以下に示す - (1)物語概念表現と変奏情報を読み込み、どちらもリストとして保持する。(2)時間順序の変奏パラメータから、物語概念表現の構造を变形する。(3)対象 event の頻度・パースペクティブ以外の変奏パラメータを参照し变形を行う。また、前の event の頻度の変奏パラメータを参照し括復法なら括復法の变形を行う。(4)対象 event の頻度の変奏パラメータ

タを参照し、反復法なら反復法の方法で物語を変形する。(5)(3)と4)をすべての event について行う。(6)すべての event について、パースペクティブの変奏パラメータを参照して action の選別変形を行い、action 系列をテキストファイルとして出力する。(7)変形された物語概念表現をテキストファイルとして出力する。

音楽変奏機構から出力された変奏情報の変奏パラメータから、物語概念表現を変形するために、物語構造と音楽構造の対応付けを行った。まず A メロ B メロといった音楽の時系列的構成(時間順序の言説技法)は、物語の event の配列と対応する。また、音楽の速度、調、和声、音色といった、ある構成単位の音楽を装飾するよぶな要素(頻度、持続、距離、パースペクティブの言説技法)については、event 内の action に対応させる。以下に、各言説技法における変換方法をやや詳しく説明する。

時間順序

時間順序に関する音楽変奏方法には、先説法と後説法がある。物語への変換方法としては、先説法の場合一番最後の event を先頭へ移動し、後説法の場合最初の event を最後に移動する。これにより物語の全体構造が変化する。

頻度

頻度に関する音楽変奏方法は、単起法、反復法、括復法の三種類がある。物語への変換方法として、単起法と反復法はそれぞれ event に対応する変奏パラメータを読み取り、括復法の場合変奏パラメータが設定されている event の次の event に対して処理を行う。変奏パラメータなら 0 単起法、正の整数なら反復法でパラメータは反復回数となる。変奏パラメータが -1 括復法とする。単起法は物語に対して何も処理を行わない。反復法の場合は、対応する event を変奏パラメータで指定した数だけ複製する。括復法では event を複製し、その片方をその他の変換方法により変形し、変形した event と変形前の event における action のリストをマージする。マージの方法としては、両者の action を ID に従って羅列し、同じ action は 1 つにする、という手法を取っている。頻度の変換例を表 5-3 に示す。

持続

持続に関する変奏方法には、休止法、情景法、要約法、省略法の四種類のものがある。パラメータが 100 なら休止法、512 なら情景法、1536 もしくは 2048 なら要約法、0 なら省略法とする。

休止法では、event 内の action のリストの物語言説部分に「語りイベント」という事象の進行を停止させるイベントを挿入し、物語内容の時間経過を一時中断する。例えば、**蛇が突然現れる。蛇は皇女達を火の翼に乗せる。蛇は皇女達を火の翼で飛って行く。**という物語に休止法を適用すると次のようになる。**蛇が突然現れる。蛇は黒海に棲みついているが、なぜあんな場所に生息しているのだろう。蛇は皇女達を火の翼に乗せる。蛇は皇女達を火の翼で飛って行く。**

情景法では、物語言説の時間経過を物語内容の時間経過へ近付ける。これは、物語内容をより詳しく語ることで実現出来ると考えられ、情景法にはそのための 2 つの処理を持たせる。後に述べる要約法等により、既に action の要約が行われていた場合は、action の要約を復元し、action の要約が行われていない場合は描写のイベントを挿入する。いずれもイベント内の action リストへの操作となる。

要約法ではイベント内の action のリストにおける物語言説部分を要約する。手順は以下の通りである。(1)action リスト中の物語言説部分から最も重要度の低い action を選択する。語りイベ

ントが存在する場合は、語りイベントを選択する。(2)選択した action を物語言説とされない部分へ移動する。語りイベントを選択した場合は語りイベントを削除する。(3)パラメータが 2048 だった場合は、物語言説部分にある action の個数の半分の回数だけ、(1)と(2)を繰り返す。パラメータが 1536 なら繰り返さず終了する。

距離

距離には、大きくする、小さくする、という種類の変奏方法がある。変奏パラメータが 2 か 4 の場合距離を大きくし、-1 の場合小さくする。物語における距離の操作は、[山影 2003]で用いられている方法を参考にした。距離を大きくする場合は、物語言説部分を要約し、要約した action の数だけ語りイベントを挿入する。これらの手順は、持続の要約法や休止法と同じになる。距離を小さくする場合は、語りイベントを削除し、情景法と同じ手順で要約の復元または描写の挿入を行う。例えば、**蛇が突然現れる。蛇は皇女達を火の翼に乗せる。蛇は皇女達を火の翼で飛って行く。**という物語の距離を大きくした例は次のようになる。**蛇は火の翼を持っていて、空を飛ぶことができる。蛇は皇女達を火の翼で飛って行く。第一皇女は非常に優しい人なので、国民から慕われているだろう。**

パースペクティブ

パースペクティブの変換方法は他の変換とは扱いが異なる。他の変換方法では物語概念表現を変形するのに対して、パースペクティブでは、他の変換方法によって変形された物語概念表現に対して特定の視点を採用して、視点による制限を受けた action 系列を生成する。この action 系列が物語変換機構の出力の一つとなる。パースペクティブの変奏方法には、全知(視点)、内的固定(視点)、外的(視点)の三種類がある。変奏パラメータが 0 なら全知、11 なら主格の登場人物の内的固定視点、21 なら目的格の登場人物の内的固定視点、12 なら外的視点となる。全知視点の場合は、物語言説部分にある action をそのまま出力する。内的固定視点の場合は、焦点化された登場人物が知覚していない action は出力しない。また、ある action が焦点化された登場人物によって異なる知覚をされていた場合は、その人物が知覚している action を出力する。これらの情報は、内的固定視点のための情報として、各 action に付随している。外的視点の場合は、思う、考える、といった内的な action は出力しない。内的な action は予め用意した知識ベースから検索して使用する。現在この知識ベースは、思う、考える、といった動詞が記述してあるだけの簡単なものとなっている。

次の例は、前の物語を目的格の登場人物であるフロールカの内的視点で変換したものである。「ナレーション」**皇帝はいつまでたっても皇女達がお部屋に帰って来ないので、皇帝はお傍の者を庭へ訪ねさせる。ナレーション 影も形も見えませんが、皇帝は心配になる。皇帝はフロールカと兵士とエリョーマと侍臣を集める。皇帝は 我が皇女達を探し出した者には、望み通りの褒美を取らせる。言う。」** - **皇帝はフロールカと兵士とエリョーマと侍臣を集める。皇帝は 我が皇女達を探し出した者には、望み通りの褒美を取らせる。言う。」**

4. おわりに

筆者らが提案した、物語から音楽を構成し、音楽を変奏してその結果を物語に変換し、さらにこのプロセスを反復可能なシステムの枠組みに基づいて、デザインと実装を進めた。問題点として、物語処理の部分が Common Lisp 環境で行われ、音楽処理の部分が Max/MSP 環境で行われており、両者の接合が出

来ていないので、処理が分断されてしまうことがある。また、物語概念表現を最初に用意しておく必要があるので、音楽を最初に作りそこから物語への自動変換を行う処理は出来ず、相互変換処理としては不完全なものとなっている。

今後は、以上のような問題点や、本質的な問題である物語(テキスト)と音楽との相互性や関連性についての理論的検討を進め、研究を継続して行きたい。

参考文献

- [Bod 2002] Bod, R.: A General Parsing Model for Music and Language, *Music and Artificial Intelligence*, 5-17, 2002.
- [Genette 1972] Genette, G.: *Discours du recit, essai de methode*, Figures III, Seuil, 1972. (花輪光, 和泉涼一訳, 物語のディスクール, 水声社, 1985.)
- [Kobayashi 2002] Kobayashi, F. & Ogata, T.: Automatic Melody Composition Based on the Constraint of Chord and Music Knowledge, *Proceedings of 17th Congress of the International Association of Empirical Aesthetics*, 565-568, 2002.
- [Kobayashi 2004] Kobayashi, F. and Ogata, T.: A Musical Variation System by the Structural Correspondence between Music and Narrative, *Proceedings of 18th Congress of the International Association of Empirical Aesthetics*, 555-559, 2004.
- [小方 2003] 小方孝: 物語の多重性と拡張文学理論の概念, 吉田雅明編, 複雑系社会理論の新地平, 専修大学出版局, 127-181, 2003.
- [小方 2004] 小方孝, 小林史典: 変奏からの物語生成への接近 - 物語と音楽の変換及び音楽変奏システムの試作に基づく諸考察 -, 人工知能学会第二種研究会ことば工学研究会 (第 17 回) 資料, 1-33, 2004.
- [Propp 1969] Propp, V. (): : , 2 . : , 1969. (北岡誠司, 福田美智代訳: 昔話の形態学, 白馬書房, 1987.)
- [Tarasti 1994] Tarasti, E.: *A Theory of Musical Semiotics*, Indiana University Press, 1994.
- [山影 2003] 山影沙耶夏, 小方孝: 物語における距離の計算機構の提案, 2003 年度人工知能学会全国大会 (第 17 回) 論文集, 2G2-05, 2003.