Propp's Morphology of the Folk Tale as a Grammar for Generation*

Pablo Gervás

Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
Madrid, Spain
pgervas@sip.ucm.es

Abstract

The semi-formal analysis of Russian folk tales carried out by Vladimir Propp has often been used as theoretical background for the automated generation of stories. Its rigour and its exhaustive description of the constituent elements of Russian folk tales, and the enumeration of the patterns they follow, have acted as inspiration for several story generation systems, both sequential and interactive. Yet most of these efforts have attempted to generalize Propp's account to types of stories beyond the corpus that it arose from. In the process, a number of the valuable intuitions present in the original work are lost. The present paper revisits Propp's morphology to build a system that generates instances of Russian folk tales. Propp's view of the folk tale as a rigid sequence of character functions is employed as a plot driver. Unification is used to incrementally build a conceptual representation of discourse by adding to an ongoing draft story actions that instantiate the character functions. Story actions are defined by pre and post conditions on the state of the plot to account for the causal relations crucial to narrative. The potential of the resulting system for providing a generic story generation system is discussed and possible lines of future work are discussed.

1998 ACM Subject Classification I.2.4 Knowledge Representation Formalisms and Methods

Keywords and phrases narrative generation, story grammar, unification

Digital Object Identifier 10.4230/OASIcs.CMN.2013.106

1 Introduction

At the start of the 20th century, Vladimir Propp identified a set of regularities in a subset of the corpus of Russian folk tales collected by Afanasiev. Propp set out to study a subset of the corpus already classified as fairy tales, and concentrated on 100 of those tales to carry out this study. Over these tales he carried a systematic analysis in terms of character functions, understood as acts of the character, defined from the point of view of its significance for the course of the action. The conclusions of his study where that, for the given set of tales, the number of such functions is limited, the sequence of functions was always identical, and all these fairy tales could be considered instances of a single structure. His book "Morphology of the Folk Tale" [19] describes this set of character functions, the sequence in which they appear, and the overall structure of this archetype of fairy tale. Propp's work was intended to provide insights for the description and classification of folk tales. It has in fact been used in this way by many researchers [26, 14, 17].

* This work was partially supported by the Ministerio de Educación y Ciencia (TIN2009-14659-C03-01).

Workshop on Computational Models of Narrative 2013.
Editors: Mark A. Finlayson, Bernhard Fisseni, Benedikt Löwe, and Jan Christoph Meister; pp. 106–122

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

However, the fact that it decomposes a tale into restricted set of elementary components, and outlines a procedure for putting them together to construct further tales has made it very appealing for researchers hoping to construct systems capable of generating stories automatically, both for sequential stories [24, 25, 8, 10, 26] and interactive ones [9, 7]. Yet most of these efforts have attempted to generalize Propp's account to types of stories beyond the corpus that it arose from, or combine it with additional techniques that had not been considered by Propp. A brief review of some of these efforts included in section 2.3 discusses the way Propp's work has been extended and adapted in some existing storytelling systems. In the process of these extensions and adaptations, a number of the valuable intuitions present in the original work are lost. As the field of computational narratology matures, it has become generally accepted that Propp's formalism can only be stretched so far, and that using it beyond its intended setting leads to severe limitations to story generation abilities of the resulting system. The story generation systems based on Propp's formalism have not managed to provide generic story telling capabilities. As a result both Propp's formalism and the goal of achieving generic story telling capabilities stand discredited.

In view of this situation, research in story generation has a chance to shift towards generation of quality stories for very specific domains [1]. It is in this last direction that Propp's work may find a different application niche. One of the reasons that made Propp's work so atractive to researchers in story generation is that Propp actually describes how his formalism might be used for the generation of tales. Seen in this light, Propp's formalism constitutes a blue-print for a story generation system intended to reproduce a particular model of story, while strongly adhering to specific genre and domain conventions. It is in this spirit that the present paper revisits Propp's morphology as a story generation procedure, exploring its potential for building a system that generates instances of Russian folk tales faithful to Propp's description. In this endeavour, a principle of economy is followed, considering the extension of the system with additional technologies only where their absence would clearly result in poorer stories. Additionally, an attempt has been made wherever possible to model Propp's formalism for this task in a declarative manner. This is with a view to hopefully replace in the future these declarative descriptions of Propp's formalism for Russian folk tales with a different set of descriptions, possibly capturing different types of story.

2 Previous Work

Before the proposed system can be described, a number of issues addressed by previous work must be presented: basic elements of Propp's morphology, Propp's description of how his morphology could be used to generate stories, Propp's influence on existing automated storytellers, and relevant insights from existing story generators even though not explicitly considered by Propp.

2.1 Elements of Propp's Formalism Relevant for Computational Implementation

The collection of tales that Propp focuses on involves stories built on combinations of a number of narrative ingredients: a protagonist sets out on a journey, usually triggered by a lack in his immediate environment or a villainy performed upon it, faces a villain, and in the process gets helped by a magical agent. A possible complication considered is the presence of a an additional character that competes with the protagonist for the role of hero of the story, which involves additional ingredients such as a gradual unveiling of the hero's real role

in the story, from initial presentation in disguise to the obtention of a reward towards the end, and usually involving recognition as a result of success on a difficult task.

The two corner stones of Propp's analysis of Russian folk tales are a set of roles for characters in the narrative (which he refers to as *dramatis personae*), and a set of character functions. These two concepts serve to articulate the morphology as an account of the elementary structure of the tales. Both of these concepts are constructed specifically for the family of tales begin considered. Therefore the set of roles includes fundamental elements such as the hero (who sets out on a journey), the dispatcher (who dispatches the hero on his journey), the villain (that the hero faces during the story), the donor (who provides the magical agent to the hero), the false hero (who competes with the protagonist for the role of hero of the story). The set of character functions includes a number of elements that account for the journey, a number of element that detail the involvement of the villain, including the villainy itself, some possible elaborations on the struggle between hero and villain, and a resolution, a number of elements that describe the dispatching of the hero, a number of elements that describe the acquisition of a magical agent by the hero, a number of elements concerned with the progressive unveiling of the hero's role in opposition to the false hero.

The sequence of character functions described by Propp is supposed to apply to all stories of the type described, so that any story will include character functions from this sequence appearing in the given order. With respect to the relative ordering, some deviation is allowed in that tales may depart from it by shifting certain character functions to other positions in the sequence.

Character function are sometimes repeated three times. This is a widely spread feature fo fairy tales, called *trebling*, where upon three instances of a particular event occur in sequence (stepmother tries to kill Snow White in three different ways but only the last one succeeds, Cinderella attends the Prince's ball under disguise on three consecutive nights but only on the last one does she forget to leave before midnight,...). For the purposes of the system described in this paper, this type of refinement may be left for consideration at a later stage.

Character functions in a given narrative are related to one another by long range dependencies related to motivation and correference.

Propp says:

> "The majority of character's acts in the middle of a tale are naturally motivated by the course of the action, and only villainy, as the first basic function of a tale, requires a supplementary motivation." [19, p. 75]

The concept of motivation that is referred to here concerns the network of causal relations between the different events of a story that a reader usually provides during comprehension [23]. This network representation determines the overall unity and coherence of the story. When considering the procedural generation of tales based on this model, motivation introduces a significant problem. The selection of what particular instantiation of a character function to use at a particular point of the tale must take into consideration that the new character function instance appear appropriately motivated by the preceding selections already made. This is a fundamental aspect for the success of the result as a story. As shown later, in order to account for this problem additional computational mechanisms need to be added.

Some character functions are implicitly linked to one another. Propp mentions two types of link between character functions:[1] elements which are always linked with varieties

[1] Propp's own abbreviations for specific types of his character functions are used to allow reference to the original work.

corresponding to one another (alternative instantiations of struggle and victory, such as H^1 – fight in an open field – always connected to I^1 – victory in an open field, or of villainy and its liquidation, such as A^{11} – enchantment – linked to K^8 – the breaking of a spell – ...); and elements that act as necessary preconditions to others (second element cannot happen unless the first one is already present) but allow for variation (the hero can only be rescued from pursuit if a pursuit has commenced, but rescue can take several forms regarless of how the pursuit started).

These links are mostly concerned with particular instantiations of certain character functions being linked to instantiations of character functions that went before them. This is one of the ways in which overall coherence of the tale can be ensured: characters kidnapped at the beginning are freed towards the end, and so on. A computational procedure must take these links into account when deciding which characters to assign to particular roles in each new character function added to a story. If the sister of the hero was bewitched at the start, it is she that needs to be released from the spell towards the end.

Character functions are so named because, in Propp's understanding, they represent a certain contribution to the development of the narrative by a given character. When he talks about the set of characters of the story (or *dramatis personae*), Propp constantly reccurs to a set of labels to describe particular roles played by characters in tales. They are gathered together in chapter VI where he discusses the distribution of functions among dramatis personae. For simplicity I will refer to these as *role names*, though Propp does not. Some examples of these roles are: the *villain*, the *donor* (who provides the hero with a magical agent), the *helper* (usually a magical agent, that helps the hero carry out his tasks), the *dispatcher* (who sends the hero on his mission), the *hero* (the protagonist of the story), and the *false hero* (who maliciously sets himself up to usurp the protagonist as hero of the story).

Propp defines these only in terms of the set of character functions that can be grouped around each one of them, as involving the same character. This set he refers to as the *sphere of action* for a particular role name. In the description of each character function in chapter III, Propp mentions how the character fulfilling a particular named role is involved in the various actions that can instantiate that character function (the villain carries out the villainy, the dispatcher sends the hero on his mission, the hero departs from home,...). If a procedural solution is sought that attempts to model closely the vision of tales that Propp had, these narrative roles must be explicitly defined, and some means of explicitly defining their participation in each type of character function should be provided, to ensure that these participations are instantiated by particular characters in a coherent manner throughout the tale.

Finally, Propp considers how more complex stories can be seen to conform to his morphology. To achieve this, the proposed sequence is considered as the skeleton for a single narrative thread, and complex stories may be composed of several such threads combined in different ways. In Propp's terminology, these narrative threads are called *moves* of a tale. Even though part of the same story, different moves may each involve different heroes or villains. The present paper is concerned with the construction of single move tales. The construction of multiple move tales may be addressed at a later stage both as a way of combining single moves and as a refinement of the construction procedure to produce individual moves suitable for combination with others.

2.2 Propp's Description of Tale Generation

Propp provides in his book a very clear description of how his morphology could be used for story generation:

> "In order to create a tale artificially, one may take any A, then one of the possible B's then a $C\uparrow$, followed by absolutely any D, then an E, the one of the possible F's, then any G, and so on. In doing this, any elements may be dropped, or repeated three times, or repeated in various forms. If one, then distributes functions according to the dramatis personae of the tale's supply of by following one's own taste, these schemes come alive and become tales. Of course, one must also keep motivations, connections, and other auxiliary elements in mind" [19, pp. 111–112]

In addition to this clearly procedural description he provides a number of constraints that a potential storyteller should obey and an emumeration of the points where a storyteller has freedom to decide.

The constraints on the story teller are:

1. "The storyteller is constrained [...] in the overall sequence of functions, the series of which develops according to the above indicated scheme." [19, p. 112]
2. "The storyteller is not at liberty to make substitutions for those elements whose varieties are connected by an absolute or relative dependence." [19, p. 112]
3. "In other instances, the storyteller is not free to select certain personages on the basis of their attributes in the event that a definite function is required." [19, p. 112]

The points where Propp considers that a storyteller has a certain freedom are:

1. "In the choice of those functions which he omits, or, conversely, which he uses" [19, p. 112]
2. "In the choice of the means (form) through which a function is realized." [19, p. 112]
3. in the assignment of story characters to particular slots in functions: "If one then distributes functions according to the dramatis personae of the tale's supply or by following one's own taste, these schemes come alive and become tales" [19, pp. 111–112] and "The storyteller is completely free in his choice of the nomenclature and attributes of the dramatis personae. Theoretically the freedom here is absolute." [19, pp. 112–113]
4. "The story teller is free in his choice of linguistic means." [19, p. 113]

On the third point, Propp follows on to discuss in rather vague terms that people do not make wide use of this freedom, preferring to let personages recur much as functions do. So there is a typical villain, a typical donor... Given the level of uncertainty involving this description, it has been decided not to consider it in the present system. The fourth point surely underlies Propp's decision not to address linguistic issues in his morphology at all. We follow this decision in deciding not to address the linguistic rendering of the tales in the initial implementation of our system.

The remaining insights are considered in a computational implementation in section 3.

2.3 Propp in Existing Automated Storytellers

Lang [13] developed the Joseph system to produce instances of stories akin to those in the Afanasiev corpus of Russian tales that Propp used as inspiration, but Lang departed from Propp's formalism in favour of a story grammar closer to Thorndyke's model [22], coupled with a complex network of logical procedures for modelling time and rational intention. Lang explicitly identifies some of the difficulties inherent in trying to formalize Propp's account with a view to generation [19, section 2.1.1]. The grammar used by Lang represents a story as a sequence of episodes, each one involving an initial event, a reaction by the protagonist to the event, and an outcome.

Turner [24] mentions Propp's work as an inspiration for his thesis, precisely for its potential as a story generator (as described in the introduction). But he also specifically claims that there were no traces of Propp's work in the final version of his story writing program. For his MINSTREL system, Turner claims to have been inspired by Propp's work, but then developed a system that combined case-based reasoning and planning to produce stories about King Arthur and his knights, without resorting to Propp's ideas at all.

Peinado [8] developed a description logic ontology for Propp's set of character functions, but then focused on exploring the potential of description logic ontologies for providing a knowledge intensive case-based solution for tale generation, based on reusing structure from existing tales into new ones. This ontology is a valuable resource that could have been used to implement further systems based on Propp's formalism. However, description logic ontologies have proven to be very good at representing the world as it is, and not so good at representing a world liable to change. Representation of change is a fundamental aspect of narrative. For this reason we have preferred to rely on a different representation mechanism for the effort reported in this paper.

Grasbon and Braun [9] and Fairclough and Cunnigham [7] adapted some of Propp's ideas to the realm of interactive story telling. They rely on character functions based on Propp's work to develop a story engine for interactive narrative. They extend the concepts of Propp with the idea of polymorphic functions, which can have different outcomes depending on user interaction.

The Proppian fairy tale Markup Language (PftML) [16] is an XML application developed by University of Pittsburgh's researchers based on Propp's work. PftML utilizes a Document Type Definition (DTD) to create a formal model of the structure of Russian magic tale narrative and to help standardize the tags throughout a corpus when analyzing it. PftML has been tested on a subset of the same Russian language corpus from which Propp drew has been used, as an empirical test of the conclusions of Propp's initial analysis against the original data. PftML constitutes a formal grammar for Propp's morphology of the folk tale. However, it is not well geared towards generation, and it misses many of the subtleties uncovered in section 2.1.

Fairclough and Cunningham [6] implement an interactive multiplayer story engine that operates over a way of describing stories based on Propp's work, and applies case-based planning and constraint satisfaction to control the characters and make them follow a coherent plot. They define a plot as a series of character functions and a series of complication-resolution event pairs, where a complication occurs whenever a character performs a function that alters the situation of the hero. A case based reasoning solution is used for storyline representation and adaptation. They use 80 cases extracted from 44 multi-move story scripts given by Propp. There are stories composed of one, two or more moves. A case is a move, seen as a story template, to be filled in by a constraint satisfaction system that chooses which characters perform the functions.

2.4 Relevant Insights of Existing Story Generators

There are a large number of story generators in existence, relying on a multiplicity of techniques. For the sake of brevity only those specific ingredients of some of them that are relevant for the solution proposed in this paper are listed here.

Although Propp never described his formalism as a grammar for stories, it has often been described as such [24]. The popularity of grammars as a representation mechanism peaked in the seventies and early eighties as a result of Noam Chomsky's work on formal grammars. A popular technique for modelling common phenomena was to develop a grammar for them.

This lead to the concept of a story grammar, pioneered by Rummelhart [21] and later taken up by Thorndyke [22] and many others. The very concept of story grammar was questioned by Black and Willensky [3], and a debate around story grammars went on for many years and lead to the discreditation of the concept as an actual model of human cognitive processing of stories. Nevertheless, story grammars remained a popular technique with researchers in story generation. The Joseph system [13] and the BRUTUS system [4] were based on story grammars. They both produced a succesful number of stories of high quality. In this sense, the concept of story grammar for the generation of stories can be considered validated as a sound and successful technology.

A different concept related with the implementation of narrative systems is that of story actions as operators that change the world. Actions in a story are applicable if certain conditions hold in the state of the world before they happen, and after they happen they change the state of the world. This idea has been represented by defining actions with an associated set of preconditions and another of postconditions or effects. This approach to defining actions is important because it constitutes a possible way of capturing the causal dependencies that constitute a fundamental ingredient of narrative as it is understood by people [23]. It has become popular in story generation through the numerous research efforts that use planning techniques [20, 2, 11], which are inherently based on this concept. Even systems based on alternative generation technologies include the possibility of associating pre and postconditions to actions, such as the information on emotional links between characters considered in the MEXICA system [18] or the preconditions added to the representation of actions in the Joseph system [13].

3 A Computational Solution for Proppian Story Generation

The first step for considering Propp's formalism as a computational procedure would be to define specific representations for the concepts involved. In the description of Propp's formalism given in section 2.1 we have relied on two different concepts that would need to be assigned a conceptual representation:

- *character function* (a label for a particular type of acts involving certain named roles for the characters in the story, defined from the point of view of their significance for the course of the action)
- possible instantiations of a character function in terms of specific *story actions*, involving a number of *predicates* describing events with the use of *variables* that represent the set of characters involved in the action

To fully capture Propp's restrictions (constraint 3), story actions will also include non-narrative predicates which encode constraints on the specific choice of dramatis persona that can fill particular argument slots in the predicates of the story action; for instance, the fact that the author of a villainy must be the villain.

The sequence of character functions chosen as backbone for a given story we will refer to as a *plot driver*.

The set of story actions available for instantiating a given character function, as defined by Propp, includes several variants concerning the form of the action. For instance, a villainy can take the form of kidnapping a person, seizing a magical agent, ruining the crops,... Each of these would be represented in our proposal by an action with a set of preconditions and a set of postconditions. To keep track of the effects of this actions as they are added to the story, some form of representation of the context must be employed. As the simplest possible solution, a representation of the context is considered as a set of states, each one

Table 1 Examples of story actions.

Character function	`villainy`	`liquidation`

Preconditions	`married H Y`
	`hero H`
	`villain X`
		`married H W`
		`sundered H W`
		`hero H`

Action	`makes_disappear X Y`	`resume_marriage H W`
Postconditions	`victim Y` `sundered H Y`	

representing the state of the world before a certain story action took place. A *state* of the world is represented as a set of predicates describing the facts that hold in that state. The sequence of states for a given story we call a *fabula*.

We represent a *story action* as a set of predicates that describe an instance of a character function. Links with preceding story actions are represented as dependencies of the story action with predicates that need to have appeared in previous story actions (preconditions). Therefore a story action involves a set of preconditions (predicates that must be present in the context for continuity to exist), and a set of postconditions (predicates that will be used to extend the context if the action is added to it). Some additional predicates not corresponding to events in the story are added to encode the sphere of action to which each story action belongs. These predicates explicitly link the corresponding narrative role to a particular variable in the story action The predicates in a story action are defined over free variables as arguments. This ensures that relative instantiation of the various arguments in the predicates of a story action is coherent, as discussed later. Table 1 includes example of story actions linked by preconditions.

Each successive state in a fabula contains all the predicates arising from the preceding actions that have not been retracted by a story action since they occurred. This is difficult to read. Also it is difficult to define over such a structure significant metrics on measures such as number of predicates in which a certain character appears (which we will need to consider when measuring the structural quality of a story). For this purpose we define a final structure called a *flow* for a story, which is simply an ordered sequence of all the predicates in the fabula, such that each one appears only once, and grouped into subsets according to the particular state of the fabula in which they were first introduced.

Based on this representation, the procedure originally sketched by Propp can be subdivided into the following stages, each one of which will be addressed by a different module in our proposed system:

- employ an algorithmic procedure for generating a sequence of character functions considered valid for a tale (*plot driver generator*)
- given a valid sequence of character functions, progressively select instantiations of these character functions in terms of story actions (*fabula generator*)
- given a fabula where all variables have been replaced by constants, produce a flow for the story (*flow generator*)

For each of these stages a computational decision procedure must be selected. We are considering a possible computational implementation. For this purpose we intend to consider in the first instance the simplest representation and the simplest procedures compatible with

acceptable results. To this end, a number of computational options for some of these modules have been considered, together with a knowledge engineering effort to produce the required resources. The results have been empirically tested for fulfillment of Propp's constraints. The following section report on the development, the evaluation procedures, and the results of the tests.

3.1 The Implementation

The computational solution described in this paper has been partially implemented as a working prototype written in Java and operating over a small set of resources defined as plain text files. This development involved a very small effort of simple coding of the overall algorithmic procedures, but a considerable effort of knowledge engineering over the set of resources.

3.1.1 Plot Driver Generators

Three possible implementations have been considered for plot drivers:

- a baseline plot drivers that randomly selects character functions, not necessarily in sequence, up to a randomly decided number,
- a plot driver that follows a canonical sequence of character functions, deciding at each stage whether to add it to the plot driver or not,[2] or
- a grammar based plot driver which generates based on a gramar automatically extracted from a subset of the schemes analyzed by Propp in Appendix III.

In the second case, a reference sequence of character functions is constructed following the matrix employed by Propp in Appendix III for tabulating his analyses of stories from his corpus. This sequence includes several possible placements of certain character functions in the sequence, to capture the accepted possibilities for inversion.

The grammar for the third case is built automatically from a set of Propp's schemes annotated as a grammar by considering subgroupings of character functions that recurred frequently, and using Propp's own terminology for different segments of a tale. An example of grammar is given in Table 2.

Performance results for the three plot driver generators considered are given in section 3.3.

3.1.2 Fabula Generators

A fabula generator receives a plot driver and selects story actions for the character functions given in it. To do this, the fabula generator has to define a fabula, a sequence of states that contain a chain of instances of character functions ideally somehow linked by having their preconditions fulfilled by the context. The initial state by default incorporates all predicates of the first action, and each valid action added to the fabula generates a new state that incorporates all predicates of the previous state, plus the predicates of the new action.

A mapping is established between the set of story actions and the set of character functions, so that each of the available story actions is considered a possible instantiation of a given character function.

---

[2] With the exception of the villainy/lack character functions, for one of the two is always added to a story to ensure story interest. This follows Propp's own suggestion [19, p. 102].

Table 2 A Simple Grammar.

```
FOLKTALE            =  COMPLICATION DONOR COURSE_OF_ACTION CLOSURE
FOLKTALE            =  COMPLICATION DONOR COURSE_OF_ACTION
FOLKTALE            =  COMPLICATION COURSE_OF_ACTION CLOSURE
FOLKTALE            =  COMPLICATION COURSE_OF_ACTION
FOLKTALE            =  ↑ DONOR T COMPLICATION DONOR COURSE_OF_ACTION CLOSURE

COMPLICATION        =  TRIGGER
COMPLICATION        =  TRIGGER MEDIATION ↑
COMPLICATION        =  TRIGGER MEDIATION
COMPLICATION        =  TRIGGER M MEDIATION

TRIGGER             =  A
TRIGGER             =  a

MEDIATION           =  C
MEDIATION           =  B C

DONOR               =  F
DONOR               =  F G
DONOR               =  D E
DONOR               =  D E F
DONOR               =  D E F G
DONOR               =  D E G F

COURSE_OF_ACTION    =  TASK
COURSE_OF_ACTION    =  ↓ PURSUIT
COURSE_OF_ACTION    =  K ↓ PURSUIT
COURSE_OF_ACTION    =  ↓ PURSUIT o
COURSE_OF_ACTION    =  o DONOR K T PURSUIT
COURSE_OF_ACTION    =  CONFRONTATION K ↓
COURSE_OF_ACTION    =  CONFRONTATION K ↓ PURSUIT
COURSE_OF_ACTION    =  K ↓
COURSE_OF_ACTION    =  CONFRONTATION
COURSE_OF_ACTION    =  CONFRONTATION K

PURSUIT             =  Pr Rs
PURSUIT             =  Pr DONOR Rs

CONFRONTATION       =  H I
CONFRONTATION       =  I

TASK                =  M N

CLOSURE             =  W
CLOSURE             =  Q W
CLOSURE             =  Q Ex U W
CLOSURE             =  T W
CLOSURE             =  X U W ↓ X
```

To evaluate whether the preconditions of a story action are satisfied by the context, they are unified with the set of predicates that hold in that state. This serves two purposes:

- if the preconditions are not satisfied, an alternative story action will be considered
- unification allows any of the free variables in these preconditions to unify with those in the predicates holding in the fabula

A story action is considered a valid extension of a given fabula if the set of its preconditions can be successfully unified with the predicates in the latest state of the fabula. Once the story action is added, the next state is built by extending the preceding state with the action and the postconditions of the story action.

When the preconditions unify with the state in the fabula, any replacement of free variables in the preconditions is carried over to the rest of the story action before it is added to the context. This ensures that the story action become coherent with the rest of the predicates in the fabula, creating continuity.

This enables the system to model long range dependencies between character functions. If the choice for a character function such as liquidation of misfortune or lack depends on which particular story action was chosen to instantiate the character function for lack, this procedure will both block non appropriate instantiations for liquidation (as their preconditions will not be satisfied) and will ensure the appropriate assignment of variable names to ensure coherence (for instance, that the person that was kidnapped at the beginning be freed towards the end). The additional predicates encoding the sphere of action to which each story action belongs enforce a correct distribution of functions over dramatis personae. Overall, the use of unification models Propp's constraints 2 and 3.

However, a requirement of strict unification narrows down the set of options for extending to a very small set of story actions. given the current size of the set of available story action (described in section 3.2), it was considered advisable to allow a certain relaxation of this constraint. This corresponds to an operation of accommodation [15], in which if some of the preconditions unify and some do not, those that do not can be added to the context together with the action and its postconditions. Given a story action and a fabula, the less preconditions that need to be accommodated, the more appropriate the story action is considered as an extension of the fabula.

Two fabula generators have been considered:

- a purely random fabula generator that, for a character function given by a plot driver, picks at random one of the possible story actions available for that character function (considered as baseline)
- a fabula generator that adds the story action that best unifies with the context (allowing for accommodation)

Performance results for these fabula generators considered are given in section 3.3.

3.2 The Knowledge Engineering Effort

Although the grammar proved easy to write, developing and debugging the set of story actions required for the 31 character functions proved to be an onerous task.

Propp never exhaustively described his set of instances of character functions nor his set of schemes for the structure of tales, and it is difficult to match the numbering of the tales that he uses as reference with existing compilations of translated (into English) tales from the Afanasiev corpus. There is therefore no obvious source for constructing a set of resources

to match Propp's descriptions.[3] In order to develop a set of story actions for our system, we followed a detailed procedure to guarantee close conformance to Propp's specification and reasonable coverage of his set of character functions.

A set of story actions conforming to the described representation was built following Propp's descriptions of character functions (Chapter III) and matching the set of abbreviations proposed for the analysis formalism (Appendix IV); ensuring that dependencies indicated by Propp are represented as preconditions

We then tested that this set of story actions could be used to construct fabulas matching both the structure and the specific choice of instances of each character function given in the set of schemes listed by Propp in his book. The resulting fabulas showed acceptable continuity (or impression thereof under some possible interpretation). Success in this respect was taken to constitute a certain degree of validation of both the set of story actions engineered (in as much as they provide enough links to guarantee continuity in terms of co-occcurrence of variables across the whole set of predicates for the fabula) and the unification procedure employed (validation procedure for extensions enforces continuity).

Finally we tested that this set of story actions could be used to construct fabulas for the original schemes described by Propp in Appendix III but allowing freedom of choice with respect to the specific choice of story action. Success in this respect was taken to constitute a certain degree of validation of both the set of story actions engineered (enough articulation to provide variation over a set of possible instantiation choices for each character function) and the unification procedure employed (validation procedure for extensions still enforces continuity even when free choice of instance is allowed).

A final set of 280 story actions was obtained corresponding to the restricted set of 24 story actions that Propp used to describe the schemes given in Appendix III (Propp explains that functions of the preparatory section were not included for lack of space).

3.3 Evaluation

Given that the development effort has focused at a very abstract level of representation, evaluation has to be considered at a corresponding level to provide valid feedback for the improvement of the system. As the linguistic modelling of the stories has not been addressed, evaluation by human volunteers is plagued with difficulty. Introducing some kind of rapidly constructed stage for rendering the results as text by providing text templates for each story action (as done in some existing story generators [18]) is likely to introduce noise in terms of elements present in the text and not necessarily produced by the system. Asking human evaluators to rate the quality of an abstract representation as produced by the system runs the risk of judgements being clouded by the difficulty of interpreting the representation.

Additionally, evaluations by humans necessarily have to be restricted to a small number of instances of system output. The choice of which particular instances to test is left to the designer of the experiment, and there is a risk of focusing on examples that are not representative of system performance overall.

As an alternative, quantitative procedures have been defined to measure the specific qualities desired for each stage of the representation, at a corresponding abstract level. These procedures can be applied to a large number of system results, providing a measure of the

[3] This is probably one of the reasons why no one has yet attempted to build a story generator for Russian fairy tales. Given the need to engineer the set of knowledge resources from scratch, most researchers have preferred to start from alternative material that was either more easily available or closer to their native intuitions of folklore or narrative.

Table 3 Results for plot drivers.

Random	Sequence	Grammar
59.2	100	83.48

quality of system output at the working level of abstraction and applicable to a broad range of system results, leaving no doubt as to their significance over the complete set of outputs.

3.3.1 Plot Driver Generators

Plot driver generators must obey constraint 1, as described in section 2.2. To establish the extent to which the various implementations fulfill this constraint, a measure of conformance to a reference sequence has been defined. The key measure to consider is, given a certain character function appearing in a candidate plot driver, how many of the functions preceding/following it in the plot driver are contained in the part of the reference sequence that goes before/after (the best scoring of) its appearances in the reference sequence. This value is normalised as a percentage over the length of the plot driver. This measure is 100 if all character functions before and after the one considered have the same relative order in the reference sequence. The measure for a complete driver is taken as the average value for all its functions. This is 100 if the plot driver satisfies perfectly the order in the reference sequence and degrades towards 0 if some of its character functions appear out of place with respect to the given sequence.

Results for the three different plot driver generators that have been tried are reported in Table 3. Each of the alternative implementations was run 100 times and values were averaged over the results.

These results confirm as expected that the random approach results in plot drivers that do not conform to Propp's requirement, that strict adherence to Propp's instructions results in perfect conformance. The results for the grammar approach are more surprising. The grammar obtained from the set of schemes described by Propp performs considerably better than the random baseline, but almost 20 points below the version following the sequence strictly. This is due to the fact that the set of schemes contains several exceptions to the general rules. As the grammar is extrapolated from actual tales, this result suggests that allowing a certain flexibility with respect to Propp's rules is likely to produce less rigid stories that resemble more closely the kind that might be produced by humans. However, testing this hypothesis is beyond the scope of the present paper.

3.3.2 Fabula Generators

Fabula generators must obey constraints 2 and 3, and the links between instances of character functions described in section 2.1.

In order to evaluate the extent to which the different construction algorithms fulfill these requirements, a number of metrics have been defined over the flow for a story, which is the closest representation produced by our system of the final shape of the story. Consideration of the referential chains in the flow (the set of predicates in which each character occurs) is significant of the degree of continuity achieved. These measures include: the maximum length of referential chain (%MLRC), the minimum length of referential chain (%mLRC), and the average length of referential chain (%avLRC). All of these measures are normalised

Table 4 Results for fabula generators.

	FL	%MLRC	%mLRC	%avLRC
PR	35.52	11.98	2.6	4.39
UA	37.38	47.58	2.57	7.64

Table 5 Example story.

hero 296	lack 74 75	test 284 296	disguised 296
captive 295	*	donor 284	artisan 613
asks 295 296 297	dispatches 261 296	*	apprentice 296 613
*	seeker_hero 296	finds 296 453	unrecognised 296
not_perform_service 296	banished 261	follows 296 453	*
negative_result 296	victimised_hero 261	at_target_location 296	false_hero 616
*	transported_to 261 296	*	claims 616 617
villain 73	*	arrives 296 612	unfounded 617
maims 73 74	sets_out 296	location 612	*
victim 74	*	home 612	returns 296

over the length of the flow (FL) and expressed as a percentage, to allow comparison across tales of different length.

The fabula generators are tested over plot drivers produced by generators that follow Propp's sequence strictly. Results for their evaluation are given in Table 4. Each of the alternative implementations was run 100 times and values were averaged over the results.

These results confirm that the use of unification and accommodation as techniques for ensuring continuity over stories do result in longer referential chains, and in stories that more clearly involve specific characters over a period of time.

4 Discussion

An example of a story is given in Table 5. The example appears broken down over four columns, separated by * into groups corresponding to single story actions. The arguments of predicates are represented by numbers, each corresponding to a character, location or object in the story. The story concerns character 296, who behaves badly at the start of the story, is banished, is tested by a donor, finds a trail that leads him home, arrives disguised as an apprentice to an artisan, suffers an impostor and returns. The example was picked out from the combined results of a plot driver relying on Propp's sequence and a fabula generator relying on unification with accommodation.

The story evidences some of the problems with literal implementation of Propp's algorithm: long range dependencies are captured by the unification mechanism when the two character functions involved appear in the plot driver (hero sets out and returns), but the mechanism for generating the driver does not take them into account. As a result, for instance, the villain and the false hero go unpunished in this case. This suggests that some computational means of taking the long range dependencies into account must be included to improve the performance of the system. The grammar approach has a potential for representing implicitly these long range dependencies, though empirical testing of this hypothesis is again beyond the scope of this paper.

With respect to the usefulness of Propp's various concepts, some of the roles described for dramatis personae seemed less relevant to the actual set of character functions than others (*princess*, for instance, was much more loosely linked to specific functions than *hero* or *villain*). Equivalent roles not mentioned by Propp, such as the victim of the villainy, were more useful in establishing continuity.

Preliminary results indicate that the overall quality of the generated stories is highly dependent on the quality of the set of story actions. This evidence confirms the limitations of knowledge-based computational solutions, but it also highlights a potential for generalization for the approach followed in the paper. Although the current initiative strove to build a system as faithful as possible to Propp's formalism, all references to Propp's material in the resulting implementation occur only within the set of plain text files that constitute the knowledge resources. The choice and names of the character functions, and the restrictions imposed on how they may combine are captured in the grammar. The set of story actions and the relationships between stated in terms of preconditions are also written in a separate text file. The actual Java code that exploits these resources is independent of Propp's particular solution for Russian folk tales. As a result, it would be possible to write an alternative set of resources to use a completely different grammar, over elements of a similar nature but which need no longer be called character functions, and which combine according to different rules, or which get instantiated with a completely different set of story actions, and assigned to different characters.

Because of this property, the approach presented in the paper has the potential for being ported to different domains (for instance, to science fiction stories by changing the set of story actions and the set of characters) or adapted to account for different structural analyses of narrative (by changing the grammar into one that covers, for instance, Campbell's account of the hero's journey [5] or Lakoff's account of the structure of fairy tales [12]).

5 Conclusions

The theoretical account of Russian fairy tales provided by Vladimir Propp has been revisited as potential source for a procedure for story generation. Although many research efforts in story generation have considered Propp as an inspirational source, none of them has explored the actual procedures explicitly described by Propp. By considering the simplest possible implementation of these procedures, a framework for story generation has been developed that takes full advantage of the intuitions behind Propp's account but which is built in a modular and declarative manner so that particular details arising from Russian folk tales can later be replaced with material from alternative knowledge sources.

The approach suffers from the limitations inherent to any knowledge intensive approach, in the form of a heavy knowledge engineering effort required to kick start the necessary set of resources. Preliminary results show strong coupling between the quality of these resources and the quality of the resulting stories. This can be seen as a weakness in terms of a deep adaptation curve for any new domain or new application, but also as a significant strength, in terms of the possibility of extending it to other domains and of targeted refinement until a desired level of quality is reached.

The various refinements and possible extensions described through the paper will be considered as future work.

References

1 Nicholas D. Allen, John R. Templon, Patrick Summerhays McNally, Larry Birnbaum, and Kristian Hammond. StatsMonkey: A data-driven sports narrative writer. In Mark Alan Finlayson, editor, *Computational Models of Narrative: Papers from the 2010 AAAI Fall Symposium*, number FS-10-04 in AAAI Technical Reports, pages 2–3. AAAI Press, 2010.

2 Byung-Chul Bae and R. Michael Young. A use of flashback and foreshadowing for surprise arousal in narrative using a plan-based approach. In Ulrike Spierling and Nicolas Szilas, editors, *Interactive Storytelling, First Joint International Conference on Interactive Digital Storytelling, ICIDS 2008, Erfurt, Germany, November 26-29, 2008, Proceedings*, number 5334 in Lecture Notes in Computer Science, pages 156–167. Springer, 2008.

3 John B. Black and Robert Wilensky. An evaluation of story grammars. *Cognitive Science*, 3:213–230, 1979.

4 Selmer Bringsjord and David A. Ferrucci. *Artificial Intelligence and Literary Creativity: Inside the Mind of BRUTUS, a Storytelling Machine*. Erlbaum, 1999.

5 Joseph Campbell. *The Hero with a Thousand Faces*. Princeton University Press, Princeton, second edition, 1968.

6 Chris Fairclough and Pádraig Cunningham. A multiplayer case based story engine. In Quasim H. Mehdi, Norman E. Gough, and Stéphane Natkin, editors, *4th International Conference on Intelligent Games and Simulation (GAME-ON 2003), 19–21 November 2003, London, UK*, pages 41–46, London, 2003. EUROSIS.

7 Chris Fairclough and Pádraig Cunningham. A multiplayer O.P.I.A.T.E. *International Journal of Intelligent Games & Simulation*, 3(2):54–61, 2004.

8 Pablo Gervás, Belén Díaz-Agudo, Federico Peinado, and Raquel Hervás. Story plot generation based on CBR. *Knowledge-Based Systems*, 18:235–242, 2005.

9 Dieter Grabson and Norbert Braun. A morphological approach to interactive storytelling. In Monika Fleischmann and Wolfgang Strauss, editors, *CAST 2001. Living in Mixed Realities: Conference on Artistic, Cultural and Scientific Aspects of Experimental Media Spaces, September 21–22, 2001, Schloss Birlinghoven, Sankt Augustin*, netzspannung.org event, pages 337–340, 2001.

10 Shohei Imabuchi and Takashi Ogata. Story generation system based on propp theory as a mechanism in narrative generation system. In Masanori Sugimoto, Vincent Aleven, Yam San Chee Chee, and Baltasar Fernández-Manjón, editors, *2012 IEEE Fourth International Conference On Digital Game And Intelligent Toy Enhanced Learning, DIGITEL 2012, Takamatsu, Japan, March 27-30, 2012*, pages 165–167, Los Alamitos, CA, USA, 2012. IEEE Computer Society.

11 Arnav Jhale and R. Michael Young. Cinematic visual discourse: Representation, generation, and evaluation. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(2):69–81, 2010.

12 George P. Lakoff. Structural complexity in fairy tales. *The Study of Man*, 1:128–150, 1972.

13 R. Raymond Lang. *A Formal Model for Simple Narratives*. PhD thesis, Tulane University, 1997.

14 Piroska Lendvai, Thierry Declerck, Sándor Darányi, Pablo Gervás, Raquel Hervás, Scott A. Malec, and Federico Peinado. Integration of linguistic markup into semantic models of folk narratives: The fairy tale use case. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pages 1996–2001, Paris, 2010. European Language Resources Association.

15 David Lewis. Scorekeeping in a language game. *Journal of Philosophical Logic*, 8:339–359, 1979.

16 Scott A. Malec. Proppian structural analysis and XML modelling. Presented at *Computers, Literature and Philology (CLiP 2001)*, 2001.

17 Scott A. Malec. AutoPropp: Toward the automatic markup, classification, and annotation of Russian magic tales. In Sándor Darányi and Piroska Lendvai, editors, *Proceedings of the First International AMICUS Workshop on Automated Motif Discovery in Cultural Heritage and Scientific Communication Texts*, pages 112–115, Szeged, 2010. University of Szeged, Faculty of Arts, Department of Library and Human Information Science.

18 Rafael Pérez y Pérez. *MEXICA: A Computer Model of Creativity in Writing.* PhD thesis, University of Sussex, 1999.

19 Vladimir Yakovlevich Propp. *Morphology of the Folktale.* University of Texas Press, Austin, TX, 2nd edition, 1968. transl. L. Scott.

20 Mark Owen Riedl and R. Michael Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268, 2010.

21 David E. Rumelhart. Notes on a schema for stories. In Daniel G. Bobrow and Allan Collins, editors, *Representation and Understanding: Studies in Cognitive Science*, pages 211–236, New York, 1975. Academic Press, Inc.

22 Perry W. Thorndyke. Cognitive structures in comprehension and memory of narrative discourse. *Cognitive Psychology*, 9:77–110, 1977.

23 Tom Trabasso, Paul van den Broek, and So Young Suh. Logical necessity and transitivity of causal relations in stories. *Discourse Processes*, 12(1):1–25, 1989.

24 Scott R. Turner. *MINSTREL: a computer model of creativity and storytelling.* PhD thesis, University of California at Los Angeles, Los Angeles, CA, USA, 1993.

25 Cati Vaucelle. Générateur d'histoires. Master's thesis, Université Paris VIII, 2000.

26 Takenori Wama and Ryohei Nakatsu. Analysis and generation of Japanese folktales based on Vladimir Propp's methodology. In Paolo Ciancarini, Ryohei Nakatsu, Matthias Rauterberg, and Marco Roccetti, editors, *New Frontiers for Entertainment Computing*, number 279 in IFIP International Federation for Information Processing, pages 129–137. Springer, 2008.