Reviewing Propp's Story Generation Procedure in the Light of Computational Creativity

Pablo Gervás [1]

Abstract. Vladimir Propp's analysis of Russian folk tales is known to have produced a semi-formal description of the structure of these tales that has acted as inspiration for several story generation systems, both sequential and interactive. Its exhaustive description of the constituent elements of tales of this kind, and the enumeration of the patterns they follow provided a very useful starting point for researchers looking for computational implementations of story generators. However, it is less generally known that in his book Propp also proposed a procedure for the generation of new tales based on his analytical framework. Although this generative procedure is much less formal than its analytical counterpart, it is one of the first existing instances of a creative process described procedurally. Of particular interest for the field of creative story generation is the number of issues that are declared relevant but not explored in detail. The present paper revisits Propp's description and focuses on the task of generating the sequence of character functions that determine the plot of the tale. For this task, a number of possible computational implementations are explored, in search for those that produce better results in terms of a number of simple evaluation metrics inspired by Propp's formalism.

1 Introduction

Vladimir Propp's "Morphology of the Folk Tale" [11] has been a preferred model used by computer scientists trying to model human story telling ability, mainly because it it decomposes a tale into a restricted set of elementary components, and outlines a procedure for putting them together to construct further tales.

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*↑, 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" p. 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.

[1] Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid, Spain, email: pgervas@ucm.es

Out of the three constraint on the story teller mentioned by Propp, two are particularly relevant to the role of dependencies in establishing a valid sequence of character functions:

1. "The storyteller is constrained (...) in the overall sequence of functions, the series of which develops according to the above indicated scheme." 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." 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" p. 112
2. "In the choice of the means (form) through which a function is realized." p. 112
3. in the assignment of story characters to particular slots in functions p. 112-113
4. "The story teller is free in his choice of linguistic means." p. 113

The second, third and fourth points are beyond of the scope of the present paper. The first point is considered in a computational implementation in section 3.

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 influence on existing automated storytellers, some basic points on computational creativity, and two outstanding problems with Propp's procedure: the existence of long range dependencies between character functions and the appropriate ending of stories.

2.1 Elements of Propp's Formalism Relevant for Computational Implementation

Vladimir Propp [11] set out to study a subset of a corpus of Russian folk tales collected by Afanasiev, and concentrated on 100 of those tales to carry out this study. Over these tales he identified a set of regularities in terms of character functions, understood as acts of the character, defined from the point of view of their significance for the course of the action. He concluded that, for the given set of tales, the number of such functions is limited, the sequence of functions was always identical, and all these folk tales could be considered instances of a single structure, an archetype of a folk tale.

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 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 achievement of a reward towards the end, and usually involving recognition as a result of success on a difficult task.

The two cornerstones 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 being 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). 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. In this paper we will be relying on this abstraction to focus on plot centred construction, giving the characters themselves a secondary role. This is not intended as a statement on the relative importance of these elements but as simplification to allow a partial improvement of one aspect while postponing treatment of the other. 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 of the villainy, 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.

2.2 Propp in Existing Automated Storytellers

There have indeed been several attempts to use Propp's formalism as a basis for story generation. However, with only one exception [6], most of these attempts either were loosely inspired by Propp (Lang's Joseph system [8], Turner's MINSTREL system [14]) or relied on the part of Propp's framework designed for analyzing / describing folk tales, which they used to specify the building blocks for their systems but then combined with additional constructive techniques that had not been considered by Propp (such as case-based reasoning or interactive storytelling [5, 7, 4, 3]. Gervás [6] provides more detailed argumentation of how these various storytelling systems differ from Propp's description of the generative procedure he proposed based on his analytical framework.

The system developed by Gervás [6] does address a computational implementation of a story generator that more closely ressembles Propp's description of how his morphology might be used to generate stories.

It relies on the following specific representations for the concepts involved:

- a *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
- a sequence of character functions chosen as backbone for a given story

To fulfill Propp's description of the morphology of a folk tale, the sequence of character functions that acts as backbone for a story has to be a subset of the character functions listed by Propp, appearing in a relative order that conforms with a given canonical sequence. Propp refers to this canonical sequence but mentions several possible ways of constructing it. The sequence of occurrences of Propp's characer functions taken as canonical sequence for this paper 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 actual set of character functions employed as canonical sequence is given in Table 1.

test by donor	difficult task
hero reaction	branding
acquisition magical agent	victory
villainy / lack	task resolved
hero dispatched	trigger resolved
begin counteraction	return
acquisition magical agent	hero pursued
departure	rescue from pursuit
test by donor	unrecognised arrival
hero reaction	unfounded claims
acquisition magical agent	false hero exposed
transfer	transfiguration
trigger resolved	branding
unrecognised arrival	villain punished
unfounded claims	hero marries
struggle	

Table 1. Set of character functions employed as canonical sequence.

Character functions are presented in two columns by their abbreviated name. A key point in the canonical sequence is the **villainy / lack** pair of character functions written in bold. These differ from all the others in that only one of them is ever included in any single story, and all stories must contain either one or the other.

Over these representations, an evaluation metric was developed to check for fulfillment of Propp's constraints. This metric involved a measure of conformance to a reference sequence of character functions (given a certain character function appearing in a candidate sequence of character functions, how many of the functions preceding/following it in the sequence are contained in the part of the reference sequence that goes before/after (the best scoring of) its appearances in the reference sequence; normalised as a percentage over the length of the sequence of character functions.

Based on this representation, the procedure originally sketched by Propp was subdivided into a number of stages, each one of which will be addressed by a different module in the folk tale generation system. Of these, the one concerned with producing the sequence of character functions employs an algorithmic procedure for generating a sequence of character functions considered valid for a tale. For this stage, Gervás considered a a number of computational options which were empirically tested for fulfillment of Propp's constraints as captured by the described metrics:

- a baseline character function sequence generator that randomly selects character functions, not necessarily in sequence, up to a randomly decided number
- a character function sequence generator generator that follows a canonical sequence of character functions, deciding at each stage whether to add an instance of the next character function in the canonical sequence to the sequence under construction or not[2]

2.3 Computational Creativity

Wiggins [15] takes up Boden's idea of creativity as search over conceptual spaces [1] and presents a more detailed theoretical framework intended to allow detailed comparison, and hence better understanding, of systems which exhibit behaviour which would be called creative in humans. This framework describes an exploratory creative system in terms of a septuple of elements, which include elements for defining a conceptual space as a distinct subset of the universe of possible objects, the rules that define a particular subset of that universe as a conceptual space, the rules for traversing that conceptual space, and an evaluation function for attributing value to particular points of the conceptual space reached in this manner. Wiggins goes on to provide refinements that cover issues such as the differences between exploratory and transformational creativity expressed in terms of this framework. However, the systems considered in this paper are far too humble to require these refinements.

Ritchie [12] addresses another important issue in the development of creative programs, that of evaluating when a program can be considered creative. He does this by outlining a set of empirical criteria to measure the creativity of the program in terms of its output. He makes it very clear that he is restricting his analysis to the questions of what factors are to be observed, and how these might relate to creativity, specifically stating that he does not intend to build a model of creativity. Ritchie's criteria are defined in terms of two observable properties of the results produced by the program: novelty (to what extent is the produced item dissimilar to existing examples of that genre) and quality (to what extent is the produced item a high-quality example of that genre). To measure these aspects, two rating schemes are introduced, which rate the typicality of a given item (item is typical) and its quality (item is good). Another important issue that affects the assessment of creativity in creative programs is the concept of an inspiring set, the set of (usually highly valued) artifacts that the programmer is guided by when designing a creative program. Ritchie's criteria are phrased in terms of: what proportion of the results rate well according to each rating scheme, ratios between various subsets of the result (defined in terms of their ratings), and whether the elements in these sets were already present or not in the inspiring set.

For the analysis of complex creative acts in terms of their constituents elements, a recent theoretical proposal for understanding computational creativity software will be useful. The FACE model [2, 9] presents a framework to understand creative acts performed by software. It defines a creative act as a non-empty tuple containing exactly zero or one instances of eight types of individual generative acts. The eight types are defined in terms of four different target types: the expression of a concept, a concept, an aesthetic measure, or framing information. A *concept* is a procedure which is capable of taking input and producing output, the *expression of a concept* is an instance of an (input, output) pair produced when a concept is run. An *aesthetic measure* is a function which takes as input a concept or an expression and outputs a numerical score. *Framing information* is a comprehensible explanation (in natural language) of some aspect of the tuple. The eight types arise by distinguishing, for these four target types, between artefacts (generating instances of them) and processes (generating methods for producing instances).

2.4 The Problem of Long Range Dependencies

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." 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 [13]. 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:[3] elements which are always linked with varieties 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 regardless 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.

Gervás [6] uses unification between successive story actions during construction of a tale hoping to model Propp's constraint 2 on dependencies between instances of character functions within the same

[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 in page 102.

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

tale as described in section 1. However, this only worked to a certain extent. The use of unification / accommodation enables the system to partially 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 problems that arise are apparent in the example of story output given in Gervás [6]. In this example, as a result of the unification procedure employed, the hero that departs is associated with the hero that returns, and also with the character that performs all the tasks associated with the hero in between those two events. This does indeed capture some of the dependencies implicit in Propp's analysis. However, in the same story, a villain maims a certain victim, but the story neither resolves this villainy nor punishes the villain. Yet this is surely the kind of dependency between character functions that Propp is considering when he mentions "Of course, one must also keep motivations, connections, and other auxiliary elements in mind" p. 111-112.

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 such systems.

2.5 The Problem of Endings

A different point to consider is whether a sequence of functions generated in this way allows for a story with a satisfactory ending. This important point was not considered in detail by Propp, possibly due to the fact that his main goal was to propose an analytical framework to help classify folk tales. The proposal of a related generative procedure was a side product, and Propp never considered the problem of when to end a story. From a computational point of view, however, the need for a clear stopping condition on the construction procedure is paramount.

The most relevant mentions of endings in Propp's book occur in pages 58 ("A great many tales end on the note of rescue from pursuit.") and 64 (on the subject of the reward / marriage character function: "At this point the tale draws to a close.").

3 Missing Pieces in Propp's Generative Procedure

Existing efforts (such as Gervás [6]) have demonstrated the potential of Propp's generative procedure for story construction as a blueprint for a computation solution to story generation. But they have also uncovered a number of points where the description of the procedure and/or the required operations is vague. The main computational contribution of this paper involves a closer study of two important points for computational story construction: the existence and management of dependencies between character functions and the need for a stopping condition to determine when a satisfactory sequence of character functions has been obtained. As explained above, Propp mentions dependencies in various ways, but he does not go into detail of how they may be treated during generation. On the subject of endings, he says very little. His proposed procedure does implicitly contain a solution: as it involves following the canonical sequence, deciding for each character function whether to include it or not, the procedure ends when the end of the sequence is reached. This will be our baseline solution, but we also want to consider if more informed solutions might perform better with respect to the potential of the resulting sequence to support a satisfactory ending.

Following Gervás [6] the present paper will focus on a representation of stories at a conceptual level, as sequences of character functions, and omit issues concerning the more detailed elements of Propp's analysis.

3.1 Long Range Dependencies between Character Functions

As described in section 2.4, the existence of long range dependencies between character functions is noted by Propp, but not actually addressed in terms of how it should affect the computational procedure described for story generation. To establish that, we need to identify these long range dependencies, how their satisfaction might be evaluated over sequences of character functions, and how they might be taken into consideration during construction of these sequences.

3.1.1 Identifying Long Range Dependencies

Dependencies can be of two types. Some dependencies are such that a sequence is only acceptable if both character functions involved are present (for instance, if the hero is tested he has to react, and if he reacts it is because the has been tested, or a struggle and victory, or the setting of a difficult task and its resolution, or a pursuit and a rescue from pursuit). This is equivalent to each character function being a necessary and sufficient condition for the other. But there is also a different type of dependency where the presence of one character function suggest that another one may follow, but that one can occur without the previous one (for instance, if the hero is branded at some stage during the tale, it is very likely that he will be recognised by the brand later in the tale; however, he may also be recognised by some other means). In this case, the first character function is a sufficient (but not necessary) condition for the second one.

Some character functions have several possible dependents. For instance, the presence of character function *unrecognised arrival* (the hero arrives at a new place in disguise) early in the sequence suggest that the character function *unfounded claims* (a false hero tries to claim merit on some of the hero's actions) may appear later, but also *hero recognised* (the hero is recognised and his merits are recognised). This type of dependencies create difficulties for simple ways of measuring satisfaction of dependencies.

3.1.2 Evaluating Satisfaction of Long Range Dependencies in Sequences of Character Functions

In addition to the metric proposed by Gervás [6] to measure conformance with Propp's canonical sequence, we want to consider two additional metrics.

To measure satisfaction of long range dependencies, we consider a metric that computes the number of dependencies that are actually satisfied out of the set that might have been satisfied. This is done by collecting the set of character functions present in the sequence that may have dependencies with other functions, and for each one, checking whether the character function that it depends

test by donor	=	hero reaction
hero reaction	-	acquisition magical agent
villainy	-	trigger resolved
lack	-	trigger resolved
hero dispatched	-	begin counteraction
hero dispatched	-	departure
begin counteraction	-	departure
branding	-	unrecognised arrival
branding	-	hero recognised
unrecognised arrival	-	false hero exposed
unrecognised arrival	-	unfounded claims
unrecognised arrival	-	hero recognised
unfounded claims	-	false hero exposed
struggle	=	victory
difficult task	=	task resolved
departure	-	return
hero pursued	=	rescue from pursuit

Table 2. List of long range dependencies between character functions: necessary conditions are indicated with a - sign, necessary and sufficient conditions with an = sign

upon is present in the sequence (before or after it, depending on the direction of the dependency). Bidirectional dependencies are counted twice if they are not satisfied. To normalise over a large set of tales, the metric currently returns 100 if there are no dependencies or if all dependencies are satisfied, and otherwise a number between 100 and 0 corresponding to the percentage of the dependencies present that have been satisfied.

3.1.3 Considering Dependencies in Generation of Sequences of Character Functions

Three different strategies for building sequences of character functions are proposed as alternatives of how long range dependencies can be taken into account:

- a baseline approach that follows to the letter the procedure described by Propp: builds a sequences of character functions by randomly deciding whether or not to include character functions from the canonical sequence in the appropriate order (save for the trigger, which is forcefully included, either as a villainy or as a lack)
- an extension of that procedure that takes into account dependencies between character functions by superimposing an alternative procedure for character functions in the sequence that may have dependents earlier in the result sequence accumulated so far; for these, the dependent is always inserted whenever the antecedent is already included
- a refinement on the alternative procedure for character functions that have dependents earlier in the sequence: for character functions subject to dependencies, inserts them if antecedent is already included and no alternative consequents already present (considers the fact that in cases where a character function gives rise to several possible dependents, the presence of one of them may be enough for a satisfactory result)

3.2 Considering Endings

Stopping conditions are crucial in any computational procedure. Story endings are also fundamental in the perception of the quality and the success of a story. In the process of building a sequence of character functions that will give rise to a story, it is important to consider whether the final character function of the sequence is likely to provide support for a satisfactory ending.

3.2.1 Identifying Character Functions that Support Satifactory Endings

As described in section 2.5, Propp does not explictly provide much information on the subject of endings. To obtain guidance on this issue one must turn to the set of examples of folk tales he considers in his book. By studying these we can come to some conclusions as to what character functions constitute suitable candidates to end a tale. The examples of tales in Propp's book come in two forms. One is the set of examples of analyses of tales given in Appendix II. The other is the set of schemes for tales tabulated in Appendix III.

Data has been collected for these two sources, and the results are presented in Table 3. Propp considers instantiations of his canonical scheme as the elementary unit for tales, which can be combined into more complex stories. Each instantiation of the canonical scheme is considered a move within the larger tale. The table lists both cases where character functions occur at the end of a tale and where character functions occur at the end of a move within a tale. It seems reasonable to assume that moves within a larger tale may finish in a character function that does not support a satisfactory ending. Yet suitability for ending a move may also be a merit in terms of ability to resolve a narrative thread.

	FE	FS	AE	AS
hero marries	5	27	7	34
hero recognised	1	1	1	1
villain punished	1			1
acquisition magical agent	1		4	4
return	1	8	2	21
rescue from pursuit		6	1	12
trigger resolved	1		2	
unrecognised arrival			1	1
difficult task			1	
villainy/lack	1		1	5
transfer				

Table 3. Frequency data for character functions occurring in final positions for tale examples and schemes: FE are final moves in examples, FS final moves in schemes, AE any move in example, AS any move in scheme.

3.2.2 Evaluating the Potential of a Sequence to Support a Satisfactory Ending

To measure the potential of a sequence to support a satisfactory ending, we consider a metric that computes whether the sequence ends in a character function that has been recorded to occur at the very end of a tale (not at the end of internal moves). The metric assigns a score of 100 it the last character function is within the collected list, and 0 otherwise.

4 Experimental Results

Results for the three different strategies for the generation of character sequences that have been tried are reported in Table 4. Each of the alternative implementations was run 100 times and values were averaged over the results.

These data show some interesting results. Given that all the strategies employed are based on following Propp's canonical sequence, the fact that they all achieve top score on the corresponding metric

Sequence 1 (Propp's baseline)	Sequence 2 (bl + strict dependencies)	Sequence 3 (bl + lax dependencies)
test by donor	lack	test by donor
acquisition magical agent	acquisition magical agent	hero reaction
lack	departure	acquisition magical agent
hero dispatched	transfer	lack
acquisition magical agent	trigger resolved	hero dispatched
transfer	unfounded claims	departure
victory	return	trigger resolved
trigger resolved	unrecognised arrival	difficult task
return	unfounded claims	task resolved
unfounded claims	hero recognised	return
	false hero exposed	transfiguration
	transfiguration	hero marries

Table 5. Examples of sequences of character functions

	Random	Sequence Strict	Sequence Lax
Conformance	100.0	100.0	100.0
Dependencies	48.0	96.7	92.7
Endings	70.0	58.0	73.0

Table 4. Results for different strategies for the generation of character sequences

is no surprise. The baseline predictably gets a low score on satisfaction of dependencies but a surprisingly high score on endings. The two alternative strategies for taking long range dependencies into account fare differently with respect to dependencies and endings. The strategy that imposes strictly the presence of any character function linked by a dependency achieves very high results on dependency satisfaction as expected. However, it achieves a very low score for satisfactory endings. The lax strategy of allowing only one dependent to be introduced whenever several are possible suffers a decrease in the degree of satisfaction of dependencies, but achieves a much higher score on satisfactory endings.

Overall, the values for satisfactory endings are low. This raises two different questions. One relates to the suitability of the rough procedure adopted towards the evaluation of this feature. In view of the results, it may pay to consider more fine grained metrics for the quality of endings. The other one points directly to the fact that the default approach to stopping conditions suggested by strict adherence to Propp's generative procedure as he described it may be insufficient to achieve tales with satisfactory endings. Better results would be obtained if some stopping condition is added to the procedure, so that the strategy for deciding whether or not to add a character function to the sequence takes into account some kind of criterion based on whether the last character function added is already a good candidate for a satisfactory ending.

Both of these points are beyond of the scope of the current paper and will be addressed as future work.

Examples of sequences of character functions resulting from these strategies are presented in Table 5.

Sequence 1 was produced by the baseline strategy following Propp's procedure strictly. It obtained a score of 100 on conformance to the canonical sequence, 20 on dependency satisfaction and 0 on potential for satisfactory endings. The low score on dependency satisfaction can be understood seeing that, for instance, the hero is tested and he acquires a magical agent without his reaction to the test being mentioned, the hero is dispatched but he does not actually leave, a victory is mentioned but no preceding struggle is described, and unfounded claims are made and not resolved. The fact that these unfounded claims are made right at the end of the sequence also explains the low score on potential for satisfactory endings.

Sequence 2 was produced by the strategy imposing strictly satisfaction of all dependencies. It obtained a score of 100 on conformance to the canonical sequence, unsurprisingly 100 on dependency satisfaction and 0 on potential for satisfactory endings. The low score on potential for satisfactory endings is explained by the fact that it ends with the character function for transfiguration. This occurs late in the canonical sequence but is followed by other character functions more suited to end the tale (namely marriage / reward, which in this particular case happen to have been ommitted by the random decision procedure). This sequence shows how the imposition of the dependencies forces very coherent subsequences of character functions, even though dependencies are stated only in terms of pairs of functions. This is because the pairs sometimes chain up to produce longer subsequences.

Sequence 3 was produced by the strategy that allows lax satisfaction of dependencies. It obtained a score of 100 on conformance to the canonical sequence, 100 on dependency satisfaction and 100 on potential for satisfactory endings. This is because it finishes the sequence with the most frequently used final character function in the set of stories examined: marriage of the hero. It also happens to read nicely as a coherent story, though this is a feature not actually considered by the metrics in use. Future work should be devoted to capturing this type of feature in an automatically computable metric.

5 Discussion

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 [10]) 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 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.

In terms of the criteria defined by Ritchie, the metrics presented in this paper for the evaluation of sequences of character functions are clearly instances of ratings of typicality, rather than novelty of these sequences. Conformance to a canonical sequence, satisfaction of dependencies, and provision for satisfactory endings constitute valuable features of an acceptable story. In fact, stories are more likely to be considered novel the further away they are from a canonical sequence, or if they allow for some unresolved dependencies, or if they opt for an unconventional ending. In this sense, the framework described in the present paper is unlikely to be considered creative by any standards. Nevertheless, it addresses fundamental issues concerning computational attempts to generate stories.

The framework proposed by Wiggins can be used to analyse the type of creative system that is being considered. The procedures described in this paper for generating sequences of character functions can be understood as defining a conceptual space of sequences of character functions. For each different strategy, the resulting conceptual space is different. The description of the strategy itself constitutes an instance of the traversal function that Wiggins defines to traverse the conceptual space. The definition of the conceptual space is implicit in the description of each strategy, as each one rules out different kinds of sequence of character function. For instance, all the strategies discussed in this paper rule out sequences of character functions that do not conform with Propp's canonical sequence (as shown by the results given in Table 4).

The metrics defined over sequences of character functions constitute instances of evaluation functions as defined by Wiggins. These metrics assign different values to elements of these conceptual spaces. The fact that the metrics are applicable to elements of all the different conceptual spaces is an important insight. In truth, these metrics are defined in such a way that they pick out elements of particular conceptual spaces by assigning high scores to them. In particular, the metric for satisfactory endings assigns top scores to sequences from a particular conceptual space that would only include sequences with satisfactory endings.

As mentioned above, the procedures for generating character functions described in this paper exhibit very low indices of creativity. However, they constitute an elementary exploration of what the conceptual spaces are for this particular kind of artefact, what traversal functions can be defined, and what metrics might be useful to capture features that humans consider typical for this domain. Once these basic elements have been established, more elaborate generative procedures may be explored. These can involve systematic exploration of how these basic elements can be progressively modified. The degree of transgression of the modifications considered would likely determine the perception of creativity arising from the results obtained. Along these lines, Propp himself considers some simple transgressions of his framework as possible when he talks about the possibility of having *inverted sequences* (p. 107), where character functions occur out of order but this is not considered a transgression of the basic rule. More elaborate transgressions are likely to lead to conceptual spaces further away from those considered here. As this happens, more refined evaluation functions will be required. This type of progression can be observed between the system presented by Gervás [6] and the one presented in this paper. Where the paper by Gervás addresses the construction of sequences of character functions, the strategies he defines determine a conceptual space of character functions different from those considered here (except for the baseline solution following Propp's procedure strictly, which is similar in both). To the extent that the present paper addresses issues not considered by Gervás [6], such as long range dependencies and endings, new extensions of the metrics are needed. The conceptual spaces defined by the strategies presented here constitute subsets of the more generic conceptual space defined by the baseline.

With respect to Colton's FACE model, Propp's generative procedure would constitute a *concept* of the process type. The particular implementation of that procedure described here would constitute an *expresion of that concept*, different from the expression of that same concept described in Gervás [6]. Interestingly enough, each sequence of character functions obtained by either of these procedures would itself be a concept of the artefact type, susceptible of being expressed in different ways. Gervás [6] provides more detail on how a sequence of character functions can, by means of an additional computational procedure, come to be expressed as a set of predicates describing specific instantiations of the character functions, all linked by a set of shared variables that represent the characters in the story. The procedure for generating such a sequence of predicates from a sequence of character functions would be a concept of the process type. Similar considerations can be made about the final stage in the construction of a story, that of rendering the set of predicates as text. This final text would itself be an artefact with an associate process to produce it.

With respect to the creative potential of the proposed framework, two different issues remain to be addressed. One is the need to avoid repetition, so that once a particular sequence of character functions has been produced by the system, the following attempts by the system try to avoid using the same character functions in the same order. Another is the possibility to break away from the rigid procedure for constructing stories, so that good stories beyond the set defined by the given rules can be achieved. The first issue corresponds to providing the system with better opportunities to explore regions of the conceptual space being searched that have not been visited already. The second issue corresponds to exploring possibilities of transforming the conceptual space to be explored beyond the one defined by the rigid rules set by Propp. According to Wiggins' framework, this would require the development of both new procedures for constructing stories (which would transform the definition of the conceptual space and the function for traversing) but most importantly, the definition of an evaluation that captures this concept of a *good* story beyond obeying structural constraints and conforming with Propp's morphology of the folk tale. Both of these remaining issues will be addressed as further work.

6 Conclusions

The generative procedure for constructing instances of Russian fairy tales provided by Vladimir Propp and based on his analytical framework does provide a very insightful model of the task of story generation. The division it provides of the task into the construction of an abstraction of the story in terms of character functions, the instantiation of these character functions with particular forms, the assignment of characters, and the rendition as text, provides a modular approach for addressing the overall task in engineering terms. Yet the brevity in which this generative procedure is described in Propp's book inevitably leaves many things unsaid and a large number of open problems. The present paper focuses on the first stage of constructing sequences of character functions, and addresses two open questions: the existence of long range dependencies between character functions and the potential of a sequence to provide support for a satisfactory ending.

For these particular problems computational solutions have been produced based on Propp's own analysis, convering both strategies for generating adequate sequences of character functions and metrics for evaluating their suitability. Results indicate further work is needed in refining both the strategies and the metrics.

The overarching framework of Propp's generative procedure, and the particular solutions presented in this paper have been analysed in terms of relevant theoretical advances in computational creativity. Although the proposed system has no claim to being considered creative, it has been argued that it constitutes a first step in a long road towards understanding the procedures involved in story generation, with a view to finding how and where these procedures can be infused with the spark of creativity.

Acknowledgements

This paper has been partially supported by the project WHIM 611560 funded by the European Commission, Framework Program 7, the ICT theme, and the Future Emerging Technologies FET program.

REFERENCES

[1] M. Boden, *Creative Mind: Myths and Mechanisms*, Routledge, New York, NY, 10001, 2003.

[2] Simon Colton, John Charnley, and Alison Pease, 'Computational creativity theory: The face and idea descriptive models', in *In 2nd International Conference on Computational Creativity*, (2011).

[3] Chris Fairclough and Pádraig Cunningham, 'A multiplayer case based story engine', in *GAME-ON Conference*, pp. 41–, (2003).

[4] Chris Fairclough and Padraig Cunningham, 'A multiplayer o.p.i.a.t.e.', *Int. J. Intell. Games & Simulation*, **3**(2), 54–61, (2004).

[5] P. Gervás, B. Díaz-Agudo, F. Peinado, and R. Hervás, 'Story Plot Generation Based on CBR', *Knowledge-Based Systems. Special Issue: AI-2004*, **18**, 235–242, (2005).

[6] Pablo Gervás, 'Propp's morphology of the folk tale as a grammar for generation', in *Workshop on Computational Models of Narrative, a satellite workshop of CogSci 2013: The 35th meeting of the Cognitive Science Society*, Universität Hamburg Hamburg, Germany, (08/2013 2013). Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany, Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany.

[7] D. Grasbon and N. Braun, 'A morphological approach to interactive storytelling', in *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*, (2001).

[8] R. Raymond Lang, *A Formal Model for Simple Narratives*, Ph.D. dissertation, Tulane University, 1997.

[9] Alison Pease and Simon Colton, 'Computational creativity theory: Inspirations behind the face and the idea models', in *In 2nd International Conference on Computational Creativity*, (2011).

[10] R. Pérez y Pérez, *MEXICA: A Computer Model of Creativity in Writing*, Ph.D. dissertation, The University of Sussex, 1999.

[11] Vladimir Propp, *Morphology of the Folktale*, University of Texas Press, Austin, 1968.

[12] Graeme Ritchie, 'Some Empirical Criteria for Attributing Creativity to a Computer Program', *Minds & Machines*, **17**, 67–99, (2007).

[13] T. Trabasso, P. van den Broek, and S.Y. Suh, 'Logical necessity and transitivity of causal relations in stories', *Discourse Processes*, **12**, 1–25, (1989).

[14] Scott R. Turner, *Minstrel: a computer model of creativity and storytelling*, Ph.D. dissertation, University of California at Los Angeles, Los Angeles, CA, USA, 1993.

[15] G. Wiggins, 'A preliminary framework for description, analysis and comparison of creative systems', *Knowledge-Based Systems*, **19(7)**, (2006).