A Case Based Reasoning Approach to Story Plot Generation

Belén Díaz-Agudo, Pablo Gervás and Federico Peinado

Dep. Sistemas Informáticos y Programación
Universidad Complutense de Madrid, Spain
{belend,pgervas}@sip.ucm.es, fpeinado@fdi.ucm.es
http:\\gaia.sip.ucm.es

Abstract. Automatic construction of story plots has always been a longed-for utopian dream in the entertainment industry, especially in the more commercial genres that are fueled by a large number of story plots with only a medium threshold on plot quality, such as TV series or video games. We propose a Knowledge Intensive CBR (KI-CBR) approach to the problem of generating story plots from a case base of existing stories analyzed in terms of Propp functions. A CBR process is defined to generate plots from a user query specifying an initial setting for the story, using an ontology to measure the semantical distance between words and structures taking part in the texts.

1 Introduction

The explosion of the information society and the various communication technologies have progressively shifted the bottleneck for the entertainment industry from technological issues to content production. Whether in the web site industry, the game industry, or the animation industry, companies employ larger numbers of screen writers and content providers than actual technicians or programmers (or devote higher portions of their budgets to buying ready made content elsewhere). Automatic construction of story plots has always been a longed-for utopian dream in the entertainment industry, specially in the more commercial genres that are fueled by a large number of story plots with only a medium threshold on plot quality, such as TV series or video games. Although few professionals would contemplate full automation of the creative processes involved in plot writing, many would certainly welcome a fast prototyping tool that could produce a large number of acceptable plots involving a given set of initial circumstances or restrictions on the kind of characters that should be involved. Such a collection of plots might provide inspiration, initiate new ideas, or possibly even include a few plot sketches worthy of revision. Subsequent selection and revision of these plot sketches by professional screen writers could produce revised, fully human-authored valid plots. By making such a collection of tentative plots available to company screen writers, a smaller number of writers might be able to provide the material needed to keep the technical teams in work.

In order for an automated plot generation tool to meet the requirements described above, its results would have to exhibit a degree of unpredictability. In his work on story generation, Turner [23] advocates the use of CBR in plot generation to provide a level of automated "creativity" to systems of this kind.

We propose a Knowledge Intensive CBR (KI-CBR) approach to the problem of generating story plots from a case base of Propp functions. A CBR process is defined to generate plots from a user query specifying an initial setting for the story, using an ontology to measure the semantical distance between words and structures taking part in the texts. This constitutes an improvement on previous work [10, 3] similar solutions had been applied to generate poetic texts using CBR but employing only syntactic information during adaptation.

2 Previous Work

In this section, we outline here previous relevant work on story generation, and issues concerning natural language generation (NLG) that are relevant to our method of rendering plots for ease of comprehension.

2.1 Story Generation

In his Morphology of the Fairy Tale [21], Propp derives a morphological method of classifying tales about magic, based on the arrangements of "functions". This provides a description of the folk tales according to their constituent parts, the relationships between those parts, and the relations of those parts with the whole. Propp's work has been used as a basis for a good number of attempts to model computationally the construction of stories.

The main idea is that folk tales are made up of ingredients that change from one tale to another, and ingredients that do not change. According to Propp, what changes are the names - and certain attributes - of the characters, whereas their actions remain the same. These actions that act as constants in the morphology of folk tales he defines as *functions*.

For example, some Propp functions are: Villainy, Departure, Interdiction, Interdiction Violated, Acquisition of a Magical Agent, Guidance, Testing of the hero, etc. There are some restrictions on the choice of functions that one can use in a given folk tale, given by implicit dependencies between functions: for instance, to be able to apply the *Interdiction Violated* function, the hero must have received an order (*Interdiction* function).

PftML, a Proppian fairy tale Markup Language[18]utilizes a Document Type Definition (DTD) to create a formal model of the structure of narrative and to standardize the tags throughout a corpus – a subset of the corpus Propp used – when analyzing it. This allows for an empirical test of the conclusions of Propp's initial analysis against the original data. We have used Propp's original work and PftML as the basic sources for building the ontology that underlies our system.

There have been various attempts in the literature to obtain a computational model of story generation. Important efforts along these lines are presented in [22, 13]. Tale-Spin [20] was the first approximation to story generation based on the transformational paradigm. The system has predefined rules that simulate a small environment that evolves while characters in it – based on Aesop's fables – try to achieve certain goals. The underlying mechanism is based on a planning engine that works on chcaracter goals. Resulting stories are not necessarily very interesting. Universe [14] defines characters in terms of frames, with a richer representation of their goals and restrictions. Particulary oriented towards generating plots in an ongoing serial. Teatrix [16] is a software application that helps children to develop narrative abilities. It offers a 3D environment where stories – actually small school plays – can be created cooperatively, intended for young children. The GEIST project [11] presents a story engine that generates digital interactive narrative in a mixed reality environment. It creates historic characters as ghostly apparitions that are combined with the users perception of reality. Automatic generation of stories is considered too complex, and the computer only takes part in the high level management of the story. They use a simplified version of Propp's morphology. IDA (I-Drama Architecture) [17] is an architecture incorporates both agents as actors that can be directed – but have a certain degree of autonomy – and a story director. It uses first order logic to represent the story and classic AI planning algorithms for reasoning. Minstrel [23] includes a CBR solution for generating stories about Arthurian characters. A moral for the story is used a a seed, and it attempts to model the way human bards compose their stories. It uses a semantic network as knowledge representation. The Proppian Fairy Tale Generator [18] [1], works by randomly choosing a possible interpretation of a function – as a previously written piece of text, a fragment of a folk tale – and stringing them together to create a story. In the resulting stories characters appear and disappear from one passage to the next, resolutions occur for conflicts never mentioned, and events have to be extrapolate between passages to make sense of the plot. This suggests that it is not enough to find combinations of Propp's functions to create viable stories. From a CBR point of view, this implies that no simple retrieval process will address the issue, and specific adaptation processes must be devised to obtain a useful solution.

Fairclough and Cunningham [7, 8] 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. A story director agent manages the plot using case-based planning.

Of particular interest is their definition of 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. Also interesting is their basic description of the world model in terms of characters, locations and objects. To be able to model them dynamically through the plot, characters are represented in terms of an internal state, which changes as a result of the interaction with other characters. Stories are controlled by a story director agent. This agent selects which functions should take place by ap-

[1] *http://www.brown.edu/Courses/FR0133/Fairytale_Generator/theory.html*

plying a case based planning system. The full CBR life cycle is used for storyline representation and a strategy formalization that allows for storyline adaptation. They use 80 cases extracted from 44 story scripts given by Propp. These scripts are defined as lists of character functions. There are stories composed of one, two or more moves. A case is a move, but there is no real option to change the order between moves, because a given story can only be resolved with its own particular choice of moves. Cases are seen as story templates, to be filled in by a constraint satisfaction system that chooses which characters perform the functions - *casting* - by taking into account the constraints imposed by the definitions of the functions and the models of the characters. Retrieval is done by means of a k-nearest neighbor algorithm that takes into account the functions performed so far and the character and object resources needed to execute a story script.

2.2 Natural Language Generation

The most natural format for presenting a plot to users is to describe it - or rather narrate it - in natural language. Obtaining a high quality natural language text for a story is itself a subject of research even if the plot is taken as given [2]. This paper is concerned strictly with the process of generating valid plots, and only the simplest sketch of a natural language rendition is attempted as means of comfortably presenting the results. Even so, there are certain technical issues concerning NLG that are relevant to this effort.

The most relevant line of research for our purpose is that concerned with NLG based on templates. The basic idea is that texts often follow conventionalized patterns. These patterns can be encapsulated in *schemas* [19], which are template programs which produce text plans. Schemas are derived from a target text corpus, by breaking up these texts into messages, and trying to determine how each message can be computed from the input data. This process may be easier to carry out if messages are organized into a taxonomy. This is currently the most popular text-planning approach in applied NLG

The schema-based approach to NLG has striking parallelism to CBR approaches to problem solving, in that existing previous solutions, such as those obtained from a corpus of target texts, are extracted and prepared so they can be reused to solve future problems, much in the same way as cases in a case-base are prepared from previous problem solutions.

This has lead to attempts to apply explicitly CBR solutions to the task of text generation. In [3] poetry generation is chosen as an example of the use of the COLIBRI (*Cases and Ontology Libraries Integration for Building Reasoning Infrastructures*) system. COLIBRI assists during the design of KI-CBR systems that combine cases with various knowledge types and reasoning methods. It is based on CBROnto [4–6], an ontology that incorporates reusable CBR knowledge, including terminology plus a library of reusable Problem Solving Methods (PSMs).

The transformation of a given prose text into a poem can be represented as a CBR process [10], where each case contains a sample – a sentence – of the source text as case description and a sample of the object poem as case solution. The current version of this system selects the required word at each stage solely based on its syntactic category and its relative position in either user proposal, case description or case solution. This works reasonably well for words originating from these sources, but not so well for the additional vocabulary.

3 Knowledge Representation for Plot Generation

Knowledge representation in our system is based on a basic ontology which holds the various concepts that are relevant to story generation. This initial ontology is subject to later extensions, and no claim is made with respect to its ability to cover all the concepts that may be necessary for our endeavour.

3.1 The Ontology

The ontology has been designed to include various concepts that are relevant to story generation. Propp's character functions are used as basic recurrent units of a plot. In order to be able to use them computationally, they have been translated into an ontology that gives semantic coherence and structure to our cases.

We have implemented this ontology using the last release of the Protégé ontology editor that was developed at Stanford University [9]. It can manage ontologies in OWL [1], a new standard that has recently reached a high relevance. The choice of OWL as a representation language provides the additional advantage, that it is designed to work with inference engines like RACER [12].

Although the functions of the *dramatis personae* are the basic components, we also have other elements. For instance, conjunctive elements, motivations, forms of appearance of the dramatis personae (the flying arrival of a dragon, the meeting with a witch), and the attributive elements or accessories (a witch's hat or her clay leg) [21]. An additional ontology provides the background knowledge required by the system, as well as the respective information about characters, places and objects of our world. This is used to measure the semantical distance between similar cases or situations, and maintaining an independent story structure from the simulated world. The domain knowledge of our application is the classic fairy tale world with magicians, witches, princesses, etc. The current version of the ontology contains a number of basic subconcepts to cover this additional domain knowledge that needs to be referred from within the represented function.

Propp functions In our approach, Propp's character functions act as high level elements that coordinate the structure of discourse. Each function has constraints that a character that is to perform it must satisfy. A view of the top of the function hierarchy is given in Figure 1 (left).

The contents of a function are the answers to the Wh-questions: what (the symbolic object), when, where (the place), who (who are the characters of the function) and why.

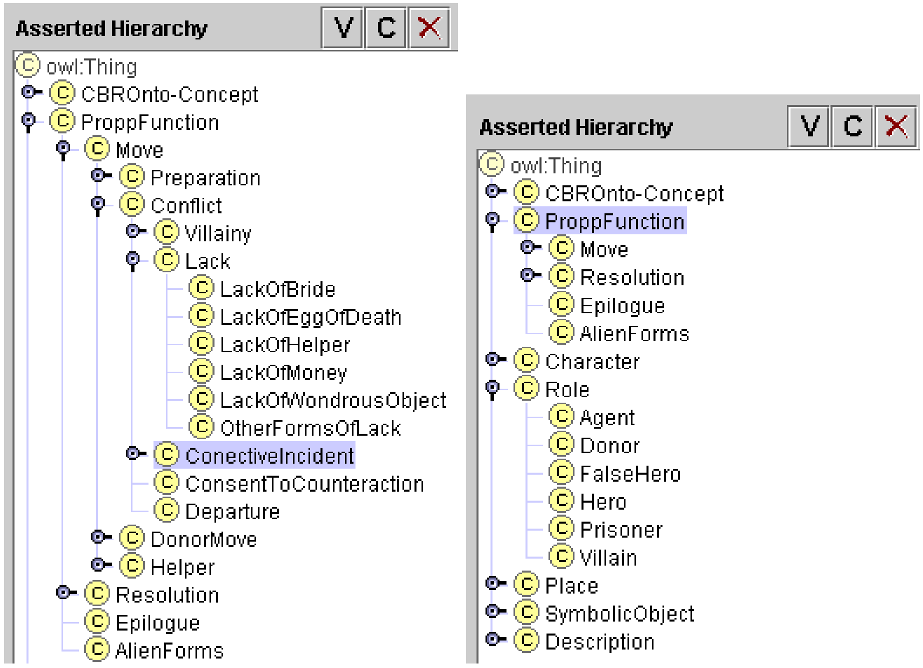

Fig. 1. Function and role sub-hierarchies in the ontology as modeled in Protégé.

Moves Morphologically, a tale is a whole that may be composed of *moves*. A move is a type of development proceeding from villainy or a lack, through intermediary functions to marriage, or to other functions employed as a *denouement* (ending or resolution). Terminal functions are at times a reward, a gain or in general the liquidation of a misfortune, an escape from pursuit, etc. [21].

One tale may be composed of several moves that are related between them. One move may directly follow another, but they may also interweave; a development which has begun pauses, and a new move is inserted.

We represent tales and their composing moves using structured descriptions. A tale is related with an ordered sequence of complete moves. We represent the temporal sequence between these moves using the CBROnto temporal relations. These representation issues are described in Section 3.2.

Character The roles in the story must be filled by characters. Each character is defined by a set of relationships with other characters, objects in his possession, location... These characters are one of the elements that the user can choose to customize a story.

Properties of the characters By properties or attributes of the characters, we mean the totality of all the external qualities of the characters: their age, sex, status, external appearance, peculiarities of this appearance... These attributes provide the tale with its brilliance, charm and beauty. However, one character in a tale is easily replaced by another (permutability law) [21].

In our ontology we represent a character's attributes using three basic headings: external appearance (and nomenclature), particularities of introduction into the narrative, and dwelling (described by a relation with a place). Also there are other auxiliary elements.

Roles Propp describes a number of 'spheres of action' that act as roles that certain characters have to fulfill in the story. The role sub-hierarchy of our ontology is given in Figure 1 (right).

Places and objects Certain locations (outdoors, indoors, countries, cities...) and symbolic objects (towels, rings, coins...) can be significant to the way a story develops, and any sort of substitution during adaptation must take this into account. Our ontology must have the ability to classify such locations and objects.

Descriptions Since our system generates text by filling in templates with selected descriptions that correspond to instances of particular concepts, it was considered convenient to have these descriptions represented in the ontology in such a way that their relations with the relevant concepts can also be modelled and the inference mechanisms available can be employed in their selection.

3.2 The case base

The case base is built up of texts from the domain of fairy tales, analyzed and annotated according to Propp's morphology.

A selection of stories from the original set of the Afanasiev compilation originally used by Propp are taken as sources to generate our initial case base.

In the CBR literature there are different approaches to case representation and, related to that, different techniques for Case Based Reasoning: the *textual* CBR approach, the *conversational* CBR approach, and the *structural* CBR approach. In the textual CBR approach, cases are represented in free-text form. In the conversational CBR approach, cases are lists of questions and answers. For every case, there can be different questions. In the structural CBR approach, the developer of the case-based solution decides ahead of time what features will be relevant when describing a case and then stores the cases according to these.

We use a structural CBR approach that relies on cases that are described with attributes and values that are pre-defined, and structured in an object-oriented manner. This structural CBR approach is useful in domains (like the one we are considering) where additional knowledge, beside cases, must be used in order to produce good results. The domain ontology assures the quality of new cases (regarding the ontology commitments) and the low effort in maintenance.

Within the defined case structures we represent the plots of the fairy tales. Besides this structural representation of the cases we also associate a textual representation to each case that can be used to generate texts from the plot descriptions (see Section 4.2).

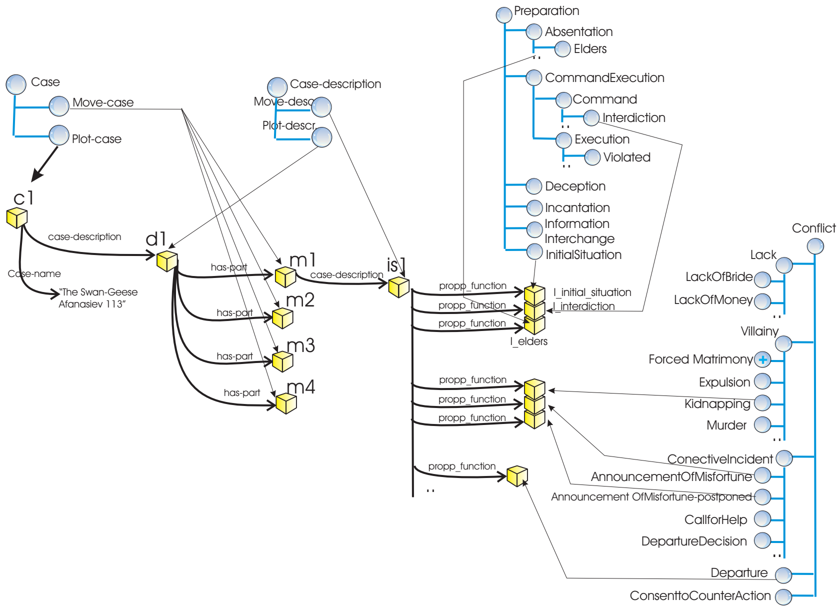

Fig. 2. Case Structure

We facilitate the case structure authoring tasks by proposing a framework to represent cases that is based on the Description Logics (DLs) instance definition language and the CBROnto terminology. Besides, we define a reasoning system (based on generic CBR PSMs) that works with such representations [6].

Cases are built based on CBROnto case representation structure [4–6] using the vocabulary from the domain ontology described in Section 3.1. The semantic constraints between scene transitions are loosely based on the ordering and co-occurrence constraints established between Proppian functions. Because the case base is made using cases proposed by Propp, we know that the system makes correct versions of folk tales.

CBROnto provides a *primitive* concept `CASE`. Subconcepts of `CASE` are referred to as *case-type* concepts. Cases of different types are represented as instances of different `CASE` subconcepts, so they will not have, in general, the same structure. The designer will define case-type concepts to represent the new types of cases.

Each case-type concept (more exactly its set of instances) defines a case base where CBR processes can be applied. This mechanism allows having different level of abstraction where there are cases that are part of other cases (see Figure 2).

In our application each case represents a complete tale that is typically composed of one or more interrelated moves (that are also cases). For representational purposes, relations between moves are basically of two types: *temporal* relations (before, after, during, starts-before, ends-before, ...) or *dependencies* (meaning

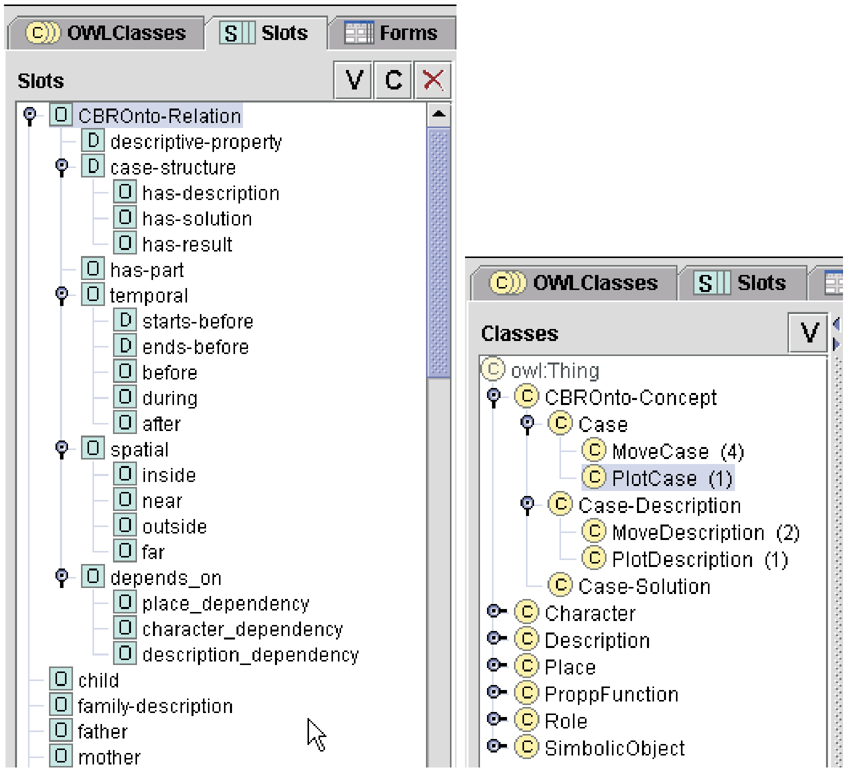

Fig. 3. CBROnto relation (left) and concept (right) hierarchies in Protege

that a change in one of them strongly affects the other) like *place-dependency*, *character-dependency* and *description-dependency* [5].

DLs allows representing hierarchies between relations (see Figure 3) what eases defining reasoning methods (using the top level relation) that are applicable (and reusable) with all the sub-relations.

As an example of the type of stories that are being considered, the following outline of one of the tales that Propp analyzes is given below [2]. The main events of the plot are described in terms of character functions (in bold) :

> *The Swan Geese (113 of Afanasiev Collection).* **Initial situation** (a girl and her small brother). **Interdiction** (not to go outside), **interdiction violated**, **kidnapping** (swan geese take the boy to Babayaga's lair), **Competition** (girl faces Babayaga), **Victory**, **Release from captivity**, **Test of hero** (swan geese pursue the children), **Sustained ordeal** (children evade swan geese), **Return**.

[2] Complete text in *http://gaia.sip.ucm.es/people/fpeinado/swan-geese.html*

4 How the system works

We propose a Knowledge Intensive CBR approach to the problem of generating story plots from a set of cases consisting of analyzed and annotated fairy tales. Our system operates in two phases: an initial one that applies CBR to obtain a plot plan from the conceptual description of the desired story provided by the user, and a final phase that transforms the resulting plot plan into a textual rendition by means of template based NLG.

4.1 The First Stage: Description to Plot Plan

We use the descriptive representation of the tale plots with a CBR system, that retrieves and adapts these plots in several steps and using the restrictions given in the query.

Query specification, similarity assessment and case retrieval A query determines the components of the tale we want to build. For example, its characters, descriptive attributes, roles, places, and the Propp functions describing the actions involved in the tale. Although there are roles whose existence (a character that plays that role) is mandatory in every plot, like the hero and the villain, they are not required in the query as they can be reused from other plots (cases).

Query description is an iterative and interactive process. The user first describes: the characters in the tale , their roles and attributes, the places where the story occurs, and so on. The user can do this, either by selecting individuals (predefined instances) from the ontology or creating new ones (new instances of the ontology concepts). Afterwards, the user determines the set of character functions that are going to be involved in the story. The user selects these functions from the given set of Propp functions represented in our ontology (see Figure 1 (left)). Optionally the user indicates which characters take part in each function. The knowledge in the ontology (and the associated reasoning processes) helps the user to fill the attributes of each function while maintaining the corresponding restrictions between the concepts and individuals.

If the function set is incompatible the system guides the user toward an appropriate query definition. For Propp the plot is a causal closed system where adequate characters, places, etc. are selected.

We propose an interactive case retrieval process taking into account progressive user inputs. The system retrieves the more similar case with the restriction of Propp morphology and characters available. As CBROnto provides with a general test-bed of CBR methods we have made different tests with different similarity measures between the complex descriptions that represents the plots([5]).

Each retrieved case constitutes a plot-unit template. The retrieved case components are *hot-spots* (or flex points) that will be substituted with additional information obtained from the context, i.e. the query, the ontology and other cases, during the adaptation process. Similarity measures should guarantee that

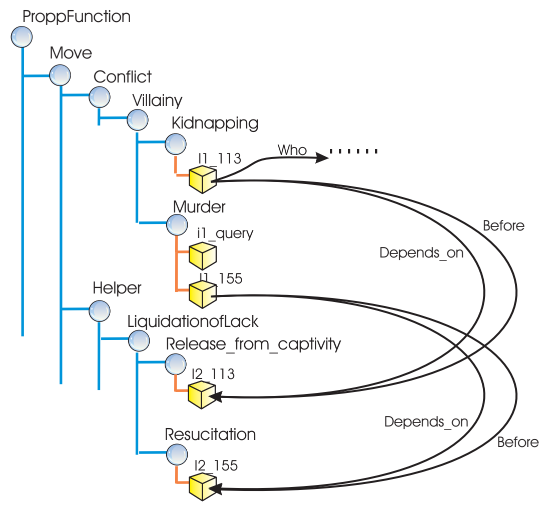

Fig. 4. Substitution example

(when possible) all the query elements are valid elements to be allocated in the retrieved cases.

For instance, let us say we want a story about a *princess*, where **murder** occurs, where an **interdiction** is given and **violated**, there is a **competition**, and a **test of the hero**. We can use that information to shape our query. The system retrieves the case story number 113, Swan-Geese (whose analysis has been given in the previous section).

Retrieval has occurred because the structure of this story satisfies straight away part of the conditions (interdiction, competition, test of hero) imposed by the query. No murder appears, but there is a *similar* element: a kidnapping. **Kidnapping** and **murder** are similar because they are different types of villainies; so, they are represented as children of the same concept **Villainy** in the ontology.

Adaptation The retrieval process provides with the plot skeleton where the system makes certain substitutions. A basic and simple initial adaptation step is to substitute the characters given in the query into the template provided by the retrieved case. This is equivalent to Fariclough and Cunnigham's process of *casting*. The retrieval process described above together with this simple adaptation mechanism provides a starting point for the plot generation, but it will not produce original results.

We also propose an adaptation method based on substitutions, but more creative results may be achieved by generating a solution as a mixture of the ingredients from various cases. During the adaptation of our *plot case*, we use additional retrieval steps (defining adequate queries) over the case base of *move cases* (that are part of the plot cases (Figure 2)) to find appropriate substitutes maintaining the dependencies and temporal relations.

In our example, the system should suggest an adaptation where **murder** is substituted for the **kidnapping**. However, the **kidnapping** in the retrieved case has *dependencies* with the **release from captivity** that appears later on (which is a **liquidation of lack** according to the ontology) (see Figure 4). To carry out a valid adaptation, the adaptation process is forced to define a query and retrieve cases in which **murder** appears with a *similar* dependency (i.e. dependency with another **liquidation of lack**).

The following case is retrieved (only a part of which is relevant to the issue):

> (*155 of Afanasiev Collection*). (...) **Absentation** of the hero (brother goes hunting), **Deception** of the villain (beautiful girl entices him), **Murder** (girl turns into lioness and devours him), (...) **Consent to counteraction** (other brother sets out), **Competition** (faces beautiful girl), **Victory** (kills lioness), **Resurrection** (revives brother), **Return**.

In this case there is a dependency between the **murder** and the **resuscitation**. The adaptation system can therefore substitute the kidnapping-release pair in the first retrieved case with the murder-resuscitation pair in the second, obtaining a better solution for the given query. Additional adaptations can be carried out to substitute the hero of the first case (the girl) or the prisoner (the boy) for the princess specified in the query. Besides, the swan-geese character in the retrieved case can be substituted for a similar element (for instance, another animal like the lioness that appears in the second retrieved case).

Additionally, those character functions that appear in the retrieved case but not in the query and that are not relevant for the story as developed so far can be filtered out during adaptation. This involves taking into account the explicit representation of dependencies between functions represented in the cases.

The resulting plot, showing how the two cases are combined, could be a story like this:

> *The Lioness (new fairy tale)*. **Initial situation** (a knight and his beloved princess). **Interdiction** (not to go outside), **Interdiction violated**, **Murder** (a lioness devours her), **Competition** (knight faces the lioness), **Victory** (kills lioness), **Resurrection** (revives the princess), **Return**.

4.2 The Second Stage: Plot Plan to Textual Sketch

This corresponds to the natural language generation basics to be applied to the plot plan in order to obtain a readable rendition of it. The current version is not focused on these issues, so this stage is currently covered by a skeleton solution to the problem. The NLG is simple, it uses templates without a grammar. If the CBR process of the first stage has taken place successfully, the second stage will accept as input a data structure satisfying the following constraints:

- The case that has been selected during retrieval, has been pruned or combined with other cases retrieved during adaptation and strung into a case sequence that makes up a plot skeleton.

- The character functions, acting as templates for the basic units of the plot, have been filled in during adaptation with identifiers for the characters described in the query

This result corresponds to the sort of structure that in NLG is termed a text plan, and we will refer to it as a *plot plan*. Various operations of increasing degree of difficulty are possible from this stage.

First Approximation: Direct Surface Realization of the Plot Plan A one-to-one correspondence can be established between character functions in the plot plan and sentences to be expected in the output. In this case, a simple stage of surface realization can be applied to the plot plan. Such a stage converts the templates into strings formatted in accordance to the orthographic rules of English - sentence initial letters are capitalized, and sentences are ended with a colon.

This approach is very similar to the process of syntax-based substitution applied in [3], with a number of interesting improvements. Having access to the sort of relations embodied in an ontology provides the means for finding the most suitable substitutions during adaptation. To avoid ambiguity problems during this process, each instance of a concept in the ontology must have a unique identifier. Semantic correctness of the output text is enforced by the substitution process.

The fact that we are using an ontology to represent concepts, and not a set of axioms encoding their meaning somehow restricts the degree of correctness that can be guaranteed by the substitution process. Any checking algorithm can only test for structural equivalence within the ontological taxonomy, and it cannot carry out proper inference over the meanings of concepts.

Nevertheless, in order to solve the problem of syntactical coherence of the texts, issues like number and gender agreement, some sort of syntactic information must be available in the system. Templates partly solve the need for having an explicit grammar, but some of these questions must be taken into account and therefore must be explicitly represented in the system.

In view of the various arguments outlined so far, it is perceived that the system would greatly benefit from incorporating semantic information in the form of a knowledge rich ontology. This addition must be done in some way that allows the new information to play a role easily in the various processes carried out by the system.

A More Elaborate Approximation: Micro-plan the Sentences of the Plot Plan Additional stages of *micro-planning* (or *sentence planning*) are performed on the elements that constitute the plot plan. This involves operations like joining sentences together wherever this gives more fluency to the text, substituting character identifiers for pronouns or definite descriptions wherever appropriate, or selecting specific words to use in realizing templates whenever options are available.

Due to the inherent heuristic nature of their application in the present context, these operations are better carried out by applying rule-based solutions.

An alternative is to employ a CBR solution. This would imply associating a textual representation to each case, with adequate links established explicitly between the ingredients that play particular roles in the case - characters, locations or objects - and their textual representations. These links would have to be used during adaptation to ensure that the textual descriptions corresponding to the character identifiers that have been introduced in the case during the adaptation of the first stage are substituted in the relevant textual templates.

CBR might be applied to the process of selecting an appropriate description for a given identifier when it has to be mentioned in a specific context of the discourse. If each character - represented in our ontology - has a set of possible descriptions - also represented in our ontology - associated with it, the system could resort to such annotated cases as described above to find which of the possible descriptions might be better used in the context. For this operation to be successful, the cases employed - discourse configuration cases - would probably have to represent a larger window of the previous context than was necessary for the CBR process of the first stage, which focused on cases as basic plot-units.

5 Conclusions

The system described is an advancement

Although the system is not fully-implemented yet, the progress so far points to a reasonable solution for Story Generation. Our approach follows the lines of structuralist story generation, which distinguishes from more tranformational work [20, 17]. It constitutes an evolution on Minstrel [23] in as far as the ontology used for representation is a formal evolution of the semantic networks it used. It improves on [11] in terms of a full implementation of Propp's morphology and actual involvement of the computer in story generation.

Unlike the uses of Proppian functions described for other systems, our approach represents character functions with more granularity. This allows the establishment of relations between characters and attributes and the functions in which they appear. Using this facility, a coherent character set can be guaranteed throughout the story. Additionally, dependencies between character functions are modelled explicitly, so they can be checked and enforced during the process of plot generation without forcing the generated plots to be structurally equivalent to the retrieved cases.

The system architecture is general in as much as one accepts Propp's set of character functions as complete. In the face of disagreement, the ontology is easy to extend, and, as mentioned before, it is not intended to be complete as it is. Under these conditions, the approach described in this paper may be extend to work in other domains.

In future work we intend to address the specific problems of the second stage, involving the transition from plot plan to textual sketch, and to explore the possible interactions between the two stages. Once these issues have been solved, the integration of the generator with software applications for script writing – along the lines of classics [?,?] – can be contemplated.

Given the facilities that COLIBRI provides for reusing knowledge that exists in the form of ontologies, we contemplate using Mikrokosmos [15] as an additional resource to support richer implementations of the NLG tasks involved in the second stage. An effort to port Mikrokosmos from its frame-based format to a DL format like OWL is under way.

6 Acknowledgements

The work was partially funded by the Spanish Committee of Science & Technology (TIC2002-01961). The last author is supported by a FPI Predoctoral Grant from Complutense University of Madrid.

References

1. S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and A. Stein. OWL web ontology language reference. W3C http://www.w3.org/TR/2004/REC-owl-ref-20040210/, February 2004.
2. C. B. Callaway and J. C. Lester. Narrative prose generation. *Artificial Intelligence*, 139(2):213–252, 2002.
3. B. Díaz-Agudo, P. Gervás, and P. González-Calero. Poetry generation in COLIBRI. In S. Craw and A. Preece, editors, *ECCBR 2002, Advances in Case Based Reasoning*. Springer, 2002. Lecture Notes in Artificial Intelligence.
4. B. Díaz-Agudo and P. A. González-Calero. An architecture for knowledge intensive CBR systems. In *Advances in Case-Based Reasoning – (EWCBR 2000)*. Springer-Verlag, 2000.
5. B. Díaz-Agudo and P. A. González-Calero. A declarative similarity framework for knowledge intensive CBR. In *Procs. of the (ICCBR 2001)*. Springer-Verlag, 2001.
6. B. Díaz-Agudo and P. A. González-Calero. Knowledge Intensive CBR through Ontologies. *Expert Update*, 2003.
7. C. Fairclough and P. Cunningham. An interactive story engine. In M. O'Neill, R. Sutcliffe, C. Ryan, M. Eaton, and G. N.J.L, editors, *13th Irish International Conference on Artificial Intelligence and Cognitive Science*, volume 2464, pages 171–176, Limerick, Ireland, 2002. LNCS Springer-Verlag Heidelberg.
8. C. Fairclough and P. Cunningham. A multiplayer case based story engine. In *4th International Conference on Intelligent Games and Simulation*, pages 41–46. EUROSIS, 2003.
9. J. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu. The evolution of Protégé: An environment for knowledge-based systems development. Technical report, Stanford University, 2002.
10. P. Gervás. An expert system for the composition of formal Spanish poetry. *Journal of Knowledge-Based Systems*, 14(3–4):181–188, 2001.
11. D. Grasbon and N. Braun. A morphological approach to interactive storytelling. In M. Fleischmann and W. Strauss, editors, *Artificial Intelligence and Interactive Entertainment, Living in Mixed Realities*, Sankt Augustin, Germany, 2001.
12. V. Haarslev and R. M˙*RACER User s Guide and Reference Manual Version 1.7.7.*

13. R. R. Lang. *A Formal Model for Simple Narratives.* PhD thesis, Tulane University, 1997.
14. M. Lebowitz. Storytelling and generalization. In *Seventh Annual Conference of the Cognitive Science Society*, pages 100–109, Berkeley, California, 1987. FALTA POR LOCALIZAR. Autor del sistema UNIVERSE de generación de historias.
15. E. Lonergan. Lexical knowledge engineering: Mikrokosmos revisited. In *PACLING2001 - Pacific Association for Computational Linguistics 2001*, Kitakyushu, Japan, 2001.
16. I. Machado, R. Prada, and A. Paiva. Bringing drama into a virtual stage. In *Proceedings of CVE 2000.* ACM Press, 2000.
17. B. Magerko. A proposal for an interactive drama architecture. In *AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, Stanford, CA, 2002. AAAI Press.
18. S. A. Malec. Proppian structural analysis and XML modeling. http://clover.slavic.pitt.edu/s̃am/propp/theory/propp.html, February 2004.
19. K. R. McKeown. The text system for natural language generation: An overview. In *20th Annual Meeting of the ACL*, pages 261–265, Canada, 1982.
20. J. R. Meehan. Tale-spin and micro tale-spin. In R. C. Schank and C. K. Riesbeck, editors, *Inside computer understanding.* Erlbaum Lawrence Erlbaum Associates, Hillsdale, NJ, 1981.
21. V. Propp. *Morphology of the Folktale.* University of Texas Press, Austin, 1968.
22. D. E. Rumelhart. Notes on a schema for stories. In D. G. Bobrow and A. Collins, editors, *Representation and Understanding: Studies in Cognitive Science*, pages 211–236. Academic Press, Inc., New York, 1975.
23. S. R. Turner. Minstrel: A computer model of creativity and storytelling. Technical Report UCLA-AI-92-04, Computer Science Department, 1992.