GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

ISSN 1612-6793

Master's Thesis
submitted in partial fulfilment of the
requirements for the course "Applied Computer Science"

An Ontology-Driven Information System based on Vladimir Propp's Morphology of the Folktale for Southern African Folktales

Franziska Pannach

Göttingen Centre for Digital Humanities

Master's Thesis
of the Center for Computational Sciences
at the Georg-August-Universität Göttingen

18. March 2019

Georg-August-Universität Göttingen
Institute of Computer Science

Goldschmidtstraße 7
37077 Göttingen
Germany

☎ +49 (551) 39-172000
FAX +49 (551) 39-14403
✉ office@informatik.uni-goettingen.de
🌍 www.informatik.uni-goettingen.de

First Supervisor:	Prof. Dr. Caroline Sporleder
Second Supervisor:	Prof. Dr. Wolfgang May

I hereby declare that I have written this thesis independently without any help from others and without the use of documents or aids other than those stated. I have mentioned all used sources and cited them correctly according to established academic citation rules.

Göttingen, 18. March 2019

[illegible]

Abstract

Our ontology has been carefully modelled using a description logic foundation that defines the interactions between characters in a tale and Proppian functions. It was subsequently implemented in OWL (Web Ontology Language). In contrast to pre-existing work (Peinado et al., 2004), this ontology not only represents Proppian functions as a taxonomy of concepts but also illustrates the binary relations that represent the actions defined in those functions.

To model the strict order of functions, we introduced the functional relations FollowedBy(fn,fn) and PrecededBy(fn,fn). A small set of data properties was defined to allow representing the functions, tales and publications within the annotation. We intentionally limited the number of data properties to ensure that the ontology remains a useful annotation tool. Our ontology also models concepts that illustrate the family relationship of characters in a story (Koleva et al., 2012). While these additional concepts and relationships are only based on Propp in the sense that they appear in the functions, they are nevertheless crucial elements for the analysis of the tales.

For test purposes, twenty African tales in their English translations (Smith, 1989), (Greaves, 1990), (Scheub, 2005) were annotated by four different annotators. Although only one of the annotators had previous annotation experience, only limited assistance was needed for populating the ontology, which indicates that the system is easy to use.

An ontology-driven information system presented as a web application allows performing SPARQL queries to investigate the annotated tales. We found a strong prevalence towards tale endings that do not imply a traditional reward situation, but rather the restoration of the situation in the beginning of the tale, e.g. the rescue of the victim .

Contents

Chapter 1

Introduction

Folk and Fairy tales are a substantial part of oral folklore. They play an important role in the cultural heritage of regions, nations or cultural minorities. In European context, fairy tales have been collected and edited by the Grimm brother's in the beginning of the 19th century (Grimm and Grimm, 1857) while their Russian counterpart Alexander Afanasyev collected more than 450 folktales of Russian and Slavic origin (Afanasyev, 1916). Afanasyev's tale collection later became the foundation of Vladimir Propp's theory *Morphology of the Folk tale* (Propp, 1968).

While the collection of European tales altered the transmission of fairy tales, from mainly oral tradition towards reading and retelling variants of those collected tales, traditional oral transmission of folk tales exists up to this day in the African context. Presumably, African folk tales are therefore different from European fairy tales in terms of structure and motifs. This project aims to construct an Ontology Driven Information System for African folk tales following the morphological approach of Russian folklorist Vladimir Propp.

The goal is to collect and analyse African folk tales, but also investigate how they follow Propp's formalism and how motifs and agents are verbalised. Hence, the ontology is going to contain:

- The Proppian functions and entities encoded within.
- Specific folk tale motifs from the Thompson-Motif-Index[1] (TMI) and tale type classes according to the Aarne-Thompson-Uther Index[2] (ATU) (see Section 2.2.2).
- The representation of the functions and motifs in selected African Folktales.

[1]https://sites.ualberta.ca/~urban/Projects/English/Motif_Index.htm
[2]http://www.mftd.org/index.php?action=atu

1.1 Scope of the Project

Due to the strong formality of Propp's morphological approach, as discussed in Section 2.2.1, it is well fitted to be represented as an ontology.

This project aims to build an ontology based on preliminary work. (Abbas et al., 2018) The ontology should connect to related ontologies, such as Declerck's work on a similar ontology modelling Propp's theory (see Section 2.2.1) that used a different approach to model the same formalism. (Declerck et al., 2017a) It also aims to enrich the existing ontology by including multi-lingual descriptions of Proppian functions.

The system provides means to access and compare existing Proppian analyses of African tales. It will not, however, provide access to the full text of the tales. The first factor leading to this decision is that published tales mostly underly copyright restrictions. Secondly, following Alan Dundes speech "Folkloristics in the Twenty-First century" (Dundes, 2005), modern folkloristics should focus more on the analysis than on the collection of tales.

> "The stereotype of folklorists as simply collectors, obsessive classifiers, and archivists is strengthened each and every time yet another collection of unanalyzed folklore is published."

(Dundes, 2005, p.391)

To demonstrate how a thoroughly modelled ontology in combination with natural language processing approaches can provide a semi-automatic population of the ontology, an information extration component for folktale characters and Proppian functions has been added. This module should be seen as a proof of concept rather than an ideal tool for extracting information from folk tale texts.

Where needed, manual annotation of folk tale texts will be used to populate the ontology further.

The final goal of this project is to build an ontology-driven information system that is able to answer questions about the morphology of Southern African tales, and investigate the role of agents and the appearance of folktale motifs. The system is presented as a web application that allows users to query the ontology as well as to add their own folk tale annotations.

1.2 Structure of the Thesis

The domain for which the Ontology-Driven Information System is modelled is described in Chapter 2. It explains the theory *Morphology of the Folk Tale* by Vladimir Propp (Propp, 1968), provides information about Southern African folk tales on which Propp's theory will be applied, and gives examples where this application was made in analogue approaches. In Chapter 3 the

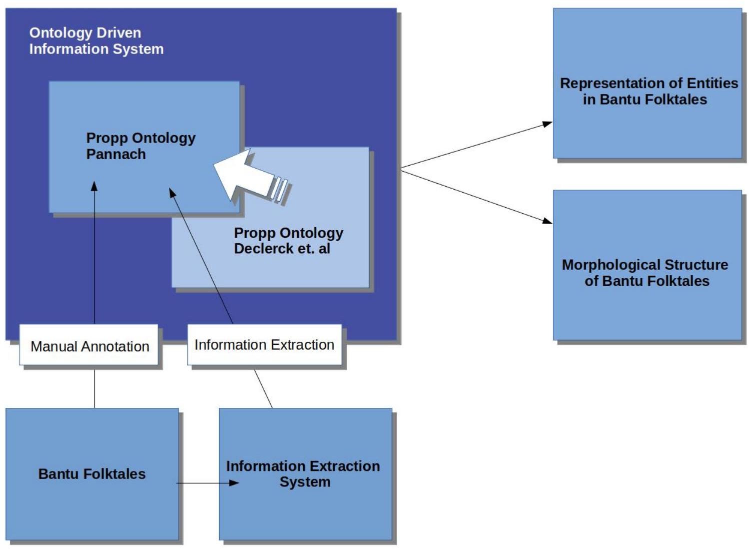

Ontology Driven Information System
Propp Ontology Pannach
Propp Ontology Declerck et. al
Manual Annotation
Information Extraction
Bantu Folktales
Information Extraction System
Representation of Entities in Bantu Folktales
Morphological Structure of Bantu Folktales

Figure 1.1: Visualisation of the Project Scope

implementation of the Ontology-Driven Information System is explained. Chapter 4 describes in detail how the ontology was defined and built. In Chapter 5, the results of the project with regard to competency questions and case studies on Southern African folktales are reported. Finally, in Chapter 6, the results are discussed, limitations of the system are evaluated and an outview regarding further work is given.

Chapter 2

Fundamentals

2.1 Ontology-Driven Information Systems

The goal of this project was to develop an ontology-driven information system that allows users to investigate the morphology of Southern African folktales.

Yildiz and Miksch define such a system as:

> "An ontology-driven information system, [...], is a system where the ontology is yet another component of the system that co-operates with other components of the IS at run time."(Yildiz and Miksch, 2007, p.36)

In our case, the ontology-driven information system is presented as a web application. It yields information accessible by user queries, but also supplies data for internal use through the Flask[1] (see Section 2.3.8) implementation. For this project, the ontology itself builds the core of the information system.

2.2 Domain

2.2.1 V. Propp's *Morphology of the Folktale*

The Russian folklorist Vladimir Propp introduced 31 invariant functions describing the morphology of the Russian magic folk tale. In his groundbreaking 1928 work *Morphology of the Folk tale*, he argues that the narrative of folk tales always follows the same pattern. The narrative functions, such as *XXXI Wedding W*, and the *Dramatis Personae* (agents in the story), such as the *Hero*, or

[1]http://flask.pocoo.org/

Villain, introduced by Propp are strictly defined and specify recurrent units from which the tales are constructed. Propp set four axioms:

1. "Functions of characters serve as stable, constant elements in a tale, independent of how and by whom they are fulfilled. They constitute the fundamental components of a tale.
2. The number of functions known to the fairy tale is limited.
3. The sequence of functions is always identical.
4. All fairy tales are of one type in regard to their structure."

(Propp, 1968, p.21-23)

The high formality of this structuralism allows something as complex and highly emotive as the folk tale to be represented in the form of rigid patterns. Thus, they can be further used in automatically processing or when generating new tales. In Computational Linguistics, Propp's functions are used in various ways, such as for automatic markup, classification and annotation (Malec, 2010), or as a foundation of an independent XML dialect (Malec, 2010), (Lendvai et al., 2010). His approach is still widely used not only in folk tale research but also applied to contemporary work such as the Star Wars Trilogy[2].

The Functions of the Dramatis Personae

Vladimir Propp defined a set of 31 invariant functions to describe the morphology of the folk tale. A list of these functions can be found in Appendix A.1. Most of these functions can occur as variants, which are defined as subfunctions of the broader function. For example, the function *Absentation* β can have the variants *Absentation of elders* $\beta 1$, *Death of parents* $\beta 2$, or *Absentation of younger people* $\beta 3$.

Proppian functions can appear in the tale as they are, with modifiers or they can be inverted, e.g. *Hero leaves home* $\rightarrow$ *Hero does not leave home* (explicitely).

The functions apply to seven types of characters: hero, (seeker hero), false hero, villain, victim helper, donor, dispatcher, princess and princess' father. This list corresponds to seven 'archtypes' because the seeker hero is a subtype of hero, and from a content perspective the princess' and princess' father belong together. One character appearing in a tale can fill multiple roles, such as *Donor* and *Dispatcher*, and not all roles have to be present in every tale. (Propp, 1968)

Some functions appear as pairs, such as *Villainy/Lack A* and *Liquidation of Lack K* as shown in Fig. 2.1. A prominent example of the A-K pair is the princess in the *Frog Prince or Iron Henry*[3] (Grimm and Grimm, 1857) losing her golden ball (*Lack*) and the frog subsequently bringing the ball back (*Liquidation of Lack*).

[2]http://jaced.com/2013/02/06/vladimir-propp-science-of-the-fairy-tale/
[3]https://en.wikisource.org/wiki/Grimm%27s_Household_Tales_(Edwardes)/The_Frog-Prince

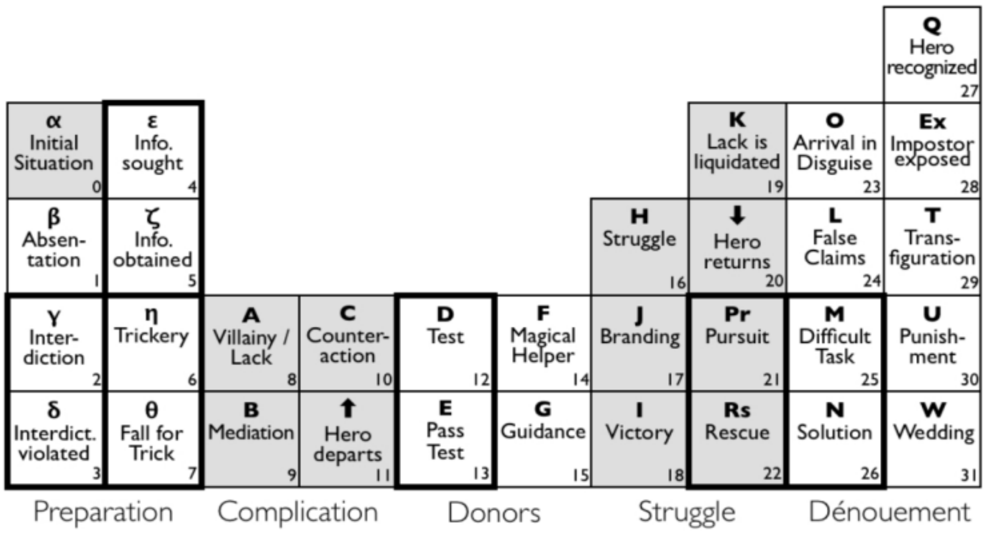


Figure 2.1: Schematic overview of Proppian functions. Bold lines represent function pairs, numbers on the lower right represent the order in which the functions appear. Contrasting colors indicate neighboring functions belonging to the same group. Source: (Scheidel, 2010, p.4)

A story consists of one or more sets of functions in sequential order, called moves. While one tale can constist of multiple moves, the functions forming the move have to follow the order as presented in Appendix A.1. Propp clusters functions into five contentual segments: Preparation, Complication, Functions of the Donor, Struggle and Dénouement.

Relations between Proppian functions

In addition to function pairs within the 31 upper function classes, subclasses can also be connected to each other. Especially for the classes *First function of the Donor F* and *Receipt of Magical Agent D*, possible connections have been defined (Propp, 1968). The following example illustrates these connections between Proppian functions:

> In the fairy tale of the Golden Goose (Grimm and Grimm, 1857), the *Hero* Simpleton meets an on old man (*Donor*). After an initial greeting, the old man asks for food and drink from Simpleton's proviant (*D7 Other requests*). Simpleton shares his sour beer and *Aschekuchen*[4] with the old man (*E7 Hero performs some other service*). The donor rewards the good deed by pointing out a tree to Simpleton, advising him to cut it down in order to find an item which he does not specify further (*F2 Agent is pointed out*). After logging the tree down, Simpleton finds the Golden Goose (*Magical Agent*) within the roots.

Propp stressed the importance of the *Donor* in the Russian magic tale. (Propp, 1968). He defined

[4]biscuit that has been made from dough fallen into the ashes of the hearth

which functions of the donor correspond to which kind of transferrance of the magical agent, as shown in Fig. 2.2.

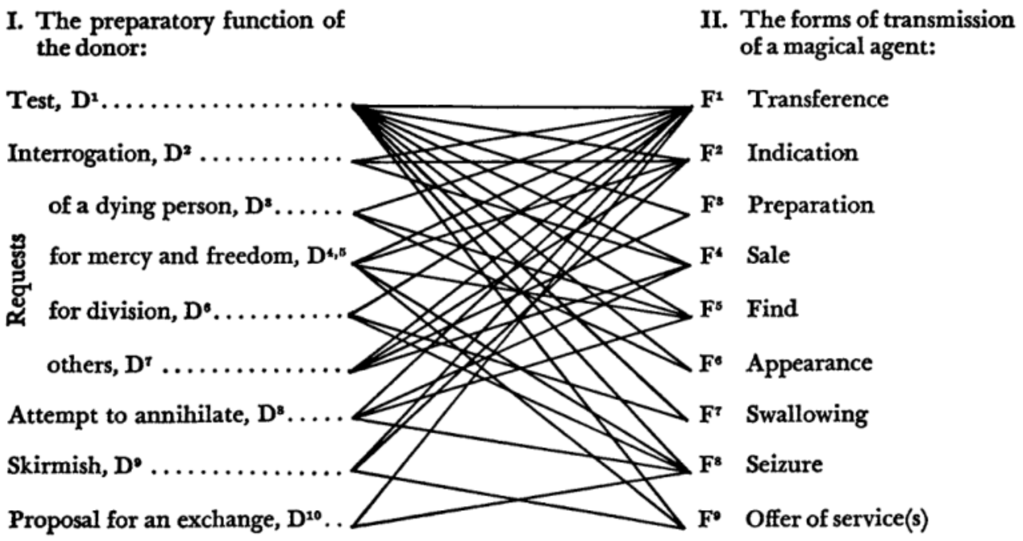


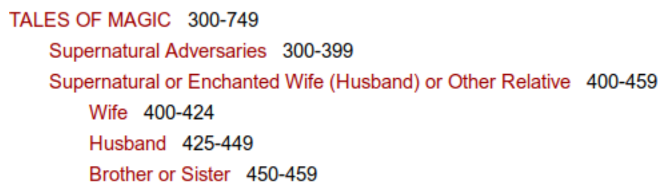
Figure 2.2: Connection between *First Function of the Donor* and *Receipt of Magical Agent* Source: (Propp, 1968)

2.2.2 Folktale Motif Indices

Two common indices are applied for folk and fairy tale classification and analysis. The Aarne-Thompson-Uther index (ATU)[5] is used to classify a tale into exactly one class, the tale type. The forementioned tale of the *Frog Prince or Iron Henry* (Grimm and Grimm, 1857) falls into the ATU class 440 - The Frog King, which belongs into the broader category Magic Tales, as shown in Fig. 2.3. Type classes are relatively wide, describing the main story line of the tale. Therefore, each tale can only have one ATU type. Tale types also indicate the relation of tales that belong to the same class. Crowley states that, "Two occurrences of the same type are considered to have a common origin." (Crowley, 1969, p.127-128) The Aarne-Thompson-Index was first published in 1910, revised by Hansjörg Uther and republished as ATU-Index in 2004.

In contrast, the Thompson-Motif-Index (TMI) is more fine grained, describing single motifs, i.e. "recognizable object[s], character[s], or event[s]" (Crowley, 1969, p.127), such as characters, actions, or numerical patterns. A tale can contain more than one TMI motif.[6], e.g. the *Frog Prince* tale includes motifs such as *B211.7.1 Speaking Frog*, *P40 Princesses*, *P23 Children and Parents*, *P320 Hospitality*, or *D935 Transformation: Frog to Person*.

[5]`http://www.mftd.org/`
[6]`https://sites.ualberta.ca/urban/Projects/English/Motif_Index.htm`



TALES OF MAGIC 300-749
- Supernatural Adversaries 300-399
- Supernatural or Enchanted Wife (Husband) or Other Relative 400-459
  - Wife 400-424
  - Husband 425-449
  - Brother or Sister 450-459

Figure 2.3: ATU Type of the Frog Prince

The TMI was first published in 1932. (Dundes, 2005) Both the TMI motifs and the ATU types are organized in a hierarchical structure.

2.2.3 Southern African Tales

Social and Scholarly Significance of African Tales

The collection of African folk tales has a long history. Western folklorists compiled anthologies of tales from colonialised country from the beginning of the 19th century. The first African tale collection was published by Le Bon Roger in 1828.[7] (Crowley, 1969) In contrast to the European tale collections, like Grimm's *Kinder- und Hausmärchen*, these African collections did not influence the oral transmission of folk tales as much. Harold Scheub describes the relevance of tales in African oral culture as follows:

"They said that tales were as common and routine as washing cloths, as working in the fields as raising children. And I said that that was precisely why I was interested in gathering tales." (Scheub, 2010, p.104)

Regardless of cultural background, telling folktales to children or adults is far more than a leisure activity, but contributes to various aspects of socio-cultural life. Laurence M. Porter stresses the social importance of tales by stating:

"All tales, in short, serve some social function beyond that of entertainment, be it educational task or emotional maintenance." (Porter, 1995, p.237)

The analysis of oral African folklore also plays an important role in investigating how cultures defied colonialism. For example, Harold Scheub describes the importance of folktales during the apartheid regime as follows:

> "On the surface, these stories seem to be typical fairy tale-type narratives, with the usual folkloric motifs, the ogres and the fantasy helpers, the Cinderella characters moving up from their status as least likely heroes. And that is the way many of those in the white ruling class of apartheid South Africa viewed these stories. They were

[7] Le Bon Roger, "Fables Sénégalaises Recueillies de l'Oulof".

> activities of the subjugated Africans and were not to be taken seriously, certainly not as weapons in the antiapartheid struggle.” (Scheub, 2010, p.xii-xiii)

Scheub further stresses the significance of folktales that withstood 350 years of apartheid as a “testament to the power and lasting effects of storytelling.” (Scheub, 2010, p.xiii)

Motifs in Southern African Tales

Alexander McCall Smith elaborates that African tales share common motifs with folk tales from other cultures, such as the reward of moral integrity, exposing jealousy and selfishness, punishment of curiosity, or treachery. (Smith, 1989, p.1) However, Smith defines some characteristic motifs for African tales, such as the act of singing, which might be derived from performance aspects that found their way into written variants of a tale, sharing of food and other important aspects of “communal social values”, or childlessness as a distinctive motif for misfortune in African tales. (Smith, 1989, p.2-3)

Scheub claims that “the oral stories and poems of the San and Nguni peoples of southern Africa are essentially constructed around transformation patterns.” (Scheub, 2010, p.19)

With regard to the Merwa people in Tanzania, Uta Reuster-Jahn analysed folk tale endings, conluding that they essentially lack reward situations as typical for European tales, but result in a gain in morality of an entire community. The raise in communal moral should be regarded as the reward in the African tale, even if the tale ends with an unfortunate event for the main character, such as his or her death. (Reuster-Jahn, 2002)

This analysis of tale endings goes in line with Scheub’s transformation hypothesis, as the communal advancement could be seen as the ultimate transformation.

As storytelling transitions into the 21st century, modern motifs find their way into the tales. Singers and storytellers, “speak unselfconsciously of computers and jet planes." (Porter, 1995, p.237) This indicates, that modern African tales are constantly subject to change and adapt to new socio-cultural phenomena, like technological advancement.

In the results section of this thesis (Chapter 5), we will investigate if our system can reconfirm these qualities with regard to the tales that were studied in the context of this thesis.

Transmission and Translation of Southern African Tales

For the scope of this project, the focus lies on English translation of folk and fairy tales that were originally told in Bantu languages, such as Zulu, Xhosa, or Ndebele. However, translations of oral folkloric material need to be interpreted as its own source from which only limited interpretations regarding the original material can be inferred.

In the context of West African tales, Laurence M. Porter stated that when an oral tale has been derived "from a culture that has been overwhelmed", translations can be used as means to "justify to members of the dominant culture the enduring importance of the surviving remnents of the submerged culture." (Porter, 1995, p.229) In order to achieve this justification, Porter claims that the tales undergo "a process of compromise and modification that renders the cultural object less alien and exotic." (Porter, 1995, p.229)

Therefore, there is need to stress that the material we are working with, especially in Chapter 5, is at least twice distorted from the original. The first change took place during the transition from oral tradition to written texts while the tales were modified a second time due to the translation from the original language to English. Daniel J. Crowley goes as far as to declare, he has "little faith that any important stylistic elements can be preserved in translating an African language into a European one." (Crowley, 1969, p.123) Additionally, even when a tale is told by the same storyteller who retells a story in a different environment, e.g. at home and in school, a "transition from rural to urban culture" (Porter, 1995, p.231) can become apparent.

As Porter stresses, "[t]o forget this fundamental principle leads to an artificial, sanitized vision of culture." (Porter, 1995, p.229)

However, when we discuss the outcome of this project in Chapter 5, we will show how the approach we followed with the ontology-driven information system can help adressing this particular problem.

With the translation and readaption of the tales, folklore studies were able to do extensive research on motifs and narrative patterns. However, in the context of Digital Humanities, digital folk and fairy tale studies – especially with regard to African folktales – is still rare. Traditional folklore studies still rely on hardcopied books as their main instrument of scholarly communication. Commonly, African tale collections are published in first world countries, making them – for the most parts – unavailable to interested non-scholarly readers in the country where the tales were collected.

The ontology-driven information system will be publicly available. Furthermore, it was designed in such a way that users with different levels of digital proficiency are able to use it and access the data. Following the principle of Open Science, we believe that research findings need to be publicly available for all, especially for those people who provided the data – in this case the native South African people.

Moreover, tale telling is mainly a verbal art (Crowley, 1969). When a story undergoes the transition from oral to written, performance aspects such as onomatopoeia, are lost, unless they are actively preserved either by linguistic cues or – in a digital context – by extensive use of metadata. Due to a lack of "details of performance and ethnography, virtually all this material of the past is good for one thing only, the study of the diffusion of tales types and motifs." (Crowley, 1969, p.119). Modelling verbal art in an ontologically modelled context, such as the presented system, can be a way of preserving performance features alongside their textual interpretations.

2.2.4 Propp and African Folktales

Structural analysis of African folktales and the applicability of Proppian functions to them has not been introduced without critical side-eyeing. In 1969, a year after the *Morphology of the Folk Tale* was published in its second, revised edition (Propp, 1968), Daniel J. Crowley found clear words for the Proppian approach in the context of African folklore:

> "If Propp's thirty-one "functions" can be shown to happen in the sequence he believes, or that this is significant, so much the better, but it hardly seems likely at this reading. If truly comparable units could be found without doing too much violence to the variant nature of tales, we could indeed solve many problems that have long been with us. But first the structuralists must prove their case, and elucidate their method much more clearly than they have been able to do so far. Otherwise I feel sure I speak for the vast majority of professional folklorists when I say that we will in all likelihood not pursue this direction."

(Crowley, 1969, p.130)

Since then, studies that investigate the fitness of Propp's theory for African Tales have been conducted. A prominent example is the "Morphology of the Igbo Folktale" by Chukwuma Azuonye (Azuonye, 1990). Since the Igbo are a group of people in Nigeria, their tales strictly speaking do not fall into the scope of this project. However, Azuonye's work has been included into the ontology to show some of the advantages and limitations of our system.

Azuonye published a morphological analysis of the *Obaraedo* tale in 1990. (Azuonye, 1990) The same tale has been analysed by Ikechukwu Okodo in 2012. (Okodo, 2012) Both analyses explain how Propp's functions are represented in the *Obaraedo* tale. While Azuonye did not provide the full text of the tale, Okodo included the text translated to English in his article. From the explanations Azuonye gave on how the Proppian functions appear in the untranslated text, it becomes apparent that both of them worked on the same tale.

However, Azuonye's findings regarding the structure of the tale are significantly different from Okodo's analysis. Due to Propp's formalistic approach, we can easily compare both findings. Azuonye defines the function sequence of the *Obaraedo* tale as

$$\alpha\beta\gamma\delta\theta aBC \uparrow HIK \downarrow T. \tag{2.1}$$

While Okodo defines the sequence as

$$\beta\gamma\delta\epsilon\eta\theta aBC \uparrow FGHIK \downarrow \tag{2.2}$$

The first difference that we encounter when we compare both analyses is that the *initial situation* α does not appear in equation 2.2. The initial situation is often omitted since according to Propp it should not be regarded as an own function, but as "an important morphological element", which rather introduces the hero and the circumstances in which the tale takes place. (Propp, 1968, p.25)

Another apparent difference between the two function sequences is the appearance of the function *Transfiguration T* at the end of 2.1. Azuonye sees the function fit to describe a moral transformation undergone by the children in the tale, who after the evil spirit is defeated, "obey their parents without question." (Azuonye, 1990, p.41) However, Propp clearly described the function as *Transfiguration*, including the function description "The hero is given a new appearance." (Propp, 1968, p.62) Not only does the function imply a comparatively strict focus on physical appearance, it is also clearly directed towards the hero. Therefore, we believe in Azuonye's case, the function was not assigned correctly. However, the ontology-driven information system allows to compare different analyses of the same tale, therefore holds potential to spark scientific discourse, providing a platform for different interpretations of Proppian functions.

In equation 2.2, the preparatory functions include the tuple *Reconnaissance* ϵ, and *Trickery* η while in equation 2.1 *Violation* δ is directly followed by *Complicity* θ.

Okodo argues that the *Reconnaissance* function appears in the tale, since the evil spirit locates the victim, which in the text is encoded as "The spirit was coming close to her." (Okodo, 2012, p.104) Azuonye (Azuonye, 1990) sees the meeting as a consequence of the violated interdiction "The woman warned Obaraedo not to go out onto the frontage of their compound in the afternoon because a certain spirit that paraded the frontage of their house attacked anybody he met by making the person lose his or her nose" (Okodo, 2012, p.103), and the girl meeting the spirit as part of *Complicity*.

The text does not clearly state that the evil spirit makes any effort to gain information about the victim, which by definition of the *Reconnaissance* function is a key element. Furthermore, in Propp's examples, the reconnaissance takes place through questioning, e.g. "Who will tell me what has become of the tsar's children? Where did they disappear to?" (Propp, 1968, p.28)

As for the *Trickery* function appearing in equation 2.2, Okodo sees the discourse between the spirit and the girl in which both ask each other to jump as "double actions of trickery." (Okodo, 2012, p.106) Azuonye subsumes this "verbal duel" under the *Complicity* function.(Azuonye, 1990, p.41)

Following the *Departure* function, Okodo finds the *Provision or Receipt of Magical Agent* function F. Propp's description of the tale defines it as "The hero acquires the use of a magical agent." (Propp, 1968, p.43) Therefore, we can assume that Okodo sees the herbalist as the hero of the tale. In that case, Okodo's analysis of the tale is contradicting, since he uses the *Departure* function when Obaraedo's father leaves the village to summon the herbalist. The *Departure* function, however, is designed for the departure of the *hero*. Both Azuonye and Okodo use the functions *Struggle H* and *Victory I*, when the herbalist/dibia, fights the spirit, indicating again that he fulfills the role of the hero in the tale. Additionally, they both define the *Departure* function as departure of the girl's father, but the *Return* function as a function of the herbalist/dibia. In that sense, they are both separating the action described in the functions from the Dramatis Personae who fulfill them. This shows a rather free interpretation of what Propp clearly defined as "The Functions of the Dramatis Personae." (Propp, 1968)

These differences between two folkloric analyses show that the interpretation of Propp's functions is not universal, nor is there only one correct sequence of functions per tale.

The ontology-driven information system allows users to query Proppian functions and their verbalisations, hence, providing them with the neccessary information to form an opinion which of the both analyses is more suitable for the given tale. Furthermore, for this specific case, the author's comments on why a function was chosen were added to the function instance as an rdfs:comment. This way, users can comprehend the authors' reasoning.

The ontology-driven information system can put these existing analyses into context of analyses of tales from other regions, and help compare tales on the basis of Propp's theory. Existing annotations can be accessed easily through SPARQL queries or by accessing the triple search. Therefore, the system allows the study of Proppian morphology interculturally and language-independently which might lead to new findings in folktale research.

2.3 Technical Foundations

2.3.1 RDF/XML

Resource Description Framework (RDF) is a specification for interchanging data[8]. RDF data is stored in triples in the form (resource, relationship, resource) that can be represented by a directed graph where the relationship links represent the edges and the resources represent the nodes. Fig. 2.4 shows an example RDF graph. RDF/XML is a format for the serialisation of RDF graphs as an XML document. [9] Code example 16 shows how the graph in Fig. 2.4 is represented as RDF/XML serialisation.[10]

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:dc="http://purl.org/dc/elements/1.1/"
            xmlns:ex="http://example.org/stuff/1.0/">

  <rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar"
```

[8]https://www.w3.org/RDF/
[9]https://www.w3.org/TR/rdf-syntax-grammar/
[10]Source: https://www.w3.org/TR/rdf-syntax-grammar/#example7

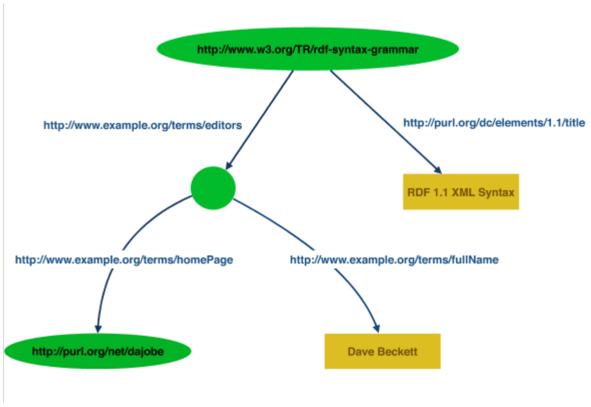

http://www.w3.org/TR/rdf-syntax-grammar
http://www.example.org/terms/editors
http://purl.org/dc/elements/1.1/title
RDF 1.1 XML Syntax
http://www.example.org/terms/homePage
http://www.example.org/terms/fullName
http://purl.org/net/dajobe
Dave Beckett

Figure 2.4: Example RDF graph
Source: `https://www.w3.org/TR/rdf-syntax-grammar/#figure1`

```
             dc:title="RDF1.1 XML Syntax">
    <ex:editor>
      <rdf:Description ex:fullName="Dave Beckett">
        <ex:homePage rdf:resource="http://purl.org/net/dajobe/" />
      </rdf:Description>
    </ex:editor>
  </rdf:Description>

</rdf:RDF>
```

Listing 2.1: Example RDF/XML serialisation of Fig. 2.4

2.3.2 RDFS

While RDF describes a data model for exchanging world knowledge, RDF Schema (RDFS) [11] is a specified vocabulary for modelling said knowledge. RDFS defines three main concepts:

- Classes describing resources, such as rdfs:Class, rdfs:Resource, or rdfs:Literal
- Properties defining the relation between subject resources and object resources, such as rdfs:range, rdfs:domain, or rdfs:subClassOf
- Other vocabulary, such as containers (e.g. rdfs:bag), utility vocabulary (e.g. rdfs:seeAlso), collections (e.g. rdfs:list), and reification vocabulary (e.g. rdf:subject)

[11]`https://www.w3.org/TR/rdf-schema/`

2.3.3 OWL 2

The Web Ontology Language 2 (OWL 2) is a specification language for ontologies.[12] An OWL ontology can be represented as a RDF graph, as a direct mapping between both languages is possible. For interoperability purposes, OWL ontologies can be translated into different syntaxes in order to be exchanged between applications. RDF/XML is the most common syntax for owl serialisation. In order to be OWL 2 conform, a tool must account for RDF/XML serialisation as a minimal standard.[13]. Fig. 2.5 shows the relationship between OWL, RDF graphs, and different syntaxes.

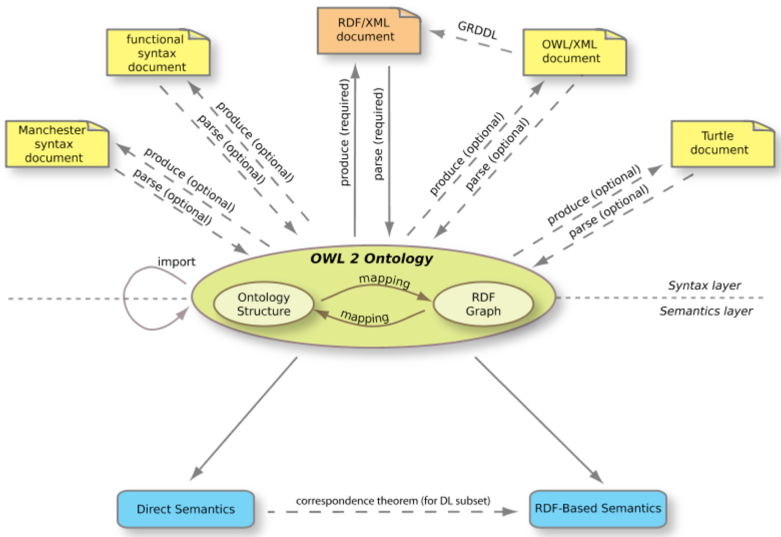


Figure 2.5: Relationship between OWL representation, RDF graph, and serialisation syntaxes, Source: `https://www.w3.org/TR/owl2-overview/`

2.3.4 SKOS

Simple Knowledge Organization System (SKOS) is a data model for the representation of controlled vocabulary, such as thesauri or classification systems, as linked data in semantic web contexts.[14] SKOS refines OWL and RDF properties and classes, but can also be used independently. Within the context of this project, SKOS was mainly used for representing concept descriptions, especially for the Proppian functions, by using the skos:prefLabel and skos:altLabel properties (see Section 3.4).[15]

[12]`https://www.w3.org/TR/owl2-overview/`
[13]`https://www.w3.org/TR/2012/REC-owl2-conformance-20121211/`
[14]`https://www.w3.org/2004/02/skos/`
[15]`https://www.w3.org/TR/skos-reference/#labels`

2.3.5 SPARQL

SPARQL Protocol And RDF Query Language (recursive acronym) is a RDF based query and manipulation language. It can be used to access data, delete and updata data in an RDF graph.[16] In order to improve the readability of the queries, prefixes for namespaces, such as *rdfs* representing *http://www.w3.org/2000/01/rdf-schema#*, can be defined. SPARQL queries also allow filters such as regex or groupby operations. The SPARQL protocol allows different exchange formats for query results, namely XML, JSON, CSV, and TSV.

The following examples illustrate the three main functionalities query, update, and delete.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name (COUNT(?friend) AS ?count)
WHERE {
    ?person foaf:name ?name .
    ?person foaf:knows ?friend .
} GROUP BY ?person ?name
```

Listing 2.2: "Example query Source: `https://www.w3.org/TR/sparql11-overview/#sparql11-query`"

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>

DELETE DATA
{
  <http://example/book2> dc:title "David Copperfield" ;
                         dc:creator "Edmund Wells" .
}
```

Listing 2.3: "Example delete operation Source:`https://www.w3.org/TR/sparql11-update/#deleteData`"

[16] `https://www.w3.org/TR/sparql11-overview/`

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
INSERT DATA
{
  <http://example/book1> dc:title "A new book" ;
                         dc:creator "A.N.Other" .
}
```

Listing 2.4: "Example insert operation Source: `https://www.w3.org/TR/sparql11-update/#insertData`"

2.3.6 REST

Representational State Transfer (REST) is an architectural paradigm for distributed systems. It defines the interaction principles between different components of webservices. In order to be considered RESTful, a system has to fulfill six constraints.

- Client-server: separating the user interface from the data storage, in order to improve the portability of the individual components
- Stateless: requests from a client need to provide all the necessary information the server needs to understand and handle them without additional knowledge about the context on the server or the session state on the client side
- Cache: data that is delivered as a response to a request must be labelled as cacheable, i.e. the client is given the right to reuse the data for equivalent requests at a later time, or non-cacheable
- Uniform interface: data is transferred in a standardized form
- Layered System: restricting knowledge of the system to the layer it is interacting with, allowing to encapsulate legacy services
- Code on Demand (optional): allows client functionality to be extended by the server by downloading and executing code, e.g. Java applets

(Fielding and Taylor, 2000, p.78-85)

2.3.7 Apache Jena and Fuseki

Apache Jena is an open source Java framework for semantic web applications. Jena internally represents data as RDF graph models, which can originate from OWL data. It constist of a set of application programming interfaces (API) for various purposes, such as RDF serialisation, inference, or triple store.[17] Apache Jena's architecture is illustrated in Fig. 2.6.

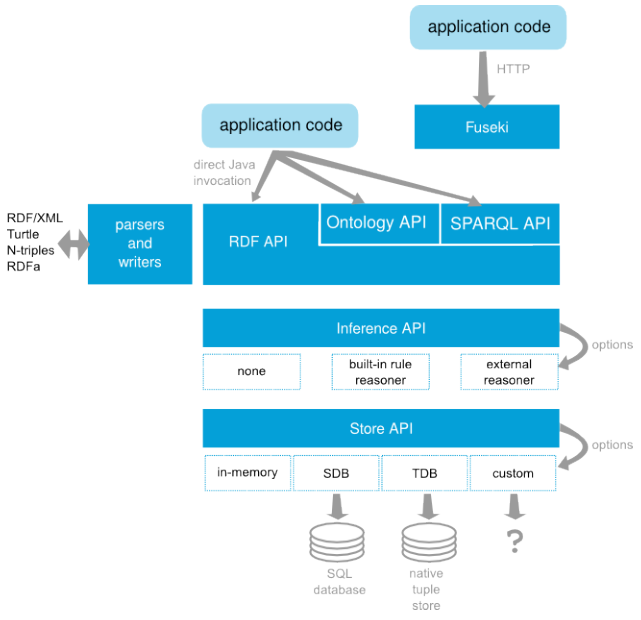


Figure 2.6: Apache Jena architecture
Source: `https://jena.apache.org/getting_started/index.html`

Fuseki is a Jena based SPARQL server. Fuseki allows CRUD[18] operations as a RESTful web service, such as GET, POST, or DELETE via HTTP. It can be run as a standalone server, as a service or a web application. [19]

[17] `https://jena.apache.org/getting_started/index.html`
[18] Create, Read, Update, Delete
[19] `https://jena.apache.org/documentation/fuseki2/`

2.3.8 Flask

Flask is a Python based micro-framework[20], which depends on the Jinja2 template engine[21], and the WSGI[22] utility library Werkzeug[23]. Flask allows rendering HTML templates for static websites while providing access to Python libraries and their functionality. This way, web applications can conveniently use functionality provided by Python libraries such as the Natural Language Toolkit (NLTK)[24].

Data can be effortlessly passed from a Python module to the rendered templates by using delimiters { and } for variables, and {% and %} for control flow statements, as illustrated by the following example. [25]

```
<!doctype html>
<title>Hello from Flask</title>
{% if name %}
  <h1>Hello {{ name }}!</h1>
{% else %}
  <h1>Hello, World!</h1>
{% endif %}
```

Listing 2.5: Example Jinja2 template

[20]http://flask.pocoo.org/
[21]http://jinja.pocoo.org/docs/2.10/
[22]Web Server Gateway Interface
[23]http://werkzeug.pocoo.org/
[24]https://www.nltk.org/
[25]Example code from http://flask.pocoo.org/docs/1.0/quickstart#static-files

Chapter 3

Design of the Ontology

3.1 Motivation

The choice to model Propp's theory by using an ontology has two main motivations. Firstly, the functions are highly hierarchical as they are divided in categories, functions and subfunctions. Secondly, the use of RDF triples allows us to model the functions not only as classes within the ontology, but additionally represent them as object properties between two instances of subclasses of *Dramatis Personae*. This approach allows us to query not only instances of functions, but also the relation they represent between characters in a tale. After all, the functions are defined as "Functions of the Dramatis Personae" (Propp, 1968) and should therefore not be separated from the characters in a tale. To our knowledge, the representation of functions as object properties as followed in this project is a novel approach.

3.2 Competency Questions

For the design of the ontology, we followed Noy and MacGuinness' approach and defined a set of competency questions. (Noy and McGuinness, 2001) If these questions can be answered by the final ontology, it has fulfilled its expressive purpose.

1. Which folk tales fall into a given motif class, e.g. ATU 70-99 Other Wild Animals?
2. Which Dramatis Personae appear in a given tale?
3. Which Proppian functions appear in African folk tales?
4. How are Dramatis Personae interacting in the African folk tales, e.g. which figures use the "interdiction" relation?

5. Which sequences of Proppian functions appear in a given tale? Which sequences appear in tales in general?
6. Which Proppian functions follow a given function predominantely, i.e. are there patterns withing the Proppian sequences?
7. Who is the editor of an anthology with folk tales from a given origin?
8. How are Proppian functions verbalised, i.e. which words are used to describe events that fall into a given function class?
9. Is there a dominating interaction between certain classes of Dramatis Personae?

If and how these questions can be answered by the ontology-driven information system will be discussed in Chapter 5.

3.3 Formal Definition of the Ontology

Following Noy and MacGuiness design pipeline (Noy and McGuinness, 2001), a set of axioms was defined before the implementation of the ontology.

3.3.1 Axioms

We define some axioms for the publication and the classification of the fairy tales.

- Each tale is published in an anthology, or as part of a journal article.
- Each anthology has at least one editor, a title, a publisher, and a date of publication.
- Each tale has a title.
- A tale has an author, and can have an origin.
- Each tale has a set of Dramatis Personae.
- Each fictional character is represented by one or more verbalisations.[1]
- Each tale falls into one of the ATU classes.
- Each ATU class has an ATU number and a description.
- If a Proppian function applies for a tale, there is some verbalisation in the text.
- Proppian functions follow a specific order (see below), this order is represented by a sequence.
- Each Proppian function has a symbol.

[1]e.g. in Snow White 'the stepmother' and 'the evil queen' describe the same individual

Furthermore, following Propp's approach, we derive axioms for the description of the narrative, see appendix A.1. If a function applies to a tale, the axiom holds. Not all of the functions/axioms need to be fulfilled by every tale.

3.3.2 Description Logic

The following tables describes the most important concepts of the core Propp ontology as Description Logic statements. For the additional ontologies that were imported (see Section 3.8), this kind of logical foundation has not been published. Therefore, we cannot make any assumptions about it. The Protégé plugin reasoner Pellet processed the ontology merged with Koleva's family ontology (see Section 3.8.2) without raising an error.

Table 3.1: Most important axioms and concepts of the core ontology in DL

	Axioms of Class Hierarchy	Important Concepts
Content	FictionalCharacter $\sqsubseteq$ Person	FictionalCharacter $\sqsubseteq \exists$ appearsIn.Tale
	DramatisPersonae $\sqsubseteq$ FictionalCharacter	$\exists$ FollowedBy.ProppianFunction $\sqsubseteq$ ProppianFunction
	Animal $\sqsubseteq$ FictionalCharacter	$\exists$ PrecededBy.ProppianFunction $\sqsubseteq$ ProppianFunction
	Human $\sqsubseteq$ FictionalCharacter	$\exists$ correspondsTo.ProppianFunction $\sqsubseteq$ ProppianFunction
	SocialAbstraction $\sqsubseteq$ FictionalCharacter	ProppianFunction $\sqsubseteq \exists$ applies.Tale
	Plant $\sqsubseteq$ FictionalCharacter	FictionalCharacter $\sqsubseteq \neg$ RealPerson
	Speaker $\sqsubseteq$ FictionalCharacter	Hero $\sqsubseteq \forall$ acquires.MagicalAgent
	Supernatural $\sqsubseteq$ FictionalCharacter	Hero $\sqsubseteq \forall$ acquiresFrom.Donor
	Family Member $\sqsubseteq$ FictionalCharacter	Hero $\sqsubseteq \forall$ attackedBy.Donor
	Princess $\sqsubseteq$ DramatisPersonae	Donor $\sqsubseteq \forall$ attemptsHarm.Hero
	Princess′ Father $\sqsubseteq$ DramatisPersonae	Donor $\sqsubseteq \forall$ begsForFreedom.Hero
	Hero $\sqsubseteq$ DramatisPersonae	Villain $\sqsubseteq \forall$ causesHarm.FamilyMember
	False Hero $\sqsubseteq$ DramatisPersonae	False Hero $\sqsubseteq \neg$ Hero
	Victim $\sqsubseteq$ Hero $\sqcup$ FamilyMember $\sqcup$ Princess	FamilyMember $\sqsubseteq \forall$ relatedTo.Hero
	Dispatcher $\sqsubseteq$ DramatisPersonae	Villain $\sqsubseteq \neg$ (Victim $\sqcup$ Hero)
	Villain $\sqsubseteq$ DramatisPersonae	Hero $\sqsubseteq \forall$ brandedBy.Object
	Donor $\sqsubseteq$ DramatisPersonae	Hero $\sqsubseteq \forall$ combats.(Villain $\sqcup$ Donor)
	Helper $\sqsubseteq$ DramatisPersonae	Reward $\sqsubseteq \exists$ isRewardedTo.Hero
	Seeker $\sqsubseteq$ Hero $\sqcup$ seeks.DesiredObject	Reward $\sqsubseteq \neg$ MagicalAgent
	DesiredObject $\sqsubseteq$ Object	Villain $\sqsubseteq \forall$ deceives.Victim
	MagicalAgent $\sqsubseteq$ Object	Princess $\sqsubseteq \neg$ Princess′Father
		Hero $\sqsubseteq \forall$ defeats.Villain
		Hero $\sqsubseteq \forall$ devidesProperty.Donor
		Hero $\sqsubseteq \forall$ frees.Victim
		Villain $\sqsubseteq \forall$ gainsInformation.Victim
		FictionalCharacter $\sqsubseteq \forall$ interdicts.FictionalCharacter
		Hero $\sqsubseteq \forall$ interrogated.Donor
		Hero $\sqsubseteq \forall$ marries.Princess
		Hero $\sqsubseteq \forall$ offered.MagicalAgent
		MagicalAgent $\sqsubseteq \forall$ offeredBy.Donor
		Villain $\sqsubseteq \forall$ persuades.Victim
		Task $\sqsubseteq \exists$ proposedBy.FictionalCharacter
		Task $\sqsubseteq \exists$ proposedTo.Hero
		Hero $\sqsubseteq \forall$ pursuedBy.FictionalCharacter
		Hero $\sqsubseteq \forall$ reactsTo.Donor
		Hero $\sqsubseteq \forall$ recognizedByMeansOf.Object
		Villain $\sqsubseteq \forall$ reconnaissance.(Victim$\sqcup$ Object)
		Donor $\sqsubseteq \forall$ requests.Task
		Donor $\sqsubseteq \forall$ requests.Object
		Donor $\sqsubseteq \forall$ requestsFrom.Hero
		Donor $\sqsubseteq \forall$ requestsMercy.Hero
		Donor $\sqsubseteq \forall$ requestsService.Hero
		Hero $\sqsubseteq \forall$ rescuedBy.FictionalCharacter
		Hero $\sqsubseteq \forall$ resolves.Task
		Villain $\sqsubseteq \forall$ taskesPosessionFrom.Victim
		Hero $\sqsubseteq \forall$ testedBy.Donor
		Villain $\sqsubseteq \forall$ threatens.Victim
Metadata	RealPerson $\sqsubseteq$ Person	Anthology $\sqsubseteq \exists$ editedBy.Editor
	Editor $\sqsubseteq$ RealPerson	Tale $\sqsubseteq \exists^{=1}$ hasClass.ATU_Class
	Author $\sqsubseteq$ RealPerson	Tale $\sqsubseteq \exists$ publishedIn.Publication
	Storyteller $\sqsubseteq$ RealPerson	JournalArticle $\sqsubseteq \exists$ publishedInJournal.Journal
	Anthology $\sqsubseteq$ Publication	
	Journal $\sqsubseteq$ Publication	
	JournalArticle $\sqsubseteq$ Publication	

3.4 Classes

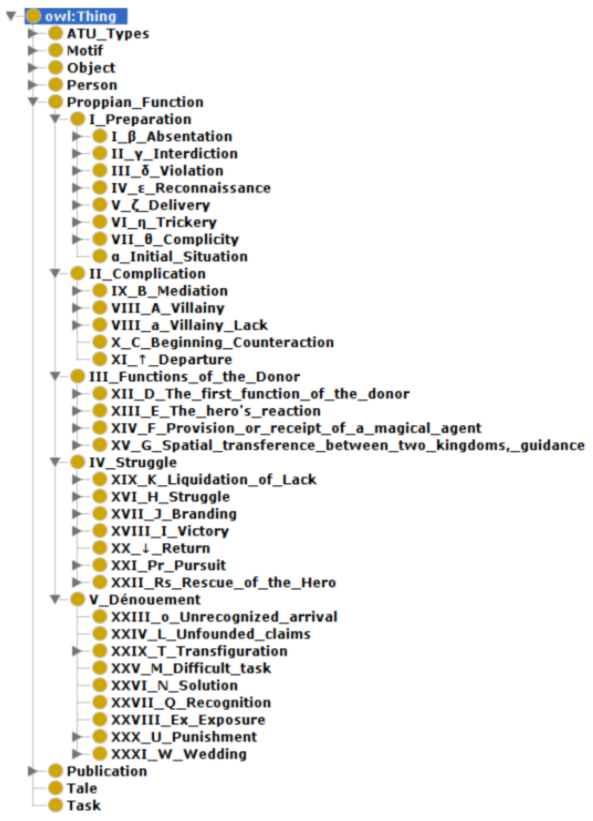
As for the current version of the ontology (see Section 3.9), the classes as shown in Fig. 3.1 and 3.2 were defined or imported into the ontology.

In contrast to ProppOnto by (Declerck et al., 2017a), the classes modelling the Proppian functions and their subfunctions have been named after their original description as published in (Propp, 1968). Furthermore, in our case the types of Dramatis Personae are modelled as subclasses and not as individuals of the Dramatis Personae class. This way, we can assign characters appearing in a tale as individuals of character classes, such as *O_Obaradeo* as *Victim*.

Fig. 3.3 shows an example class definition for the Proppian function *Punishment U*, including skos labels, comments, class restrictions and disjoint classes. Alternative labels consist of translations of the prefLabels in different languages, such as German, Russian, and Bulgarian, that were either imported from the Family Ontology (see Section 3.8.2) or the ProppOnto (see Section 3.8.1), provided by native speakers of isiZulu for the possible application of the system for African tales in their native languages, or created ourselves. Some English altLabels have been derived from WordNet synsets via the NLTK wordnet interface[2], in order to increase number of matches between the folktale text and the prefLabels for the information extraction.

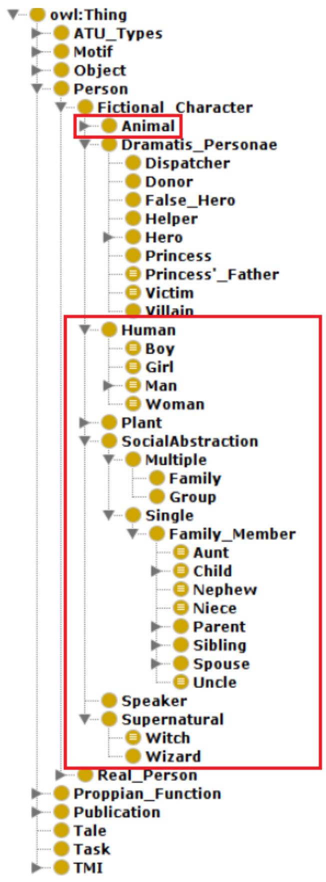
Following Propp's naming conventions, the subfunctions are named following the same pattern as the parent function, e.g. $\delta 1\_Interdiction\_violated$. The ontology was designed in OWL format[3] using the Protégé desktop application (Musen, 2015).

[2]http://www.nltk.org/howto/wordnet.html
[3]https://www.w3.org/2001/sw/#owl



```
owl:Thing
  ATU_Types
  Motif
  Object
  Person
  Proppian_Function
    I_Preparation
      I_β_Absentation
      II_γ_Interdiction
      III_δ_Violation
      IV_ε_Reconnaissance
      V_ζ_Delivery
      VI_η_Trickery
      VII_θ_Complicity
      α_Initial_Situation
    II_Complication
      IX_B_Mediation
      VIII_A_Villainy
      VIII_a_Villainy_Lack
      X_C_Beginning_Counteraction
      XI_↑_Departure
    III_Functions_of_the_Donor
      XII_D_The_first_function_of_the_donor
      XIII_E_The_hero's_reaction
      XIV_F_Provision_or_receipt_of_a_magical_agent
      XV_G_Spatial_transference_between_two_kingdoms,_guidance
    IV_Struggle
      XIX_K_Liquidation_of_Lack
      XVI_H_Struggle
      XVII_J_Branding
      XVIII_I_Victory
      XX_↓_Return
      XXI_Pr_Pursuit
      XXII_Rs_Rescue_of_the_Hero
    V_Dénouement
      XXIII_o_Unrecognized_arrival
      XXIV_L_Unfounded_claims
      XXIX_T_Transfiguration
      XXV_M_Difficult_task
      XXVI_N_Solution
      XXVII_Q_Recognition
      XXVIII_Ex_Exposure
      XXX_U_Punishment
      XXXI_W_Wedding
  Publication
  Tale
  Task
```

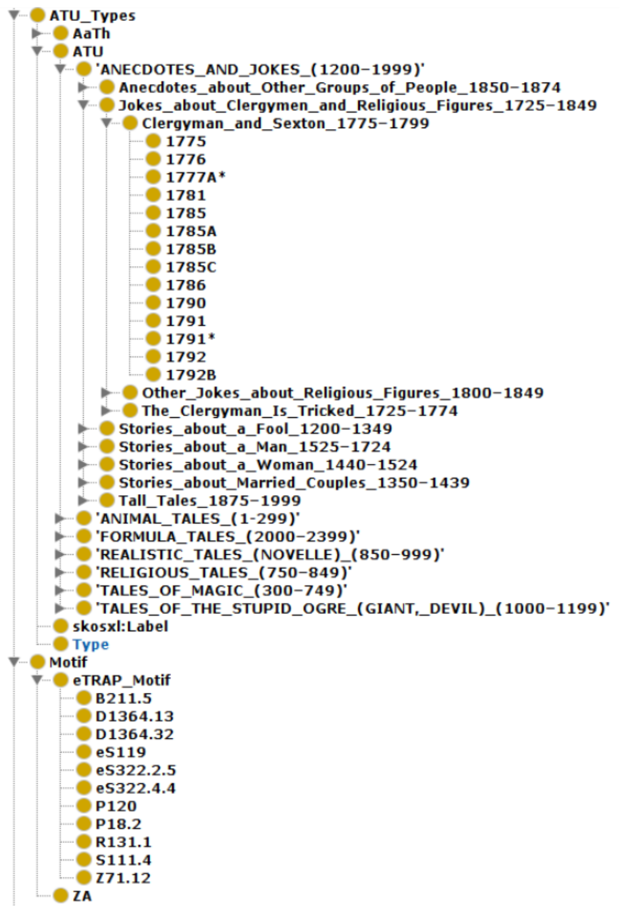
(a) Subclasses Proppian Functions



```
owl:Thing
  ATU_Types
  Motif
  Object
  Person
    Fictional_Character
      Animal
      Dramatis_Personae
        Dispatcher
        Donor
        False_Hero
        Helper
        Hero
        Princess
        Princess'_Father
        Victim
        Villain
      Human
        Boy
        Girl
        Man
        Woman
      Plant
      SocialAbstraction
        Multiple
          Family
          Group
        Single
          Family_Member
            Aunt
            Child
            Nephew
            Niece
            Parent
            Sibling
            Spouse
            Uncle
      Speaker
      Supernatural
        Witch
        Wizard
    Real_Person
  Proppian_Function
  Publication
  Tale
  Task
  TMI
```

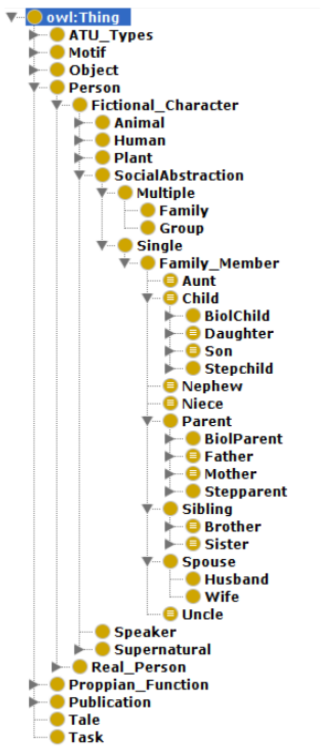
(b) Subclasses of the Dramatis Personae, red boxes indicate classes imported from the Family Ontology (Koleva et al., 2012)

Figure 3.1: Subclasses Function and Dramatis Personae



```
ATU_Types
    AaTh
    ATU
        'ANECDOTES_AND_JOKES_(1200-1999)'
            Anecdotes_about_Other_Groups_of_People_1850-1874
            Jokes_about_Clergymen_and_Religious_Figures_1725-1849
                Clergyman_and_Sexton_1775-1799
                    1775
                    1776
                    1777A*
                    1781
                    1785
                    1785A
                    1785B
                    1785C
                    1786
                    1790
                    1791
                    1791*
                    1792
                    1792B
                Other_Jokes_about_Religious_Figures_1800-1849
                The_Clergyman_Is_Tricked_1725-1774
            Stories_about_a_Fool_1200-1349
            Stories_about_a_Man_1525-1724
            Stories_about_a_Woman_1440-1524
            Stories_about_Married_Couples_1350-1439
            Tall_Tales_1875-1999
        'ANIMAL_TALES_(1-299)'
        'FORMULA_TALES_(2000-2399)'
        'REALISTIC_TALES_(NOVELLE)_(850-999)'
        'RELIGIOUS_TALES_(750-849)'
        'TALES_OF_MAGIC_(300-749)'
        'TALES_OF_THE_STUPID_OGRE_(GIANT,_DEVIL)_(1000-1199)'
    skosxl:Label
    Type
Motif
    eTRAP_Motif
        B211.5
        D1364.13
        D1364.32
        eS119
        eS322.2.5
        eS322.4.4
        P120
        P18.2
        R131.1
        S111.4
        Z71.12
    ZA
```

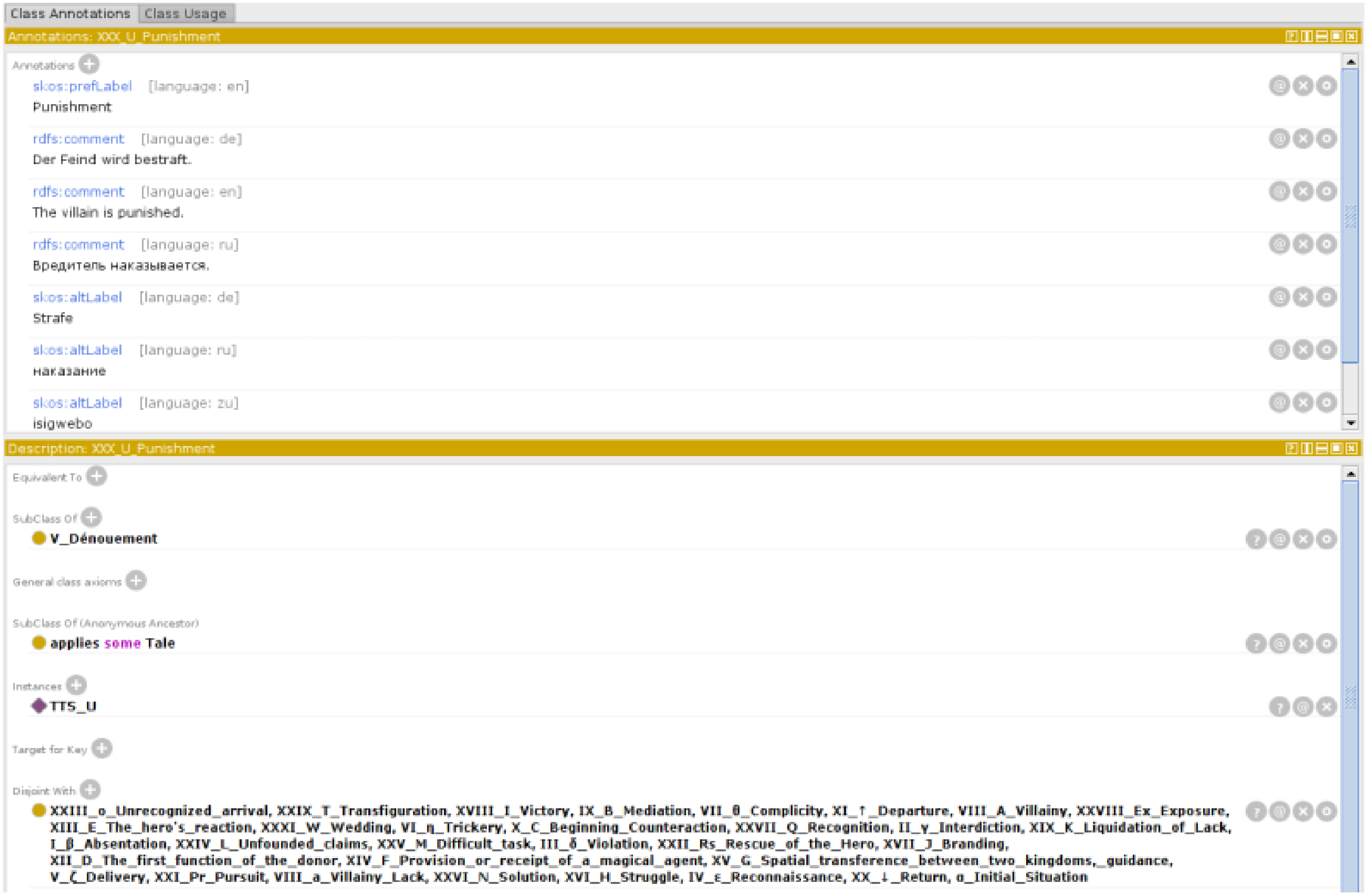
(a) Subclasses Motifs and ATU Types (Declerck et al., 2017b)



```
owl:Thing
    ATU_Types
    Motif
    Object
    Person
        Fictional_Character
            Animal
            Human
            Plant
            SocialAbstraction
                Multiple
                    Family
                    Group
                Single
                    Family_Member
                        Aunt
                        Child
                            BiolChild
                            Daughter
                            Son
                            Stepchild
                        Nephew
                        Niece
                        Parent
                            BiolParent
                            Father
                            Mother
                            Stepparent
                        Sibling
                            Brother
                            Sister
                        Spouse
                            Husband
                            Wife
                        Uncle
            Speaker
            Supernatural
        Real_Person
    Proppian_Function
    Publication
    Tale
    Task
```

(b) Subclasses of the Family Ontology (Koleva et al., 2012)

Figure 3.2: Ontology Classes



Annotations: XXX_U_Punishment

Annotations

skos:prefLabel [language: en]
Punishment

rdfs:comment [language: de]
Der Feind wird bestraft.

rdfs:comment [language: en]
The villain is punished.

rdfs:comment [language: ru]
Вредитель наказывается.

skos:altLabel [language: de]
Strafe

skos:altLabel [language: ru]
наказание

skos:altLabel [language: zu]
isigwebo

Description: XXX_U_Punishment

Equivalent To

SubClass Of
V_Dénouement

General class axioms

SubClass Of (Anonymous Ancestor)
applies some Tale

Instances
TTS_U

Target for Key

Disjoint With
XXIII_o_Unrecognized_arrival, XXIX_T_Transfiguration, XVIII_I_Victory, IX_B_Mediation, VII_θ_Complicity, XI_↑_Departure, VIII_A_Villainy, XXVIII_Ex_Exposure, XIII_E_The_hero's_reaction, XXXI_W_Wedding, VI_η_Trickery, X_C_Beginning_Counteraction, XXVII_Q_Recognition, II_γ_Interdiction, XIX_K_Liquidation_of_Lack, I_β_Absentation, XXIV_L_Unfounded_claims, XXV_M_Difficult_task, III_δ_Violation, XXII_Rs_Rescue_of_the_Hero, XVII_J_Branding, XII_D_The_first_function_of_the_donor, XIV_F_Provision_or_receipt_of_a_magical_agent, XV_G_Spatial_transference_between_two_kingdoms,_guidance, V_ζ_Delivery, XXI_Pr_Pursuit, VIII_a_Villainy_Lack, XXVI_N_Solution, XVI_H_Struggle, IV_ε_Reconnaissance, XX_↓_Return, α_Initial_Situation

Figure 3.3: Example of a class definition

```
<owl:Class rdf:about="https://teaching.gcdh.de/ProppOntology/1.0.1#
    A11_Villain_casts_spell_upon_someone_or_something">
    <rdfs:subClassOf rdf:resource="https://teaching.gcdh.de/ProppOntology/1.0.1#
        VIII_A_Villainy"/>
    <skos:altLabel xml:lang="de">Verzauberung, Verwandlung</skos:altLabel>
    <skos:altLabel xml:lang="en">cast, casting</skos:altLabel>
    <skos:altLabel xml:lang="en">enchantment, spell, trance</skos:altLabel>
    <skos:altLabel xml:lang="en">transformation, transmutation, shift</skos:altLabel
        >
    <skos:prefLabel xml:lang="en">the casting of a spell, a transformation</skos:
        prefLabel>
    <skos-xl:literalForm>A11</skos-xl:literalForm>
</owl:Class>
```

Listing 3.1: Example specification of a Proppian function

```
<owl:Class rdf:about="https://teaching.gcdh.de/ProppOntology/1.0.1#Princess">
    <rdfs:subClassOf rdf:resource="https://teaching.gcdh.de/ProppOntology/1.0.1#
        Dramatis_Personae"/>
    <owl:disjointWith rdf:resource="https://teaching.gcdh.de/ProppOntology/1.0.1#
        Princess&apos;_Father"/>
    <dc:source xml:lang="en">Vladimir Propp: Morphology of the Folktale. Austin,
        Texas 1968</dc:source>
    <rdfs:comment xml:lang="en">Propp claims that sometimes functionally the
        Princess and her father cannot be distinguished, i.e. they fulfill the same
        role on an abstract level.</rdfs:comment>
    <skos:altLabel xml:lang="de">Prinzessin</skos:altLabel>
    <skos:prefLabel xml:lang="en">Princess</skos:prefLabel>
</owl:Class>
```

Listing 3.2: Example specification of a Character class

3.5 Object Properties

The ontology currently holds 137 object properties from three sources, own properties, properties from Koleva's family ontology (Koleva, 2011) and properties from the TMI-ATU ontology (Declerck et al., 2017b). The following table lists object properties with their corresponding source.

Table 3.2: Object properties by source

ProppOntology	Family Ontology	TMI-ATU Ontology
relatedTo	hasBiolChild	linkFromAaThToATU
correspondsTo	hasNephew	verbalizes
reconnaissance	into	represents
requestsMercy	hasResult	isRepresentedBy
seeks	hasStepChild	partOfCollection
gainsInformation	contains	containsMotif
ledTo	theWifeIs	hasVIAF_ID
recognizedByMeansof	hasSubevent	isLinkFromATU
attemptsHarm	after	hasPart
PreceededBy	finishes	linkToTMI
isReversed	hasBiolFather	verbalizedAs
rescuedBy	ntpp	appearsInTale
hasClass	hasBiolParent	linkFromTMIToATU
applies	on	
editedBy	owns	
publishedInJournal	hasUncle	
helpsEnemy	hasMother	
interrogated	under	
resolves	hasLocation	
pursuedBy	sameAs	
causesHarm	hasBiolMother	
brandedBy	hasHusband	
devidesProperty	starts	
testedBy	ec	
publishedIn	before	
acquires	startedBy	
requestsService	marriedTo	
proposedTo	beyond	
acquiresFrom	equals	
appearsIn	hasSibling	
attackedBy	meets	

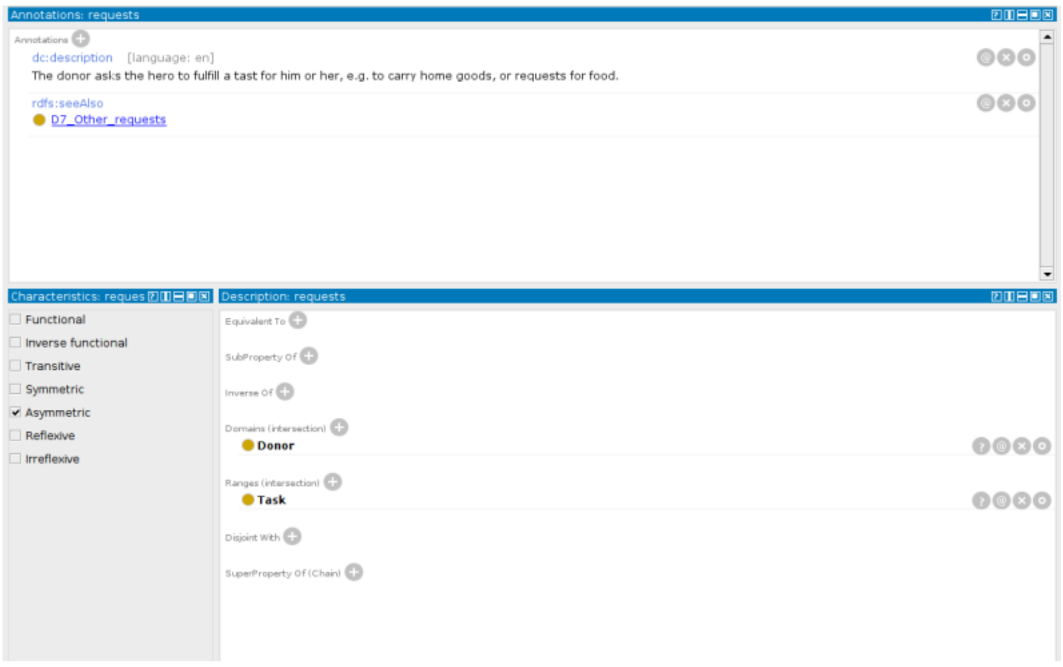
Table 3.2: Object properties by source

ProppOntology	Family Ontology	TMI-ATU Ontology
combats	hasRecipient	
offered	during	
FollowedBy	overlaps	
frees	rcc8	
isRewarded	theHusbandIs	
defeats	hasPatient	
proposedBy	allen	
begsForFreedom	dc	
takesPosessionFrom	hasPart	
persuades	hasFather	
isRewardedTo	isPart	
requests	metBy	
interdicts	isLocation	
offeredBy	po	
deceived	hasOwner	
hasAuthor	hasParticipant	
requestsFrom	tpp	
marries	eq	
threatens	hasNiece	
reactsTo	isYoungerThan	
	hasWife	
	isOderThan[4]	
	hasParent	
	hasStepfather	
	hasMember	
	hasStepmother	
	hasBrother	
	hasAunt	
	isPlace	
	hasChild	
	hasSister	
	hasStepparent	
	hasInstrument	
	hasTheme	
	hasProtagonist	

[4] Typographical errors by (Koleva, 2011) have been preserved.

Table 3.2: Object properties by source

ProppOntology	Family Ontology	TMI-ATU Ontology
	hasAgent	
	overlapedBy[4]	
	finishedBy	
	hasTime	
	hasValidPeriod	
	hasDirection	

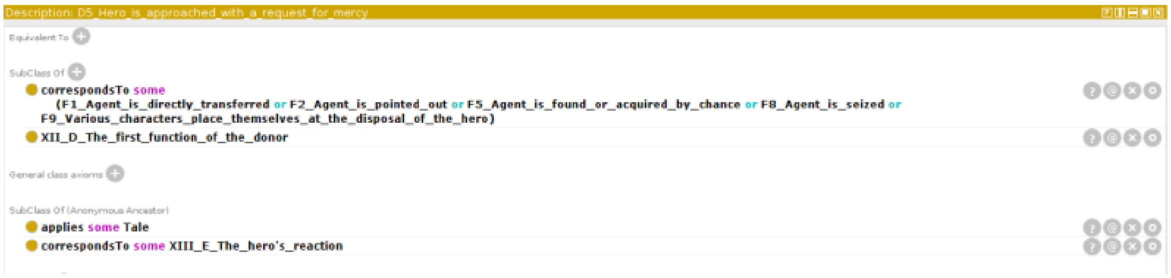


```
Annotations: requests
Annotations
  dc:description  [language: en]
  The donor asks the hero to fulfill a tast for him or her, e.g. to carry home goods, or requests for food.

  rdfs:seeAlso
  D7_Other_requests

Characteristics: reques          Description: requests
☐ Functional                     Equivalent To
☐ Inverse functional             SubProperty Of
☐ Transitive                     Inverse Of
☐ Symmetric                      Domains (intersection)
☑ Asymmetric                       Donor
☐ Reflexive                      Ranges (intersection)
☐ Irreflexive                      Task
                                 Disjoint With
                                 SuperProperty Of (Chain)
```

Figure 3.4: Protégé view of object property *requests* with range and domain

Functions of the Donor

As shown in Fig. 2.2, the *First Function of the Donor D* corresponds to the *Receipt of Magical Agent F*. Propp carefully defined which subfunctions appear together. For the design of the ontology, we added this restriction in the form of 'F correspondsTo some D' subclass for each of the subfunctions. Figure 3.5 shows this correspondence for the example of the subfunction *Hero is approached with a request for mercy D5*.



```
Description: D5_Hero_is_approached_with_a_request_for_mercy
Equivalent To
SubClass Of
  correspondsTo some
    (F1_Agent_is_directly_transferred or F2_Agent_is_pointed_out or F5_Agent_is_found_or_acquired_by_chance or F8_Agent_is_seized or
  F9_Various_characters_place_themselves_at_the_disposal_of_the_hero)
  XII_D_The_first_function_of_the_donor
General class axioms
SubClass Of (Anonymous Ancestor)
  applies some Tale
  correspondsTo some XIII_E_The_hero's_reaction
```

Figure 3.5: Correspondence restriction for the function D5

For the function pairs as indicated by the bold lines in Fig. 2.1, the same approach has been followed.

3.6 Data Properties

The ontology currently holds 36 owl:Datatype properties, originating from (Koleva, 2011) or from our own implementation. The TMI-ATU ontology provided no datatype properties. An overview of the properties can be found in Table 3.3.

Table 3.3: Datatype properties by source

ProppOntology	Family Ontology
Year	hasNumber
Volume	isYoung
Verbalisation	isOld
Title	begins
Publisher	lasts
Pages	hasAge
Origin	ends
Name	isAdult
lifeDates	hasGender
Language	sole
Key	isEvil
Journal_Name	isGood
Issue	happens
ISSN	isAnimated
Gender	
FunctionSequence	
died	
Date	
Country	
born	
ATU_Number	

3.7 Ontology Metrics

The following table shows insightful statistics on the sizes of the resources of the combined ontology, the ontology combinded with the family ontology without the TMI-ATU ontology, and the core Propp ontology. The term *metrics* was taken from the Protégé context (Tudorache et al., 2008).

Table 3.4: Ontology Metrics

	Propp+Family+ATU-TMI	Propp+Family	Propp
Class Count	15236	284	232
Object Property Count	138	126	52
Data Property Count	37	37	24
Individual Count	51635	310	242
Annotation Property Count	24	24	24
SubClassOf	15263	313	252
Equivalent Classes	34	34	21
Disjoint Classes	38	38	32
SubObjectPropertyOf	93	84	38
InverseObjectProperty	22	16	1
DisjointObjectProperty	1	1	1
FunctionalObjectProperty	5	5	4
InverseFunctionalObjectProperty	1	1	1
TransitiveObjectProperty	4	4	3
SymmetricalObjectProperty	7	6	3
AsymmetricalObjectProperty	31	31	29
ObjectPropertyDomain	71	62	51
ObjectPropertyRange	87	75	53
SubDataPropertyOf	17	17	16
FunctionalDataProperty	18	18	13
DataPropertyDomain	38	38	27
DataPropertyRange	25	25	13
ClassAssertion	98688	354	236
ObjectPropertyAssertion	15888	554	550
DataPropertyAssertion	365	345	325
Same Individual	16	16	0

As Table 3.4 shows, the merged ontology is excessively large. The Protégé reasoner plugin Pellet is not able to process an ontology that large without causing buffer overflow. However, the ontology file that combines the core Propp ontology and Koleva's family ontology is processed just fine. The reasoner does not yield any errors in the consistency of the merged ontology.

3.8 Import of Supplementary Ontologies

3.8.1 Alternative Labels

Declerck et al. (Declerck et al., 2017a) created an ontology that modelled Proppian functions as classes with a vast set of rdfs labels in English, German and Russian. The Internationalized Resource Identifier (IRI) of their classes is a short description of the function. However, they named subfunctions only according to their literal, e.g. *Delta1*. In the approach followed by our project, IRIs were named as close to the original description as possible. The functions in Declerck's work are not grouped into the five segments defined by Propp. Furthermore, they did not provide object properties for functions, such as those modelling sequential order or those that connect a function with the corresponding tale it appears in. The extensive labels and rdf comments provided by (Declerck et al., 2017a) were found a very useful addition to our existing ontology. Therefore, they were added to the ProppOntology and converted to skos prefLabels for the English labels, and skos altLabels for the German and Russian labels.

3.8.2 Family-Ontology

Nikolina Koleva (Koleva, 2011) built a folktale ontology that modelled family relations of characters. She used SWRL[5] rules for the classes to allow ontology reasoning. These rules were then used to automatically populate the ontology by iterating through a text, searching for semantic cues that introduce fairy tale characters. Koleva used NooJ grammars to detect the entities and the OWL API to populate the ontology. The automatic extraction of characters and role attribution from fairy tale texts seemed to work comparatively well.

However, her characters lack certain features such as their verbalisation, or information about the tale they appear in. Therefore, in this project, we tried to implement the semi-automatic population differently. Nonetheless, the class hierarchy of the characters was imported to the ProppOntology as it is a valuable addition to the ontology beyond Propp's definition of character roles, especially with regard to the repeated motif of family relations in folk tales from different origins. Additionally, Koleva provided labels, although mistakingly annotated as dc:language fields, for the character classes in German, English, Russian and Bulgarian. Those fields were carefully transformed into skos:prefLabel and skos:altLabel fields.

[5]Semantic Web Rule Language, `https://www.w3.org/Submission/SWRL/`

Even though the family ontology provided object properties *hasWife* and *hasHusband*, the classes modelling spousal relationship were missing, and had to be added to the ontology.

3.8.3 TMI-ATU-Ontology

Declerck et al. provided an ontology modelling types of folktales according to the Aarne-Thompson-Uther classification and motfis according to the Thompson-Motif-Index. They modelled ATU and TMI classes, with rdfs:labels in English and German, both as classes in the ontology and as individuals. (Declerck and Schäfer, 2017) Their ontology also includes a set of additional motifs as defined by the ETrap project[6]. As for now, only the core ATU classes and TMI motifs have been imported into the ProppOntology. While the population of the ontology with motifs and tale types as individuals technically does not follow the approach in our case, they were nevertheless imported to allow possible reuse at a later stage in the project. Following the approach of our project, motifs and tale classes should have only been added as individuals if they occur in a tale that is annotated, with a distinction in naming, such as *O_ 2012_F* for the appearance of the evil spirit in the *Obaradeo* tale. This specific individual can then hold further information on how it is verbalised in the given tale. However, leaving the individuals as they are provided, it would be possible to link tales that share a common motif.

3.9 Ontology Versioning

As of publication of this thesis, the ontology that the system is based on, has undergone a series of major and minor changes. Ontology versioning is a crucial requirement in the design of ontology-driven information systems. (Yildiz and Miksch, 2007) Therefore, all important versions are reported in Fig. 3.6.

3.10 Properties of the Ontology

Yildiz and Miksch articulate the importance of *confidence level* as a property for ontology components to indicate "how sure the ontology developer or an automated learning algorithm is about the existence of the component in the conceptualisation." (Yildiz and Miksch, 2007) However, we refrained from using this particular property, as the theory that was modelled is clearly documented, and has been academically accepted for more than 50 years. As other ontological approaches to Propp's theory exist for the representation of functions as classes[7], we can be confident that our approach is academically sound.

[6]`https://www.etrap.eu/`

[7]See for instance, (Peinado et al., 2004)

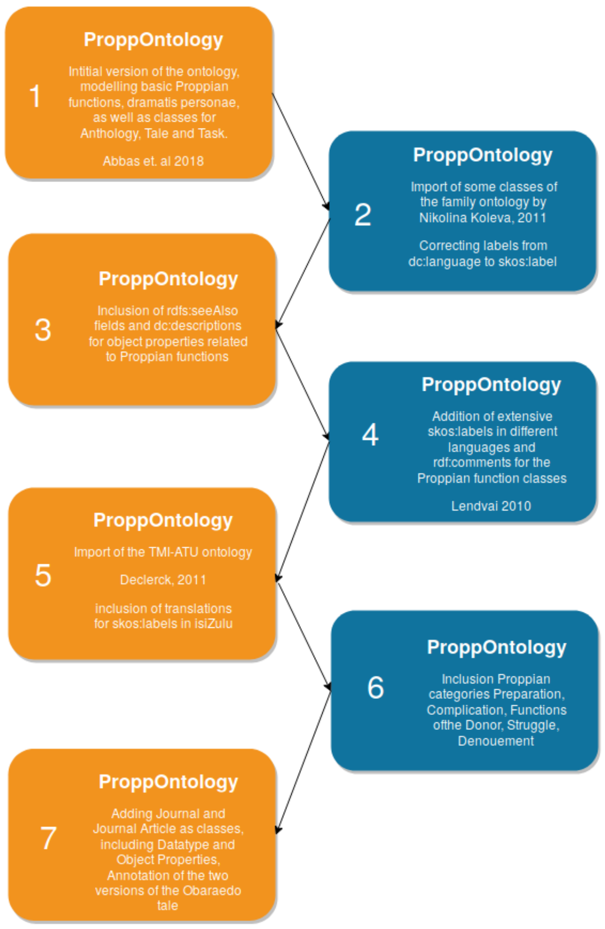

1
ProppOntology
Intitial version of the ontology, modelling basic Proppian functions, dramatis personae, as well as classes for Anthology, Tale and Task.
Abbas et. al 2018
2
ProppOntology
Import of some classes of the family ontology by Nikolina Koleva, 2011
Correcting labels from dc:language to skos:label
3
ProppOntology
Inclusion of rdfs:seeAlso fields and dc:descriptions for object properties related to Proppian functions
4
ProppOntology
Addition of extensive skos:labels in different languages and rdf:comments for the Proppian function classes
Lendvai 2010
5
ProppOntology
Import of the TMI-ATU ontology
Declerck, 2011
inclusion of translations for skos:labels in isiZulu
6
ProppOntology
Inclusion Proppian categories Preparation, Complication, Functions ofthe Donor, Struggle, Denouement
7
ProppOntology
Adding Journal and Journal Article as classes, including Datatype and Object Properties, Annotation of the two versions of the Obaraedo tale

Figure 3.6: Version history of the Propp ontology

.

We also refrained from using the *value change frequency* property, since Propp's Dramatis Personae and their functions are not object to change. While the components themselves do not change in a structural sense, they are likely to be enriched by metadata, such as rdf:comments or skos:labels (see Section 3.9).

Some *value constraint properties* have been introduced to the data properties to ensure the consistency of annotations. For further elaboration on data properties, please refer to Section 3.6.

Furthermore, to supply these and other additional properties in an efficient manner as suggested by (Yildiz and Miksch, 2007), the use of an Ontology Management Module is advised. For the further evolution of the ontology, such a module would certainly be useful, and with its introduction, the provision of more ontology properties might become desirable.

3.11 Annotation Scheme for Folktales

Folktales, Proppian functions, dramatis personae and real persons are annotated as individuals in the ontology.

The *Tale* class is the domain of a number of datatype properties, such as title, origin, key and function sequence as strings. A number of object properties connects the tale to other individuals in the ontology, such as *publishedIn Publication*.

Proppian functions are annotated as instances of the corresponding class. The IRI is created by using the corresponding *Tale*'s key and the literal representing the function. The key is usually the intial letters of the title, such as *COW* for the tale *Children of Wax*. If the same tale *Stoff* is annotated multiple times, as in the *Obaradeo* tale discussed in Section 2.2.4, the versions should be distinguished by adding _ year_ of_ publication to additional versions, e.g. *Rotkäppchen_ 1857*.

This way, the instances can be differentiated and associated with the correct folk tale immediately. For Proppian functions, the information about the preceding and following functions is annotated by using the dedicated object property, either *FollowedBy* or *PrecededBy*. No restriction rule of the form 'Proppian Function FollowedBy Exactly 1 Proppian Function' could be used when we defined the classes, since no prediction about the start or end of a sequence can be made. Therefore, any function can be the start or end function of a function sequence. The object properties *FollowedBy* and *PrecededBy* have been defined as functional properties, i.e. every Proppian function has one direct predecessor and one successor. This rule can be used for reasoning, and eventually creating function sequences automatically. Therefore, users need to be careful only to assign these properties once per Proppian function and to be coherent (A FollowedBy B $\rightarrow$ B PrecededBy A).

The following example illustrates how the function sequence of a tale can be inferred:

If we consider the function sequence $\beta AC \uparrow HIK \downarrow W$ of the tale *Brave Hunter* (Smith, 1989). We can derive all successors of the function *Absentation* β as an unordered list by querying for the transitive shell (*) of the FollowedBy relation:

$$FollowedBy(\beta) \equiv \{K, I, H, \downarrow, \uparrow, A, C\} \tag{3.1}$$

Since Propp's theory defines that the functions appear in sequential order, we know the direct successor of β in the set must be *Villainy/Lack A*. According to the Proppian order, no other function within the set can appear before *A*. For the description of a tale this means that if a hero leaves homes ($\uparrow$) and returns ($\downarrow$) before the initial harm (*A*) is inflicted, the functions *Departure* $\downarrow$ and *Return* $\uparrow$ do not apply for the specific tale.

The naming convention for characters in the tale follow the same approach as for Proppian functions. IRIs should be named *key_Name of the character*, e.g. *O_ Obaradeo*, or *key_first verbalisation* *of the character*, e.g. *O_ Mother*.

Since the family ontology (Koleva, 2011) was imported, a character can belong to multiple classes, e.g. *O_ Father* can belong to the classes *Father*, *Husband*, *Seeker* and *Man*. In the first version of the ontology, a character could only belong to one class of fictional characters, that is one of the dramatis personae defined by Propp.

Chapter 4

Ontology-Driven Information System

4.1 Preliminary Considerations

The first approach to developing the ontology-driven information system was to extend an existing, established tool.

When working with ontologies, the first tool that comes to mind is the Protégé software, that comes either as a desktop application or as the Webprotégé client (Musen, 2015). The Termine plugin for Protégé would have been a suitable candidate (Frantzi et al., 2000) to build up upon, especially for the information extraction. However, the software is not open source and contact attempts to people involved in the project were fruitless.

The second approach was to extend the Fuseki GUI, making use of existing features such as their comparatively advanced text editor for the SPARQL queries.

However, after a series of attempts to achieve the latter, it was decided that a division of the system into three different parts would be the most sustainable approach.

4.2 System Design

The activity diagram in Fig. 4.1 depicts the general functionality of the different system components.

The core of the system is a Flask[1]-based web application that provides three major functionalities: queries, annotation and ontology browsing. While most modern web applications are developed using programming languages like PHP or Ruby, Python was used in the context of this project because of the extensive availability of libraries and toolkits especially for the information extraction.

[1] http://flask.pocoo.org/

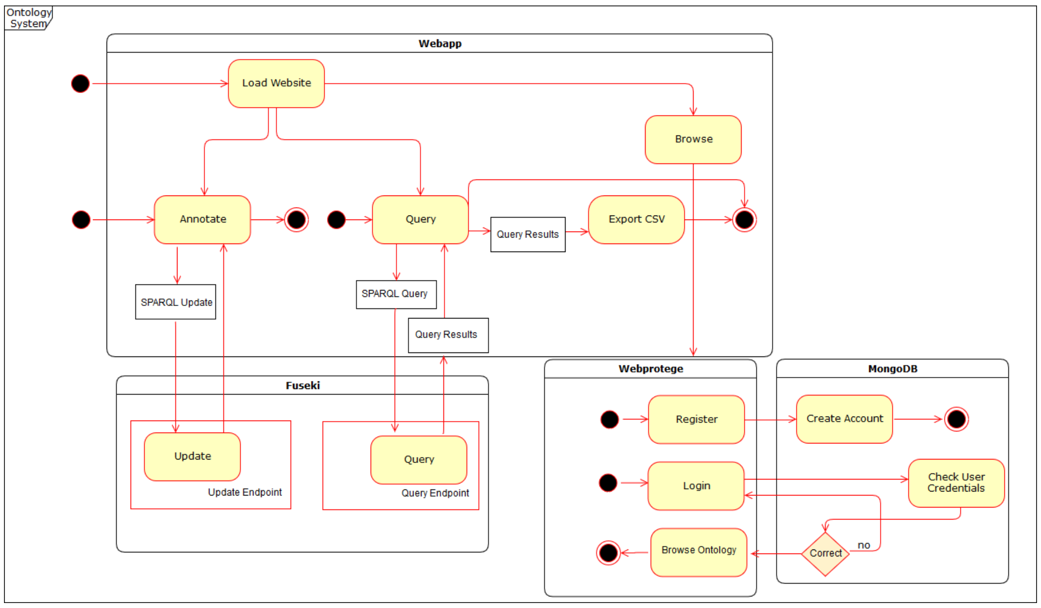


Figure 4.1: Activity Diagram of the Information System's functionality

This way, the system was developed in one language, avoiding the need to exchange data back and forth between different applications written in different programming languages.

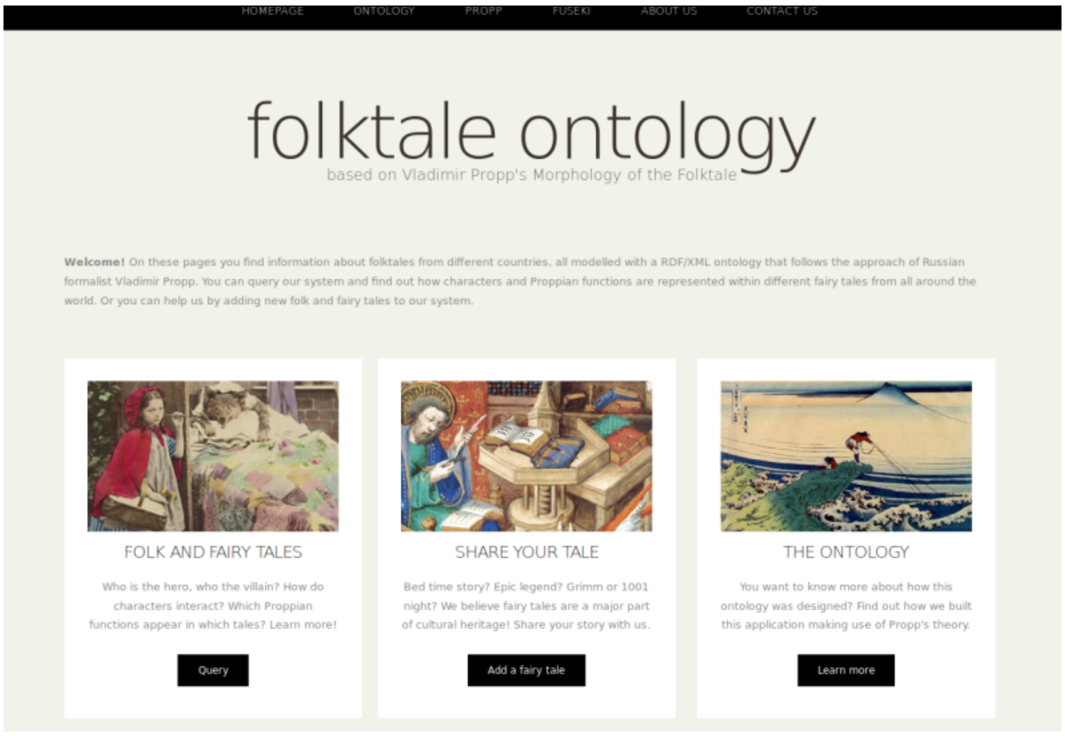
The Flask application builds the webpages from HTML templates, and communicates with the Fuseki webserver via a RESTful API. The Fuseki server processes the SPARQL queries and sends the results back to the Flask application. The Webprotégé instance is not directly connected to the Flask application but merely linked from the HTML file of the landing page as shown in Fig. 4.2. A MongoDB database is used for managing user accounts.

4.3 Fuseki

For the ontology processing, an Apache Jena Fuseki server application is used. It provides comfortable handling of SPARQL updates and queries via a RESTful API. [2]fuseki Fuseki runs on an Apache Tomcat server. For development purposes, the usage of Fuseki came with the advantage that its interface could be used to check if the ontology-driven information system that was developed behaves as desired, especially for the verification of the queries.

For the productive system, the Fuseki server is hosted on a port that is only accessible from the server on which the Flask application is deployed. That way, we ensure that no requests, especially no SPARQL updates, are sent to the RESTful API except those that come from the Flask application.

[2]https://jena.apache.org/documentation/fuseki2/



folktale ontology

based on Vladimir Propp's Morphology of the Folktale

Welcome! On these pages you find information about folktales from different countries, all modelled with a RDF/XML ontology that follows the approach of Russian formalist Vladimir Propp. You can query our system and find out how characters and Proppian functions are represented within different fairy tales from all around the world. Or you can help us by adding new folk and fairy tales to our system.

FOLK AND FAIRY TALES

Who is the hero, who the villain? How do characters interact? Which Proppian functions appear in which tales? Learn more!

SHARE YOUR TALE

Bed time story? Epic legend? Grimm or 1001 night? We believe fairy tales are a major part of cultural heritage! Share your story with us.

THE ONTOLOGY

You want to know more about how this ontology was designed? Find out how we built this application making use of Propp's theory.

Figure 4.2: Homepage of the Ontology System

This way, the risk of harmful injections into the ontology is reduced. Nevertheless, wrong or harmful user inputs could still be sent through the HTML forms, even though auto-escaping for HTML and XML is used.

4.4 Webprotégé

Webprotégé is used for the ontology browsing part of the system. (Tudorache et al., 2008) While querying in itself already provides a lot of insight, especially a good overview of individuals that were added, users might want to see how classes and subclasses are defined. Unfortunately, the current version of Webprotégé needs a MongoDB database for handling user accounts. Earlier versions provided means to view an ontology without being logged in. However, this particular functionality has been deprecated.[3]

For the time being, users are not allowed to make changes to the ontology. Changing the ontology within Webprotégé and via SPARQL updates at the same time would necessarily lead to versioning problems. However, in the future it could be useful to refrain from annotating via the Flask

[3]`http://protege-project.136.n4.nabble.com/WebProtege-home-page-Public-projects-td4668809.html`

application entirely, and allow registered users to add individuals in Webprotégé. In this case, it has to be taken into consideration that when users are allowed to make changes to the ontology, they could also change classes, and are not restricted to only adding instances. This feature is currently not wanted, as the ontology has been carefully modelled. A desirable feature in terms of user rights would be if the changes to the ontology could be restricted to specific tasks, such as creating instances, or adding labels, but not deleting them. Webprotégé also allows user discussions and comments, which is an interesting feature that the current system does not hold.

4.5 Flask Implementation

Flask is a microframe work for Python, extending the template language Jinja 2[4] and Werkzeug[5]. It was used to process the HTML templates that build the webpages, and handle the communication with the Fuseki endpoints ds/update and ds/sparql (for queries).

4.5.1 SPARQL Queries

The users of the ontology-driven information system have three means of querying the ontology. A basic text field can be used for advanced queries, triple queries can be used to investigate relations between rdf triples, and single queries provide means to investigate single classes.

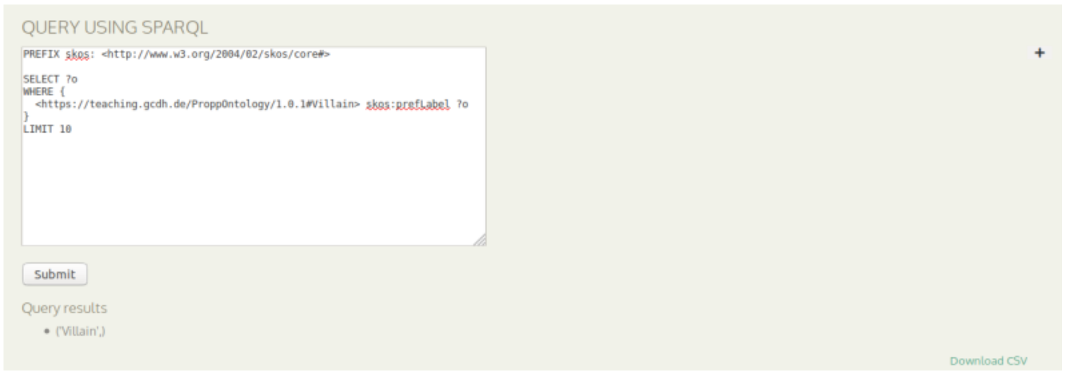
Conveniently, Jinja 2 provides means to auto-escape user inputs in a way that prevents potentially harmful inputs.[6]. To reduce the risk further, the Flask module which delivers the query pages only allows GET requests to the Fuseki API.

The user can state a SPARQL query, including prefixes, interpunctuation, query limits or regex restrictions in the text field. Figure 4.3 shows a text based query for the skos:prefLabel for the ontology class *Villain*.

All queries are processed through the Python package SPARQLWrapper2[7] which is used by the Flask application to send the query to Fusekis sparql endpoint (ds/sparql) as a GET request.

The second way of querying the ontology is by using the triple query. Users can enter either one or two classes, leaving the ones empty that would be represented by the placeholders in a SPARQL query. The first and third field are dedicated to classes, while the second field is assigned to the relation. When the query page is loaded, the most recent relations, ranges and domains are queried from the ontology to create a dropdown menu for each of the fields. A star at the end of a class

[4]http://jinja.pocoo.org/
[5]http://werkzeug.pocoo.org/
[6]`http://jinja.pocoo.org/docs/2.10/templates/`
[7]`https://rdflib.github.io/sparqlwrapper/doc/latest/SPARQLWrapper-module.html`



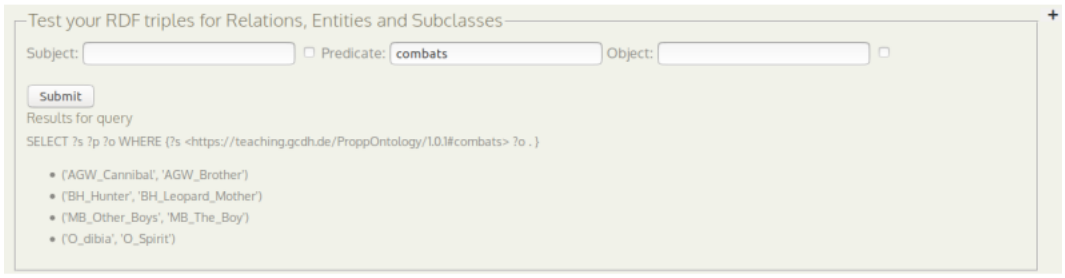
```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT ?o
WHERE {
  <https://teaching.gcdh.de/ProppOntology/1.0.1#Villain> skos:prefLabel ?o
}
LIMIT 10
```

- ('Villain',)

Figure 4.3: Text-based query and results list

name is used as a flag to query not only the class itself but also its subclasses. If the checkbox behind the first or the last of the fields is ticked, the query looks for individuals instead of classes.

All query results from either of the three ways to query the ontology can be exported as a CSV file.



Test your RDF triples for Relations, Entities and Subclasses
+

Subject: ☐ Predicate: combats Object: ☐

Results for query

SELECT ?s ?p ?o WHERE {?s <https://teaching.gcdh.de/ProppOntology/1.0.1#combats> ?o . }

- ('AGW_Cannibal', 'AGW_Brother')
- ('BH_Hunter', 'BH_Leopard_Mother')
- ('MB_Other_Boys', 'MB_The_Boy')
- ('O_dibia', 'O_Spirit')

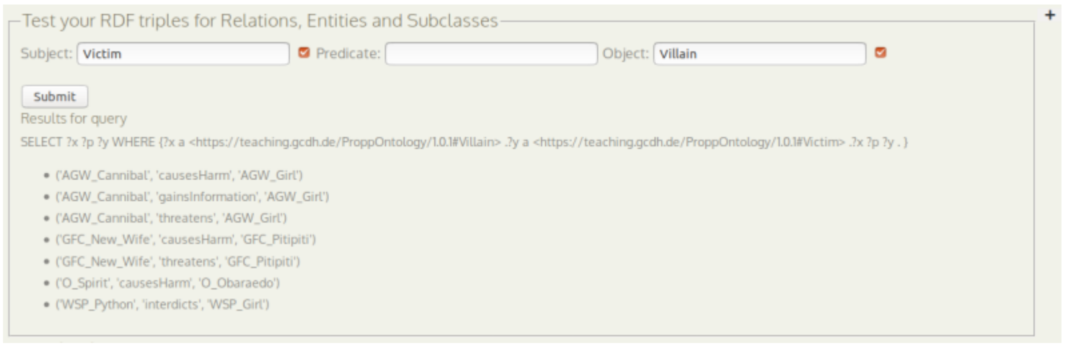
Figure 4.4: Triple query for relation

Single classes, subclasses and instances can be queried to find out more about one particular class. Again, a * after the class name indicates the query for subfunctions, the checkbox indicates the query for individuals.

4.5.2 SPARQL Update

The second functionality of the application is to provide means to annotate tales. This functionality should be seen as a proof of concept rather than a bullet-proof way of implementing the functionality in an ontology driven way. For considerations to use Webprotégé instead, please see Section 4.4.

When a user accesses the annotation functionality as shown in Fig. 4.2, he or she can either upload a tale to the textbox and get candidate classes derived from the information extraction as described in Section 4.6 or they can select which classes they want to annotate manually.

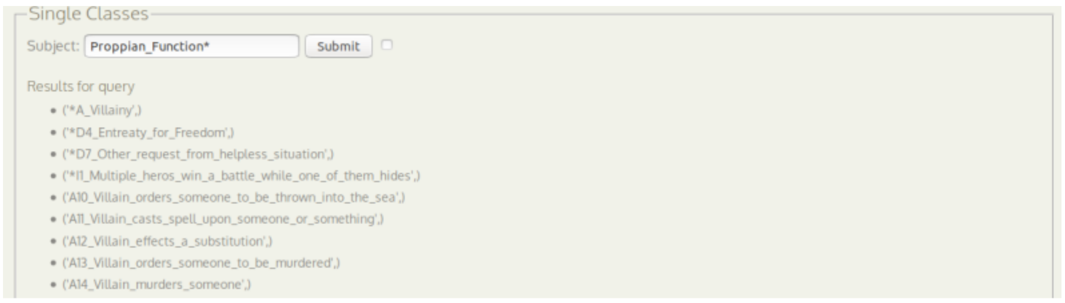

Test your RDF triples for Relations, Entities and Subclasses
Subject: Victim
Predicate:
Object: Villain
Submit

Results for query

```
SELECT ?x ?p ?y WHERE {?x a <https://teaching.gcdh.de/ProppOntology/1.0.1#Villain> .?y a <https://teaching.gcdh.de/ProppOntology/1.0.1#Victim> .?x ?p ?y . }
```

- ('AGW_Cannibal', 'causesHarm', 'AGW_Girl')
- ('AGW_Cannibal', 'gainsInformation', 'AGW_Girl')
- ('AGW_Cannibal', 'threatens', 'AGW_Girl')
- ('GFC_New_Wife', 'causesHarm', 'GFC_Pitipiti')
- ('GFC_New_Wife', 'threatens', 'GFC_Pitipiti')
- ('O_Spirit', 'causesHarm', 'O_Obaraedo')
- ('WSP_Python', 'interdicts', 'WSP_Girl')

Figure 4.5: Triple query for instances of Villain and Victim and their connecting relations



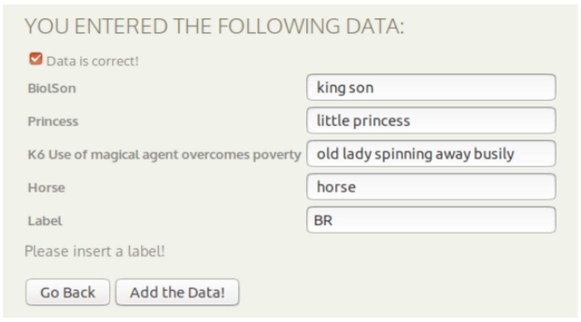
Results for query

- ('*A_Villainy',)
- ('*D4_Entreaty_for_Freedom',)
- ('*D7_Other_request_from_helpless_situation',)
- ('*I1_Multiple_heros_win_a_battle_while_one_of_them_hides',)
- ('A10_Villain_orders_someone_to_be_thrown_into_the_sea',)
- ('A11_Villain_casts_spell_upon_someone_or_something',)
- ('A12_Villain_effects_a_substitution',)
- ('A13_Villain_orders_someone_to_be_murdered',)
- ('A14_Villain_murders_someone',)

Figure 4.6: Single query for the subclasses of Proppian Function

If candidate classes are obtained by using the information extraction, the user confirms the classes by ticking a checkbox. After clicking the submit button, he or she is then guided to an overview page where changes to the text fields can be made. The user is asked to provide a label for the tale to be annotated. This label is used to create the instances of the classes. If the label already exists, a warning will be issued and the user will be asked if the tale that already exists under that label should be updated.

To send the update to the Fuseki server, the user has to confirm that the data is correct, and click the 'Add the Data' button. For each class, an own update query is created.



YOU ENTERED THE FOLLOWING DATA:

☑ Data is correct!

BiolSon	king son
Princess	little princess
K6 Use of magical agent overcomes poverty	old lady spinning away busily
Horse	horse
Label	BR

Please insert a label!

Figure 4.7: Update page after information extraction

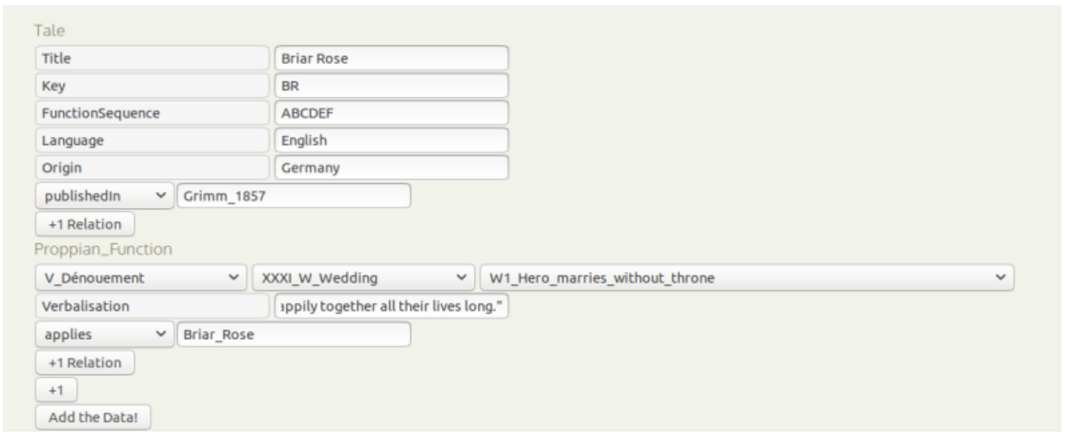
The corresponding SPARQL update for the first of the candidate classes:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

INSERT DATA { <https://teaching.gcdh.de/ProppOntology/1.0.1#BR_BiolSon> a owl:
    namedEntity .
 <https://teaching.gcdh.de/ProppOntology/1.0.1#BR_BiolSon> rdf:type <https://teaching.
     gcdh.de/ProppOntology/1.0.1#BiolSon> .
<https://teaching.gcdh.de/ProppOntology/1.0.1#BR_BiolSon> <https://teaching.gcdh.de/
    ProppOntology/1.0.1#Verbalisation> "king son" .
}
```

Listing 4.1: "Example SPARQL update after information extraction"

The update string is then sent to Fuseki's update endpoint as a POST request and processed there. After successful update, the user sees a message “Update successful”, and the new instance can be accessed in the query search.

If the user decides to enter a manual annotation, he or she is asked to click a set of checkboxes for the specific classes. Subsequently, a set of input fields is generated from the ontology by a query for related data and object properties. Fields correspond to data properties with the class as their domain. The user can choose from a dropdown list of relations, corresponding to the object properties that belong to the class as their domain. For annotations of Proppian functions or persons, a “+1” button allows the user to create another form for additional instances of that class. However, the users should annotate one tale at a time, therefore other classes such as *Anthology* or *Author* can only be added once. Due to performance considerations, the lists for the subclasses of *Proppian_ Function* and *Fictional_ Character* are not generated at runtime, but initialised once when the Flask application is started on the server. If the user does not check the checkbox for *Tale*, he or she will be asked to provide a label for the classes that describe the content of a tale, such as



Tale

Title	Briar Rose
Key	BR
FunctionSequence	ABCDEF
Language	English
Origin	Germany
publishedIn	Grimm_1857

Proppian_Function

V_Dénouement | XXXI_W_Wedding | W1_Hero_marries_without_throne

Verbalisation	ıppily together all their lives long."
applies	Briar_Rose

Figure 4.8: Example of manual annotation of the tale *Briar Rose* and one Proppian function

Proppian_ Function, *Fictional_ Character* or *Task*. If the *Tale* class is included in the form, the label will be taken from the data property *Key*. The label is crucial for the successful annotation of content classes, because it is used to match them to a specific tale.

Conveniently, when an update is sent to the Fuseki server for an individual that already exists, the existing data is not overwritten. New object properties or datatype properties with different values are added to the existing dataset. If the data of a datatype property field is the same as an already existing entry, the value is not added again. Therefore, it is possible for two users to work on an annotation for the same tale.

After the user presses the 'Add the data'-button, an overview page is generated which has to be confirmed before the data is converted to a SPARQL update and sent to the Fuseki update endpoint.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
INSERT DATA {
<https://teaching.gcdh.de/ProppOntology/1.0.1#Briar_Rose> a owl:namedEntity . <https://
    teaching.gcdh.de/ProppOntology/1.0.1#Briar_Rose> rdf:type <https://teaching.gcdh.de/
    ProppOntology/1.0.1#Tale> .
 <https://teaching.gcdh.de/ProppOntology/1.0.1#Briar_Rose> <https://teaching.gcdh.de/
     ProppOntology/1.0.1#Title> "Briar Rose" .
 <https://teaching.gcdh.de/ProppOntology/1.0.1#Briar_Rose> <https://teaching.gcdh.de/
     ProppOntology/1.0.1#Key> "BR" .
 <https://teaching.gcdh.de/ProppOntology/1.0.1#Briar_Rose> <https://teaching.gcdh.de/
     ProppOntology/1.0.1#FunctionSequence> "ABCDEF" .
 <https://teaching.gcdh.de/ProppOntology/1.0.1#Briar_Rose> <https://teaching.gcdh.de/
     ProppOntology/1.0.1#Language> "English" .
 <https://teaching.gcdh.de/ProppOntology/1.0.1#Briar_Rose> <https://teaching.gcdh.de/
```

```
     ProppOntology/1.0.1#Origin> "Germany"
.  }

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
INSERT DATA { <https://teaching.gcdh.de/ProppOntology/1.0.1#BR_W1> a owl:namedEntity .
<https://teaching.gcdh.de/ProppOntology/1.0.1#BR_W1> rdf:type <https://teaching.gcdh.de/
    ProppOntology/1.0.1#W1_Hero_marries_without_throne> .
<https://teaching.gcdh.de/ProppOntology/1.0.1#BR_W1> <https://teaching.gcdh.de/
    ProppOntology/1.0.1#Verbalisation> "And then the prince and Briar Rose were married,
     and the wedding feast was given; and they lived happily together all their lives
    long." .
<https://teaching.gcdh.de/ProppOntology/1.0.1#BR_W1> <https://teaching.gcdh.de/
    ProppOntology/1.0.1#applies> <https://teaching.gcdh.de/ProppOntology/1.0.1#
    Briar_Rose>
.  }
```

Listing 4.2: SPARQL Update corresponding to Fig. 4.8

After the update is successfully processed at the Fuseki update endpoint, a message is shown and the data can be accessed via the queries.

4.6 Information Extraction from Tale Texts

We attempted to extract some of the information encoded in the text semi-automatically. Specifically, nominal phrases that describe characters or animals, and appearances of Proppian functions were of interest. However, nominal phrases of non-living objects that are repeated through the text can indicate a motif, such as the tree that *Cinderella* repeatedly visits which supplies her with the ball gown (Grimm and Grimm, 1857) corresponds to the TMI motif D950 Magic Tree.

The information extraction component of the system can be regarded as its own ontology-driven information extraction system for ontology population. Wimalasuriya and Dou define such a system as:

Definition 1 *An Ontology-based Information Extraction System: A system that processes unstructured or semi-structured natural language text through a mechanism guided by ontologies to extract certain types of information and presents the output using ontologies.*

(Wimalasuriya and Dou, 2010)

While Wimalasuriya and Dou argued that linguistic extraction rules should be part of the ontology, we implemented the natural language processing elements entirely on the Flask side of the application. With a rule based approach, e.g. using regular expressions or gazetteer lists, it would make sense to include it within the ontology. However, this project followed a machine learning approach that used the Python module NeuralCoref [8].

Entity Recognition for Folktale Characters

In a first approach, a set of semantic rules were defined to extract potential candidates of dramatis personae from the text. However, this approach did not yield satisfying results. The main reason might be that the rules for the appearence of characters in tales must naturally be relatively broad. A rule like: $NP : < DT >? < JJ > * < NN >$, would deliver correct nominal phrases, such as *the girl*, but also yield many false positives. Stricter, more sophisticated rules would likely not find entities that are verbalised in a simple manner, like *the man*.

The NLTK toolkit for Python provides a named entity chunker *ne_chunk*[9] that was tested on different texts. Expectedly, fairy tale texts do seldomly supply *named* entities, with exception to some popular tales like *Hans in Luck* or the *Obaradeo* tale discussed before. Usually, characters are introduced in a more general way, e.g. *the girl*. Therfore, the pure named entity recognition task was abandoned.

Since verbalisation of characters is one of the interesting features the ontology is supposed to supply, the focus shifted to the resolution of coreferences instead. The main idea behind using coreferences was that entities or other important features will likely be repeated throughout the text. A satisfyingly working coreference resolution tool would not only provide characters that occur in the text, it would also provide reoccuring motifs, e.g. a tale revolving around an apple tree would yield many coreferences for *apple tree* or *tree*. Using a coreference approach yields results for named entities as well as unnamed entities, which is the most significant advantage and the main reason this approach was chosen.

From the available coreference resolution approaches, the NeuralCoref[10] approach method was found to be the most promising. They trained their word embeddings model on the OntoNotes corpus. Making use of two neural networks, they calculate a score for a word either having an antecedent or not. NeuralCoref uses the Python libraries Spacy for text parsing, and Numpy for training neural networks.

Although NeuralCoref was initially designed for coreference resolution in chatbot systems, their approach seems to work reasonably well on English folktale texts.

The text is first preprocessed using Spacy's *nlp* method[11]. Subsequently, coreferences are resolved using NeuralCoref. Candidate entities of characters are identified from the text by using Spacy's named entity recognition method *ents*, finding tokens from the entire text that are labelled as

[8]https://medium.com/huggingface/state-of-the-art-neural-coreference-resolution-for-chatbots-3302365dcf30
[9]https://www.nltk.org/api/nltk.chunk.html
[10]https://medium.com/huggingface/state-of-the-art-neural-coreference-resolution-for-chatbots-3302365dcf30
[11]https://spacy.io/api/doc

Person. Since this list alone yields very noisy results, the candidate tokens are then compared to the antecedents in the coreference clusters. Candidates that do not appear in the coreference clusters are abandoned. Certainly, this comes with losing some of the classes. However, only considering the tokens labelled as 'Person' yields too many false positives.

Unfortunately, with the comparatively small number of annotated texts, no gold standard can be defined. So far, the information extraction system still needs to be evaluated by eyeballing. For examples of the information extraction, please refer to Section 5.2.

Extraction of Instances of Proppian Functions

For extracting occurences of Proppian Functions, the extensive skos labels provided by the ontology were used. For the time being, only skos:prefLabel fields are used. However, in the future skos:altLabels could be used to identify instances for classes in different languages.

For the information extraction, the text is preprocessed as described above. A SPARQL query yielding the values of all pref:Labels and their corresponding classes is sent to the Fuseki server at the beginning of the text processing. In an earlier attempt, before the Fuseki server was deployed, the list of prefLabels was obtained by parsing the ontology .owl-file. However, this approach was soon found to be too time consuming especially with the import of the TMI-ATU ontology. In addition, the ontology file needs to be replaced each time it undergoes any changes. By using the Fuseki server to retrieve the labels, the system is more sustainable and less prone to errors.

After the coreferences are identified, a list of starting phrases of all the coreference chains is created. Each starting phrase is tokenized and stripped of punctuation. A list of tokenized prefLabels is created. Both lists are then lemmatized using the NLTK WordNetLemmatizer and compared. If one antecedent matches a token in a prefLabel, it is added to the list of potential candidates for that particular class.

The results of both approaches are then handed back to the Flask application, which creates an input form. If a potential person is found, a dropdown list allows the user to select the correct ontology class. If a candidate class is found by the second approach, the class name is shown next to the input field. Users can then change the data and create their own annotation.

Chapter 5

Results

This section reports the results of the application of the Ontology-Driven Information System applied to Southern African Folk Tales. However, to ensure we do not fall into the trap to believe we as non-folklorists "are qualified to speak authoritatively about folkloristic matters" (Dundes, 2005, p.401), we focus on reporting the verifiable results, leaving deeper interpretations of our findings to the interested folkloristically educated scholar. We believe the results the ontology-driven information system can yield are potentially interesting for folklorists who want to compare Proppian analyses of African tales. Furthermore, existing theories about African tales can be reconfirmed.

5.1 Evaluation of Competency Questions

To determine if the Ontology Driven Information System fulfills its purpose, we defined a number of competency questions in Section 3.2. This section shows which SPARQL queries are needed in order to answer these questions. The respective answer is given below the query.

1. Which folk tales fall into a given motif class, e.g. ATU 70-99 Other Wild Animals?
   SPARQL:

```
SELECT ?s WHERE {
  ?s <https://teaching.gcdh.de/ProppOntology/1.0.1#hasClass> <https://teaching.gcdh
      .de/ProppOntology/1.0.1#Wild_Animals> .
        }
```

   Answer:

   - A Bride For The Hare
   - The Old Woman Her Sons And The Python

- The Tale Of Ngcede

2. Which Dramatis Personae appear in a given tale?
   SPARQL:

```
SELECT ?s ?y WHERE {
?s <https://teaching.gcdh.de/ProppOntology/1.0.1#appearsIn> <https://teaching.gcdh.
    de/ProppOntology/1.0.1#When_Hippo_was_Hairy>
  . }
```

   Answer:

   - WHH_ Flames
   - WHH_ Hare
   - WHH_ Hippo

3. Which Proppian functions appear in African folk tales?
   SPARQL:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT  DISTINCT ?t
WHERE {
        ?fun rdf:type ?t.
        ?fun <https://teaching.gcdh.de/ProppOntology/1.0.1#applies> ?tale.
        ?t rdfs:subClassOf* <https://teaching.gcdh.de/ProppOntology/1.0.1#
            Proppian_Function> .
}
```

   Answer: 71 Proppian functions appear in the folktales annotated within the ontology. For a list of all functions, please refer to Appendix A.2

4. How are Dramatis Personae interacting in the African folk tales, e.g. which figures use the "interdiction" relation?
   SPARQL:

```
SELECT  DISTINCT ?s ?p ?o
WHERE {
        ?s <https://teaching.gcdh.de/ProppOntology/1.0.1#interdicts> ?p.
}
```

   Answer:

   - AGW_ Mother → AGW_ Girl
   - BAG_ OLD_ MAN → BAG_ MAN

- O_ Mother → O_ Obaraedo
- RWD_ Hare → RWD_ Zebra
- WSP_ Python → WSP_ Girl

5. Which sequences of Proppian functions appear in a given tale? Which sequences appear in tales in general?
SPARQL (all tales):

```
SELECT  DISTINCT ?inst ?p ?o
WHERE {
   ?inst a <https://teaching.gcdh.de/ProppOntology/1.0.1#Tale> .
   ?inst <https://teaching.gcdh.de/ProppOntology/1.0.1#FunctionSequence> ?o .
}
```

Answer:

- $\gamma 1\delta 1B4Pr6$
- $\gamma 2\delta 2B4E7F1$
- $\beta AC\uparrow HIK\downarrow W$
- $\alpha\beta\gamma\delta\theta aBC\uparrow HIK\downarrow T$
- $\beta 3\gamma 2\delta 2\epsilon 3\eta 1\eta 3\lambda A14E1I1I2W1$
- $\beta\gamma\delta\epsilon 1\zeta 1\eta 3\theta A17B7HI5K10\downarrow$
- $\beta\gamma\delta\epsilon\eta\theta ABC\uparrow FGHIK\downarrow$
- $\beta\epsilon 2\zeta 2aC\uparrow FK6U$
- $\beta\epsilon 2\zeta 2\eta 3\lambda HI\downarrow$
- $\beta\epsilon\zeta 3a2C\uparrow HIK\downarrow$
- $\gamma 2DEFJ\downarrow W$
- $\gamma 2\delta 1\epsilon 2\zeta 1\eta 3\theta 2A2aE1I1$
- $\gamma 2\zeta 2a6G1Pr1Rs2Rs4o$
- $\gamma\delta 2A3A8A14GI1K9$
- $\gamma\delta AIK$
- $\gamma\delta aCK$
- $\eta 1\theta 1A6J1T$
- $\eta 1\theta 1E1Pr5Rs8$
- $\eta 3\theta 1A6J1T$
- $\eta 3\theta 1G3H1J1$

SPARQL (tale given):

```
SELECT  DISTINCT ?o
WHERE {
   <https://teaching.gcdh.de/ProppOntology/1.0.1#How_Jackal_got_his_Markings> <
       https://teaching.gcdh.de/ProppOntology/1.0.1#FunctionSequence> ?o .
}
```

Answer: $\eta 3\theta 1A6J1T$

6. Which Proppian functions follow a given function predominantely, i.e. are there patterns

withing the Proppian sequences?

SPARQL:

```
SELECT  DISTINCT ?s ?p ?o
WHERE {
   ?s a <https://teaching.gcdh.de/ProppOntology/1.0.1#Tale> .
   ?s <https://teaching.gcdh.de/ProppOntology/1.0.1#FunctionSequence> ?o .
FILTER regex(?o, "AC", "i") }
```

Answer:

- $\beta AC \uparrow HIK \downarrow W'$
- $\beta\epsilon 2\zeta 2aC \uparrow FK6U$
- $\gamma\delta aCK$

7. Who is the editor of a tale from a given origin?
SPARQL:

```
SELECT  DISTINCT ?s
WHERE {
   ?s a <https://teaching.gcdh.de/ProppOntology/1.0.1#Editor> .
   ?tale <https://teaching.gcdh.de/ProppOntology/1.0.1#Origin> ?o .
FILTER regex(?o, "NDebele", "i") }
```

Answer:

- Alexander McCall Smith
- Harold Scheub
- Nick Greaves
- Phyllis Savoury

8. How are Proppian functions verbalised, i.e. which words are used to describe events that fall into a given function class?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT  DISTINCT ?s ?o WHERE {
        ?fun rdf:type ?s.
        ?fun <https://teaching.gcdh.de/ProppOntology/1.0.1#applies> ?tale.
        ?s rdfs:subClassOf* <https://teaching.gcdh.de/ProppOntology/1.0.1#
            Proppian_Function> .
    ?fun <https://teaching.gcdh.de/ProppOntology/1.0.1#Verbalisation> ?o.
}
```

Answer: The query delivers a list of 159 functions and their verbalisations. A list limited to 25 functions can be found in Appedix A.3

SPARQL (for a specific function and its subfunctions)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT  DISTINCT ?s ?o
WHERE {
        ?fun rdf:type ?s.
        ?fun <https://teaching.gcdh.de/ProppOntology/1.0.1#applies> ?tale.
        ?s rdfs:subClassOf* <https://teaching.gcdh.de/ProppOntology/1.0.1#
            VIII_A_Villainy> .
    ?fun <https://teaching.gcdh.de/ProppOntology/1.0.1#Verbalisation> ?o.
}
```

Answer:

- A14 Villain murders someone → The lion killed the queens son
- A14 Villain murders someone → The python then started eating the men, woman and children of the village.
- A17 Villain makes a threat of cannibalism → The cannibal, though, was too quick and had seized her before she could seal off the cave mouth.
- A2a Forcible seizure of magical helper → The first weapon breaks
- A3 Villain pillages or spoils crops → Gradually it ate up all the grain in the village.
- A6 Villain causes bodily injury → Hare crept up and threw in the burning embers, blowing on them until he had a fine blaze going.
- A6 Villain causes bodily injury → He struck Jackal on his flanks with the fire, which set the animal’s coat ablaze.
- A8 Villain demands or entices his victim → The monster asked her for more food, saying it’s hunger was not nearly satisfied.
- VIII A Villainy → She would pass by Pitipiti’s fields and jeer at her, asking her why she grew crops if she had no mouths to feed.
- VIII A Villainy → Then, quickly substituting his own bird, he passed it to the blind man and put the coloured bird into his own pouch.

9. Is there a dominating interaction between certain classes of Dramatis Personae?
   SPARQL:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT  DISTINCT ?s ?p ?o
WHERE {
        ?s rdfs:subClassOf* <https://teaching.gcdh.de/ProppOntology/1.0.1#
            Dramatis_Personae> .
        ?o rdfs:subClassOf* <https://teaching.gcdh.de/ProppOntology/1.0.1#
            Dramatis_Personae> .
        ?t rdf:type ?s.
        ?t2 rdf:type ?o.

  ?t ?p ?t2 .
}
```

   Answer:

   - Dispatcher interdicts Hero
   - Helper combats Villain
   - Helper defeats Villain
   - Helper interdicts Hero
   - Hero attackedBy Villain
   - Hero combats Villain
   - Hero defeats Villain
   - Hero frees Victim
   - Hero pursuedBy Villain
   - Hero reactsTo Helper
   - Hero testedBy Donor
   - Seeker seeks Helper
   - Seeker seeks Hero
   - Victim hasFather Seeker
   - Victim helpsEnemy Villain
   - Victim relatedTo Hero
   - Villain causesHarm Victim
   - Villain combats Hero
   - Villain deceived Hero
   - Villain gainsInformation Victim
   - Villain interdicts Victim
   - Villain threatens Victim

5.2 Information Extraction

The information extraction system is designed to find occurences of persons or agents in stories, and to identify potential candidates of Proppian functions. In the following section, the results of the information extraction for two different fairy tales in English translations are given. The first tale, *Jabu and the Lion*[1] is a traditional Zulu story, in which a number of named entities occur. The narration of the story is mainly transported by direct speech. The text contains utterances in isiZulu, such as *Ngane*[2] or *Mfana*[3], traditional Zulu names and their phonetic description, and some Afrikaans expressions like *kraal*[4] or *koppie*[5]. It contains human and animal characters (lion, jackal, donkey) and belongs into the category of trickster tales.

The information extraction system identifies the classes shown in table 5.1.

Table 5.1: Information extraction results for the tale *Jabu and the Lion*

Class	Value	Correct
Entity	Thabo	yes
Entity	Jabu	yes
Entity	Bhubesi	yes
Entity	Sipho	yes
Entity	Nkosi	yes
Cow	father's cow	yes
BiolFather	father's cow	no
Human	they, these stupid humans	yes
Function Rs6	my king, the king of the animals	no
Function Pr3	my king, the king of the animals	no
Animal	my king, the king of the animals	yes
Princess' Father	father's cow	no

[1]http://www.canteach.ca/elementary/africa3.html
[2]Come on!
[3]Boy
[4]Homestead
[5]Hill

A complete manual annotation would yield the following classes and values:

Table 5.2: Manually derived results for the tale *Jabu and the Lion*

	Classes	Value
Entities	Victim, Boy, Son	Jabu, young herdboy, he, friend, the boy, their shepherd
	Villain, Animal	Bhubesi, the lion, king of beasts, my king, king of the animals, a majestic animal, NKosi[6], he, Lion
	Hero, Animal	Mpungushe, the jackal, Jackal, he
	Dispatcher, Boy	Sipho, he, Jabu's friend
	Animal	his father's cattle, the cows, a large herd, they
	Boy	Thabo, a sloppy herdboy, a fellow who ran with his head in the clouds
Proppian Functions	*Intitial Situation α*	"There was a young herdboy named Jabu ..."
	Absentation β	Jabu takes the cows to the river
	Reconnaissance $\epsilon 3$	Jabu approaches the lion to find out why he is crying.
	Delivery (of information) $\zeta 3$	Jabu sees that the lion stepped into a trap
	Persuasion $\eta 1$	The lion convinces Jabu to free him from the trap.
	Villainy A	Once freed, the lion comes towards Jabu in order to eat him.
	Mediation B	Jabu tells the jackal how the lion is treatening him.
	Beginning Counteraction C	The jackal asks to see the trap.
	Struggle H	The jackal asks the lion to stick his head into the trap, then closes the bar.
	Victory I	The lion's head is stuck in the trap. He is no longer a threat to Jabu and his herd.

The second tale for the investigation of the information extraction system, is the English translation of Grimm's *Rumpelstilzchen*, *Rumpel-stilts-kin.*[7]. The tale contains the villain's song, and a number of different names, which the queen/miller's daughter guesses in order to prevent Rumplestilskin to take her child.

The information extraction tools yields the following candidate classes.

Class	Value	Correct
Man	he, the little man	yes
Function M	your task, a fresh task	no
BiolDaughter	poor miller daughter	yes
Task	your task, a fresh task	yes
Daughter	poor miller daughter	yes

Manual annotation of Proppian functions, and character classes with their textual representation yields the classes shown in Table 5.2.

[7]`https://en.wikisource.org/wiki/Grimm%27s_Goblins_(1876)/Rumpel-Stilts-Kin`

	Classes	Value
Entities	Hero, Wife, Woman	beautiful daughter, the girl, she, the maiden, the miller's daughter, the queen, lady
	Villain, Man	a droll-looking little man, he, the dwarf, funny little man the little gentleman, Rumplestiltskin, her little visitor
	Helper, Man	one of the messengers
	Man, Husband	the king, he
Proppian Functions	*Initial Situation α*	In a certain kingdom once lived a poor miller ...
	Absentation β	The king orders the miller's daughter to be brought before him
	Reconnaissance $\epsilon 1$	The little man asks the girl why she is crying.
	Villain receives information $\zeta 1$	The little man learns about the impossible task to spin gold from straw.
	Complicity θ	The miller's daughter agrees to let the little man help her.
	Villainy A	Although, she has nothing left to offer as a payment, the little man helps the girl a last time, but asks for her first-born as a reward.
	Liquidation of Lack K	After the queen begs the little man not to take her child, he agrees to leave it with her for three more days.
	Difficult Task M	Within three days, the queen must guess the little man's name or her child will be taken away.
	Solution N	After the third day, a messenger tells the queen about the song, and she guesses the name correctly.
	Exposure Ex	The villain is exposed, and leaves the court.

5.3 Case Studies

5.3.1 Representation of Characters in African Tales

A query for individuals of fictional characters in all tales that have been annotated, their corresponding class and their verbalisation can be made very easily.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?s ?p ?o
WHERE {
  ?s a ?p .
  ?p rdfs:subClassOf* <https://teaching.gcdh.de/ProppOntology/1.0.1#Fictional_Character>
       .
  ?s <https://teaching.gcdh.de/ProppOntology/1.0.1#Verbalisation> ?o .

}
```

Listing 5.1: Query for characters in tales, their corresponding class and verbalisation

The entire list of results can be found in Appendix A.4. Characters in the annotated tales mainly belong to three upper classes, *Animal* (34), *Family Member* (25) and *Dramatis Personae* (45). Of course, one character can belong to multiple of those upper classes. The dramatis personae fall into seven categories as defined by (Propp, 1968). Fig. 5.1 shows the distribution of Proppian characters in the corpus.

The classes *Villain* and *Hero* appear 17 and 16 times in the corpus of annotated tales. Five instances of *Victim*, three instances of *Donor*, and two instances of *Helper* occur, *Seeker* and *Dispatcher* both appear exactly once. There was no instance of either *Princess* or *Princess' Father*.

The distribution of the subclasses of *Family Member* are shown in Fig. 5.2. The most prevalent class in the family subtree is *Father*, which appears four times in all tales. The classes *Mother*, *Wife*, *Daughter*, *Child* and *Brother* each occur three times. Two instances of *Husband* and one instance of each *Sister* and *BiolMother* can be found in the corpus.

Since one character can belong to multiple classes, we can investigate the distribution of Proppian roles among other classes.

While the distribution of the *Hero* and *Villain* instances is predominantly within the *Animal* category, it can be seen that the *Victim* instances mainly belong into the female classes.

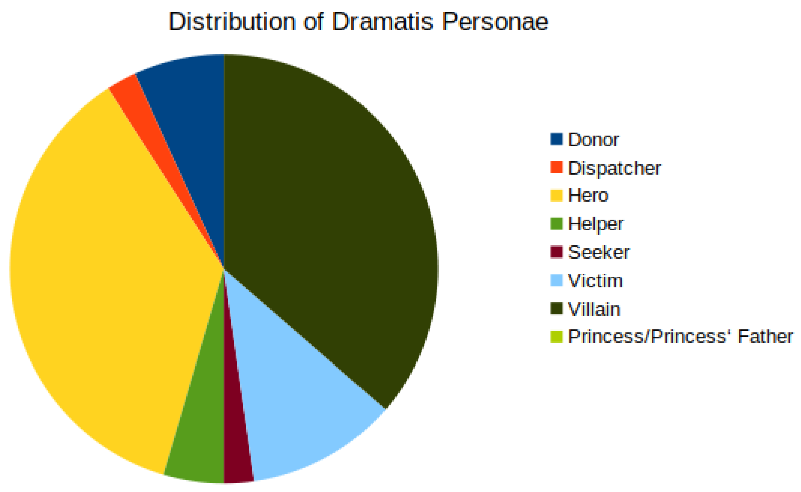


Figure 5.1: Distribution of Dramatis Personae

From the list of verbalisations of characters in A.1, we can see that seldomly a character is introduced by name. They are rather verbalised in a more general, descriptive way, like *the girl*. The relation between characters plays a significant role in the verbalisation. Characters are often described as someone's relative, such as *his mother*. For the verbalisation of animals, two patterns are a observable. Animals are either introduced in a general way, such as *the hare*, or the characterisation of the animals is treated as a name, such as *Lion* without any article.

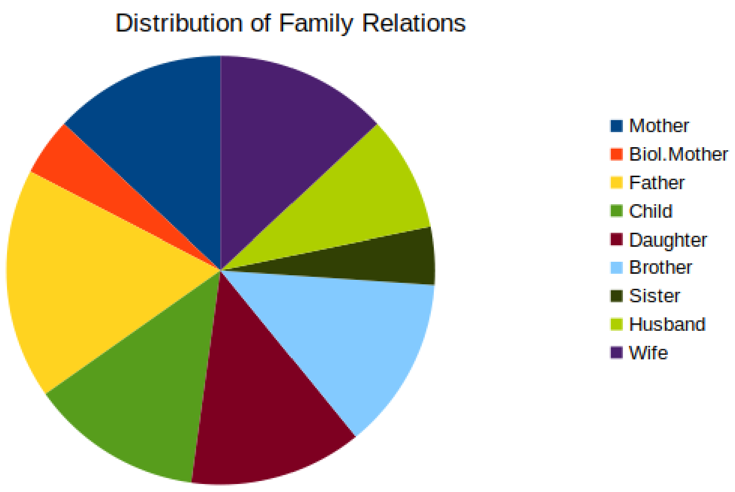


Figure 5.2: Distribution of Subclasses of Family Member

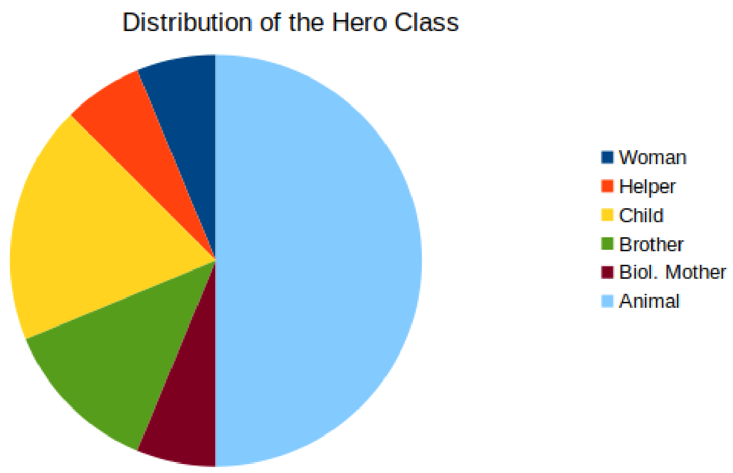


Figure 5.3: Distribution of the Hero class among other character classes

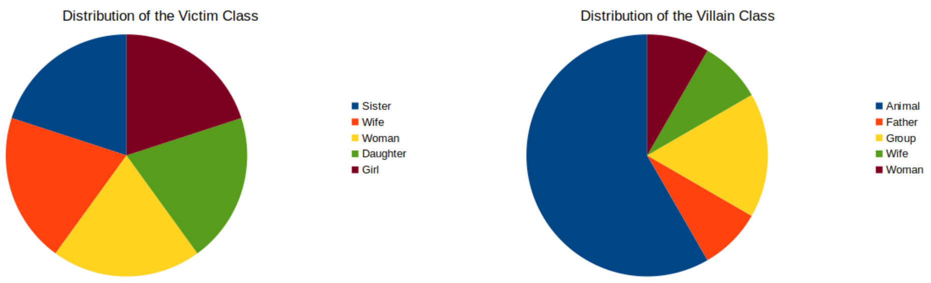


Figure 5.4: Distributions of the classes Victim and Villain among the other character classes

5.3.2 Structure of Southern African Tales

Distribution of Functions

From 20 tales that have been annotated so far, we can derive some structural features of Southern African tales. If we neglect the one case of the *Obaradeo* tale, where the function sequence starts with the *initial situation* α, as discussed in Section 2.2.3 and assume the sequence starts with the following function *Absentation* β, the tales start with three different functions. The ending of the tales is more diverse, with ten different functions.

Table 5.3: Distribution of introductory and conclusive functions

Beginning	End
Interdiction γ: 8	*Return* $\downarrow$: 4
Absentation β: 8	*Liquidation of Lack K*: 3
Trickery η: 4	*Transfiguration T*: 3
	Wedding W : 3
	Pursuit P: 1
	Provision of Magical Agent F: 1
	Unrecognized Arrival o: 1
	Rescue Rs: 1
	Punishment U: 1
	Victory I: 1
	Branding J: 1

Propp divided the 31 functions into five categories, *Preparation*, *Complication*, *Functions of the Donor*, *Struggle*, and *Dénouement*. The annotated tales have been investigated as to how prevalent these five categories are. The following table shows the mean division of functions relative to sequence length.

While 40 % of the mean function sequence length consists of preparative functions ($\alpha - \theta$), and 27 % consist of function from the *Struggle* category, only 5 % of the sequence length is made up from *Denouement*. In Table 5.3, only four of the eleven functions fall into that category. The rest of the end functions belong into the *Struggle* category, with exception of *Provision of Magical Agent F* (*Functions of the Donor*).

Table 5.4: Mean division of functions by category relative to sequence length

Preparation	*Complication*	*Functions of the Donor*	*Struggle*	*Denouement*
0.40	0.19	0.1	0.27	0.05

Sequence	Length
$\gamma 1\delta 1B4Pr6$	4
$\gamma 2\delta 2B4E7F1$	5
$\beta AC\uparrow HIK\downarrow W$	9
$\alpha\beta\gamma\delta\theta aBC\uparrow HIK\downarrow T$	14
$\beta 3\gamma 2\delta 2\epsilon 3\eta 1\eta 3\lambda A14E1I1I2W1$	12
$\beta\gamma\delta\epsilon 1\zeta 1\eta 3\theta A17B7HI5K10\downarrow$	13
$\beta\gamma\delta\epsilon\eta\theta ABC\uparrow FGHIK\downarrow$	16
$\beta\epsilon 2\zeta 2aC\uparrow FK6U$	10
$\beta\epsilon 2\zeta 2\eta 3\lambda HI\downarrow$	8
$\beta\epsilon\zeta 3a2C\uparrow HIK\downarrow$	10
$\gamma 2DEFJ\downarrow W$	7
$\gamma 2\delta 1\epsilon 2\zeta 1\eta 3\theta 2A2aE1I1$	9
$\gamma 2\zeta 2a6G1Pr1Rs2Rs4o$	8
$\gamma\delta 2A3A8A14GI1K9$	8
$\gamma\delta AIK$	5
$\gamma\delta aCK$	5
$\eta 1\theta 1A6J1T$	5
$\eta 1\theta 1E1Pr5Rs8$	5
$\eta 3\theta 1A6J1T$	5
$\eta 3\theta 1G3H1J1$	5
Mean	8.15

Sequence Length

The length of the 20 function sequences ranges between 16 and five functions. The average sequence length is 8.15 functions.

Patterns of Functions

Spatial distance seems to play a certain role in African tales. The functions *Departure* $\uparrow$ and *Return* $\downarrow$ appear alone or together in eight of 20 tales. In four tales, both *Departure* $\uparrow$ and *Return* $\downarrow$ can be found as a pair. The *Departure* $\uparrow$ function appears without a corresponding *Return* $\downarrow$ once, while *Return* $\downarrow$ appears on its own three times.

Another prominent pattern is the pair *Villainy A/Villainy Lack a* and the corresponding function *Liquidation of Lack K* and their subfunctions. The pair appears together in nine tales. Additionally, *Villainy A/Villainy Lack a* appears alone in five sequences, with one sequence where three different subfunctions of *Villainy A* (A3, A8, and A14) appear consecutively. This indicates that in 25 % of the analysed tales, some form of harm is done to the hero or his/her family members without being resolved later. Consistently to Propp's theory, there is no occurence of *Liquidation of Lack* without a preceding *Villainy A/Villainy Lack a*. The distance between *Villainy A/Villainy Lack a* and *Liquidation of Lack K* ranges between one function and seven functions.

Chapter 6

Discussion

In this chapter, we want to briefly discuss the findings as presented in Chapter 5, elaborate on the limitations of the system and give an outview on future work.

In Chapter 5, we showed that all the competency questions that were defined before the initial ontology was created, can be answered by applying SPARQL queries. More advanced investigations, especially when analysing function sequences of Proppian functions, could be made by application of Regex patterns.

6.1 Characters in African tales

Regarding the representation of agents in the corpus of tales that were annotated, we see that the Dramatis Personae mainly consist of the Proppian roles *Hero* and *Villain*. Especially the lack of *Donor* figures seems to be characteristic for African tales. As expected, animal characters play a dominant role in the tales. Noticably, they do not seem to fulfill roles beyond *Villain* and *Hero*. This could be an indicator that a clear separation of characters into good and bad is characteristic for animal tales.

6.2 Proppian Functions

The African tales that were investigated showed a clear preference for the Proppian functions *Interdiction* γ, *Absentation* β and *Trickery* η at the beginning of the tales. This indicates, that the preparatory functions Propp defined fit relatively well when applied to Southern African tales. Interestingly, the diversity of ending functions as depicted in Table 5.3 might allow some intriguing interpretations. While the functions belonging to the *Dénouement* class symbolise some sort of reward for the hero's struggles, they only appear four times as final functions in the tales that were studied in this project. This could mean that the reward for heros in African tales is not to gain something, e.g. a throne, the princess, monetary reward, or fame, but to restore the status from the beginning of a tale, e.g. returning home, liquidation of lack brought onto the hero by the villain, victory, punishment of the villain, the rescue of the victim. This particularity should further be investigated as the population of the ontolotgy grows.

The appearance of the functions *Return* $\downarrow$ and *Departure* $\uparrow$ on their own could be an indicator towards the prevalence of transformation patterns, in this case spatial transformation, which Harold Scheub found "reveal[ing] the way people of the region survived the onslaught of colonialism." (Scheub, 2010, p.20)

Interestingly, functions belonging into the *Preparation* category take up the most room in African tales. While functions of *Struggle* are prominent as well, almost no space is dedicated to *Functions of the Donor* or *Dénouement*. This indicates, that the mediator function of the donor is not as prevalent in African tales as it is for Russian magic tales, which Propp analysed (Propp, 1968). While the *Dénouement* segment groups more functions than any other segment, it still does not play a role for the African tale. This might indicate that the functions within this group, maybe with exception of *Difficult Task* and *Solution* are too specific to be applied to the Southern African tale.

It might be worth studying the function sequence endings as presented in Chapter 5 in greater detail. Folklorists might come to the conclusion that an alternative to *Dénouement* with an new set of functions might be worth defining for African tales. Our findings indicate a lack of individual reward, e.g. monetary, which is in line with previous analyses. (Reuster-Jahn, 2002)

While for the Igbo tale, Nwachukwu states: "Part of the aesthetics of the Igbo tale resides in its stock characterization as well as in the naming of the dramatis personae. Characterization and naming enhance the symbolic complexity of a tale." (Nwachukwu-Agbada, 1991, p.29), no focus on named characters in Southern African tales could be found in our investigation. He further claims that "The significance of certain numbers in Igbo rituals somehow finds a similar expression in folktales. The most recurring numbers are two, three, four, and seven."(Nwachukwu-Agbada, 1991, p.32)

From the tales studied, we could not find a similar dominance of numbers for Southern African tales.

6.3 Information Extraction

The results presented in Section 5.2 indicate that the information extraction system works reasonably well for finding folk tale characters. While the tale *Jabu and the Lion* contained a number of unfamiliar words in isiZulu, only traditional names were taken into consideration for possible characters. Furthermore, while mentioned in the text, the main character's brother, sister and mother, did not occur in the results, since all of them were only mentioned once. These characters do not play a significant role for the tale, and are therefore consequently omitted for the annotatation. For the processing of the traditional German tale *Rumplestiltskin* in it's English translation, the villains's name was not among the candidate entities. Given that the name only appears at the end of the tale as part of a song and an utterance of the hero character, this result is not suprising. On the other hand, names that were uttered by the queen in an attempt to guess the villain's name were correctly omitted.

These findings indicate that the approach to perform a coreference resolution to find persons occurring in a tale, makes the system robust against picking up characters that do not play a role in the tale.

Regarding Proppian functions, the system does not perform well. Most of the function candidates were incorrect. This is mainly due to the approach of parsing the skos:prefLabels for information about the classes. However, as the ontology grows and more tales are annotated, we could make use of already existing verbalisations of Proppian functions which would presumably yield better results.

A large enough data set with annotations of Proppian functions could be used to train a machine learning algorithm that classifies segments of folk tale texts into classes of Proppian functions.

6.4 Limitations

6.4.1 Parallel Textprocessing

The Ontology-Driven information system has some limitations with regard to parallel usage of the information extraction system. Unfortunately, some of the classes needed for the text preprocessing, especially the WordNetLemmatizer does not provide means to run in a multithread environment. Therefore, only one user at a time can extract candidate classes from a given text.

6.4.2 Additional Classes for the Description of African Tales

The system cannot provide means to investigate one major feature of African storytelling, that is performance. Especially "call and response features where the audience echoes and encourages the speaker" (Porter, 1995) do not find their way into the rather rigid structuralistic approach of Vladimir Propp.

6.4.3 Fulltexts

As by design, no fulltexts are stored in the context of the ontology. We do not see this as a limitation of the usability of the system. However, first time users might expect to be able to access the entire tale and not only the verbalisations stored when annotating functions and characters. This could be achieved by storing the fulltexts as annotated XML-TEI files and referencing the verbalisations by using pointers to the specific parts of the document. However, copyright aspects need to be taken into consideration when following this approach.

6.4.4 Natural Language Questions

Efforts have been made to generate the SPARQL queries answering the competency questions in Section 3.2 automatically. However, a natural language to SPARQL system would either have to reply on a extensive rule system or needs to be trained on a large set of questions and corresponding queries if following a machine learning approach. Unfortunately, the implementation of this feature would exceed the scope of this project. However, for the system at hand such a feature would certainly be useful, especially since it would allow users with lower levels of IT-profiency to use it in a more sophisticated way. Related attempts have been made by the ORAKEL project (Cimiano et al., 2008) or (Kim and Cohen, 2013).

6.5 Future Work

In the second chapter of this thesis, we discussed the importance of performance in African tales. As the ontology grows, maybe with regard to additional media type such as video and voice recordings, it should be considered if features like facial expressions, reactions of the audience, interaction between narrator and audience, degree of attention, and composition of the audiencce "from the standpoint of age, sex, class or other social division" (Crowley, 1969) should be added as datatype properties.

Initially, it was planned to add the possibility to visualize findings, e.g. by showing origins of tales on a map. However, during the course of the project, the focus needed to remain on building the system itself. Existing visualisation libraries for javascript could be used to create visually appealing graphics from the query results on browser side.

Measuring occurence of function pairs and their distance, as discussed in Chapter 5, could be automated with relatively low effort. This feature would certainly become more interesting as the population of the ontology grows.

6.6 Conclusion

We showed how an ontology-driven information system for the domain of Southern African folktales in the context of Vladimir Propp's *Morphology of the Folktale* can be built, and which expressive value such a system can have. We showed that all investigations that were initially anticipated in form of comptency questions can be easily translated into SPARQL queries. In order to make the system more attractive for users with lower IT-profiency, queries for triples and single classes can be made from an easy-to-use input form. We investigated the structure of the Southern African folktales that were annotated and discussed some particularities, such as the low significance of the *Functions of the Donor*. The Ontology-Driven Information System has the potential to grow when new tales are added. When a large corpus of analysed tales is available, new investigations in terms of structure, characters or comparison of analyses, as in the case of the *Obaradeo* tale discussed in Chapter 2.2.4, can be made. Since the system is freely accessible, it could spark the scholarly discourse when it comes to applying Propp's theory to African folklore.

Chapter 7

Acknowledgements

This work would not have been possible without the help of a number of people.

First of all, I would like to thank Prof. Caroline Sporleder for her interest in my idea to work on African tales and her support and encouragement throughout my Master's degree. Furthermore, I would like to thank Prof. Wolfgang May, who agreed to be my second supervisor and who saw potential in my ontology for interesting applications in the future. I am deeply grateful to Thierry Declerck for his openess, his helpful insights into ontologies and their application in folktale research.

Many thanks to Prof. Fonou Dombeu from the University of KwaZulu-Natal, who first sparked the idea in me to build an ontology about folktales.

Tobias Schäfer and Stefan Ziehe, who were incredibly helpful whenever I had technical questions. Siyabonga to Siyanda Sikobi and Nokubonga Mkhize for their help with the translation of some of the labels into isiZulu.

Furthermore, I would like to thank Yuvika Singh and Danielle Russell for help with the initial annotations as well as Yasar Mahomed Abbas for his help with the first conceptualisation of the initial ontology.

Bibliography

Yasar Mahomed Abbas, Franziska Pannach, Danielle Russel, and Yuvika Singh. Report ontologies and knowledge bases, 2018. Unpublished Student Research Paper.

Alexander Nikolaevich Afanasyev. *Russian Folk-Tales*. E. P. Dutton & Company, New York, 1916.

Chukwuma Azuonye. Morphology of the Igbo folktale: Ethnographic, historiographic and aesthetic implications. *Folklore*, 101(1):36–46, 1990.

Philipp Cimiano, Peter Haase, Jörg Heizmann, Matthias Mantel, and Rudi Studer. Towards portable natural language interfaces to knowledge bases–the case of the ORAKEL system. *Data & Knowledge Engineering*, 65(2):325–354, 2008.

Daniel J Crowley. The uses of African verbal art. *Journal of the Folklore Institute*, 6(2/3):118–132, 1969.

Thierry Declerck and Lisa Schäfer. Porting past classification schemes for narratives to a linked data framework. In *Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage*, pages 123–127. ACM, 2017.

Thierry Declerck, Anastasija Aman, Martin Banzer, Dominik Machácek, Lisa Schäfer, and Natalia Skachkova. Multilingual ontologies for the representation and processing of folktales. In Anca Dinu, Petya Osenova, and Cristina Vertan, editors, *Proceedings of the First Workshop on Language technology for Digital Humanities in Central and (South-)Eastern Europe*, pages 20–24. INCOMA Ltd, 9 2017a.

Thierry Declerck, Antónia Kostová, and Lisa Schäfer. Towards a linked data access to folktales classified by Thompson's motifs and Aarne-Thompson-Uther's types. In *Proceedings of Digital Humanities 2017*. ADHO, 8 2017b.

Alan Dundes. Folkloristics in the twenty-first century (AFS invited presidential plenary address, 2004). *The Journal of American Folklore*, 118(470):385–408, 2005.

Roy T Fielding and Richard N Taylor. *Architectural styles and the design of network-based software architectures*. University of California, Irvine Irvine, USA, 2000.

Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries*, 3(2):115–130, Aug 2000. ISSN 1432-5012. doi: 10.1007/s007999900023. URL `https://doi.org/10.1007/s007999900023`.

Nick Greaves, editor. *When the Hippo Was Hairy*. Lutterworth Press, 1990.

Jacob Grimm and Wilhelm Grimm, editors. *Kinder- und Hausmärchen*. Verlag der Dieterichschen Buchhandlung, Göttingen, 7. edition, 1857.

Jin-Dong Kim and Kevin Bretonnel Cohen. Natural language query processing for SPARQL generation: A prototype system for SNOMED CT. In *Proceedings of BioLINK 2013*, pages 32–38, 2013.

Nikolina Koleva. *Ontology-based iterative detection of characters and their recognition in folktales*. Bachelor Thesis, Saarland University, 2011.

Nikolina Koleva, Thierry Declerck, and Hans-Ulrich Krieger. An ontology-based iterative text processing strategy for detecting and recognizing characters in folktales. In Jan Christoph Meister, editor, *Digital Humanities 2012 Conference Abstracts*, pages 467–470. University of Hamburg, Hamburg University Press, 2012.

Piroska Lendvai, Thierry Declerck, Sándor Darányi, and Scott Malec. Propp revisited: Integration of linguistic markup into structured content descriptors of tales. In *Digital Humanities 2010*. Oxford University Press, 7 2010.

Scott Malec. Autopropp: Toward the automatic markup, classification, and annotation of Russian magic tales. In P. Lendvai, editor, *Proceedings of the First International AMICUS Workshop on Automated Motif Discovery in Cultural Heritage and Scientific Communication Texts*, pages 112–115, 2010.

Mark A. Musen. The protégé project: a look back and a look forward. *AI Matters*, 1(4):4–12, 2015. doi: 10.1145/2757001.2757003. URL `https://doi.org/10.1145/2757001.2757003`.

Natalya Noy and Deborah L McGuinness. Ontology development 101: A guide to creating your first ontology. KSL-01-05, Knowledge Systems Laboratory, Stanford University, 2001.

Justus Obi Joseph Nwachukwu-Agbada. The Igbo folktale: Performance conditions and internal characteristics. *Folklore Forum*, 24(1):19–35, 1991.

Ikechukwu Okodo. Obaraedo: Conformity to Proppian morphology. *AFRREV IJAH: An International Journal of Arts and Humanities*, 1(2):100–111, 2012.

Federico Peinado, Pablo Gervás, and Belén Díaz-Agudo. A description logic ontology for fairy tale generation. In *Procs. of the Workshop on Language Resources for Linguistic Creativity, LREC*, volume 4, pages 56–61, 2004.

Laurence M Porter. Lost in translation: from orature to literature in the West African folktale. In *Symposium: A Quarterly Journal in Modern Literatures*, volume 49, pages 229–239. Taylor & Francis, 1995.

Vladimir Propp. *Morphology of the Folktale*, volume 10. University of Texas Press, 1968.

Uta Reuster-Jahn. Gute und schlechte Ausgänge in europäischen Märchen und in den Volkserzählungen der Mwera in Tansania. *Märchenspiegel*, 13(2):17–18, 2002.

Antonia Scheidel. *An Augmented Annotation Scheme for Fairy Tales*. Bachelor Thesis, Saarland University, 2010.

Harold Scheub, editor. *African Tales*. Univ. of Wisconsin Press, 2005.

Harold Scheub. *The uncoiling python: South African storytellers and resistance*. Ohio University Press, 2010.

Alexander McCall Smith, editor. *Children of Wax: African Folk Tales*. Canongate, 1989.

Tania Tudorache, Jennifer Vendetti, and Natalya Fridman Noy. Web-protege: A lightweight OWL ontology editor for the web. In *OWLED*, 2008.

Daya C. Wimalasuriya and Dejing Dou. Ontology-based information extraction: An introduction and a survey of current approaches. *Journal of Information Science*, 36(3):306–323, 2010. doi: 10.1177/0165551509360123. URL `https://doi.org/10.1177/0165551509360123`.

Burcu Yildiz and Silvia Miksch. Ontology-driven information systems: Challenges and requirements. In *International Conference on Semantic Web and Digital Libraries. Indian Statistical Institute Platinum Jubilee Conference Series*, pages 35–44, 2007.

Appendix A

Appendix

A.1 Proppian Functions

- α The initial situation. A text may only have a single Initial Situation function.

1. A member of a family leaves home (the hero is introduced);
2. An interdiction is addressed to the hero ('don't go there', 'go to this place');
3. The interdiction is violated (villain enters the tale);
4. The villain makes an attempt at reconnaissance (either villain tries to find the children/jewels etc; or intended victim questions the villain);
5. The villain gains information about the victim;
6. The villain attempts to deceive the victim to take possession of victim or victim's belongings (trickery; villain disguised, tries to win confidence of victim);
7. Victim taken in by deception, unwittingly helping the enemy;
8. Villain causes harm/injury to family member (by abduction, theft of magical agent, spoiling crops, plunders in other forms, causes a disappearance, expels someone, casts spell on someone, substitutes child etc, commits murder, imprisons/detains someone, threatens forced marriage, provides nightly torments); Alternatively, a member of family lacks something or desires something (magical potion etc);
9. Misfortune or lack is made known, (hero is dispatched, hears call for help etc/ alternative is that victimized hero is sent away, freed from imprisonment);
10. Seeker agrees to, or decides upon counter-action;
11. Hero leaves home;

12. Hero is tested, interrogated, attacked etc, preparing the way for his/her receiving magical agent or helper (donor);
13. Hero reacts to actions of future donor (withstands/fails the test, frees captive, reconciles disputants, performs service, uses adversary's powers against them);
14. Hero acquires use of a magical agent (directly transferred, located, purchased, prepared, spontaneously appears, eaten/drunk, help offered by other characters);
15. Hero is transferred, delivered or led to whereabouts of an object of the search;
16. Hero and villain join in direct combat;
17. Hero is branded (wounded/marked, receives ring or scarf);
18. Villain is defeated (killed in combat, defeated in contest, killed while asleep, banished);
19. Initial misfortune or lack is resolved (object of search distributed, spell broken, slain person revived, captive freed);
20. Hero returns;
21. Hero is pursued (pursuer tries to kill, eat, undermine the hero);
22. Hero is rescued from pursuit (obstacles delay pursuer, hero hides or is hidden, hero transforms unrecognizably, hero saved from attempt on his/her life);
23. Hero unrecognized, arrives home or in another country;
24. False hero presents unfounded claims;
25. Difficult task proposed to the hero (trial by ordeal, riddles, test of strength/endurance, other tasks);
26. Task is resolved;
27. Hero is recognized (by mark, brand, or thing given to him/her);
28. False hero or villain is exposed;
29. Hero is given a new appearance (is made whole, handsome, new garments etc);
30. Villain is punished;
31. Hero marries and ascends the throne (is rewarded/promoted).

A.2 Answer to Competency Question 3

Which Proppian functions appear in African folk tales?

- A14 Villain murders someone
- A17 Villain makes a threat of cannibalism
- A2a Forcible seizure of magical helper
- A3 Villain pillages or spoils crops
- A6 Villain causes bodily injury
- A8 Villain demands or entices his victim
- B4 Misfortune is announced
- B7 A lament is sung
- E1 Hero withstands or does not withstand a test
- E7 Hero performs some other service
- F1 Agent is directly transferred
- G1 Hero flies through the air
- G3 He is led
- H1 They fight in an open field
- I1 The villain is beaten in open combat
- I2 He is defeated in a contest
- I5 He is killed without a preliminary fight
- III δ Violation
- II γ Interdiction
- IV ϵ Reconnaissance
- IX B Mediation
- I β Absentation
- J1 Brand is applied to the body
- K10 A captive is freed
- K6 Use of magical agent overcomes poverty
- K9 A slain person is revived
- Pr1 Pursuer flies after the hero
- Pr5 Pursuer tries to devour the hero
- Pr6 The pursuer attempts to kill the hero
- Rs2 Hero flees, placing obstacles in the path of his pursuer
- Rs4 Hero hides himself during his flight
- Rs8 He does not allow himself to be devoured
- VIII A Villainy
- VIII a Villainy Lack
- VII θ Complicity
- VI η Trickery
- W1 Hero marries without throne
- XIII E The hero's reaction
- XII D The first function of the donor
- XIV F Provision or receipt of a magical agent
- XIX K Liquidation of Lack
- XI $\uparrow$ Departure
- XVIII I Victory
- XVII J Branding
- XVI H Struggle

- XV G Spatial transference between two kingdoms, guidance
- XXIII o Unrecognized arrival
- XXIX T Transfiguration
- XXXI W Wedding
- XXX U Punishment
- XX $\downarrow$ Return
- X C Beginning Counteraction
- a2 Lack of magical agent
- a6 Various other forms
- α Initial Situation
- $\beta 3$ Absentation of younger people
- $\gamma 1$ Interdiction
- $\gamma 2$ Order or Command
- $\delta 1$ Interdiction violated
- $\delta 2$ Order or command carried out
- $\epsilon 1$ Reconnaissance by the villain to obtain information about the hero
- $\epsilon 2$ Inverted Reconnaissance,Reconnaissance by the hero to obtain information about the villain
- $\epsilon 3$ Reconnaissance by other persons
- $\zeta 1$ Villain receives information about the hero
- $\zeta 2$ Hero receives information about the villain
- $\zeta 3$ Information received by other means
- $\eta 2$ Persuasion
- $\eta 3$ Deception or coercion
- $\theta 1$ Hero agrees to all the persuasions
- $\theta 2$ Hero mechanically reacts to the employment of magical or other means
- λ Preliminary misfortune caused by a deceitful agreement

A.3 Answer to Competency Question 8

Which Proppian functions appear in African folk tales?

Proppian Function: $I\_\ \beta\_Absentation$
Verbalisation: "For this reason, he announced to all his friends that at the ceremony he would take the life of a leopard cub, as a sacrifice for the future of his young son."

Proppian Function: $VII\_\theta\_Complicity$
Verbalisation: "The spirit focused on her face and left her."@en

Proppian Function: $I\_\beta\_Absentation$
Verbalisation: "The girl could not believe that the family would be leaving the place where they had lived for so long [. . .]"

Proppian Function: $\eta3\_Deception\_or\_coercion$
Verbalisation: "He appeared before Jackal one day, disguised as a young boy."

Proppian Function: $Rs4\_Hero\_hides\_himself\_during\_his\_flight$
Verbalisation: "The hero clung onto the villians shoulder"

Proppian Function: $\gamma2\_Order\_or\_Command$
Verbalisation: "The queen ordered for the lions death"

Proppian Function: $XIX\_K\_Liquidation\_of\_Lack$
Verbalisation: "At once they knew it was the milk bird. Gently, the boy lifted up the milk bird [. . .]"

Proppian Function: $III\_\delta\_Violation$
Verbalisation: "[. . .] but he did not see that he had failed to do so and any bird could tell that there was a trap there."

Proppian Function: $J1\_Brand\_is\_applied\_to\_the\_body$
Verbalisation: "Lion was stung so many times and was in such pain that his soft cries soon swelled to a thunderous roar that could be heard for miles around!" Proppian Function: $XVIII\_I\_Victory$
Verbalisation: "This time, the spirit staggered before falling like a dead breadfruit tree and died."@en

Proppian Function: $\zeta2\_Hero\_receives\_information\_about\_the\_villain$
Verbalisation: "When she saw the tracks that the hunter had made as he left the forest, she knew in her heart that a terrible thing had happened to her children [. . .]"

Proppian Function: $IV\_\epsilon\_Reconnaissance$
Verbalisation: "The spirit was coming close to her."@en

Proppian Function: $\eta3\_Deception\_or\_coercion$
Verbalisation: "The hare tried the lion and his family into getting into the bag"

Proppian Function: $I\_\beta\_Absentation$
Verbalisation: "One morning, when the woman wanted to go to the market place in another town, she gave Obaraedo a tuber of yam and a snail which she would roast for her lunch."@en

Proppian Function: $J1\_Brand\_is\_applied\_to\_the\_body$
Verbalisation: "the fire reached his skin, and in agony he charged away towards the waterhole."

Proppian Function: $XXXI\_W\_Wedding$
Verbalisation: "She now owned her husband's cattle and because of this there were many men waiting to marry her."

Proppian Function: $G1\_Hero\_flies\_through\_the\_air$
Verbalisation: "The villian spread his wings to fly and the hero shot like a little arrow"

Proppian Function: $IV\_\epsilon\_Reconnaissance$
Verbalisation: "This girl, who began to watch her father getting fatter and fatter, began to wonder how it was that a man could grow fat if he never ate."

Proppian Function: $VIII\_A\_Villainy$
Verbalisation: "Then, quickly substituting his own bird, he passed it to the blind man and put the coloured bird into his own pouch."

Proppian Function: $I\_\beta\_Absentation$
Verbalisation: "Each morning, Sibanda would sneak off to his food tree and sing to it about the greediness of his wife and family."

Proppian Function:$A6\_Villain\_causes\_bodily\_injury$
Verbalisation: "Hare crept up and threw in the burning embers, blowing on them until he had a fine blaze going."

Proppian Function: $\epsilon2\_Inverted\_Reconnaissance, Reconnaissance\_$ $by\_the\_hero\_to\_obtain\_information\_about\_the\_villain$
Verbalisation:"The hare found out that the monster was his friend after killing him"

Proppian Function: $\epsilon2\_Inverted\_Reconnaissance, Reconnaissance\_$ $by\_the\_hero\_to\_obtain\_information\_about\_the\_villain$
Verbalisation:"The hare found out that the monster was his friend after killing him"

Proppian Function: $I5\_He\_is\_killed\_without\_a\_preliminary\_fight$
Verbalisation: "The many skins which the cannibal was wearing soon caught fire and he ran wildly away, letting out strange cries as he ran."

Proppian Function: $II\_\gamma\_Interdiction$
Verbalisation: "The girl's mother pleaded with her to go, [. . .]"

Proppian Function: $\delta1\_Interdiction\_violated$

Verbalisation: "Before he could remember Hare’s warning, he panicked, turned, and ran for home."

A.4 Verbalisation of Characters

Table A.1: Query results for characters, their corresponding classes and verbalisations

Individual Name	Proppian Role	Verbalisation	
ABF_ Hare	Animal	Hare	
ABF_ Hare	Hero	Hare	
ABF_ Lion	Animal	the Lion	
ABF_ Lion	Villain	the Lion	
AGW_ Brother	Boy	he	
AGW_ Brother	Boy	the boy	
AGW_ Brother	Boy	the girl's brother	
AGW_ Brother	Brother	he	
AGW_ Brother	Brother	the boy	
AGW_ Brother	Brother	the girl's brother	
AGW_ Brother	Hero	he	
AGW_ Brother	Hero	the boy	
AGW_ Brother	Hero	the girl's brother	
AGW_ Cannibal	Villain	He	
AGW_ Cannibal	Villain	the man	
AGW_ Girl	Sister	fat girl	
AGW_ Girl	Sister	his sister	
AGW_ Girl	Sister	nice	fat girl
AGW_ Girl	Sister	she	
AGW_ Girl	Sister	the girl	
AGW_ Girl	Victim	fat girl	
AGW_ Girl	Victim	his sister	
AGW_ Girl	Victim	nice	fat girl
AGW_ Girl	Victim	she	
AGW_ Girl	Victim	the girl	
AGW_ Mother	Mother	the girl's mother	
BAG_ MAN	Hero	He	
BAG_ MAN	Hero	The boy	
BAG_ OLD_ MAN	Helper	He	
BAG_ VERY_ OLD_ MAN	Donor	He	
BAG_ VERY_ OLD_ MAN	Donor	Old man	
BAG_ WOMAN	Family_ Member	His wife	
BAG_ WOMAN	Family_ Member	She	
BH_ Hunter	Villain	a brave hunter	he

Table A.1: Query results for characters, their corresponding classes and verbalisations

Individual Name	Proppian Role	Verbalisation
BH_ Leopard_ Mother	BiolMother	a beautiful woman
BH_ Leopard_ Mother	BiolMother	she
BH_ Leopard_ Mother	BiolMother	the leopard mother
BH_ Leopard_ Mother	Hero	a beautiful woman
BH_ Leopard_ Mother	Hero	she
BH_ Leopard_ Mother	Hero	the leopard mother
BH_ Leopard_ Mother	Woman	a beautiful woman
BH_ Leopard_ Mother	Woman	she
BH_ Leopard_ Mother	Woman	the leopard mother
GFC_ Guinea_ Fowl	Animal	he
GFC_ Guinea_ Fowl	Animal	the bird
GFC_ Guinea_ Fowl	Animal	the child
GFC_ Guinea_ Fowl	Child	he
GFC_ Guinea_ Fowl	Child	the bird
GFC_ Guinea_ Fowl	Child	the child
GFC_ Guinea_ Fowl	Hero	he
GFC_ Guinea_ Fowl	Hero	the bird
GFC_ Guinea_ Fowl	Hero	the child
GFC_ New_ Wife	Villain	new young wife
GFC_ New_ Wife	Villain	she
GFC_ New_ Wife	Wife	new young wife
GFC_ New_ Wife	Wife	she
GFC_ New_ Wife	Woman	new young wife
GFC_ New_ Wife	Woman	she
GFC_ Pitipiti	Victim	Pitipiti
GFC_ Pitipiti	Victim	his mother
GFC_ Pitipiti	Victim	she
GFC_ Pitipiti	Wife	Pitipiti
GFC_ Pitipiti	Wife	his mother
GFC_ Pitipiti	Wife	she
GFC_ Pitipiti	Woman	Pitipiti
GFC_ Pitipiti	Woman	his mother
GFC_ Pitipiti	Woman	she
HCS_ Cheetah	Animal	cheetah
HCS_ Cheetah	Animal	he
HCS_ Cheetah	Hero	cheetah

Table A.1: Query results for characters, their corresponding classes and verbalisations

Individual Name	Proppian Role	Verbalisation	
HCS_ Cheetah	Hero	he	
HCS_ Creator	Donor	He	
HCS_ Tsessebe	Animal	Tsessebe	
HCS_ Tsessebe	Animal	he	
HCS_ Tsessebe	Victim	Tsessebe	
HCS_ Tsessebe	Victim	he	
JGM_ Creator	Villain	boy	
JGM_ Creator	Villain	He	
JGM_ Jackal	Hero	he	
MB_ Other_ Boys	Group	they	
MB_ Other_ Boys	Villain	they	
MB_ The_ Boy	Hero	he	
MB_ The_ Boy	Hero	the boy	her brother
MB_ father	Father	a man	
MB_ father	Father	the boy's father	his father
MB_ father	Father	the father	
O_ Spirit	Supernatural	ayaghayagha	
O_ Spirit	Villain	ayaghayagha	
O_ dibia	Helper	dibia	
O_ dibia	Helper	the herbalist	
O_ dibia	Hero	dibia	
O_ dibia	Hero	the herbalist	
RWD_ Hare	Animal	he	
RWD_ Hare	Animal	the Hare	
RWD_ Hare	Dispatcher	he	
RWD_ Hare	Dispatcher	the Hare	
RWD_ Wild_ Dog	Animal	he	
RWD_ Wild_ Dog	Animal	the wild dog	
RWD_ Wild_ Dog	Villain	he	
RWD_ Wild_ Dog	Villain	the wild dog	
RWD_ Zebra	Animal	he	
RWD_ Zebra	Animal	the zebra	
RWD_ Zebra	Hero	he	
RWD_ Zebra	Hero	the zebra	
TCJ_ Jackal	Animal	Jackal	he
TCJ_ Jackal	Hero	Jackal	he

Table A.1: Query results for characters, their corresponding classes and verbalisations

Individual Name	Proppian Role	Verbalisation
TCJ_ Wild_ Dogs	Group	the wild dogs
TCJ_ Wild_ Dogs	Villain	the wild dogs
TCJ_ Wildcat	Animal	Wildcat
TCJ_ Wildcat	Donor	Wildcat
TPO_ Elephant	Animal	the elephant
TPO_ Elephant	Villain	the elephant
TTO_ Ngcede	Hero	Ngcede
TTS_ Father	Father	Sibanda
TTS_ Father	Father	the father
TTS_ Father	Husband	Sibanda
TTS_ Father	Husband	the father
TTS_ Father	Villain	Sibanda
TTS_ Father	Villain	the father
WHH_ Hare	Animal	Hare
WHH_ Hare	Animal	he
WHH_ Hare	Villain	Hare
WHH_ Hare	Villain	he
WHH_ Hippo	Animal	Hippo
WHH_ Hippo	Animal	he
WHH_ Hippo	Hero	Hippo
WHH_ Hippo	Hero	he
WLR_ Bees	Animal	the bees
WLR_ Bees	Villain	the bees
WLR_ Hare	Animal	Hare
WLR_ Hare	Animal	he
WLR_ Hare	Villain	Hare
WLR_ Hare	Villain	he
WLR_ Lion	Animal	Great One
WLR_ Lion	Animal	Lion
WLR_ Lion	Animal	he
WLR_ Lion	Hero	Great One
WLR_ Lion	Hero	Lion
WLR_ Lion	Hero	he
WSP_ Father	Family_ Member	he
WSP_ Father	Family_ Member	old man
WSP_ Girl	Victim	child

Table A.1: Query results for characters, their corresponding classes and verbalisations

Individual Name	Proppian Role	Verbalisation
WSP_ Girl	Victim	she
WSP_ Mother	Mother	she
WSP_ Mother	Mother	the mother
WSP_ Older_ Brother	Boy	brother
WSP_ Older_ Brother	Boy	child
WSP_ Older_ Brother	Boy	he
WSP_ Older_ Brother	Brother	brother
WSP_ Older_ Brother	Brother	child
WSP_ Older_ Brother	Brother	he
WSP_ Older_ Brother	Child	brother
WSP_ Older_ Brother	Child	child
WSP_ Older_ Brother	Child	he
WSP_ Older_ Brother	Hero	brother
WSP_ Older_ Brother	Hero	child
WSP_ Older_ Brother	Hero	he
WSP_ Python	Animal	beast
WSP_ Python	Animal	it
WSP_ Python	Animal	monster
WSP_ Python	Villain	beast
WSP_ Python	Villain	it
WSP_ Python	Villain	monster
WSP_ Younger_ Brother	Boy	brother
WSP_ Younger_ Brother	Boy	he
WSP_ Younger_ Brother	Child	brother
WSP_ Younger_ Brother	Child	he
WSP_ Younger_ Brother	Hero	brother
WSP_ Younger_ Brother	Hero	he