

A model for verbalising relations with roles in multiple languages

C. Maria Keet and Takunda Chirema

Department of Computer Science, University of Cape Town, South Africa
mkeet@cs.uct.ac.za, CHRTAK003@myuct.ac.za

Abstract. Natural language renderings of ontologies facilitate communication with domain experts. While for ontologies with terms in English this is fairly straightforward, it is problematic for grammatically richer languages due to conjugation of verbs, an article that may be dependent on the preposition, or a preposition that modifies the noun. There is no systematic way to deal with such ‘complex’ names of OWL object properties, or their verbalisation with existing language models for annotating ontologies. The modifications occur only when the object performs some *role* in a relation, so we propose a conceptual model that can handle this. This requires reconciling the standard view with relational expressions to a positionalist view, which is included in the model and in the formalisation of the mapping between the two. This eases verbalisation and it allows for a more precise representation of the knowledge, yet is still compatible with existing technologies. We have implemented it as a Protégé plugin and validated its adequacy with several languages that need it, such as German and isiZulu.

1 Introduction

Natural language interfaces to ontologies are used both to ameliorate the knowledge acquisition bottleneck and for user interaction with so-called ‘intelligent’ systems, with the most popular application scenarios in healthcare, weather forecast bulletins, and querying of information systems and question generation in education. This involves mainly knowledge-to-text from OWL files [1, 36, 37], but also bi-directional in ontology authoring systems [11, 14] and the Manchester syntax used since Protégé 4.x. This is done mostly for English, but there are also some works on Latvian [16], Greek [1], and isiZulu [25]. A hurdle for such other languages is the correct ‘verbalisation’, i.e., a natural language rendering of an axiom, when the name of an OWL object property is not a simple verb in the 3rd person singular. For instance, *works for*, *located in*, and *is part of* all have a dependent preposition, the former two have different verb tenses, and the latter a copulative and noun rather than a regular verb. Regarding the verb tenses, even one tense already raises problems for languages in the Bantu language family, such as isiZulu, which is widely spoken in South Africa. IsiZulu has no single 3rd pers. verb regardless the subject—as in English with, say, ‘eats’—but a ‘3rd pers.’ for each noun class (nc); e.g., if a grandmother (nc1a) ‘eats’ something is it

udla, but if an elephant (nc9) ‘eats’ something it is *idla*. This raises the question of how to model that in an ontology or associated language file, or both.

Further, especially Natural Language Generation (NLG) is expected to take into account prepositions [2]. Prepositions are used in various constructions that may imply a certain relation [35], with the one relevant for ontologies mainly being the dependent prepositions—also called ‘deep prepositions’ [29] or ‘co-verbs’ [28]—that in some languages have the preposition associated not with the verb but with the noun. The three principal issues to solve for such prepositions are phonological conditioning, declensions, and noun modifiers. An example of phonological conditioning is preposition contraction in Portuguese, as in *de+a=da* (e.g., *da mesa* ‘of the table’) [33]. Prepositions may change the article of the noun, as in German and Greek; e.g., the article *der* for *Betrieb* (m.) ‘company’ together with *arbeitet für* ‘works for’ results in *arbeitet für den Betrieb*. The lists of prepositions that go with which case are known, yet this has to be encoded somewhere so as to generate the grammatically correct sentence from an ontology. Prepositions may also modify the noun, as happens in Lithuanian and Latvian [16] and in isiZulu and related languages [24]; e.g., the ‘of’ in ‘part of’ is handled by the possessive concord for the noun class of ‘part’ (*ingxenye*, in nc9), *ya-*, that is attached to the object, generating, e.g., *ya+umuntu=yomunto* ‘of the human’ [24]. Although verb conjugation and prepositions could be devolved to the individual language and language-specific implementations, a generic approach that works across languages will facilitate reusability.

To solve these issues, we first take a theoretical approach to achieve a solid foundation conceptually. Both the issue with conjugation and the prepositions can be solved with the so-called positionalist ontological commitment embedded in a representation language, exploiting 1) the *role* an object plays in the relation and 2) the distinction between *relation(ship)* and *relational expression*. As the preposition and its effects on the surface realisation belongs to neither the verb nor the noun *per sé*, the role conveniently can be adorned with such information. The second feature serves as solution to conjugation as well. This is captured at the metamodel layer of the representation language. Therefore, a formal mapping between their respective formalisations in OWL and \mathcal{DLR} is provided, to ensure a rigorous well-founded implementation. The model thus improves both the natural language generation and it provides for a more precise representation of the knowledge. The model has been implemented as a plugin for Protégé. Its adequacy has been validated with isiZulu, chiShona, and German use cases.

In the remainder of the paper we first describe the main language requirements in Section 2, which are assessed against related works in Section 3. The implementation and validation of the model is described in Section 5. We discuss in Section 6 and conclude in Section 7.

2 Language requirements and motivational use cases

This section summarises the requirements for verbs, which are straightforwardly problematic, and for prepositions in the context of relating objects, which are challenging on the whole.

2.1 Verbs in isiZulu and related languages

Linguistically, isiZulu (the Zulu language) is a member of the Bantu language family that has a characteristic *noun class system* that categorises each noun into a noun class that determines the agreement with other words in a phrase and exhibits a strong agglutinative character. It determines, among others, the singular/plural form, verb conjugation, and agreement with some prepositions; e.g., *umfundisi* ‘student’ is in noun class (nc) 1 and its plural, *abafundisi*, is in nc2, and *inja* ‘dog’ is in nc9 and its plural *izinja* is in nc10. IsiZulu has 17 noun classes. Because of the noun class-driven agreement system, any language annotation model must have some way of processing noun classes.

The nc determines verb conjugation using a subject concord (SC) that is prefixed to the verb stem. Therefore there is no single conjugated verb for 3rd pers. sg./pl., and verbalising an axiom is thus context dependent. That is, for an axiom of the form $C \sqsubseteq \exists R.D$ in an OWL ontology, the noun class of the noun/name of C determines the surface realisation of R . For instance, it is $u-+-dla=udla$ ‘eats’ for *umfundisi* (nc1) and $i-+-dla=idla$ ‘eats’ for *inja* (nc9); the respective plurals are $ba-+-dla=badla$ and $zi-+-dla=zidla$. There are only 10 different SCs, as some noun classes have the same one, with 5 variants for the sg. and 5 for the pl. This brings afore the requirements to generate, store, and access those variants somewhere, and to generate or select the right one when verbalising the axiom, and a decision how to name the object property.

Verb negation uses a negative subject concord (NEG SC), which is also determined by the noun class, and the final vowel of the verb stem changes from $-a$ to $-i$. So, a ‘does not eat’ is $aba-+-dli=abadli$ for nc1 nouns and $ayi-+-dli=ayidli$ for nc9 nouns, and so on for the other noun classes; thus, merging the negation with the verb (as in Japanese [31]). There are 10 different forms of the negated verb for singular and plural nouns.

2.2 Challenges with prepositions

Prepositions in ‘English ontologies’ are put together with the verb in the object property (OP) name, yet in multiple other languages they go with, or affect, the noun in the object position in a sentence. The issue is explained easier by referring to a Controlled Natural Language (CNL). For instance, take the axiom of the type as in (A) below in Description Logics (DL) notation, a corresponding *template* (T), and a few examples as *verbalisations* of particular axioms using that template, which generate a *reading* or controlled natural language sentence:

A: $C \sqsubseteq \exists R.D$

T: Each $\langle C \rangle$ $\langle R \rangle$ some $\langle D \rangle$

E1: Each heart is part of some human

E2: Each employee works for some organisation

This works, regardless the nouns and verbs involved. Let us now take the same axiom type (A) for isiZulu, when the verb is ‘simple’ (teaches, eats, etc.): there is no template but a *pattern* (P) instead (extended from [25]):

P: <QCall for NC_x >onke <pl. of C , is in NC_x > <SC of NC_x >< R_{root} > < D in NC_y > <RC for NC_y ><QC for NC_y >dwa

E3: Zonke izindlovu zidla ihlamvana elilodwa. ‘all elephants eat at least one twig’

E4: Bonke abantu badla ummbila owodwa. ‘all humans eat some maize’

Here, the plural of C , *izindlovu*, is in nc10 which has the SC *zi-* and *abantu* ‘humans’ is in nc2 with SC *ba-* that are added to the verb stem *-dla*. Thus, for patterns, there are variables with any number of terminals that are selected based on some criterion, which is here the noun class of the noun.

Let us extend this now such that R ’s verb in the ontology would have a (dependent) preposition squeezed in the name, such as ‘works *for*’ and ‘part *of*’. First, a few examples (regardless whether they are ontologically the best way of modelling things), with the preposition component underlined:

E5: zonke izazi zomnyuziki ziyingxeny ye-okhestra elilodwa. ‘all musicians are a member of some orchestra’

E6: onke amavazi akhiwe ngobumba ‘all vases are constituted of clay’

E7: zonke izincwadi zisemvilophini eyodwa ‘all letters are contained in some envelope’

The *ye-* in E5 is the result of the phonologically conditioned possessive concord for nc9, determined by *ingxeny* ‘part’: *ya-+i-=ye-*. The ‘of’ of ‘constituted of’ in E6 is dealt with by the preposition *nga-* regardless the noun class, but it is also phonologically conditioned (*nga-+u=-ngo-*). The containment in E7 is a locative (spatial), so those rules apply: a locative prefix *e-* and locative suffix, *-ini*, modify the noun *imvilophu* ‘envelope’ to *emvilophini* ‘(located/contained) in the envelope’.

This problem is not unique to isiZulu and related languages. Take, for instance, German and again the same axiom type. A template (T), as proposed in [19], reads awkwardly and would be better served by a pattern (P), with “ G_C ” the gender of C and “ IA_{G_D} ” the indeterminate article for D ’s gender:

T: Jeder/s < C > < R > mindestens 1 < D >

P: <Qall G_C > < C > < R > mindestens < IA_{G_D} > < D >

E8: T: Jeder/s Arbeiter arbeitet für mindestens 1 Betrieb

P: Jeder Arbeiter arbeitet für mindestens einen Betrieb
‘each worker works for at least one company’

noting that the pattern generates a more acceptable sentence. Besides the article, the noun may change as well. For instance, with R a parthood relation, then the ‘of’ (underlined) in ‘part of’ can formulated as:

E9: Jedes Herz ist Teil eines Tieres. ‘each heart is part of some animal’

E10: Jedes Herz ist ein Teil von mindestens einem Tier. ‘each heart is part of at least one animal’

Finally, observe that some verbs with dependent prepositions in English may not be so in other languages, be this a ‘co-verb’ [28], extended verb [25], or integrated in the noun. For instance, ‘part of’ in ‘part of the body’ is *Körperteil* (DE) or *Lichaamsdeel* (NL), the ‘for’ in ‘works for’ can modify the verb (*-el-* is added to the verb root *-sebenza*, resulting in the extended stem *-sebenzela* (ZU)), or the preposition is incorporated in the tense (‘made by’ *-akhiwe* (ZU)). Overall, there are gradations from no effect where a preposition can be squeezed in with

the verb in naming an OP, to phonological conditioning, to modifying the article of the noun to modifying the noun. So, a preposition does belong neither to the verb nor to the noun uniquely across languages, but, typically, to the *role* that the object plays in the relation described by the verb in the sentence; e.g., it is *yomuntu* only if it plays the role of the whole in a part-whole relation like ‘heart is part of a human’ (*inhliziyoyiyingxenyeyomuntu* (ZU)).

Thus, we have seen that a ‘3rd pers. sg.’ may be context-dependent, and notions of prepositions may modify the verb or the noun or the article of the noun, or both.

3 Related works assessed against the requirements

Several approaches have been proposed and used to ‘stretch’ OWL’s object property (OP) usage. We structure them along 5 principal options in two categories from simple to comprehensive and add CNL systems to it, whilst assessing them against the requirements.

‘Hacks’ in OWL. Although it is well-known that OWL on its own is limited [5], three different workarounds are being used. **OPTION 1: IDENTIFIERS.** Give the OP a system-generated identifier as ‘name’ (the IRI), add one or more labels, alike in the OBO ontologies or by overloading the annotation property, and in the application interface layer, such as OBOEdit and Protégé, one has to have an option to select the right label to use in an axiom (e.g., one of *badla*, *idla*, *adla*, *zidla*, *kudla* for ‘eat’ in isiZulu). This separates the ontology component from the natural language. It requires a guarantee that each OP must have at least one label, which is not required by OWL, and it should override the notion of preferred vs. alternative label. **OPTION 2: VERB STEM OR INFINITIVE ONLY.** Name the OP with the verb stem or its infinitive and conjugate everything as appropriate in the verbalisation interfaces to display it, as in the ACE system [20]. Thus, there is only one IRI for the OP, with as many relational expressions as needed. In isiZulu, an infinitive can also be a noun (e.g., *ukudla* ‘to eat’ or ‘food’). However, one cannot reuse names in the ontology other than for punning [32]. Some noun stems can be classified into multiple noun classes, and the meaning is determined only with the complete word including the prefix (e.g., *umuntu* ‘human being’ and *ubuntu* ‘humanity’ have both *-ntu* as stem), so OWL classes need the complete word for isiZulu and related languages, leaving the verb stem as only option for naming OPs. This, then, assumes an extra rule-based layer for the conjugation and prepositions. **OPTION 3: INCLUDE ALL.** Name positive and negative verb stems, add all the variants for the positive and negative, i.e., each variant has its own IRI, declare the positives equivalent and the negatives equivalent, and declare the positive and negative stem disjoint. For isiZulu, they are *dla* and *dli* for ‘eat’ and ‘does not eat’, equivalences as *badla* \equiv *idla* etc. and *abadli* \equiv *ayidli* etc. (despite that they are essentially synonyms), and disjointness for *dla* \sqsubseteq \neg *dli*. This option is only possible in a language where one can express OP equivalence and disjointness. Of the DL-based OWL species, only OWL 2 QL, 2 RL, and 2 DL permit this [32]. Thus, it is not a widely appli-

cable solution. There also will be performance consequences from ‘blowing up’ the RBox five times in size in the worst case. Further, it conflates the difference between relational expression and relation to the extreme, so it is ontologically a bad choice even if it were to perform well in a particular implementation.

Comprehensive linguistic options outside OWL. Options 1 and 2 require that at least some of the linguistic knowledge be dealt with outside OWL, for which there are two elaborate proposals. OPTION 4: LANGUAGE MODEL. One could use a language model such as *lemon* [30]. Previous work showed that *lemon* was insufficient for the Bantu language family however [9], and the recent W3C community report [<https://www.w3.org/2016/05/ontolex/>] does not address them: i) it needs an extension for the noun class information, ii) it needs to avail of the *lemon* morphology module, and iii) it was feasible for properties only when the domain and range were fixed and it and its subclasses would have names whose nouns are in the same noun class. Further, LexInfo and ISOcat are used for the linguistic annotation in *lemon*, but they miss both the noun class system information and the system of concordial agreement that requires rules. More generally, descriptive models for annotation are not suited for dealing with rules, for which rule languages exist. This brings us to OPTION 5: GRAMMAR. The grammar rules can be a tailor-made implementation or one can use one of the myriad formal grammars; within CNLs and OWL, there are GF and Codeco [26], possibly together with *lemon* as described in [10]. The OP naming of GF with ACE follows that of ACE (i.e., infinitive). While examples use an ‘English ontology’ as basis, it could be any with a resource grammar, and subsequently using a translator service either for the terms only as in [3] or to delegate the machine translation to GF [10, 15, 21]. Translation services are not available for isiZulu, and developing a full resource grammar for GF is unlikely for the foreseeable future, simply because of the limited documentation and investigation into isiZulu grammar. Even then, it still does not resolve the prepositions.

CNL-inspired approaches. Very few works take the simplistic approach of just reusing the name of the relationship or relational expression [19]. Stevens et al. [36] has one rule for processing OP names, being removing “_” (e.g., **derives_from** into **derives from**), which was feasible because all relations of the Relation Ontology adhere to a restricted naming scheme. In contrast, Hewlett et al. [18] accept incoherent naming and identified seven phrase structure categories of naming OPs in ontologies—(has) NP, V, (is) NP P, (is) VP P, VP NP, is NP, (is) AdjP—and availed of a POS tagger to verbalise them more natural language-like, so that, e.g., a **hasColor** OP verbalises into **has a colour**. SWAT NL [37] does a similar text-based processing of the OP name. ACE limits the naming scheme of OWL OP names to their infinitive form [20], with the processing happening independent from OWL, as is the case also in [36]. While ACE has a grammar module to do this, the lexical information for NaturalOWL [1] is provided by the domain expert in a Protégé plugin. The separate lexical layer on top of OWL by [1, 20, 36] have their own data structures rather than a known language model. Another strand of work seeks to link OWL to the Grammatical Framework (GF) [<http://www.grammaticalframework.org>] with, e.g., AceOwl [15,

21]. Overall, there are two extremes in approach: either working with comprehensive top-down annotation frameworks and grammars (e.g., *lemon* [30], GF, Codeco [26]) or a bottom-up approach [1, 16, 23, 36, 37]. The few works on languages other than English take, at first at least if not throughout, a bottom-up approach. There are domain-independent solutions for notably Greek [1], Latvian [15, 16], and AceWiki was tested with German and Spanish [21], where [15, 21] use a ‘detour’ through GF. Neither of the two recent surveys on NLG and CNLs for OWL address issues of conjugation or prepositions [4, 34].

Thus, none of the current approaches caters for the case where there are multiple words for a ‘3rd pers. sg./pl.’ and have flexibility on prepositions.

4 Conceptual model and mappings for relations

In order to obtain the technology-independent model to deal with verb conjugation and prepositions to support also languages other than English, we draw from several sources, which are described first in the preliminaries, after which the model is introduced, and finally the formalisation.

4.1 Preliminaries

From a language viewpoint, it may seem that the pair ‘teaches’ and ‘taught by’ or the pair ‘works for’ and ‘employs’ are all different relations, for they are different words. This is called the “standard view” on relations in philosophy [13, 27]. However, there is only one state of affairs between the professor and the course, or between the worker and the company, respectively, so then there ought to be only one relation for one state of affairs. This is solved by *positionalism*, which relegates ‘teaches’ etc. to being *relational expressions*, and introducing a different notion of relation(ship). In this case there is one n -ary *relation(ship)* that has n unordered argument places, also called *roles*, in which the objects participate, and to which any number of relational expressions can be attached [13, 27]. For instance, a relationship named **teaching** with the roles [lecturer] and [taught] such that the **Professor** participates in **teaching** by playing the [lecturer] role and **Course** plays the [taught] role.

Positionalism is the underlying commitment of the relational model and a database’s physical schema, as well as of the main conceptual modelling languages. It has been employed in Object-Role Modelling (ORM) and its precursor NIAM for the past 40 years [17], UML Class Diagram notation requires *association ends* as roles, and Entity-Relationship (ER) Models have relationship *components* [12]. To illustrate the positionalism, let us take an example in ORM depicted in Fig. 1. It has a binary relationship (ORM fact type) **eat** with two participating entity types, **Lion** and **Gazelle**, where the lion plays the [predator] role and the gazelle plays the [prey] role, and a number of fact type readings, such as ... *eats* ..., where the ellipses are filled with the entity types. Together with the fact type reading, it is verbalised as *Each lion eats at least one gazelle*. In the other reading direction, there is no constraint, which is verbalised with

‘it is possible’ or ‘may’, so we obtain the sentence *It is possible that a gazelle is eaten by a lion*. The ‘by’ is only needed when the [prey] role in eat is used to verbalise the axiom. The same mechanism holds for, say, a parthood relationship, with [part] the role that, say, Lecture plays and [whole] the role that Course plays, and a surface reading in both directions may be ... *part of* ... and ... *has part* ...: the ‘of’ preposition is only used in one reading direction, so is used with one role in that context. Put differently, the preposition is conceptually associated with neither the verb nor the noun, but with the role that an object referred to by the noun plays in the relation.

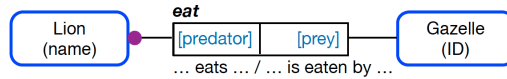


Fig. 1. Example ORM diagram with two entity types, the role names in the role-boxes of the fact type, and the fact type readings below the fact type. The name of the fact type was added to the figure for clarity (typically hidden from view).

An important advantage of positionalism is the separation of relation and reading, though roles are also useful for declaring more precise constraints; e.g., an object may not be allowed to perform two roles at the same time, which cannot easily be asserted with a standard view language. As we shall see, roles are also useful to attach information to for conjugation and prepositions.

There is already a unifying metamodel for the positionalist UML Class Diagrams v2.4.1, EER, and ORM2 [22], which can be extended with an orthogonal component for natural language annotations. This metamodel unifies their language features, the constraints that have to hold when using them, and harmonises their respective terminology. A small extract of this metamodel is depicted in the top-part of Fig. 2: each relationship contains at least two roles, whereas a role is part of exactly one relationship, and each role must have an entity type that plays it (though an entity type does not need to play a role).

4.2 Metamodel for processing properties

The task now, then, is to reconcile the positionalism, standard view, and the surface realisations or relational expressions. We first relate the components for positionalism to those of the standard view. This means mapping a relationship with at least two roles that the entities play into a predicate with entity types in a fixed order. These links and types of entities are shown with dashed lines in Fig. 2: it forces an order onto the entities and removes the role elements. If the language has binary relations only, one may simplify this to annotating it with a natural language sentence’s nominative and dative/accusative positions. Or, informally: the ‘subject’ that does the thing and the ‘object’ that has something done to it, respectively; e.g., the lion (nominative) does the eating and the gazelle (dative) is the one that is eaten, regardless the order of the two elements.

The second step in model development is to consider whether to show in an ontology development environment or domain experts' interface elements with constraints only, or also the elements themselves as being typed. That is, whether from some actual ontology, it should generate *Heart is an Entity type* (indicating type of element) and *Human has part Heart* (without any constraints) as well, or only when they appear in some axiom with constraints. The metamodel in Fig. 2 is permissive of both, through *Axiom type*. This means that it can take care of those essentially second order statements, like $\text{EntityType}(\text{Heart})$, the typing of a relationship (e.g., in DL notation, $\exists \text{haspart} \sqsubseteq \text{Heart}$ and $\exists \text{hasPart}^- \sqsubseteq \text{Human}$), and those axioms denoting constraints, such as of type $C \sqsubseteq = 1 R.D$ (e.g., $\text{Human} \sqsubseteq = 1 \text{hasPart.Heart}$).

Third, the natural language sentence. This may be split up in a reading *pattern* or *template* and the actual natural language sentences, or *readings*, that are generated from either. The main reason for this is to cater for different natural language grammars. In a 'simple' natural language, such a pattern may well be a straightforward template for the axiom where the nouns for the class and verb for the relation are simply plugged in on the fly, taken from the ontology file. For grammatically richer languages, the pattern requires additional grammar rules to generate the sentence, as is the case for isiZulu [25], or processing those prepositions (recall Section 2). The elements to be plugged into the reading pattern are of a specific POS category, such as noun, verb, possessive concord and so on. This is included on the left-hand side of Fig. 2.

Finally, one can add a myriad of properties or attributes to the classes in Fig. 2, where the main selection of attributes of the classes is included in Fig. 3. These properties are general in the sense of regardless the implementation choices, yet their datatype and value ranges can vary because of that, such as implementing them in a relational database, XML document or linking to the linguistic Linked Open Data cloud. For instance, for tense, case, gender, and grammatical number, it does not matter which language model is chosen as source for interoperability. For the noun class system, it does matter, for no

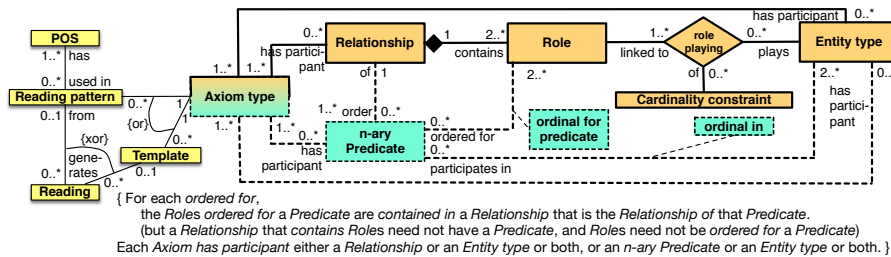


Fig. 2. Simplified depiction in UML Class Diagram notation of the main components (attributes suppressed), linking a section of the unifying metamodel (classes with thick lines; positionalist commitment) to predicates (classes with dashed lines; standard view) and their verbalisation (classes with thin lines).

Role	Entity type	Reading pattern				
identifier: Integer {id} name: String tense: <i>{/list from ISOcat}</i> case: <i>{/list from ISOcat}</i> hasPRE: Boolean PRE: String [0..1]	identifier: Integer {id} name: String gender: <i>{/list from ISOcat}</i> [0..1] noun class: <i>{/list from NCS ontology}</i> [0..1] grammNumber: Integer	identifier: Integer {id} pattern: String language: <i>{/ISO abbrev.}</i>				
		<table border="1"> <thead> <tr> <th>Axiom type</th> <th>Reading</th> </tr> </thead> <tbody> <tr> <td> identifier: Integer {id} positionalist: Boolean </td> <td> identifier: Integer {id} sentence: String </td> </tr> </tbody> </table>	Axiom type	Reading	identifier: Integer {id} positionalist: Boolean	identifier: Integer {id} sentence: String
Axiom type	Reading					
identifier: Integer {id} positionalist: Boolean	identifier: Integer {id} sentence: String					

Fig. 3. Several suggested implementation extensions to the metamodel (see text for details).

source other than the Noun Class System ontology has sufficient information about noun classes [9], in particular on which noun classes there are and the singular/plural pairs. Note that gender and noun class are optional. To cater for both cases where a preposition is squeezed into the name of the relationship, as is customary for object properties in OWL in English, and to record this separately for languages such as German and isiZulu, both presence/absence of a preposition can be recorded and the actual preposition itself when it does not fit in the relationship’s name. Because the latter may not be relevant for some languages, such as English, it is made an optional attribute.

4.3 Formalisation

Given the conceptual links between the standard view and positionalism, we now specify this formally for the knowledge-to-text case. This means that the bottom-part with the standard view relates with elements from, e.g., OWL, Common Logic, and First Order Predicate Logic, and the top-part with the conceptual modelling languages and its logic-based reconstruction with a positionalist commitment. The latter typically use a language in the \mathcal{DLR} family of Description Logic languages [7], which has been applied first to ER [8] and subsequently in many variants to UML and ORM. What only has to be done is to specify the associations indicated with dashed lines in the UML Class Diagram in Fig. 2. We link the relevant parts of OWL 2 DL to \mathcal{DLR} [8], both of whom have a model-theoretic semantics. The syntax for \mathcal{DLR} is as follows, where P is an atomic relationship and A an atomic entity type (class), based on [8]:

$$R ::= \top_n \mid P \mid (\$i/n : C) \mid \neg R \mid R_1 \sqcap R_2$$

$$C ::= \top_1 \mid A \mid \neg C \mid C_1 \sqcap C_2 \mid \exists[\$i]R \mid (\leq k[\$i]R)$$

where i denotes a role (if it is not named, then integer numbers between 1 and n_{max} are used); n is the arity of the relation; the $(\$i/n : C)$ denotes all tuples in \top_n that have an instance of C as their i -th component; k is a nonnegative integer for cardinality constraints). It uses the usual notion of interpretation, where $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and the interpretation function $\cdot^{\mathcal{I}}$ assigns to each concept C a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and to each n -ary R a subset $R^{\mathcal{I}}$ of $(\Delta^{\mathcal{I}})^n$, such that the conditions are satisfied following Table 1.

For OWL, instead of the lengthy OWL 2 DL standard, we present here only the relevant fragment of it (effectively \mathcal{ALNH}). With A in the set of named classes and R in the set of named (simple) object properties in OWL, then:

Table 1. Semantics of \mathcal{DLR} (source: based on [8]).

$\top_n^{\mathcal{I}} \subseteq (\Delta^{\mathcal{I}})^n$	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
$P^{\mathcal{I}} \subseteq \top_n^{\mathcal{I}}$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$(\neg R)^{\mathcal{I}} = \top_n^{\mathcal{I}} \setminus R^{\mathcal{I}}$	$(C_1 \sqcap C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
$(R_1 \sqcap R_2)^{\mathcal{I}} = R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}}$	$(\$/i/n : C)^{\mathcal{I}} = \{(d_1, \dots, d_n) \in \top_n^{\mathcal{I}} \mid d_i \in C^{\mathcal{I}}\}$
$\top_1^{\mathcal{I}} = \Delta^{\mathcal{I}}$	$(\exists[\$/i]R)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \exists (d_1, \dots, d_n) \in R^{\mathcal{I}}. d_i = d\}$
	$(\leq k[\$/i]R)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \{(d_1, \dots, d_n) \in R_1^{\mathcal{I}} \mid d_i = d\} \leq k\}$

$C ::= \top \mid A \mid \forall R.A \mid \exists R.A \mid \leq k R \mid \geq k R \mid C_1 \sqcap C_2$

$R ::= \top_n \mid P \mid P^-$

The semantics is like for \mathcal{DLR} , where “ $\exists R.A$ ” has a semantics $(\exists R.A)^{\mathcal{I}} = \{x \mid \exists y. R^{\mathcal{I}}(x, y) \wedge A^{\mathcal{I}}\}$.

To declare the equivalence mappings, we first use [7, 12] for typing of the DL roles/OWL OPs and their DL role components:

Standard view to positionalism:

$$\begin{aligned} \exists P.C &\Longrightarrow \exists[\$/1](P \sqcap (\$/2 : C)) & \exists P^-.C &\Longrightarrow \exists[\$/2](P \sqcap (\$/1/2 : C)) \\ \forall P.C &\Longrightarrow \neg \exists[\$/1](P \sqcap (\$/2/2 : \neg C)) & \forall P^-.C &\Longrightarrow \neg \exists[\$/2](P \sqcap (\$/1/2 : \neg C)) \end{aligned}$$

Thus, from standard view to positionalist, we add argument places based on the typing of the relation or the use of the class constructors, by numbering the roles but bearing in mind that they do not have to appear in that order once represented in \mathcal{DLR} . In the other direction, we choose the following mapping, which is based on the motivation and algorithm in [12], restricted to binaries only, for OWL has only binary OPs:

Positionalism to standard view:

$$\begin{aligned} P \sqsubseteq [role]A \sqcap [elor]C &\Longrightarrow \exists role.A \sqsubseteq C \\ &\exists elor.C \sqsubseteq A \\ role &\equiv elor^- \end{aligned}$$

There is one final step to the mappings, which is when there are no domain or range restrictions, as is allowed in ontologies; e.g., there is only some axiom of pattern $C \sqsubseteq \exists R.D$ or $C \sqsubseteq \forall R.D$. This can be linked to a positionalist representation by introducing a property R' as subproperty of R , and make C and D the domain and range of R' , and by adding the two roles:

$$\begin{aligned} C \sqsubseteq \forall R.D &\Longrightarrow R' \sqsubseteq [\$/1/2]C \sqcap [\$/2/2]D \\ &R' \sqsubseteq R \\ C \sqsubseteq \exists R.D &\Longrightarrow R' \sqsubseteq [\$/1/2]C \sqcap [\$/2/2]D \\ &R' \sqsubseteq R \\ &C \sqsubseteq \exists[\$/1/2]R' \end{aligned}$$

These mappings cover the core possibilities for mappings between a positionalist and standard view logic. DLs were used for the clear link to applications (OWL, Semantic Web technologies), for having a readily available positionalist logic, and for notation convenience, yet it equally well can be cast in other languages, such as plain first order logic and the relational model.

5 Implementation and testing

We have implemented the model with the mapping as a plugin to Protégé. It was developed in Java and avails of the OWL API for reading the OWL file and it writes into an XML file, which is graphically rendered in the plugin. A screenshot of the plugin is shown in Fig. 4 and it can be downloaded from the project page at <http://www.meteck.org/files/geni>, together with examples.

Regarding the implemented functionality, it specifically handles the interaction between the standard view OWL and the positionalist elements (Fig. 2, Section 4.3) and the annotations/attributes from Fig. 3, plus the additional feature that one can add new linguistic annotation properties. The ISOcat values are used and the noun class numbers were added, which are selectable through drop-down lists. The current version has a relevant subset of the possible axiom types, in particular: the *all-all* ($\text{AllValuesFrom}, C \sqsubseteq \forall R.D$) and *all-some* ($\text{SomeValuesFrom}, C \sqsubseteq \exists R.D$) patterns, and in anticipation of the verbaliser, also *subsumption* ($C \sqsubseteq D$), *union* ($C \sqsubseteq E \sqcup D$), *intersection* ($C \sqsubseteq E \cap D$) and *complement* ($C \sqsubseteq \neg D$), where C , D and E may be anonymous classes (though the plugin is easier to use with named classes). Each pattern is represented by a single element in the XML for annotations. This enables the user to insert also the desired name in the constructor for verbalisation; e.g., *noma* ‘or’. The mapping from the OWL ontology view to the positionalist view is done by the Relationships, which are then used in the all-all and the all-some patterns. The plugin shows this by placing the attribute ‘actorName’ in the referencing XML element. Verbalisation may then be done by using the noun class of the actor according to the role that the actor is playing in the relationship.

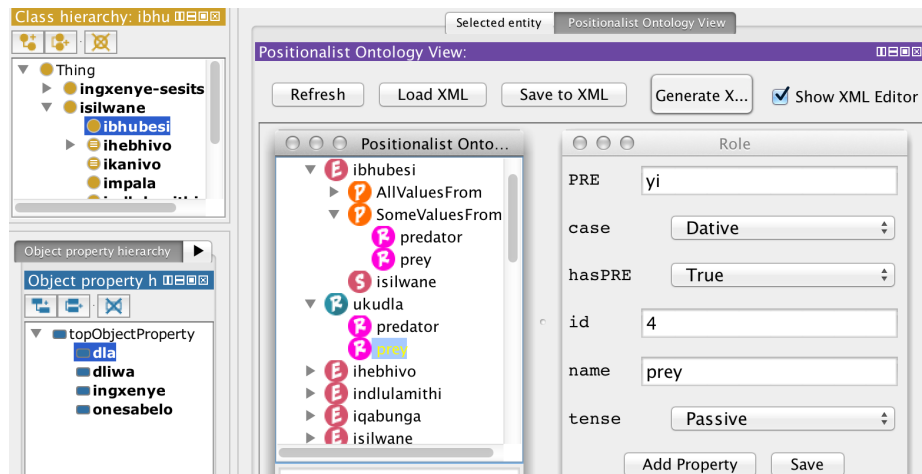


Fig. 4. Screenshot of the plugin with a section of the isiZulu African Wildlife Ontology (left), the positionalist representation (middle), and annotations (right), showing the prey role in the relationship ukudla ‘to eat’, with passive tense and yi ‘by’.

Testing of the model focussed on validation and verification, i.e., on the basis of covering use cases. It was first tested on the positionalism and axiom types functionality. Second, a real modelling scenario was used: a basic isiZulu version of the African Wildlife ontology was created, which includes *ingxenye + ya* ‘part of’ and *dla + yi* ‘eaten by’ (see Fig. 4). The German examples from Section 2 were modelled in a test ontology. Finally, an ontology about pets was created in chiShona, which has grammar features like isiZulu, that also illustrates naming of intersection (*uyezve*) and complement (*zvisiri*) in anticipation of verbalisation.

6 Discussion

As noted in Section 3, currently popular language models, in particular *lemon* [30] and its W3C version, do neither have a way to address noun class information, nor (deep) prepositions other than adding a ‘marker’ on the *lemon* annotation of an object property. Extending them limits one to a single technology and, moreover, it is still tailored to what in philosophy is called the ‘standard view’ of relations (roughly: predicates) that do not cater for roles and properties thereof. Also, there was no functional *lemon*-based ontology annotation tool, so it would have to be developed anyway. In contrast, the model proposed here is, by design, technology-independent and the mapping between a logic with standard view commitment and one with a positionalist stance can be implemented for any combination of languages. For instance, one could also link, say, the OWL or Common Logic Interchange Format to the language of UML Class Diagram notation so as to have a better interaction between the logic and conceptual models, thereby enhancing ontology-driven information systems. The proposed model offers a more precise representation of the knowledge, the natural language, and the interaction between the two.

Further, now one can add noun class, case, gender, tense, and prepositions in a simple annotation interface that guarantees syntactic correctness of the XML file, rather than manually writing in text files. These grammar features are present also in other languages, such as Greek [1], Latvian [16], Chinese [28], and languages related to isiZulu, hence, the here presented model can be reused for languages other than the isiZulu focussed on in this paper.

Our next step is to use it with isiZulu and Runyankore so as to generate more correct sentences from the patterns developed by [6, 25] and for part-whole relations in particular [24] and evaluate it more comprehensively.

7 Conclusions

A model that reconciles standard view and positionalist commitments was proposed, which is the first precise implementation that maps between representation languages committing to either. Precision was achieved with a formal mapping with OWL and *DLR* for logical correctness. The ‘roles’ (description logic role components) serve as the main vehicle for managing the annotations needed for elaborate conjugation and for prepositions that belong to it. The model with

mappings was implemented as a Protégé plugin to validate its adequacy, using examples from isiZulu, chiShona, and German.

Acknowledgments This work is based on research supported by the National Research Foundation of South Africa (Grant Number 93397).

References

1. Androutsopoulos, I., Lampouras, G., Galanis, D.: Generating natural language descriptions from OWL ontologies: the NaturalOWL system. *JAIR* 48, 671–715 (2013)
2. Baldwin, T., Kordoni, V., Villavicencio, A.: Prepositions in applications: A survey and introduction to the special issue. *Comp. Ling.* 35(2), 119–149 (2009)
3. Bosca, A., Dragoni, M., Francescomarino, C.D., Ghidini, C.: Collaborative management of multilingual ontologies. In: *Towards the Multilingual Semantic Web*, p. (in press). Springer (2014)
4. Bouayad-Agha, N., Casamayor, G., Wanner, L.: Natural language generation in the context of the semantic web. *Semantic Web Journal* 5(6), 493–513 (2014)
5. Buitelaar, P., Cimiano, P., Haase, P., M., S.: Towards linguistically grounded ontologies. In: *Proc. of ESWC'09. LNCS*, Springer (2009)
6. Byamugisha, J., Keet, C., DeRenzi, B.: Bootstrapping a Runyankore CNL from an isiZulu CNL. In: Davis, B., et al. (eds.) *Proc. of CNL'16. LNAI*, vol. 9767, pp. 25–36. Springer (2016), 25-27 July 2016, Aberdeen, UK
7. Calvanese, D., De Giacomo, G.: *The DL Handbook: Theory, Implementation and Applications*, chap. Expressive description logics, pp. 178–218. Cambridge University Press (2003)
8. Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., Rosati, R.: Description logic framework for information integration. In: *Proc. of KR'98*, pp. 2–13 (1998)
9. Chavula, C., Keet, C.M.: Is lemon sufficient for building multilingual ontologies for Bantu languages? In: *Proc. of OWLED'14. CEUR-WS*, vol. 1265, pp. 61–72 (2014), riva del Garda, Italy, Oct 17-18, 2014
10. Davis, B., Enache, R., van Grondelle, J., Pretorius, L.: Multilingual verbalisation of modular ontologies using gf and lemon. In: *Proc. of CNL'12. LNCS*, vol. 7427, pp. 167–184. Springer (2012), August 29-31, 2012
11. Denaux, R., Dimitrova, V., Cohn, A., Dolbear, C., Hart, G.: Rabbit to OWL: ontology authoring with a CNL-based tool. In: *Proc. of CNL'09*, pp. 246–264. LNAI, Springer (2009)
12. Fillottrani, P.R., Keet, C.M.: Evidence-based languages for conceptual data modelling profiles. In: *Proc. of ADBIS'15. LNCS*, vol. 9282, pp. 215–229. Springer (2015), 8-11 Sept, 2015, Poitiers, France
13. Fine, K.: Neutral relations. *The Philosophical Review* 109(1), 1–33 (2000)
14. Fuchs, N.E., Kaljurand, K., Kuhn, T.: *Discourse Representation Structures for ACE 6.6*. Tech. Rep. ifi-2010.0010, Dept of Informatics, University of Zurich, Switzerland (2010)
15. Gruzitis, N., Barzdins, G.: Towards a more natural multilingual controlled language interface to OWL. In: *Proc. of IWCS'11*, pp. 335–339. ACL, Stroudsburg, PA, USA (2011)
16. Gruzitis, N., Nespore, G., Saulite, B.: Verbalizing ontologies in controlled Baltic languages. In: *4th Int. Conf. on HLT –“The Baltic Perspective”*. FAIA, vol. 219, pp. 187–194. IOS Press (2010)

17. Halpin, T., Morgan, T.: Information modeling and relational databases. Morgan Kaufmann, 2nd edn. (2008)
18. Hewlett, D., Kalyanpur, A., Kolovski, V., Halaschek-Wiener, C.: Effective NL Paraphrasing of Ontologies on the Semantic Web. In: Proc. of WS on End-User Semantic Web Interaction. CEUR-WS, vol. 172 (2005)
19. Jarrar, M., Keet, C.M., Dongilli, P.: Multilingual verbalization of ORM conceptual models and axiomatized ontologies. Starlab technical report, Vrije Universiteit Brussel, Belgium (February 2006)
20. Kaljurand, K., Fuchs, N.E.: Verbalizing OWL in Attempto Controlled English. In: Proc. of OWLED'07. CEUR-WS, vol. 258 (2007), 6-7 June 2007, Innsbruck, Austria
21. Kaljurand, K., Kuhn, T., Canedo, L.: Collaborative multilingual knowledge management based on controlled natural language. *Semantic Web* 6(3), 241–258 (2015)
22. Keet, C.M., Fillottrani, P.R.: An ontology-driven unifying metamodel of UML Class Diagrams, EER, and ORM2. *DKE* 98, 30–53 (2015)
23. Keet, C.M., Khumalo, L.: Toward verbalizing logical theories in isiZulu. In: Proc. of CNL'14. LNAI, vol. 8625, pp. 78–89. Springer (2014), 20-22 Aug 2014, Galway, Ireland
24. Keet, C.M., Khumalo, L.: On the verbalization patterns of part-whole relations in isiZulu. In: Proc. of INLG'16. pp. 174–183. ACL (2016), 5-8 Sept, 2016, Edinburgh, UK
25. Keet, C.M., Khumalo, L.: Toward a knowledge-to-text controlled natural language of isiZulu. *LRE* pp. in print (DOI 10.1007/s10579-016-9340-0) (2016)
26. Kuhn, T.: A principled approach to grammars for controlled natural languages and predictive editors. *Journal of Logic, Language and Information* 22(1), 33–70 (2013)
27. Leo, J.: Modeling relations. *J. Phil. Logic* 37, 353–385 (2008)
28. Li, C.N., Thompson, S.A.: Co-verbs in Mandarin Chinese: verbs or prepositions? *J. Chinese Ling.* 2(3), 257–278 (1974)
29. Mathonsi, N.N.: Prepositional and adverb phrases in Zulu: a linguistic and lexicographic problem. *S. Af. J. African Lang.* 2, 163–175 (2001)
30. McCrae, J., et al.: Interchanging lexical resources on the semantic web. *LRE* 46(4), 701–719 (2012)
31. McCrae, J., et al.: The Lemon cookbook. Tech. rep., Monnet Project (2012)
32. Motik, B., Patel-Schneider, P.F., Parsia, B.: OWL 2 web ontology language structural specification and functional-style syntax. W3c recommendation, W3C (27 Oct 2009), <http://www.w3.org/TR/owl2-syntax/>
33. de Oliveira, R., Sripada, S.: Adapting simplenlg for brazilian portuguese realisation. In: Proc. of INLG'14. pp. 93–94. ACL, Philadelphia, USA (June 2014)
34. Safwat, H., Davis, B.: CNLs for the semantic web: a state of the art. *LRE* in print, DOI: 10.1007/s10579-016-9351-x (2016)
35. Schneider, N., Srikumar, V., Hwang, J.D., Palmer, M.: A hierarchy with, of, and for preposition supersenses. In: Proceedings of LAW IX - The 9th Linguistic Annotation Workshop. pp. 112–123 (2015), June 5, 2015, Denver, USA
36. Stevens, R., Malone, J., Williams, S., Power, R., Third, A.: Automating generation of textual class definitions from OWL to English. *J. Biomed. Sem.* 2(Suppl 2), S5 (2011)
37. Third, A., Williams, S., Power, R.: OWL to English: a tool for generating organised easily-navigated hypertexts from ontologies. poster/demo paper at iswc'11, Open University UK (2011), 23-27 Oct 2011, Bonn, Germany.