概念体系と結び付いたプロップに基づくストーリー生成システム
A Story Generation System based on Propp Combined with a Conceptual System

今渕 祥平[†]，大石 顕祐[‡]，小野寺 康[‡]，小方 孝[†]
Shohei Imabuchi, Kensuke Oishi, Kou Onodera, Takashi Ogata

[†]岩手県立大学，[‡]岩手県立大学大学院
Iwate Prefectural University, Graduate School of Iwate Prefectural University
g031g017@s.iwate-pu.ac.jp

Abstract

Propp's folktale theory has been a theoretical foundation for narrative or story generation system design, and we have also proposed some kinds of systems introduced the narratological idea. We think that Propp based mechanism or module is one of elements in the framework of integrated narrative generation system. In order to introduce the Propp module into it, we use the conceptual system to be used commonly in the integrated narrative generation system. This paper reports the implementation of the pilot version.

Keywords — Story Generation System, Propp, Propp Based Story Grammar, Conceptual System, Narrative Generation System,

1. 背景

筆者らはプロップの昔話の形態学(Propp, 1969)を参考に物語生成システムの一機構を担うシステムを研究している．ロシアの民俗学者ウラジーミル・プロップの昔話の形態学は，構造主義物語論（ナラトロジー）における物語内容研究の古典である．これを利用したストーリー生成システムの試みは従来からある（Klein et. al, 1974; Peinado & Gervás, 2005）．筆者らの先行研究(Ogata & Terano, 1991; 小方, 2007)では，物語生成システム(小方, 2003a, 2003b 等)への導入という観点から，物語の構成要素への解体としてプロップの昔話の形態学を整理し，コンピュータシステムとしての再構成例を提案した．

ここで物語生成システムの全体像を図 1 に示す．これは大きく概念構造(深層構造)の生成部分と表現(表層構造)の生成部分に分かれる．概念構造の生成部分は物語の中で時間順に生起する出来事を表現する物語内容，その語り方の構造を規定する物語言説に分かれる．システムのこのような大きな枠組みの中に相対的に自律したたくさんの小システムが含まれ，相互作用することで物語テクストが生成される．プロップに基づくシステムは物語生成システムにおいては物語内容機構の中に位置づけられる．

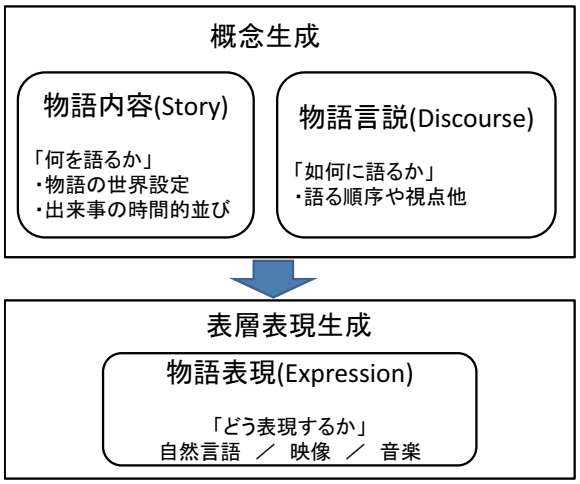

図 1　物語生成システム

2. 目的

上記のように，プロップに基づくシステムは全体としての物語生成システムの構成要素として構想されているが, 先行研究で提案したシステムは, データ構造の違い等により物語生成システムのその他の部分と統合できない状態であった．この問題点を解決すべく筆者らは概念体系と結び付いたプロップに基づくストーリー生成システムを提案する．現状では統合的な物語生成システムの開発自体も並行して進めているところで，既存のシステムに単に統合すれば良いということではないが, 他の機構との結合方式を考慮しながらプロップに

基づくストーリー生成の拡張を図ることで，統合された物語生成システムのあり方を考えることもここでの目的のひとつである．

具体的には，他の機構でも共通に用いる動詞/名詞概念体系と結合して後述のストーリーグラマー等で使用する諸概念をそれと結び付け，また事象やフレーム等の記述形式を他の機構と統一する．また，旧来のシステムは捨てて Common Lisp によってゼロから実装し直した．

なお以下で実装するシステムはストーリーグラマーに基づくシステムであるが，このストーリーグラマーは認知科学で提唱されたストーリーグラマー(スキーマ)と比較してより内容志向の特徴を持っている．また，小方 (2007)が示すようにプロップに基づく生成方法はストーリーグラマーだけではなく，その意味で本研究はまだプロップに基づくシステムとしても部分的な実現である．

3. システムの概要

新しく作り直したシステム構成図を図 2 に示す．システムはストーリーグラマーとそれを参照し処理する部分，プロップ DB から成り，動詞/名詞概念体系を参照して動作する．ストーリーグラマーは，先行研究で昔話の形態学の「機能」と例の記述から小方が生成文法風に整理したものを参考にして多少手直しして作成した．プロップによる「機能」とは物語の構造を規定する最も基本的な要素であり，ロシア昔話では表 1 に示す 31 個の「機能」が原則としてこの順序で並んでいるとされる．また，プロップは各「機能」の下位レベルに複数の実現方法を掲げている．筆者らはその実現方法を「副機能」と呼ぶ．表 2 はストーリーグラマーの実際の記述の一部であるが，これはレベル④までの多層的モデルを成す．レベル①は物語内容全体を 4 つの要素に分けて示す．レベル②は 31 個の「機能」をいくつかのグループに分けて示す．レベル③は各「機能」の「副機能」を示す．最下層レベル④は「副機能」の具体的な実現形態を示し，動詞に基づく格構造で表す．プロップ DB にはシステムで生成される事象概念やフレームが格納される．また物語生成システムの動詞/名詞概念体系は，事象概念の生成に必要な動詞/名詞情報を概念構造の生成部分に提供する．

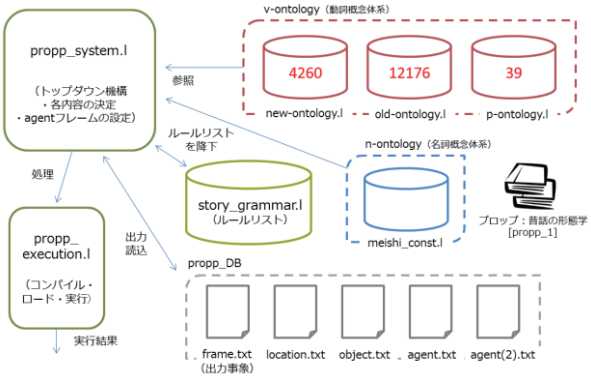

図 2　システムの構成図

表 1　31 個の「機能」

(0)予備部分（導入の状況）	(16)闘い
(1)留守	(17)標づけ
(2)禁止	(18)勝利
(3)違反	(19)不幸・欠如の解消
(4)探り出し	(20)帰還
(5)情報漏洩	(21)追跡
(6)謀略	(22)救助
(7)幇助	(23)気づかれざる到着
(8)加害（欠如）	(24)不当な要求
(9)仲介	(25)難題
(10)対抗開始	(26)解決
(11)出立	(27)発見・認知
(12)贈与者の第一「機能」	(28)正体露見
(13)主人公の反応	(29)変身
(14)呪具の贈与・獲得	(30)処罰
(15)二国間の空間移動	(31)結婚

動作の流れを説明する．まずユーザがレベル①からレベル④の物語長をパラメータとして入力する．次に「*主人公*」，「*敵対者*」等の登場人物の名前を入力する．これは agent フレームとして保存される．処理部分は入力された物語長に応じて再帰的にストーリーグラマーを参照する．参照の結果，最下層に到達した場合は事象概念を生成し(表 3)，それ以外の場合はトップダウンに参照を続ける．物語長に応じたストーリーグラマーを参照し終わると動作は終了する．最後に生成された事象概念の格情報の，登場人物を agent フレーム，物を object フレーム，場所を location フレームとして保存する．これは事象概念と共にプロップ DB に格納される．

表 2　ストーリーグラマーの一部

```
(defvar *propp-level1-list*
  '((ロシア民話 (00_導入 問題 試行 解決))))

(defvar *propp-level2-list*
  '((問題 (予備部分 発端))
    (試行 (OR (予備試練 闘いと勝利)
              (予備試練 難題解決)))
    (解決 (問題解消 到着と試練 終結))
    (予備部分 (OR (01_留守 02_禁止 03_違反)
                  (01_留守 04_探り出し 05_情報漏洩)
                  (01_留守 06_謀略 07_幇助)))
    (発端 (OR (08_加害 09_仲介 10_対抗開始 11_出立)
              (08a_欠如 09_仲介 10_対抗開始 11_出立)))
    (予備試練 (12_贈与者の第一機能 13_主人公の反応 14_呪具の贈与・獲得 15_空間移動))
    (闘いと勝利 (16_闘い 17_標づけ 18_勝利))
    (難題解決 (25_難題 26_解決))
    (問題解消 (OR (19_不幸・欠如の解消 20_帰還 21_追跡 22_救助)
                  (19_不幸・欠如の解消 20_帰還)))
    (到着と試練 (23_気付かれざる到着 24_不当な要求))
    (終結 (27_発見・認知 28_正体露見 29_変身 30_処罰 31_結婚))
    ))

(defvar *propp-level3-list*
  '((00_導入
        (alp-1_導入))
    (01_留守                                ;家族の成員のひとりが家を留守にする.
        (OR bet-1_外出 1                    ;年長の世代に属する人物の留守(外出).
            bet-2_死                        ;両親の死.
            bet-3_外出 2))                  ;年少の世代に属する人物の留守(外出).
    (02_禁止                        ;主人公に禁を課す.
        (OR gam-1_禁止                      ;禁止.
            gam-2_命令/提案))               ;命令, または提案.
    (03_違反                        ;禁が破られる.
        (OR del-1_違反                      ;禁止を破ること.
            del-2_命令実行))                ;命令を実行すること.
    ;ここで敵対者が現れる
    (04_探り出し                            ;敵対者が探り出そうとする.
        (OR eps-1_問いただし 1              ;敵対者が, 子供達の居場所, 貴重なものその他の在り処を探り出す.
            eps-2_問いただし 2))            ;犠牲となる者の方が敵対者に問いただす.
       ;eps-3                       ;他の人物を介しての探り出し.
;中略
    (30_処罰                        ;敵対者が罰せられる.
        (OR U-1_処罰                        ;二セ主人公, あるいは敵対者が, 罰せられる.
            U-bar_容赦))                    ;二セ主人公, あるいは敵対者を, 寛大に容赦する.
    (31_結婚                        ;主人公は結婚し, 即位する.
        (OR W-**_贈与 1                     ;花嫁と国が与えられる.
            W-*_結婚                        ;結婚する.
            W-_*_贈与 2                     ;王位を得る.
            W-1s_婚約                       ;結婚の約束.
            W-2s_再婚                       ;再建された結婚.
            W-3s_報賞))                     ;金銭その他の形での報酬. 175
    ))

(defvar *propp-level4-list*
  '((alp-1_導入 ((導入の状況 (1))))
    ;*****************************************************************************
    (bet-1_外出 1 ((出かける (1) (agent 両親) (object 用事))))
    (bet-2_死 ((死ぬ (1) (agent 両親))))
    (bet-3_外出 2 ((出かける (1) (agent 子ども) (object 用事))))

    (gam-1_禁止 (OR ((禁止する (1) (agent (人間)) (object 外出) (to *主人公*)))
                    ((禁止する (1) (agent (人間)) (object 発言) (to *主人公*)))
                    ((監禁する (1) (agent (人間)) (counter-agent *主人公*)))
                    ((依頼する (1) (agent (人間)) (object 留守番) (counter-agent *主人公*)))
                    ((忠告する (2) (agent (人間)) (object リンゴを食べてはならない) (counter-agent *主人公*)))
                    ((忠告する (2) (agent (人間)) (object 金の羽毛を拾ってはならない) (counter-agent *主人公*)))
                    ((忠告する (2) (agent (人間)) (object 箱を開けてはならない) (counter-agent *主人公*)))
                    ((忠告する (2) (agent (人間)) (object 妹に接吻してはならない) (counter-agent *主人公*)))
                    ((説得する (1) (agent (人間)) (counter-agent *主人公*)))))
    (gam-2_命令/提案 (OR ((命令する (1) (agent (人間)) (counter-agent *主人公*) (object 朝飯を持って行きなさい)))
                         ((提案する (2) (agent (人間)) (counter-agent *主人公*) (object 弟を連れて森へ行きなさい)))))
;中略
    (del-1_違反 (OR ((反す (1) (agent *主人公*) (object 禁)))
                    ((破る (1) (agent *主人公*) (object 禁)))
                    ((犯す (1) (agent *主人公*) (object 禁)))))
    (del-2_命令実行 ((実行する (1) (agent *主人公*) (object 命令))))

    (eps-1_問いただし 1 ((探る (4) (agent *敵対者*) (counter-agent *被害者*))))
    (eps-2_問いただし 2 ((探る (4) (agent *被害者*) (counter-agent *敵対者*))))

    (U-1_処罰 ((罰する (1) (agent *派遣者*) (counter-agent *敵対者*))
              (罰する (1) (agent *派遣者*) (counter-agent *二セ主人公*))))
    (U-bar_容赦 ((容赦する (1) (agent *派遣者*) (counter-agent *敵対者*))
                (容赦する (1) (agent *派遣者*) (counter-agent *二セ主人公*))))

    (W-**_贈与 1 (OR ((得る (1) (agent *主人公*) (object 花嫁)))
                     ((得る (1) (agent *主人公*) (object 国)))))
    (W-*_結婚 ((結婚する (1) (agent *主人公*) (counter-agent *被害者*))))
    (W-_*_贈与 2 ((得る (1) (agent *主人公*) (object 王位))))
    (W-1s_婚約 ((婚約する (1) (agent *主人公*) (counter-agent *被害者*))))
    (W-2s_再婚 ((結婚する (1) (agent *主人公*) (counter-agent *被害者*))))
    (W-3s_報賞 ((得る (1) (agent *主人公*) (object 報酬))))
    ))
```

最下層を参照した際の事象概念生成の方法を表 3 に示す．まず最下層で参照した動詞と動詞概念体系を照合し格フレームを取得する．そして取得した格フレームに，最下層に記述されている格情

報を挿入する．この際，格情報に「*主人公*」等アスタリスクに囲まれている記述が現れた場合，agent フレームを参照しその名前を割り振る．また，格情報に「(人間)」等括弧に囲まれている記述が現れた場合，名詞概念体系を参照しその下位概念からランダムに取り出し挿入する．こうして事象概念が生成される．実行結果を表 4 に示す．

表 3　事象概念生成の方法

最下層で参照した動詞と動詞概念体系を照合し格フレームを取得する．
例：最下層レベル④…(禁止する (1) (agent (人間)) (object 外出) (to *主人公*))

ここでは「禁止する (1)」と名詞概念体系を照合する．
例：(set (intern "禁止する(1)") '((name (禁止する (1))) (sentence-pattern "N1 が N2 に/へ N3 を 禁止する") (case-cons-set ((case-frame ((agent N1) (counter-agent N3) (location nil) (object N3) (instrument nil) (from nil) (to N2))) (constraint (("主体") ("主体") ("*"))))) (is-a (精神的移動))))
動詞概念体系の「禁止する (1)」を参照し，格フレームを取得する．

取得した格フレームに，最下層に記述されている格情報を挿入する．
例：(case-frame ((agent (人間)) (counter-agent N3) (location nil) (object 外出) (instrument nil) (from nil) (to *主人公*))

格情報にアスタリスクに囲まれている記述が現れた場合，agent フレームを参照しその名前を割り振る．
例：*主人公*→agent フレームの*主人公*参照「イワン」
格情報に括弧に囲まれている記述が現れた場合，名詞概念体系を参照しその下位概念からランダムに取り出し挿入する．
例：(人間)→人間の下位概念「誰かさん」

こうして事象概念が生成される．
(event 禁止する (1) (type action) (ID 1) (time (1 2)) (agent 誰かさん) (counter-agent nil) (location nil) (object 外出) (instrument nil) (from nil) (to イワン))

4. 問題点と解決案

以上のように，試作の枠組みはほぼ出来上がったが，完成度を上げるためには様々な課題がある．まず，名詞概念体系と物語の登場人物や物，場所とが完全に結び付いていない．これは agent フレーム等各フレームに概念スロットを設けて，名詞概念体系に存在する値を挿入することで解決を図る．例えば物語に現れる「鳥」に「動物」，「花嫁」に「女」等の上位概念を与えることを目指す．そのため昔話の形態学で取り上げられているロシア昔話数話において，出現する名詞の抽出，分析を行っている．またこれに伴い，ストーリーグラマーの最下層に記されている格情報を名詞概念体系と結び付く形に改善する必要がある．たとえば表 2 では格情報「リンゴを食べてはならない」を object と定めているが，これを入れ子の事象として階層化する記述法に拡張する．また，台詞などの格情報を格納できる新たなスロットの提案が必要である．

この種の細部の問題点の抽出と解決を図る必要があるが，同時に，この機構を物語生成システム全体の中に組み込む方式の検討を行う予定である．物語生成システムが生成する概念表現は談話的な構造記述であるが，構造知識にミクロからマクロへの段階を設ける案を組織的に検討中である．本稿で述べた方法はマクロ構造に相当する．従って，大きな物語を一挙に構築する方法として上述のシステムを直接的に組み込むことが出来る．しかし，プロップのストーリーグラマーの中には，禁止ー違反，加害ー解消等，ミクロな談話関係が含まれており，これらを独立して利用することも出来る．この辺りを起点として，プロップの方法の統合物語生成システムへの組み込みについて検討を進めて行きたい．

参考文献

Klein, S., Aeschlimann, J. F., Appelbaum & M. A. et al. (1974). Modeling Propp and Levi-Strauss in a Meta-Symbolic Simulation System. Computer Sciences Technical Report 226. University of Wisconsin.

小方 孝 (2003a). 物語の多重性と拡張文学理論の概念—システムナラトロジーに向けて I—. 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.

小方 孝 (2003b). 拡張文学理論の試み—システムナラトロジーに向けて II—.『複雑系社会理論の新地平』. 専修大学出版局. 309-356.

小方 孝 (2007). プロップから物語内容の修辞学へ—解体と再構成の修辞を中心として—.『認知科学』. 14(4). 532-558.

Ogata, T. & Terano, T. (1991). Explanation-Based Narrative Generation Using Semiotic Theory. Proc. of National Language Processing Pacific Rim Symposium 91. 321-328.

Peinado, F. & Gervás, P. (2005). Creativity Issues in Plot Generation. Workshop on

Computational Creativity, Working Notes, 19th International Joint Conference on Artificial Intelligence. 45-52.

Propp, V. (Пропп, В. Я.) (1969). Морфология скаэки, Иэ, 2е. Москва: Наука.(北岡 誠司・福田 美智代 訳 (1987). 『昔話の形態学』. 白馬書房.)

表 4　実行結果

```
(propp 'ロシア民話)
==================================
デフォルトは 0 入力は 1：1
==================================
*主人公*：イワン
*敵対者*：蛇
*被害者*：姫様
*呪具*：魔法の杖
*派遣者*：王様
*贈与者*：バーバ・ヤガー
*二セ主人公*：メロス
==================================
導入の状況が示される
子どもが出かける
蛇が変身する
蛇が毒入りの錠剤をイワンに使う
イワンが眠る
蛇がひとに海に投げ込めと命令する
ひとが命令を実行する
イワンが被害を知る
イワンが決意する
イワンが対抗を開始する
イワンが出立する
二人の巨人がツァンツァを取り合う
イワンが分配する
イワンが二人の巨人を仲直りさせる
イワンが魔法の杖を発見する
イワンが魔法の杖を入手する
魔法の杖がイワンを案内する
イワンが■他国へ行く
イワンが蛇と罵り合う
イワンが蛇と競争する
イワンが傷を負う
蛇がイワンにカードで負ける
イワンが魔法の杖を使う
イワンが効果で貧しくなる
イワンが■場所へ帰還する
蛇がひとに変身する
蛇が道を妨げる
イワンが鍛冶屋の許へ隠れる
鍛冶屋たちが蛇を叩きのめす
イワンが家郷に到着する
メロスが報酬を要求する
イワンが難題を解決したことによって認知される
姫様が総てを語る
メロスが露見する
イワンが■新たな姿形に変身する
王様が蛇を罰する
王様がメロスを罰する
イワンが花嫁を得る
((event 出かける (1) (type action) (ID 1) (time (1 2)) (agent 子ども) (counter-agent nil) (location 家) (object 用事) (instrument nil)
(from nil) (to nil)) (event 変身する (1) (type action) (ID 2) (time (2 3)) (agent 蛇) (counter-agent nil) (location 家) (object ■変身)
(instrument nil) (from nil) (to nil)) (event 使う (3) (type action) (ID 3) (time (3 4)) (agent 蛇) (counter-agent nil) (location 家) (object
毒入りの錠剤) (instrument nil) (from nil) (to イワン)) (event 眠る (1) (type action) (ID 4) (time (4 5)) (agent イワン) (counter-agent nil)
(location イワン) (object nil) (instrument nil) (from nil) (to nil)) (event 命令する (1) (type action) (ID 5) (time (5 6)) (agent 蛇)
(counter-agent ひと) (location イワン) (object 海に投げ込め) (instrument nil) (from nil) (to nil)) (event 実行する (1) (type action) (ID 6)
(time (6 7)) (agent ひと) (counter-agent nil) (location イワン) (object 命令) (instrument nil) (from nil) (to nil)) (event 知る (1) (type
action) (ID 7) (time (7 8)) (agent イワ
中略
(from nil) (to nil)) (event 語る (1) (type action) (ID 32) (time (32 33)) (agent 姫様) (counter-agent nil) (location 家郷) (object 総て)
(instrument nil) (from nil) (to nil)) (event 露見する (1) (type action) (ID 33) (time (33 34)) (agent メロス) (counter-agent nil) (location
家郷) (object nil) (instrument nil) (from nil) (to nil)) (event 変身する (1) (type action) (ID 34) (time (34 35)) (agent イワン)
(counter-agent nil) (location 家郷) (object nil) (instrument nil) (from nil) (to ■新たな姿形)) (event 罰する (1) (type action) (ID 35) (time
(35 36)) (agent 王様) (counter-agent 蛇) (location ■新たな姿形) (object nil) (instrument nil) (from nil) (to nil)) (event 罰する (1) (type
action) (ID 36) (time (36 37)) (agent 王様) (counter-agent メロス) (location ■新たな姿形) (object nil) (instrument nil) (from nil) (to nil))
(event 得る (1) (type action) (ID 37) (time (37 38)) (agent イワン) (counter-agent nil) (location ■新たな姿形) (object 花嫁)
(instrument nil) (from nil) (to nil)))
```

初めにユーザは物語長をパラメータとして与える. ここでは最上層の'ロシア民話を与えている

ユーザは役者に名前を与える

パラメータに応じた長さの事象概念が文として出力される

パラメータに応じた長さの事象概念が出力される