Integration of Linguistic Markup into Semantic Models of Folk Narratives: The Fairy Tale Use Case

Piroska Lendvai[1], Thierry Declerck[2], Sándor Darányi[3], Pablo Gervás[4], Raquel Hervás[4], Scott Malec[5], Federico Peinado[4]

[1]Research Institute for Linguistics, Hungarian Academy of Sciences, Budapest, Hungary
[2]Language Technology Lab DFKI GmbH, Saarbrücken, Germany
[3]Swedish School of Library and Information Science, Boras, Sweden
[4]Universidad Complutense de Madrid, Spain
[5]Carnegie Mellon University, Pittsburgh, USA
E-mail: piroska@nytud.hu

Abstract

Propp's influential structural analysis of fairy tales created a powerful schema for representing storylines in terms of character functions, which is straightforward to exploit in computational semantic analysis and procedural generation of stories of this genre. We tackle two resources that draw on the Proppian model – one formalizes it as a semantic markup scheme and the other as an ontology – both lacking linguistic phenomena explicitly represented in them. The need for integrating linguistic information into structured semantic resources is motivated by the emergence of suitable standards that facilitate this, and the benefits such joint representation would create for transdisciplinary research across Digital Humanities, Computational Linguistics, and Artificial Intelligence.

1. Introduction

During the past decade, several computational models targeted the processing of fairy tales, which in effect yielded a number of resources of folkloric texts. These models aimed to create the means for describing narration as it occurs in this genre, typically in terms of moves and their ingredients. They often drew on the work of folklorist Vladimir Propp who claimed to identify the basic plot components of Russian folk tales as a set of irreducible narrative elements, which he called 'character functions' (Propp, 1968). We focus on two resources that employ Proppian functions: PftML (Proppian fairy tale Markup Language) (Malec, 2001) that serves for narrative text segmentation and annotation, and ProppOnto (Proppian Ontology) (Peinado et al., 2004) that represents several aspects of folk tales, utilized for story generation purposes.

Both PftML and ProppOnto involve concept categories directly exploiting the functions established by Propp, in the form of hierarchically ordered textual content objects. Despite the fact that they serve different but related goals, neither of these resources incorporate representations of linguistic properties of the annotated corpus: no guidelines are provided for the linguistic segmentation of a tale into Proppian functions, and no specification is given about how and which linguistic elements are to be associated with classes and relations of the ontology.

Based on a small initial corpus of fairy tales, we indicate in some detail how this might be achieved. Since currently no methodology in folklore studies is available that specifies the magnitude (supra-, sub- or sentence level) of linguistic annotation to mark up Proppian or other functions, we would like to show that incorporating linguistic information in these models can fill this gap, delivering improved resources for folklore texts analysis, retrieval, and generation.

We also expect that results of this work will act as an indicator whether gaining more insight into linguistic vehicles underlying Proppian motifs will generalize to content units in folklore with a much broader spectrum, to the so-called *motifs*, the focus of the AMICUS project[1]. In other words, Proppian functions as a use case can bring far-reaching consequences for the study of folk narratives in general and of motifs in particular, benefiting both folk narrative studies and computational linguistics research.

2. Proppian analysis and its semantic models

In his influential treatise on the devices of narrative (Propp, 1968), Propp analyzed a set of about 100 tales from Afanas'ev's collection (Afanas'ev, 1957) and established the basic recurrent units of the fairy tale plot, called 'character functions', as well as the ways they can be combined. Within the various types of frames of character functions, actors perform certain roles and interact with each other; the sequences of actions thereby building up the storyline. For example, the protagonist absents himself from home, there is an interdiction superimposed on him/her, which is often violated, some kind of villainy (e.g. theft) takes place, so that the hero needs to complete tasks with the help of magical objects, there might be a direct struggle with the villain, after which the hero is getting rewarded, and so on[2]. Propp's original goal with his work was to derive a morphological method of magic tale classification, based on the arrangements of functions. He described the structure of tales using combinations and sequences of elements as an alternative to Aarne-Thompsons system based on a historic-geographic method of comparative folkloris-

[1]http://ilk.uvt.nl/amicus

[2]The full list of Proppian functions is available at http://clover.slavic.pitt.edu/sam/propp/praxis/features.html

tics, which looked at themes (Aarne, 1961). Propp was trying to understand the DNA-like structure of tales as a novel way to provide comparisons.

For both the Artificial Intelligence and the Digital Humanities research community, the conceptual categories of his classification scheme serve as semantic markup of narrative events in folk tales, and are thus important metadata.

2.1. Proppian fairy tale Markup Language (PftML)

Creating PftML almost a decade ago was based on the insight that Propp's functions are renderable by hierarchically arranged elements in eXtensible Markup Language (XML) documents. PftML was developed to create a formal model of the structure of Russian fairy tale narrative, using a set of synchronized metadata labels for the annotation of a corpus of fairy tales. The 31 employed labels are based on close reading of Propp's work, covering his view of the texts.

The functional elements to be marked up answer questions that require thorough understanding of the following characteristics of a tale's content: Who is the protagonist? Was there an admonition not to do something? Where are the parents? Was there a villainy or a lack (covering Propps cardinal functions, the *sine qua non* of a tale)? Is there a donor? Did the donor give a gift? Was the protagonist asked to put up a struggle before receiving a gift? If so, what was the nature of the struggle? What is the nature of the gift? Is there a villain? Does the protagonist confront a villain? How? Is there a happy ending?

Propp was using tables to categorize his observations – what one would probably use a spreadsheet or a database for today, which additionally motivated the effort to recreate them using XML. PftML is currently meant to assign labels to texts manually. The annotated texts were created to serve in future work for training a machine learning program which could thereby acquire term co-occurrence patterns to automatically apply annotation and reliably parse a tale according to a story grammar. The corpus created holds a subset of the Russian original of the Afanas'ev collection: 29 so-called 'magic' tales, to which we here refer using the more common term 'fairy tale'.

An excerpt from the tale *The Swan-Geese* in Figure 1 is marked up with PftML using some of the Proppian functions. From a computational linguistics point of view, we can state that the text is annotated in a coarse-grained manner, because the textual chunks that are labeled with a function are relatively long. Note however that a function in fact does not always need to cover full sentences, as the evidence here suggests: it may encompass a segment smaller than the whole sentence. For example, contrary to the actual markup in the example, the *<Execution subtype="Violated">* label actually only pertains to the chunk starting with *the daughter soon enough forgot*.

We propose to investigate the automation of mechanisms underlying the assignment of a function to a span of words, so that more fine-grained textual units can be labeled with units relevant to fairy tales. We base the approach on linguistic analysis, hypothesizing that boundaries of certain linguistic objects overlap with onsets of Proppian functions. A consequence of the creation of precise function boundaries is that statistical characterization and automatic processing of texts from the tale genre will improve, among others enabling better recall on objects in search and retrieval, and the detection and marking of higher-level phenomena in storylines such as semantic cross-reference, variation, as well as core and peripheral elements of motifs.

```
<Folktale Title="The Swan-Geese" AT="480" NewAfanasievEditionNumber="113"
ProppConformity="Yes">
 - <Move>
   - <Preparation>
     - <InitialSituation>
         Once upon a time a man and a woman lived with their daughter and small son.
       </InitialSituation>
     - <CommandExecution>
       - <Command subtype="Interdiction">
           "Dearest daughter," said the mother, "we are going to work. Look after your brother!
           Don't go out of the yard, be a good girl, and we'll buy you a handkerchief."
         </Command>
       - <Execution subtype="Violated">
           The father and mother went off to work, and the daughter soon enough forgot what
           they had told her. She put her little brother on the grass under a window and ran into
           the yard, where she played and got completely carried away having fun.
         </Execution>
       </CommandExecution>
     </Preparation>
   + <Villainy subtype="Kidnapping"></Villainy>
   + <ConsentToCounteraction></ConsentToCounteraction>
   + <Departure></Departure>
     <!-- border of segment quite arbitrary here -->
     <!-- trebling of Donors -->
   + <DonorFunction subtype="TestOfHero"></DonorFunction>
   + <AcquisitionOfMagicalAgent subtype="HelperOffersServices">
     </AcquisitionOfMagicalAgent>
   + <DonorFunction subtype="TestOfHero"></DonorFunction>
   + <AcquisitionOfMagicalAgent subtype="HelperOffersServices">
     </AcquisitionOfMagicalAgent>
   + <DonorFunction subtype="TestOfHero"></DonorFunction>
   + <AcquisitionOfMagicalAgent subtype="HelperOffersServices">
     </AcquisitionOfMagicalAgent>
   + <Transference subtype="RouteShownToHero"></Transference>
   + <StruggleVictory subtype="Competition"></StruggleVictory>
   + <LiquidationOfLack subtype="ReleaseFromCaptivity"></LiquidationOfLack>
   + <PursuitRescueOfHero></PursuitRescueOfHero>
   + <Return></Return>
   </Move>
</Folktale>
```

Figure 1: PftML applied to *The Swan-Geese*. Partial view

We also note that PftML uses in-line annotation, i.e. it directly interleaves markup with text, making it difficult to be annotated in a fine-grained manner, or to annotate with information coming from various sources or semantic models, e.g. according to different views on narrative functions.

2.2. ProppOnto

The context in which ProppOnto was created is an application for the generation of new fairy tales by reusing material from a corpus of existing ones. To inform the process a large body of ontological knowledge was built. This included ProppOnto as a specific ontology for representing Propp's analysis of fairy tales, where the ontology on Propp's analysis is subsumed by a more generic ontology for representing narratological concepts, and separate ontologies are used for representing additional semantic information, for instance world concepts ('boy', 'girl', 'blonde', 'brave', etc.) or temporal concepts ('before', 'after', 'day', 'night', etc.). The scheme-like structures composed by character functions are connected with "cause" and "effect" relations in ProppOnto.

The aim of an ontology is to represent interrelated concepts (for example Proppian functions) in the form of a network, thereby constituting an elementary vocabulary to represent knowledge about a given domain. This vocabulary can be very useful as a source for labels to use in annotating a cor-

pus. However, an ontology usually takes the form of a taxonomy of concepts organised into a hierarchy. Ontologies developed in OWL include the additional option of representing instances of the concepts they define. This implies that not only a concept such as 'girl' can be represented in an ontology, but also "Cinderella" as a particular instance of the concept 'girl'.

Currently, 45 fairy tales are represented in this form in the knowledge base of ProppOnto. These thus constitute a seed for a corpus of fairy tales annotated on several semantic levels. The actual representation of the fairy tales contained in this mini corpus is detailed, containing a wealth of semantic information that goes beyond lexico-syntactic details, e.g. temporal information, semantic description of the characters and locations involved in each Proppian function, as well as the correct assignment of Proppian roles to these characters. Because of the nature of how knowledge is represented in an ontology, each tale is represented as an instance of the concept of tale. As such, it inherits a large number of structural features (e.g., decomposition into moves, which themselves are sequences of Propp functions) from its parent concept. This means that each fairy tale represented in ProppOnto can be said to be annotated with a complex narrative structure in the form of a tree. Additionally, the actual plot of the story may be represented at a conceptual level, in terms of the world concepts and temporal concepts of the other branches of the general ontology.

ProppOnto can thus provide labelled data that can be utilized in annotating raw text with the classes and relations that are in the ontology. We argue that additionally associating fine-grained linguistic objects with the classes and relations in ProppOnto will create an improved resource for both natural language processing (NLP) and generation (NLG) purposes. Figure 2 shows part of the structure of Proppian functions as represented in ProppOnto.

Figure 2: Proppian functions represented in ProppOnto. Partial view

3. Integration with linguistic information

Our proposal is to recast PftML and ProppOnto so that they allow for incorporating multi-layered linguistic markup in order to support a joint representation of domain-ontological and lexico-grammatical knowledge. In doing so we support current work in eHumanities and Cultural Heritage studies on integrating content analysis and classification schemes.

3.1. Standards

A standardized linguistic annotation ensures interoperability and reusability of linguistic information associated with various texts (in our case fairy tales) in different languages and versions. We draw on recent work of the ISO committee on Language Resources Management[3], adopting a stand-off annotation strategy as well as a multi-layered approach to linguistic markup. Annotation is thus placed outside the original text (the primary data) while they remain linked to each other by referencing mechanisms. The separate layers are (i) tokens separated by punctuation or whitespace, (ii) morpho-syntactic properties of the tokens, (iii) syntactic constituency, (iv) syntactic dependency, (v) semantic relations. Instances of semantic information layers may be progressively added to the framework, as required. Typical examples of these are noun phrase coreference, named entities, and temporal information; there are several conceptual and notational approaches (especially XML-based standards for annotating a particular layer) that have been developed by the community of researchers working on a particular problem. The combination of stand-off and multi-layered annotation thus also allows adding further (domain-specific) annotation layers, e.g. on narrative functions of fairy tales of other genres. In ongoing work we are considering adopting the best practices defined by the Text Encoding Initiative[4] for the representation of texts in digital format. Our approach to a standardised textual and linguistic representation of fairy tales is described in more detail in (Declerck et al., 2010).

3.2. Infrastructure for Language Resources and Technologies

The CLARIN[5] and D-SPIN[6] projects have been setting up an infrastructure for supporting the use of language resources and tools for e-Humanities. More specifically, D-

[3] http://www.tc37sc4.org

[4] http://www.tei-c.org/index.xml

[5] http://wwwwww.clarin.eu

[6] http://www.sfs.uni-tuebingen.de/dspin

SPIN is assembling NLP tools in a web service architecture, called WebLicht[7], in which end-users can obtain documents annotated at the desired level of linguistic information without having to install the whole linguistic processing chain on their local machines. Our work consists here in adding a web service to WebLicht that annotates fairy tales with Proppian functions on top of linguistic annotation. The WebLicht services for the time being do not support ISO and TEI standards, but the implemented annotation strategy is close enough to our wishes for allowing us to run our experiments on the integration of PftML and linguistic annotation.

3.3. Linguistic annotation steps

When a text is submitted to (a specific configuration of) WebLicht[8], e.g. the one depicted in Figure 1, the first step of the analysis procedure in WebLicht consists in wrapping the input text within a XML encoding called TCF (Text Corpus Format).
Subsequently, one can select and run one from a list of available tokenizers for English, the result of which is – partially – given below:

```
<tns:tokens>
      ...
      <tns:token ID="t34">Look</tns:token>
      <tns:token ID="t35">after</tns:token>
      <tns:token ID="t36">your</tns:token>
      <tns:token ID="t37">brother</tns:token>
      <tns:token ID="t38">!</tns:token>
      <tns:token ID="t39">Do</tns:token>
      <tns:token ID="t40">n't</tns:token>
      <tns:token ID="t41">go</tns:token>
      <tns:token ID="t42">out</tns:token>
      <tns:token ID="t43">of</tns:token>
      <tns:token ID="t44">the</tns:token>
      <tns:token ID="t45">yard</tns:token>
      <tns:token ID="t46">,</tns:token>
      ...
      ...
      <tns:token ID="t71">the</tns:token>
      <tns:token ID="t72">daughter</tns:token>
      <tns:token ID="t73">soon</tns:token>
      <tns:token ID="t74">enough</tns:token>
      <tns:token ID="t75">forgot</tns:token>
      <tns:token ID="t76">what</tns:token>
      <tns:token ID="t77">they</tns:token>
      <tns:token ID="t78">had</tns:token>
      <tns:token ID="t79">told</tns:token>
      <tns:token ID="t80">her</tns:token>
      <tns:token ID="t81">.</tns:token>
      ...
</tns:tokens>
```

The link to primary data is ensured by the attribute `ID`, pointing to the location of tokens where they occur in the text, according to the philosophy of stand-off annotation. We reproduced the original strings in the example for the sake of readability.
On the tokenized text one can next run a lemmatizer and a part-of-speech (POS) tagger:

```
<tns:lemmas>
....
      <tns:lemma tokID="t71">the</tns:lemma>
      <tns:lemma tokID="t72">daughter</tns:lemma>
      <tns:lemma tokID="t73">soon</tns:lemma>
      <tns:lemma tokID="t74">enough</tns:lemma>
      <tns:lemma tokID="t75">forget</tns:lemma>
      <tns:lemma tokID="t76">what</tns:lemma>
      <tns:lemma tokID="t77">they</tns:lemma>
      <tns:lemma tokID="t78">have</tns:lemma>
      <tns:lemma tokID="t79">tell</tns:lemma>
      <tns:lemma tokID="t80">her</tns:lemma>
      <tns:lemma tokID="t81">.</tns:lemma>
...
 </tns:lemmas>

 <tns:POStags tagset="PennTB">
...
      <tns:tag tokID="t70">CC</tns:tag>
      <tns:tag tokID="t71">DT</tns:tag>
      <tns:tag tokID="t72">NN</tns:tag>
      <tns:tag tokID="t73">RB</tns:tag>
      <tns:tag tokID="t74">RB</tns:tag>
      <tns:tag tokID="t75">VBN</tns:tag>
      <tns:tag tokID="t76">WP</tns:tag>
      <tns:tag tokID="t77">PP</tns:tag>
      <tns:tag tokID="t78">VBD</tns:tag>
      <tns:tag tokID="t79">VBN</tns:tag>
      <tns:tag tokID="t80">PP</tns:tag>
      <tns:tag tokID="t81">.</tns:tag>
...
</tns:POStags>
```

These two annotation layers refer to the tokens via the feature `tokID`, and thus add to them lemma and part-of-speech information. Constituency and dependency information is obtained and linked after running a parser on the most recent levels of annotation. Below we show the syntactic annotation of two fragments of the text: *the daughter soon enough forgot ... and ran into the yard*.

```
 ...
    <tns:constituent cat="S/fin">
     <tns:constituent cat="NP-SBJ/base">
       <tns:constituent cat="DT/the">
     <tns:tokenRef tokID="t71"/>
       </tns:constituent>
        <tns:constituent cat="NN">
         <tns:tokenRef tokID="t72"/>
        </tns:constituent>
      </tns:constituent>
     <tns:constituent cat="VP/fin">
       <tns:constituent cat="ADVP-MNR/V">
          <tns:constituent cat="RB/ADV">
            <tns:tokenRef tokID="t73"/>
          </tns:constituent>
         <tns:constituent cat="RB/mnr/ADV">
          <tns:tokenRef tokID="t74"/>
          </tns:constituent>
       </tns:constituent>
       <tns:constituent cat="VVD/n">
         <tns:tokenRef tokID="t75"/>
       </tns:constituent>
...
    <tns:constituent cat="CC">
      <tns:tokenRef tokID="t93"/>
    </tns:constituent>
    <tns:constituent cat="VP/fin">
      <tns:constituent cat="VVD/p">
        <tns:tokenRef tokID="t94"/>
      </tns:constituent>
      <tns:constituent cat="PP-DIR/V">
        <tns:constituent cat="IN/into">
          <tns:tokenRef tokID="t95"/>
        </tns:constituent>
        <tns:constituent cat="NP/base">
          <tns:constituent cat="DT/the">
            <tns:tokenRef tokID="t96"/>
          </tns:constituent>
          <tns:constituent cat="NN">
            <tns:tokenRef tokID="t97"/>
          </tns:constituent>
        </tns:constituent>
...
</tns:parsing>
```

The words in this annotation layer are grouped into syntactic categories (e.g. the nominal phrase *the daughter*). The tagset in use here in fact mixes constituency and dependency information, for example the tag `NP-SUBJ` (associated with the string *the daughter*) indicates that this NP has the grammatical subject role in the sentence. Items belong-

[7] http://weblicht.sfs.uni-tuebingen.de/englisch/weblicht.shtml

[8] the use of the WebLicht services is for the time being password protected

ing to a phrase are referred to by using the same `tokID` feature as in the case of lemma and POS annotation.
In our approach, this is the kind of linguistic annotation we consider as the basis for integrating the PftML type of annotation. We have now at our disposal linguistic information associated to both words and phrases, as well as information about the linguistic relations between words and phrases (e.g. the NP *the daughter* being the Subject of the predicate *forget*). Current work is being pursued in mapping the output of WebLicht to the pivot annotation format defined in ISO TC37/SC4 (cf. Section 3.1.).

3.4. Incorporation in PftML

We propose the integration of PftML as an additional layer into this stand-off, multi-layered annotation scheme. The tagset of annotation is for the time being defined in the DTD of PftML[9] as well as by the list of Proppian character roles[10]. Based on ProppOnto and the methods discussed below in Section 3.5., we are going to be able to improve the organization of labels, given such properties as e.g. referring to an entity or to an event, additionally taking into account the complexity of events. For the time being we manually associate with e.g. the function "Violation of an execution" verbs such as *forget*, or semantically similar ones that suggest that such a violation may occur, drawing on the DTD of PftML.
The dependency structure depicted by the linguistic annotation allows the identification of the agent of the violation (here: *the daughter*) and its object (i.e. "what they have told her", referring back to the command *Look after your brother! Don't go out of the yard ...*). The cross-reference between these two events described in the text is made easier by the fact that the sentences have been annotated respectively as *Command* and *Violation*, while the DTD of PftML formalizes that a command is followed by its execution or violation. (Obviously, ProppOnto captures this constraint as well, cf. the bottom section in Figure 2.)
The integration step of the linguistic layers and PftML is straightforward: PftML and word-level annotation are combined in one XML element, thus the specific PftML annotation obtains a precise span of textual segments associated to it.

```
<Execution subtype="Violated" id="Violated1" inv_id=
"Command1" from="t93" to="t98">
</Execution>

<Command subtype="Interdiction" id="Command1" inv_id=
"Violated1" from="t39" to="t46">
</Execution>
```

`t39`, `t46`, and `t93`, `t98` mark respectively the regions in the text for which the different functions hold. The value of `inv_id` refers to the related function label. In our example: the violating execution refers back to the function label *Interdiction* with the id `Command1`. Navigating through different types of IDs of the multi-layered annotation the user can extract all kinds of information, e.g. about the grammatical subject, the main verb and its ground form (i.e. lemma), etc., which are associated with a Proppian function.

[9] cf. http://clover.slavic.pitt.edu/sam/propp/praxis/pftml2.html
[10] cf. http://clover.slavic.pitt.edu/sam/propp/praxis/features.html

3.5. Integration with ProppOnto

When annotating a corpus based on the set of concepts held by a particular ontology, it is usually desirable that all labels at a particular level of annotation come from the same level of abstraction. This requires identifying which of the concepts appearing in an ontology are to be used for annotation. For instance, ProppOnto includes a taxonomy of Proppian functions which describes them at several levels (based on Propp's own work, listing a number of detailed functions but also classifies them into more generic types). We should specify the level we are interested in, or whether we may want different layers of information in our representation format, each documenting the narrative structure at a different level of abstraction. Alternatively, one might just annotate at the lowest possible level (most specific), and assume that anyone requiring information about higher levels can obtain it by cross-referring between the annotated text and the ontology.
As said in Section 2.2., there is no information available for users of ProppOnto on e.g. why the exactly certain spans of words are to be associated with certain functions. We suggest that this can be remedied if both a terminological layer and a linguistic layer are introduced to the ontology, where each of these separate layers are combined with the class and relation hierarchies, along the lines proposed by (Reymonet et al., 2009) and (Buitelaar et al., 2009), respectively.
The terminology layer will list typical expressions such as *don't go out*, as a term to be associated with the *Interdiction* function, whereas the linguistic layer will encode the linguistic properties of terms, in this case (in simplified form): `NEG-PART + Verb[present_tense] PrepPhrase (directional).` The *Violation* example would be encoded as `NP[pers] Verb[past_tense] PrepPhrase (directional).` Via the linguistic descriptions one can (automatically) link all the terms that correspond to the linguistic objects, in line with (Buitelaar and Declerck, 2003).
Based on the linguistically enriched resources we can obtain higher-quality input material for text planning in the NLG component, since the scheme facilitates the harvesting of lexical and morpho-syntactic information, together with syntactic structures.

3.6. Improvement in e-Humanities scenarios

We think that the integration we describe can support eHumanities researchers in many ways. With this kind of annotation they can retrieve typical expressions associated with a Proppian function, not restricted to word forms, but extended to the underlying lexical information ("give me all verbs in past tense"). This can be relevant, since e.g. often *Interdiction* is mentioned using present tense, but *Violation* is reported using past tense.
One can also retrieve and verify relational information (e.g. the grammatical subject and object of an event) and semantic relations across sentences (the patient of an *Interdiction* is most of the time agent of its *Violation*, etc.). The establishment of textual statistics can thus be greatly enriched on the base of linguistic annotation. All this type of in-

formation can also be compiled in a template-like format and used for enriching or enhancing the semantic model of ProppOnto.

Furthermore, linguistic information will enable detecting functions that refer to each other, as syntax and semantics of sentence pairs in such relations mirror – at least partly– each other, e.g. in *Don't go out of the yard* and *ran onto the street*. Detecting such cross-reference will possibly facilitate the identification of linguistic units belonging to a function's core (i.e. the strings above) as opposed to its periphery (e.g. everything else within the boundaries of the chunk annotated with the given function, i.e. *be a good girl, and we'll buy you a handkerchief*, respectively *where she played and got completely carried away having fun*). Collecting instances of the surface representation of functions, on various levels of granularity, is a crucial step in NLP and NLG, for understanding and representing the linguistic vehicles by which motifs ad narration operates, and the degree of variation and optionality they allow.

By performing higher-level text analytics (i.e., in search of correlations between linguistic and domain objects) we expect to observe phenomena that are not addressed by Propp's model. For instance, we could identify elements that would in effect be relevant (or necessary, for completeness' sake in a structured model) to add as functions, because in the corpus we find explicit evidence for it. Note that *FollowedCommand* is a Proppian function (cf. Figure 2 for its ontological representation), whereas *Followed Interdiction* is not, despite that it might be important to mark up explicitly those parts of the story where interdiction was still not violated. This would also prevent certain passages of texts from remaining unannotated or erroneously annotated, such as *The father and mother went off to work*, which is currently marked up as *<Execution subtype="Violated">* (cf. Figure 1). We thus additionally foresee ways in which our method can deliver the machinery in order to extend, refine, or verify Propp's scheme and the models built upon it.

4. Concluding remarks

In our contribution we propose incorporating linguistic information in semantic resources of the cultural heritage domain. There are computational resources relevant for humanities research, however, these currently do not include linguistic annotation. We demonstrate an approach to enrich PftML and ProppOnto with linguistic markup. In the present contribution we technically focus on PftML; it is nevertheless also outlined how the method would work for ProppOnto.

The proposed integration not only enables improved indexing, retrieval, and semantic markup of folk narratives corpora, in addition, this research line may open up possibilities in Humanities research, whereby scientific hypotheses, such as the composition and mechanism of narrative motifs or other higher-order cognitive phenomena can be better investigated.

We continue working on how to reach the integration to the full range of both PftML and ProppOnto, although already in the current study showed working examples for PftML and working models to be adapted to ProppOnto. As a major resource to be generated out of the current approach, we expect to create a corpus of folklore texts annotated on multiple layers of varying granularity, reflecting language-, domain-, and culture-specific perspectives of the folk narrative genre. Since our method is language independent, we can directly test it on (parallel) fairy tales corpora in several languages; we have started to explore Russian, Hungarian, Spanish, German, and English. This will enable the creation of a matrix of linguistic and domain-specific objects and will bring novel insights into, as well as create resources both for comparative analysis and systematic evaluation of conceptual structures in folk narratives.

5. References

A.A. Aarne. 1961. *The types of the folktale; a classification and bibliography*. Helsinki, Academia Scientarum Fennica. (Antti Aarne's Verzeichnis der Märchentypen (FF communications no. 3) translated and enl. by Stith Thompson).

A.N. Afanas'ev. 1957. *Russkie Narodnye Skazki (Russian Folk Tales)*. Moscow: Gosudarstvennoe Izdatel'stvo Khudozhestvennoi Literatury.

P. Buitelaar and T. Declerck, 2003. *Linguistic Annotation for the Semantic Web*, pages 93–111. IOS Press.

P. Buitelaar, P. Cimiano, P. Haase, and M. Sintek. 2009. Towards linguistically grounded ontologies. In *Procs. of European Semantic Web Conference*.

T. Declerck, K. Eckart, P. Lendvai, L. Romary, and T. Zastrow. 2010. Standardized linguistic annotation of fairy tales. In *7th International Conference on Language Resources and Evaluation, Procs. of the Workshop on Language Resource and Language Technology Standards*.

S. A. Malec. 2001. Proppian structural analysis and XML modeling. In *Procs. of CLiP Conference*.

F. Peinado, P. Gervás, and B. Díaz-Agudo. 2004. A description logic ontology for fairy tale generation. In *4th International Conference on Language Resources and Evaluation, Procs. of the Workshop on Language Resources for Linguistic Creativity. pp 56-61.*

V. J. Propp. 1968. *Morphology of the folktale*. University of Texas Press: Austin. (Transl. L. Scott and L. A. Wagner).

A. Reymonet, J. Thomas, and N. Aussenac-Gilles. 2009. Ontology based information retrieval: an application to automotive diagnosis. In *Procs. of International Workshop on Principles of Diagnosis*.