Faculty of Computer Science, Free University of Bozen-Bolzano, Piazza Domenicani 3, 39100 Bolzano, Italy
Tel: +39 04710 16000, fax: +39 04710 16009, http://www.inf.unibz.it/krdb/

KRDB Research Centre Technical Report:

Intelligent query interface: adding natural language support

Laura Haide Perez

Affiliation	*Free University of Bozen-Bolzano*
Corresponding author	*Laura Haide Perez* *perez.laura@gmail.com*
Keywords	
Number	KRDB09-7
Date	April 2009
URL	http://www.inf.unibz.it/krdb/

Intelligent query interface: adding natural language support

Laura Haide Perez

Free University of Bolzano

April 21, 2009

Contents

1 Project Description

1.1 Motivation

- The intelligent query interface presented in [6, 7] supports the users in formulating a precise conjunctive query where the intelligence of the interface is driven by an ontology describing the domain of the data in the information system and by reasoning services running over such domain logic-based ontology. The users can exploit the ontology's entities to formulate the query. They can specify their request by the use of generic terms, refine some terms of the query or introduce new terms, and iterate the process. To compose the query this query interface provides a set of operations namely, `add property`, `add compatible`, `substitute` and `delete` [10]. The main challenge is that the underlying conjunctive query must be presented to the user in natural language and the stepwise refinements of the query, by means of this operations, are refinements that maintain the grammaticality of the sentence representing the query.

- Two different approaches to add Natural Language (NL) to the query interface: 1. Generating the text from the entire conjunctive query each time it changes, because the user applies some operations, or 2. generating in advance all the possible phrases out of the concepts in the knowledge base in such a way that they could be used as menus shown by the query building operations and be further combined into larger phrases or sentences. In this work we explore the syntactic constructions and lexical choices that make possible the latter approach. Therefore, we define a kind of Controlled Natural Language (CNL) that will be the language used by the user when composing a conjunctive query with the basic phrases. Elements in the underlaying formal conjunctive query language, our underlying semantic representation, map into constituents in the CNL.

- The second approach is motivated by the seek of efficiency, i.e. not to generate the text everytime the user changes the query. The idea is that this approach should make use of the syntactic constructions and words of English that permits that phrases verbalising ontology concepts can be precomputed, afterward, used within the suggestions made by the query building operations, and then combined into major grammatically correct clauses or sentences, in accordance to the semantic given by conjunctive query. Once the phrases are combined the text generated so far should remain the same. Each query building operation would made available a certain list of phrases. Those phrases should be the best possible realisation for each ontology concept that is compatible with any other phrases with which could be combined within the current sentence.

1.2 Goals and objectives

The goals of this work are:

- Explore the possible English linguistic constructs namely, words and syntactic constructions, that allow the verbalisation of the concepts of a given underlying ontology (classes, properties and attributes) into syntactic constituents (e.g. noun phrases), in such a way that they could be further combined into major clauses or sentences. The combination process is guided by the underlying logical formalism, the conjunctive query language, and the query building operations.

Keep the generated text to be as close to the underlaying conjunctive query as possible, and at the same time the verbalisation must be acceptable understandable English.

Handle deep class descriptions that may also branch in complex ways. Structuring the information conveyed by the descriptions in sentences.

- Produce a sample corpus.

1.3 A brief account of the Query Interface

In this section we briefly present the descriptions of the underlying technologies and techniques, from [6], that enable the Intelligent Query Interface. We will describe the query building process together with the query language.

Initially the user is presented with a choice of different query scenarios which provide a meaningful starting point for the query construction. More precisely, the user can choose the domain (ontology) where she/he wants to query on. Then, the interface guides the user in the construction of a query by means of a diagrammatic interface, which enables the generation of precise and unambiguous query expressions.

Query expressions are compositional, and their logical structure is not flat but tree shaped; i.e. a node with an arbitrary number of branches connecting to other nodes. This structure corresponds to the natural linguistic concept of phrase structure, where a noun phrase is compound, besides the head noun, by other phrases, such as prepositional or adjectival phrases that function as noun modifiers. This other modifier phrases may contain nested noun phrases as well. A query is composed by a list of terms (classes) coming from the ontology; e.g. 'off-roader' or 'car dealer'. Branches are constituted by properties (relations or attributes) with their value restriction which is a query expression itself. Referring to the underlying ontology, a relation is an association between a concept and another generic concept, while an attribute is an association between a concept and a simple data-type class (String, Boolean, Integer,. . .) or a concept subsumed by a simple data-type class. For instance, in 'car sold by car dealer', 'sold by' is a relation, 'car' and 'car dealer' are ontology terms. *'Land Rover'* is the value (or restriction) of the concept 'model' which is subsumed by the simple data-type class 'String'.

1.3.1 Conjunctive queries

The body of a *query expression* can be considered as a graph in which variables (and constants) are nodes, and binary terms are edges. A query is connected (or acyclic) when for the corresponding graph the same property holds.

To transform any *query expression* in a *conjunctive query* we proceed in a recursive fashion starting from the top level, and transforming each branch. A new variable is associated to each node: the list of ontology terms corresponds to the list of unary terms. For each branch, it is then added the binary query term corresponding to the property, and its restriction is recursively expanded in the same way.

Let us consider the following query *'Find off-roader, make Land Rover, model Defender, sold by car dealer located in Germany'*.

Firstly, a new variable (x_1) is associated to the top level 'off-roader'. Assuming that the top level variable is by default part of the distinguished variables, the conjunctive query becomes

$$\{x_1|off\_roader(x_1), ...\},$$

where the dots mean that there is still part of the query to be expanded. Then we consider the property 'sold by', with its value restriction 'car dealer': this introduces a new variable $x_{1,1}$. The remaining properties of the concept 'car dealer' are then similarly expanded, generating the conjunctive query

(1) $\{x_1|off\_roader(x_1), soldby(x_1, x_{1,1}), cardealer(x_{1,1}), located\_in(x_{1,1}, x_{1,1,1}), Germany(x_{1,1,1}),$ $has\_make(x_1, x_{1,2}), x_{1,2}LandRover, has\_model(x_1, x_{1,3}), x_{1,3}Defender\}.$

This transformation is bidirectional, so that a connected acyclic conjunctive query can be represented as a query expression by dropping the variable names. As a matter of fact, the system is using this inverse transformation since the internal representation of queries is as conjunctive queries.

1.3.2 Query building

The manipulation of the query is always restricted to a well defined, and visually delimited, subpart of the whole query called *focus*. The compositional nature of the query language induces a natural navigation mechanism for moving the focus across the query expression (nodes of the corresponding tree). focus

Users interact with the system to rene the query by a set of operations which can be performed on nodes of the query tree. Once selected, a node becomes the focus for the operations which can be divided into substitution (when a class is substituted by more general or specic one) and incremental renement by addition of compatible classes or properties. Additionally, the system allows the deletion of part of the query.

For each focus the tool suggests the terms and/or properties which can be used to rene the query. The system suggests the operations which are not only compatible with he focus but with the current query expression, in the sense that do not cause the query to be unsatisable. This is verified against the formal model describing the data sources. This step requires the interaction with an OWL-DL reasoner in order to establish which properties or classes are compatible with the current query. This must be done in real time when the user interacts with the tool, since both the query and the focus affect the responses from the reasoner.

After focusing on an ontology term, the user can perform the following operations:

i. generalise or specialise the term,

ii. add a compatible concept,

iii. substitute term with an equivalent one,

iv. add a property (relation or attribute),

v. delete the focused concept.

With the first operation the user can choose among a list of more general or specific concepts where the selected one will substitute the focused term; e.g. in the sample conjunctive query (1) the concept 'off-roader' could be focused, substituted with a more general concept 'car' and specialised again into 'sedan'.

In the refinement by compatible terms (ii), the selected term is simply added to the focus as unary query term. The system driven by the reasoner suggests terms from the ontology whose overlap with the focus can be non-empty (the compatibility requirement). For instance, 'used car' could be among the compatible terms for the concept 'off roader', and could be added by the user. Afterwards, the concept 'new car' could be also among the compatibles, but now it is no more visible because of the disjointness with 'used car'.

The property extension (iv) enables the user to add attributes (e.g. 'car dealer with e-mail e-mail address') or relations (e.g. 'car dealer located in country'), and these actions correspond to the creation of a new branch of the query tree.

Finally (iv), the deletion of the focused term has as consequence the deletion of the incoming property connected to it.

1.4 Adding Natural language support

With the aim of enhancing the man-machine interaction point of view, instead of a tree-shape query the intention is to make it available expressed in Natural Language (NL) text. The communicative goal of our verbalisation task involves the generation of *descriptive text*. In particular, within this work we focus on an approach that consist in verbalising all the concepts of an ontology as English NL phrases (e.g. noun phrases). The QI should let the user combine those phrases into major phrases, clauses or sentences, according to both the underlaying conjunctive query language and the query building operations, described in the previous section. In natural language there are many ways to say the same thing but we will not try to support all or a collection of them in verbalising a given query. Rather, we take into account the requirements of verbalisation imposed by this approach; i.e. the phrases should be generated in advance to their combination into major clauses or sentences. Thus, for instance, since inflection is a category mediating the relation between subject and predicate those phrases should always keep this agreement. The descriptive text to be generated would make use of $3^{rd}sg.$, with some possible exceptions where it could be used the $3^{rd}pl.$ Basically, when the user composes a query he/she is describing the characteristics of a main concept he/she is looking for. Then, the text is descriptive and express expectation (***SEE). To provide modifying information about a noun associated to a concept we could use linguistic constructs, words and syntactic constructions in English, such as relative clauses with simple present verb form or adjectival phrases.

The working of the graphical user interface, the query editor, described in [1], is as follows. The pre-verbalised phrases would be shown in pop-up menus by the query building operations. Within the query editor the user would be able to compose the query in natural language text. When the user selects the main concept the first and main sentence would be generated. Then, when the user moves the mouse cursor over the current text different parts of the query would be highlighted, nodes or edges of the query, which in this case would be basic building phrases, this mouse movement is called *hovering*. This highlighted text could be also selected, when clicking over a phrase representing an edge it turns into a *sticky* state. At the left bottom part of the each noun in the text there is a button which allows the *addition* of compatible concepts or properties, when clicking on those buttons it displayed a pop-up menu with the appropriate phrases. The *deletion* and *substitution* operation are possible after selecting the text. For the substitution operation a pop-up menu is also shown.

1.5 Controlled natural language and related work

Controlled natural languages (CNLs) are subsets of natural languages, obtained by restricting the grammar and vocabulary in order to reduce or eliminate ambiguity and complexity. Traditionally, controlled natural languages fall into two major types: those that improve readability for human readers (e.g. non-native speakers), and those that enable reliable automatic semantic analysis of the language.

- The first type of languages (often called 'simplified' or 'technical' languages), are used in the industry to increase the quality of technical documentation, and possibly simplify the (semi-)automatic translation of the documentation. These languages restrict the writer by general rules such as 'write short and grammatically simple sentences', 'use nouns instead of pronouns', 'use determiners' and 'use active instead of passive', and often use a predefined vocabulary without synonyms.

- The second type of languages have a formal logical basis, i.e. they have a formal syntax and semantics, and can be unambiguously mapped to an existing formal language, such as first-order logic. Thus, those languages can be used as knowledge representation languages, and writing of those languages can be supported by fully automatic consistency and redundancy checks, query answering, etc.

Both definitions of controlled natural language involve constraining the terminology, syntax, and/or semantics.

There are several proposals [16, 8, 2] for using CNL as interface language to knowledge systems instead of using formal languages that are difficult to learn and to remember for non-specialists. Specifically, there is some work done with the aim of providing more natural representations of OWL, i.e. using controlled natural language as front-end for OWL. Those approaches propose authoring OWL ontologies in controlled English [11, 12, 4, 14, 16], CNL→OWL; and most of them also provide the 'round-trip', i.e. the verbalisation of OWL ontologies in controlled English, OWL→CNL. For those approaches which use controlled natural languages as verbalisation languages, their motivation is to provide a verbalisation that could be reversible. For instance, the mapping of OWL constructs into ACE constructs must be injective so that the resulting ACE text could be parsed and converted back into OWL, obtaining an ontology that is identical or at least semantically equivalent to the original.

In our case, we do not seek for a homomorfical translation from natural language into a formal language, but rather we go from a formal language, conjunctive queries, into natural language constructions. Then, our underlying formal language is somehow influencing our verbalisation language, in the sense that we should define how the elements of the conjunctive query (namely, classes, relations, attributes and conjunctions) should be transformed into natural language constituents. Furthermore, the concepts of the ontology would define the vocabulary. Therefore, our verbalisation would result in a language restricted by the vocabulary of the ontology, the conjunctive query language and the communicative goal of our specific verbalisation task.

2 Verbalising conjunctive queries

2.1 Introduction

In the approach in which we focus in this work, the conjunctive query NL text should be obtained by combination of NL phrases. The text generated so far, verbalising the query, should not change when applying one of the query building operations (`add compatible`, `add property`, `substitute` or `delete`); only the phrase to which these operations are applied is the one that is modified. Therefore, we will describe how elements of the conjunctive query could be mapped into natural language constituents based on both their meaning representation and the linguistic constructs in English namely, words and syntactic constructions, that allow us to verbalise phrases which could be further convined into larger clauses or sentences by the user.

The underlying query expressions can be seeing as the description of a complex concept according to the following rules:

$$C \quad \rightarrow \quad AC \quad | \quad C \;\; \textbf{.} \;\; C \quad | \quad C \;\; \mathbf{R} \;\; C$$

where, the first rule is an atomic concept (i.e. a class), the second rule's meaning is conjunction of two concepts (C **.** C) and the third rule is a binary concept (i.e. a relation or attribute).

As described in [11] for OWL ontologies, unary concepts are usually nouns or nouns preceded by an attributive adjective. We will assume that the unary concepts are conceptualisations of entities in a given domain, i.e. they describe a prototype entity. Thus, OWL named classes are chosen as the counterparts for natural language nouns. In particular, we assume that as prototype entities they should be considered to represent and consequently be mapped into singular countable nouns (e.g we would expect of an unary concept in the ontology

unary concepts

to be named 'male' instead of 'males' or 'car' instead of 'cars') or uncountable nouns. However, there would be some exceptions where plural is the best conceptualisation choice. We account for representations in the ontology of different type of nouns (countable or mass noun) and multiword units (e.g. *car dealer*). On the other hand, the counterpart knowledge represention for an adjective would be an unary concept as well. Therefore, the verbalisation of a unary concept needs for some meta information, probably given by a domain lexicon, to disambiguate its syntactic category.

The conjunction of unary concepts sharing the same variable (as described in 1.3.1) introduced by the `add compatible` operation has the meaning that a main concept is also the same kind of or behaves as its compatible concepts. For instance, the concept 'off-roader' has the compatible concept 'non-smoker_car' which means that the off-roader is at the same time a non-smoker car. This conjunction of compatibles could lead to different verbalisations. On one hand, when considering this conjunction as a kind of refinement to a main concept, as well as properties do, it could be mapped into a copula construction. On the other hand, we could assume that each compatible, unary concept, would map into a noun phrase and the conjunction of compatible concepts would be mapped into coordination of noun phrases.

conjunction of compatible concepts

The latter, coordination of noun phrases, should not have a plural reading (each phrases denoting different objects), but rather, that one object that behaves or has the property of the other objects in the conjunction. For instance, for the given object noun phrase in the sentence *I look for an off-roader* we could add the compatible concept 'non-smoker_car' which would result in the conjunction of both noun phrases *I look for an off-roader and a non-smoker car*. The meaning we want to convey by the added compatible is that the *off-roader* has also the property of being or behaving as a *non-smoker car* (the off-roader is at the same time a non-smoker car) instead of referring to two different objects. But from the syntactic point of view conjoined noun phrases denote groups of individuals. Then, we prefer the former option mapping the conjunction of compatible terms into a construction, such as the copula verb *to be*, which adds the compatible concepts as information related to the main concept. For instace, we could add the compatible 'non-smoker_car' in the following way, *I look for an off-roader which must be a non-smoker car* or *I look for a non-smoker off-roader*. With this strategy we assume that the noun phrases denoting unary concepts will remain allways singular or always plural (depending on the underlying conceptualisation and the selected verbalisation for the concept).

noun phrase coordination or copula construction

Another point is regarding the verbalisation of unary concepts as compatibles when they represent adjectives, providing within the conjunction of compatibles there already exists a main concept which is mapped into a noun. Following the last strategy, coordination of noun phrases to verbalise conjunction of compatibles, we could realise them as attributive adjectives of the noun which realises the main concept (e.g. if we have the concepts 'wine' and 'Riesling grape' and then add the compatible 'dry' we should generate the following text *a dry wine and Riesling grape*) whereas in the former strategy we could also use a copula construction as in example (26).

In existing OWL ontologies, property names are more diverse than class names because they usually involve the following patterns: Noun, Verb + Noun, Noun + Preposition, Verb + Verb + Noun, Verb. In particular, data proper-

binary concepts

ties seem to be often nouns. This properties, binary concepts, always describe characteristics, states or actions related to a concept; that is, they provide information which increases specificity of the related concept. Then, we could map them into linguistics constructs that provide modifying information about a nouns such as adjectival phrases or relative clauses.

To verbalise properties as adjectival phrases there is no need to consider subject-predicate agreement, we could use words and/or syntactic constructions which are verb-like in the sense that they take nominal arguments, but which do not require change in inflection regarding to number or person when the associated subject noun phrase changes, i.e singular noun phrase is replaced by a plural one (or vice-versa). More precisely, we could use the participle form of a verb as adjective (e.g. *a car running on...*, *a car equipped with...*), *of*-constructions with meaning of possessive, attributive and partitive genitives (e.g. *equipment of*), and prepositional phrases (e.g. 'with A/C').

Another option, as far as the main noun phrase once introduced remains always with the same person and number agreement, is to realise properties using constructions with relativisers such as *Wh*-question words together with a verb in present form (e.g. *a car which runs on...*). The word *that* may also be used instead of a *Wh*-question word (e.g. *a car that runs on...*).

In the following example we present the same query text verbalised with the two different alternatives for : conjunction of compatibles:

(2) *I look for* ***an off-roader*** *running on diesel, sold by a car dealer located in Germany, made by LandRover, of model Defender, equipped with A/C, central locking and leather seats.*

(3) *I look for* ***an off-roader*** *which runs on diesel, is sold by a car dealer located in Germany, has make LandRover, is model Defender and is equipped with A/C, central locking and leather seats.*

The object noun phrase introduced when verbalising datatype properties (e.g. *Germany*) cannot be further modified by any query building operation.

Each query building operation (described in Section 1.3.2) would suggest to the user a phrase whose meaning is semantically consistent with the underlying current query and syntactically realised to be integrated in the NL expression of the query keeping its grammaticality. We could describe the working of this operations as follows:

- Noun phrases representing concepts could be changed when generalisation or specialisation of the focus takes place (e.g. *An off-roader* may be turn into *A car* by applying generalisation operation).

- Coordinated noun phrases or relative clauses could be introduced within the current query text when selecting compatible concepts (e.g. *An off-roader* may be turned into *An off-roader and a coupe* or *An off-roader which is a coupe* by adding the compatible concept 'coupe'). Or a premodification to a noun may be introduced by adding a compatible which functions as an adjective (e.g. *A car* turns into *A blue car*).

- Coordinated adjectival phrases or relative clauses could be added when related concepts are incorporated into the query. New sentences could be written when the clauses coordination becomes a long list difficult to read, or when a new related concept is introduced and is further described, again, possibly with a long list of adjective coordination.

2.2 Language construction rules

The following construction rules define words allowed and phrases constructed as a bigger unit composed of those words.

2.2.1 Words

Function words and some fixed phrases as patterns are predefined. Predefined function words are determiners ('the', 'a', 'some'), prepositions ('by', 'with', 'of'), coordinators ('and', ',and', ',') and relative pronouns ('which','who'). In addition, we will make use of the predefined fixed phrase *I look for* that should be used at the beginning of the first and main sentence of the text to be generated.

Content words come from the vocabulary given by the underlying ontology:

- singular and plural countable nouns (*a man*, *some men*).
- mass nouns (e.g. *some sugar*, *some equipment*)
- proper names
- adjectives
- participles
- infinitive and simple present verb forms
- auxiliary verbs

The classes, properties and individuals in an ontology should be mapped into content words and/or linguistic constructions. This corresponds to the lexicalization task.

2.2.2 Phrases

The supported phrases, compounds of previously introduced words, are:

- singular countable noun phrases (*a card*, *the card*)
- indefinite noun phrase *some equipment*).
- proper names (*Venice*)
- numbers and strings (*12*, *1.5*, *'5 TDi'*)
- Noun phrase coordination (*and* and ','). Coordination of noun phrases is interpreted as creating a plural object, e.g. *The car and the coupe*.

- Modifying noun phrases:

  Attributive adjective preceding the noun. (e.g. *dry wine*)

  Non-finite phrases. Compound of participles (e.g. *burnt log*), gerunds (e.g. *betting man*) and infinitives (e.g. *work to do*) which functions as an adjective (see definitions in 5). (e.g. *A car equipped with ABS*)

  Prepositional phrases (e.g. *of*-constructions *A car of model defender*).

  Adjectival phrase coordination *and* and ','.

  Relative clauses with *wh*-question words and coordination of them *and* and ','.

Generally, the participle form of the verb can function as an adjective, in fact the meaning is verb, but it functions as a modifier. To have the participle before or after a noun could be addressed as a stylistic problem. If it is used after a noun, we can judge there has been a relative clause after the noun in which the relativisers is omitted. For instance, in *A car equipped with ABS'* where the relativisers is omitted, in fact it has been *A car which is equipped with ABS*. In our case the participles could be introduced by the `add property` operation we could add a relative clause or an adjectival phrase as noun's modifier.

In general, we can use *some* with countable nouns in the plural form with *some* meaning a few, a number of, a pair of or some but not all; but *some* should not be used when speaking about things in general. In particular, in our case, we should only use *some* with indefinite noun phrases when the noun is a mass noun (e.g. *A car equipped with some equipment*). Although, when specifying a criteria we could use an object noun in plural form, such as *A car dealer selling cars produced in Germany* or a quantified expression, *A car dealer selling some cars produced in Germany*, in our conjunctive query language we do have existentially quantified variables. In addition, we consider that classes, unary concepts in the ontology, describe prototypes entities of the world, thus, we should always use singular nouns to verbalise them.

(4) *I look for an off-roader running on diesel, sold by car-dealer located in Germany, selling cars produced in India.

(5) I look for an off-roader running on diesel, sold by car-dealer located in Germany, selling a car produced in India.

2.2.3 Sentences

The conjunctive query verbalisation language will consist only in declarative sentences (i.e. NP VP). The first sentence is the one used to introduce the main concept of the query:

- composed by the main sentence '*I look for* + NP' (i.e. *I look for* followed by a noun phrase). This object noun phrase describes the main concept of the query expression.

In addition to this first sentence, there exists other sentences which would be used to the introduce information when dividing the text. In this cases, if we want to follow a strategy of using syntactic constructions that do not show number agreement, an option would be to use use a *modal auxiliary verb*. More precisely, we could use modal auxiliary verbs, referring to notions such as necessity or obligation, accompanying a main verb in its infinitive form. For instance, we would have the following sentence form [definite-NP+*should be*+adjectival phrase] or [definite-NP+*should*+infinitive verb form].

Sentence formation. The underlying meaning given in the conjunctive query could, in principle, be verbalised in several separated short sentences. This relatively flat syntax structure means that the potential for giving clues as the relative importance of different elements of the underlying content is lost; and the user's capacity to take on board the information in larger chunks is not being exercised. The text that we would generate is meant to give a description of a main object, then, we would prefer to combine the underlying meaning in a more complex sentence structure (e.g. by aggregations such as coordination of relative clauses or modifying phrases, coordination of shared participants). However, the description of a concept (according to the rules introduced in 2.1) could be syntactically arbitrarily complex in terms of coordination of postmodifiers and relative clauses. Phrases that are post noun modifiers (e.g. prepositional phrases) could be coordinated but when the coordination grows to some extent the text may become not clear and difficult to read. This poses the need for a decision of how the concept descriptions should be distributed across sentences.

After coordinating certain number of modifiers within one sentence, the next modifiers could be introduced in a separate sentence. Given this splitting of the information to be conveyed into sentences, in subsequent sentences it could be necessary to make reference to some already introduced entity; i.e. generate a referring expression. In referring generation two issues are involved [15]: the aim is to say as little as possible, but as much as necessary to enable the identification of the intended referent. The minimal descriptive form that most languages offer as resource for referring expressions is the pronoun; in English *he, she, it* which provide virtually no content at all. However, in many discourse contexts the use of pronouns is ambiguous. Given this, it would be better to use a definite noun phrase referring expression, rather than using a pronoun as anaphoric reference. Although definite noun phrases may also be ambiguous in many cases, they still could be more clear than pronouns as they provide more information. Nevertheless, in our generation problem, the user's knowledge about the specific domain plays an important role in the interpretation of the anaphoric references generated within the text. For instance, we would prefer text (7) to (6):

pronouns vs definite descriptions

(6) *I look for **an off-roader**, running on diesel, sold by a car dealer, made by LandRover and of model Defender, equipped with A/C, central locking and leather seats. **It** should be located in Germany.*

(7) *I look for **an off-roader**, running on diesel, sold by a car dealer, made by LandRover and of model Defender, equipped with A/C, central locking and leather seats. **The off-roader** should be located in Germany.*

If our choice was to verbalise properties as adjectival phrases within the first as well as subsequent sentences, then an option would be that the subsequent sentences could be of the form [NP + *should be* + adjectival phrase] linking the subject of the sentence with the modifying phrase (e.g. *The car should be equipped* *with A/C*). However, not all the adjectival phrases built to verbalise a property appropriately fit with the construction *should be*, even though, it is known that the context of usage is describing expectations. For instace,

(8) *I look for an off-roader running on diesel, sold by a car dealer located in Germany, made by LandRover, of model Defender. The off-roader should be equipped with A/C, central locking and leather seats.*

  a. *I look for an off-roader and a coupe running on diesel, sold by car-dealer located in Germany, made by LandRover, of model Defender. The off-roader and the coupe should be equipped with A/C, central locking and leather seats.*

(9) **I look for an off-roader running on diesel, sold by a car dealer located in Germany, made by LandRover. The off-roader should be of model Defender, equipped with A/C, central locking and leather seats.*

  a. **I look for an off-roader and a coupe running on diesel, sold by a car dealer located in Germany, made by LandRover. The off-roader and the coupe should be of model Defender, equipped with A/C, central locking and leather seats.*

(10) *I look for an off-roader running on diesel, sold by a car dealer located in Germany, made by LandRover. The off-roader should be model Defender and equipped with A/C, central locking and leather seats.*

  a. *I look for an off-roader and a coupe running on diesel, sold by a car dealer located in Germany, made by LandRover. The off-roader and the coupe should be model Defender and equipped with A/C, central locking and leather seats.*

As we can see in (9), we get an ungrammatical sentence if we generate an adjectival phrase such as *of model Defender*. Then, we can have *The car should be equipped with ABS* but not **The car should be of model Defender*. That is some adjectival phrases which could be use as noun post modifier do not directly fit when combining them with the *should be* sentence structure.

A similar case would be the case of *A car maker making a car* to be use as *The car maker should be making a car*. Again, although being under a context of expressing expectations this latter sentence gives the sense that we are expressing opinion and that at the moment of the reading of the sentence the car is not being made (example (32a)).

Another possibility is to use adjectival phrases within the first sentence, but for the subsequent sentences use the auxiliary *should* followed by a verb in its infinitive form, i.e. [*NP + should + Infinite verb form*]. Then the text look like the text in (11), note that in this case we are combining adjectival phrases in the first sentence and '*should* + infinitive verb form' in the subsequent sentences. That is, the choice of realisation for a given property would depend on whether it is in the first sentence or in subsequents.

(11) *I look for an off-roader running on diesel, sold by a car dealer located in Germany, made by LandRover. The off-roader should be model Defender and equipped with A/C, central locking and leather seats.*

(12) *I look for an off-roader running on diesel, sold by a car dealer located in Germany. The off-roader should be made by LandRover, be model Defender*

and equipped with A/C, central locking and leather seats.

However, considering that the user knows that he/she is specifying or listing descriptions about some concept, we think that the length of the sentence in terms of number of modifiers' coordination would not lead to unreadable or not understandable sentence; providing its length, making an analogy with the theory of chunking information in cognitive psychology, remains under seven plus two.

There could be cases when the modifying phrase becomes unclear respect to which noun it is adding information. For instance, in (13) and (14), it is not clear if the modifier *of red colour* modifies *wine* or *grape* and *equipped with A/C* modifies the *off-roader* or *car*.

modifying phrase attachment ambiguity

(13) *I look for some wine produced by a winery, located in a french region, adjacent to Bordeaux region, made from grape Merlot, with sugar equal to sweet, of red colour.*

(14) *I look for an off-roader running on diesel, sold by a car dealer located in Germany, selling a car produced in India, equipped with A/C, central locking and leather seats, made by LandRover and of model Defender.*

Nevertheless, this does not depend on the length of the sentence but in the nature of the modifying phrase which could be interpreted as modifying different referents.

2.3 Natural language generation tasks

In this section we discuss about the NLG modules that are needed to add natural language support to the query tool.

- * `Content determination`. The content to be verbalised is the content of the conjunctive query, thus, no NLG module is needed for this task.
- * Document structuring imposes order and structure over the information to be conveyed. Usually, in this NLG task some techniques such as those based on rhetorical relations or centering theory could be applied to get a text as coherent as possible. However, in the general case of the query interface, the order should be derived from the *query expression*. That is, the order of the units of information to be conveyed, concept descriptions, would be given by the depth-first traversal of a given *query expression*, which kept the information to be verbalised within the same order as the user introduces each constituent. As a result, the information to be conveyed is both organised in a coherent way and according to the order in which the user builds the query. Particularly, in this approach through phrases combination, the order is still given by the user by the way he/she applies the operations and combines the possible phrases, the resulting text is fixed after a phrase is added. That is, the context text should not change when one of its phrases is being modified by applying the query building operations.

The order of the coordinated noun phrases which are introduced through the `add compatible` operation, should be one after the other according to the order in which the user adds each of them. Except the addition of a compatible that could be realised as attributive adjective, in this case it should be placed before the noun, in case of adding more than one adjective they should be placed one before the other (example 29).

- * `Aggregation` involves taking a set of simple phrase specifications and combine them to achieve more complex sentence structures. The strategy discussed in this work would produce some aggregated text, but driven by the way the phrases are combined into major clauses or sentences. That is, based on the working of the query building operations `add compatibles` and `add property`, adding the underlying query conjunction of two concepts and relation between two concepts which are mapped into coordination of noun phrases or modifiers, respectively. We obtain an aggregation of the type *syntactic embedding* when adding the coordination of modifiers. For instance, by phrases combination we directly produce the text in 15b:

  (15) a. *I look for an off-roader. The off-roader runs on diesel. The off-roader is sold by a car dealer. The car dealer is located in Germany. The off-roader is made by LandRover. The off-roader should be model Defender. The off-roader should be equipped with A/C. The off-roader should be equipped with central locking. The off-roader should be equipped with leather seats.*

  b. ⇒ *I look for an off-roader running on diesel, sold by a car dealer located in Germany, made by LandRover, of model Defender, equipped with A/C, central locking and leather seats.*

  Besides, within this example, there is another aggregation hold, *aggregation by shared participant* (i.e. when two entities share argument position and have the same content). For instance,

  (16) *The off-roader should be equipped with A/C. The off-roader should be equipped with central locking. The off-roader should be equipped with leather seats.*
  ⇒ *The off-roader should be equipped with A/C, central locking and leather seats.*

  However, in this example if we follow the query building operations (described in 1.3.2), in first place, for the concept *off-roader* through the `add property` operation we could add a modifier phrase obtained from the property 'equipped-with' such us *equipped with leather seats*. Following, if the user wanted to add another feature about the equipment desire for the car like A/C, and if we consider, in addition, that 'A/C' and 'leather-seat' are concepts that belong to disjoint superclasses, this other feature could not be added through `add compatible` for the first introduced object 'leader-seat', but instead applying again the `add property` operation for the concept 'off-roader'. Next, if the user was to add the concept 'central-locking', assuming that is given within the compatibles of concept 'A/C', then it would be possible to incorporate it as coordination of two noun phrase as an aggregation between *equipped with central-locking* and *equipped with A/C*. Then the resulting text would be:

(17) *The off-roader should be equipped with leather seats, equipped with A/C and central locking.*

A point to consider as well, is to take sentence-length constraints into account when making aggregation decisions, as discussed in section 2.2.3.

- `Referring expressions` involves the decision on how to refer to entities. The entities that are first introduced, i.e. initial reference, should be introduced with an indefinite noun phrase (e.g. *I am looking for an off-roader*). When entities are already introduced (i.e. subsequent references) an definite noun phrase should be used. In our case, as we discused in section 2.2.3, we could use a definite description (such as, *the off-roader*) or a pronoun (it). As previously discussed (again, in section 2.2.3), suppose that a given text for a query expression had both *first* and *sunsequent references* to a concept (i.e. indefinite and definite noun phrases). Then, if the user added a compatible concept, by applying the `add compatible` operation, as a result a noun phrase corresponding to the compatible concept would be combined by coordination with the existing noun phrases or added as a relative clause using a copula construction. It the case of noun phrase coordination it should not be necessary to propagate the change into all the subsequent references, which could remain only referring to the main concept.

- `Lexicalization` is the process of choosing words and syntactic structures to communicate the information in a text plan. One mechanism used for lexicalization is to implement a given set of lexicalization rules in procedures that based on some input information would define a lexicalization choice, such us implementing rules by means of decision trees. As a result of the lexicalization task, we could get templates associated with the underlying meaning representations or some kind of abstract linguistic structure. For the latter, information about the lexicalization choices (e.g. syntactic features such as voice, associated lexeme, syntactic category) could be given in a lexicon associated to the underlying formal domain knowledge (i.e. message to be verbalized).

  In the approach presented in this work, the lexicalization choices are influenced, on one hand, by the communicative goal, i.e. *descriptive text* expressing expectation; on the other hand, by the syntactic requirements imposed by the precomputing of the NL phrases and its further combination into major phrases.

  Usually, surface realisation systems, (e.g., KPML [13], GenI [9]), distinguish some kind of linguistically motivated relations like agent, patient, object, instrument, etc. Then, we observed that there is a gap, because properties within an ontology are typically not linguistically motivated (e.g. 'has_make', 'has_model'). In addition, the names given to ontology concepts are not always the best choice. For instance, in example (34b) eventhough the name of the property is 'has_model' non of its verbalisations contains *have*. This is because when talking about the model of a car the common expression is *to be of certain model*. Thus, in order to verbalise such properties in a given ontology we need their *lexicalization*, both for a surface realiser and template-based approach. One option could be to manually build it by a linguist or NLG-expert, which is a native speaker or knows about the grammar and usage of the language. Another option could be to do it semi-automatically with the help of linguistic resources (e.g. dictionaries).

Linguistically oriented ontologies have already been used as interface methods between generators and formal knowledge about the domain. The most known is the Generalized Upper Model [3], but its size and complexity makes the process of mapping the domain knowledge base into a linguistic specification more difficult for non-experts. In contrast, in [5] they defined a core set of 4 linguistically motivated basic properties types, and in the domain ontologies any other property is defined as a sub-property of one of these 4 generic ones. Their aim is to enable knowledge engineers rather than NLG experts to perform the mapping between properties and their linguistic expressions and to reduce the effort of customising the generator for new domains.

Particularly, as our formal knowledge about the domain consist in relations between concepts, that is binary relations. Then, the lexicalization process for a given relation could produce a linguistically motivated meaning representation (e.g. a logic formula) by identifying what kind of relation and which are the roles of the concepts involved in the relation. To this, the process should also add the specification of the chosen words and syntactic information.

- `Surface Realisation` is the task of mapping phrase specifications into surface text; it will be more or less complex depending on the input linguistic specification.

- `Structure Realisation`. Adding mark-up symbols understood by the text presentation component (e.g. in html adding paragraph or link tags). Within the text in the query interface we need to bring out which are the phrases that admit the query operations, and differenciate its appearance under different states such as hover or sticky (as discussed in Section 1.4).

Lexicon

The exact information needed in a lexicon depends on the application and on the strategies adopted for the text generation task. In general, the information contained in a lexicon could consist of: (1) the lemma (2) syntactic category membership; (3) syntactic features (e.g. tense, voice, mood, subcategorization frame, etc.); (4) semantic information (e.g. knowledge representation, theta roles, selectional restrictions) (5) morphological information.

3 Discussion

We have described some possible linguistic constructions that allow the verbalisation of ontology concepts separately into phrases that after could be combined in major phrases or sentences. In particular, we analysed how the query building operations let the user build the underlaying conjunctive query and how each operation is reflected in a NL text. We developed a sample corpus i) following examples from the current *query tool* and the available test ontologies, and ii) verbalising an ontology from the automotive domain (cars.owl) aiming at gleaning ideas about 'choices' that should be made in this particular natural language generation problem. That is, to be able to explore and, may be, derive from it which could be or not the linguistic constructions to adopt. For instance, the usage of adjectival phrases, specifically those made by gerunds, work well when writting the main sentence. However, when splitting the content into more sentences and using the auxiliary verb *should* it is more appropriate to use it followed by a verb in its infinitive form than by *be*+Adjectival Phrase, as the latter results in phrases that are not correct, not natural or not commonly used by native speakers.

We discussed the natural language generation tasks that are required for our specific problem. While content determination is not needed and the document structure is given by the *query expression*, on the other hand, the sentence aggregation and the referring expressions problems should be handled, but taking into account that decision here are influenced by the way of generating phrases in advance and its combination through the query building operations. Finally, we observed that the main challenge within this text generation application is mapping the underlying ontology concepts into words and sentence structures, i.e. lexicalization, and the linguistic realisation operation.

The use of syntactic constructions and words that enable precomputing NL phrases, which are possible to be further combined in grammatically correct sentences, may be more difficult to handle in other languages. For instance, there is an important difference concerning inflection between English and Italian: in many cases English expresses inflection through an independent word, such as *will*, called auxiliary or modal, and which does not show person nor number agreement. This is less frequent in Italian. This difference is connected to another difference dividing the two languages concerning their morphology: English is a *quasi-isolating* language vs. Italian is a *fusional-inflectional* language.

References

[1] *GRAPHICAL USER INTERFACE FOR QUERYTOOL.*

[2] A. Cregan, R. Schwitter, T. M. Sydney owl syntax - towards a controlled natural language syntax for owl 1.1.

[3] Bateman, J., Magnini, B., and Fabris, G. The generalized upper model knowledge base: Organization and use. In *Towards very large knowledge bases* (1995).

[4] Bernstein, A., and Kaufmann, E. Gino - a guided input natural language ontology editor. Springer, pp. 144–157.

[5] Bontcheva, K., and Wilks, Y. Automatic report generation from ontologies: The miakt approach. In *Natural Language Processing and Information Systems* (2004), vol. Volume 3136/2004 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 324–335.

[6] Dongilli, P., and Franconi, E. An intelligent query interface with natural language support. In *FLAIRS Conference* (2006), pp. 658–663.

[7] Franconi, E., Fillottrani, P., Dongilli, P., and Tessaris, S. A multi-agent system for querying heterogeneous data sources with ontologies. In *13th Italian Symposium on Advanced Database Systems - SEBD* (2005).

[8] Fuchs, N. E., and Schwitter, R. Attempto controlled english (ace), 1996.

[9] Gardent, C., and Kow, E. A symbolic approach to near-deterministic surface realisation using Tree Adjoining Grammar. In *ACL'07* (Prague, 2007).

[10] Guagliardo, P. An ontology based visual tool for query formulation support: theoretical foundations.

[11] Kaljurand, K. *Attempto Controlled English as a Semantic Web Language*. PhD thesis, Faculty of Mathematics and Computer Science, University of Tartu, 2007.

[12] Kuhn, T. Acewiki: Collaborative ontology management in controlled natural language. *In Proceedings of the 3rd Semantic Wiki Workshop, CEUR Workshop Proceedings, 2008*, arXiv:0807.4623 (Jul 2008).

[13] Matthiessen, C. M. I. M., Bateman, J. A., and Patten, T. Text generation and systemic-functional linguistics: Experiences from english and japanese, 1991.

[14] Polajnar, T., Tablan, V., and Cunningham, H. User-friendly ontology authoring using a controlled language. Tech. Rep. CS Report No. CS-05-10, University of Sheffield, Sheffield, UK, 2005.

[15] Reiter, E., and Dale, R. *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press, New York, 2006.

[16] Schwitter, R. Representing knowledge in controlled natural language: A case study. In *Knowledge-Based Intelligent Information and Engineering Systems*.

4 Appendix: Corpus-based analysis

The collection and use of a corpus of examples posed some common cases, special cases or problems that will help use to infer constraints of what should or not be allowed in the design of the verbalisation language. Following, we give a collection of pairs query fragments, underlaying meaning, and its corresponding output text that let us by way of examples study the requirements of our verbalisation language:

4.1 Analysis of different cases

(18) school(y)→ privateSchool(y) Specialisation. Change of the spinet, substitution operation.
school(y)→ school(y) and private(y)

(19) -A building hosting an office *located in Russia.*
-A building hosting an office and a school *located in Russia.*
The text *located in Russia* should not change, after adding the compatible concept's noun school. In fact in this case the property *located in* is related to the concept building!
Semantic: building(y), host(y,x), office(x), school(x), located(y,z)
-(op2) A building hosting an office which is a school and is located in Russia.
-(op2) A building hosting an office. The office is a school. The building is located in Russia.

(20) [A car] [equipped with] [[*air bag*] [manufactured by] [Equipment-maker]].
A car equipped with air bag manufactured by Equipment-maker.
A car equipped with air bag and ABS manufactured by Equipment-maker.
The text *manufactured by Equipment-maker* should not change after adding the second concept. In this case, the requirement of being manufactured by an specific Equipment-maker is placed on the equipments and not over the car (main concept).
Semantic: car(x), equipped(x,y), equipped(x,z), manufacture(m,y), manufacture(m,z), equipment-maker(m), air-bag(y), abs(z)

(21) [A company] [compound by] [[an office] [located] [in Russia]].
A company compound by an office located in Russia.
A company compound by a blue office located in Russia.
Semantic: company(x), compound(x,y), office(y), located(y,z), country(z)..... school(y), highSchool(y), blue(y)
In both examples, this and previous, its possible to add some additional information. In this example, the addition is respect to characteristics over the same concept, modifiers or compatibles to the concept office. In the previous, the additional information is regarding the relation of being equipped which will relate the main concept car with two different objects which in turn shared the property of being manufactured by the same equipment-maker.

(22) [A car] [running on [diesel]]

A car running on diesel.
Conjunctive query:
$\{x_1 | car(x_1), run-on(x_1, x_{1,1}), diesel(x_{1,1})\}$
Where, *car* and *diesel* are concepts and *run-on* is an object property.

(23) $\{x_1 | off{-}roader(x_1), run{-}on(x_1, x_{1,1}), diesel(x_{1,1}), sold{-}by(x_1, x_{1,2}), car{-}dealer(x_{1,2}), located{-}in(x_{1,2}, x_{1,2,1}), Germany(x_{1,2,1}), has{-}make(x_1, x_{1,3}), x_{1,3} \in \{LandRover\}, has-model(x_1, x_{1,4}), x_{1,4} \in \{Defender\}, equipped-with(x_1, x_{1,5}), A/C(x_{1,5}), equipped{-}with(x_1, x_{1,6}), central{-}locking(x_{1,6}), equipped{-}with(x_1, x_{1,7}), leather-seat(x_{1,7})\}$

a. I look for an off-roader that runs on diesel. It is sold by a car-dealer which is located in Germany, is made by LandRover and the is model Defender. The off-roader is equipped with A/C, central locking and leader seats. (I need the price, and the mileage of the off-roader, the name, the city, and the phone number of the car-dealer.)

b. I look for an off-roader running on diesel, sold by car-dealer located in Germany, made by LandRover, of model Defender, equipped with A/C, central locking and leather seats.

(24) a. $\{x_1 | coupe(x_1), off{-}roader(x_1), run{-}on(x_1, x_{1,1}), diesel(x_{1,1}), sold{-}by(x_1, x_{1,2}), car{-}dealer(x_{1,2}), located{-}in(x_{1,2}, x_{1,2,1}), Germany(x_{1,2,1}), has{-}make(x_1, x_{1,3}), x_{1,3} \in \{LandRover\}, has-model(x_1, x_{1,4}), x_{1,4} \in \{Defender\}, equipped{-}with(x_1, x_{1,5}), A/C(x_{1,5}), equipped{-}with(x_1, x_{1,6}), central{-}locking(x_{1,6}), equipped-with(x_1, x_{1,7}), leather-seat(x_{1,7})\}$

b. I look for a coupe and an off-roader running on diesel, sold by car-dealer located in Germany, made by LandRover, of model Defender, equipped with A/C, central locking and leather seats.

(25) a. $\{x_1 | Air\_bag\_system(x_1), equipment\_of(x_1, x_{1,1}), car(x_{1,1}), non{-}smoker\_car(x_{1,1})\}$
-[An air bag system] being equipment of a car and a non-smoker car.
-[An air bag system] being equipment of a car. The car must be a non-smoker car.

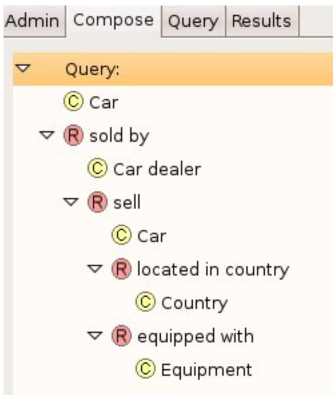
b. $\{x_1 | car(x_1), non-smoker\_car(x_1)\}$
-A car and a non-smoker car.
-The car must be a non-smoker car.

(26) Add compatible operation: noun phrase coordination and attributive adjectives.
$\{x_1 | wine(x_1), riesling-grape(x_1), dry(x_1)\}$

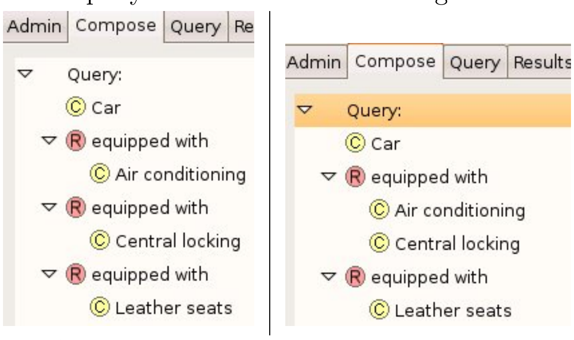
-[Some dry wine and a dry Riesling grape]
or
-[Some dry wine and a Riesling grape]
-I am looking for some wine. The wine must be Riesling grape and dry.

-I am looking for some wine being Riesling grape and dry.
-I am looking for some wine which must be a Riesling grape and dry.
-I am looking for some dry wine which must be a Riesling grape.
Verbalisation with the following sequence: first adding 'wine' main concept, then adding by `add compatibles` operation the 'riesling-grape' concept, until here we get the noun phrase *Some wine and some Riesling grape*. Next, we can adding the 'dry' concept that could be mapped into an attributive adjective we would obtain the noun phrase *Some dry wine and some dry Riesling grape*.

(27) $\{x_1|off{-}roader(x_1), run{-}on(x_1, x_{1,1}), diesel(x_{1,1}), sold{-}by(x_1, x_{1,2}), car{-}dealer(x_{1,2}), located{-}in(x_{1,2}, x_{1,2,1}), city(x_{1,2,1}), x_{1,2,1} \in \{'Germany'\}, sell(x_{1,2}, x_{1,2,2}), car(x_{1,2,2}), produced{-}in(x_{1,2,2}, x_{1,2,2,1}), x_{1,2,2,1}\{'India'\}, equipped{-}with(x_1, x_{1,3}), AC(x_{1,3}), central{-}locking(x_{1,3}), has{-}maker(x_1, x_{1,4}), x_{1,4} \in \{'LandRover'\}, has-model(x_1, x_{1,5}), x_{1,5} \in \{'Defender'\}\}$
- I look for an off-roader running on diesel, sold by car-dealer located in Germany, selling a car produced in India, equipped with A/C and central locking, made by LandRover and of model Defender.



(28) Similar query but difference in meaning:



⇒ Left query: The user always clicks over the concept 'car' and adds property 'equipped-with'
The off-roader should be equipped with A/C. The off-roader should be equipped with central locking. The off-roader should be equipped with leather seats.
⇒ Right query: The user clicks over the concept 'car' and adds property 'equipped-with' twice. Then clicks over 'A/C' and adds as compatible the concept 'central-locking'.

a. *The off-roader should be equipped with A/C and central locking. The off-roader should be equipped with leather seats.*
b. *The off-roader should be equipped with A/C which should be a central locking. The off-roader should be equipped with leather seats.*

4.1.1 Special cases

This cases derive in a difficult or unnatural verbalisation and they would not frequently happend. In most of the cases they arise because of a not appropriate ontology engineering.

(29) *DryRedWine as compatible
$\{x_1|wine(x_1), DryRedWine(x_1), dry(x_1)\}$

- Some dry wine.
- Some dry red wine. or Some wine and dry red wine.
In this case the concept *dry* was selected through the `add compatible` operation. The decision whether this concept should be verbalised as adjective should come from the lexical information associated with the concept. There are concepts like 'dry' that they cannot be used alone. May be 'dry thing'?

(30) *Property hasSugar and restriction WineSugar
$\{x_1|wine(x_1), region(x_1), hasSugar(x_1, x_{1,1}), wine{-}sugar(x_{1,1}, adjacent{-}region(x_1, x_{1,2}), region(x_{1,2})\}$
-[Some wine and a region containing some sugar and adjacent to region 'region']

4.2 Verbalisation of the cars.owl ontology

4.2.1 Object Properties

(31) a. $\{x_1|car(x_1), equipped\_with(x_1, x_{11}), equipment(x_{11})\}$

→ Past participle as adjective.
-[A car] equipped with [some equipment]. → past participle
-[The car] should be equipped with [some equipment]. → should be + adj. phrase
-[The car] should have [some equipment]. → should + Infinitive verb form
-[A car] which is equipped with [some equipment]. → relative clause *which*
Note that here the use of *some* emphasise a requirement that is general. Then the requirement would be specialised into a more specific equipment such as 'air conditioning'.

b. inverse: equipment_of
→ prepositional phrase
-[Some equipment] which [a car] has. → relative clause *which*
-[Some equipment] should be equipment of [a car]. → should be + adj. phrase
-[Some equipment] which is equipment of [a car]. → relative clause *which*
-[Some equipment] of [a car].
The lexicalization of the inverse relation by using *equipment of* or *of* is suitable when talking in the general about 'some equipment'. But after the specialisation of the subject concept into a more specific one the verbalisation of the relation may turn out to be not a natural English phrase, or in the worse case inappropriate. For instance, if the concept 'equipment' could be specialised into the concept 'radio' or 'wheel' then it would be more appropriate to verbalise them as *[A radio] in the car* and *[A wheel] on the car*. Thus, when choosing a lexicalization for a relation we should consider whether it would work for the specialisations and generalisations of the concept.
In the sample sentences it is used the definite noun phrase *the car* it could be possible to use an indefinite noun phrase as well, this would depend in the context where this is verbalised; i.e. whether the refereed entity was already introduced or not.

(32) a. $\{x_1 | car\_maker(x_1), is\_make\_of(x_1, x_{11}), car(x_{11})\}$
→ *to be* + *make* + *of*-construction
-[A car maker] making [a car]. → gerund
-[A car maker] which makes [a car]. → relative clause *which*
-[The car maker] should make [a car]. → should + Infinitive verb form
- ?[The car maker] should be making [car]. → should be + adj. phrase
In the last sentence, although the intention when using *should* is the same as in the other cases, that is to express expectation, here its use followed by a gerund, this results in a linguistic construction which should be used to express an opinion and in the case that the car is not being made.

b. inverse: has_make
→ *have* + Noun

-?[A car] having [`car-maker` make]. → gerund.
-?[The car] should have make [`car-maker`]. → should + Infinitive verb form
-[A car] which has make [`car-maker`]. → relative clause *which*
-[A car] made by [`car-maker`]. → past participle
-[A car] made by [a car maker]. → past participle
-[A car] of [`car-maker`] make. → *of*-construction
The mark '?' before the samples above, point out that even if the sentence is grammatically correct it is not the way in which in English we talk about the make of the car.

(33) a. $\{x_1 | car\_make(x_1), is\_make\_of\_model(x_1, x_{11}), car\_model(x_{11})\}$
→ *to be* + *of*-construction
-[A car make] making model [`car_model`]. → gerund
-[A car make] making [a car model]. → gerund
-[The car make] should make model [`car_model`]. → should + Infinitive verb form
-[A car make] who makes model [`car_model`]. → relative clause *who*
-[A car make] who makes [a car model]. → relative clause *who*
-?[The car make] should be making model [`car_model`]. → should be + adj. phrase
The last sample is the same as in example (32a).

b. inverse: is_model_of_make
→ *to be* + *of*-construction
-[A car model] of [`car_make`] make. → *of*-construction
-[A car model] of [a car make]. → *of*-construction
-[The car model] should be model of make [`car_make`]. → should be + *of*-construction (should + Infinitive verb form)
-[A car model] which is model of make [`car_make`]. → relative clause *which*
-[A car model] which is model of [some car make]. → relative clause *which*

(34) a. $\{x_1 | car\_model(x_1), is\_model\_of(x_1, x_{1,1}), car(x_{1,1})\}$
→ *of*-construction
-[A car model] of [a car]. → *to be* (present verb form) + *of*-construction
-[The car model] should be model of [a car]. → should be + *of*-construction (should + Infinitive verb form)
-[A car model] which is model of [a car]. → relative clause *which*

b. inverse: has_model
→ *have* + Noun
-[A car] of model [`car_model`]. → prepositional phrase
-[The car] should be model [`car_model`]. → should + Infinitive verb form
-[A car] which is model [`car_model`]. → relative clause *which*
Notice here that we do not use *to have* when talking about the model of a car.

c. $\{x_1 | car\_dealer(x_1), sell(x_1, x_{1,1}), car(x_{1,1})\}$

→ present simple verb form (transitive verb)
-[A car dealer] selling [a car]. → gerund
-?[The car dealer] should sell car. → should + Infinitive verb form
-[The car dealer] should be selling car. → should be + adj. phrase
-[A car dealer] who sells car. → relative clause *who*
The sample marked with '?' sounds more like expressing opinion or advice.
inverse: sold_by
→ past participle
-[A car] sold by [`car-dealer`]. → past participle
-[A car] sold by [a car dealer]. → past participle
-[The car] should be sold by [`car-dealer`]. → should be + adj. phrase
-[The car] should be sold by [a car dealer]. → should be + adj. phrase
-[A car] which is sold by [`car-dealer`]. → relative clause *which*
-[A car] which is sold by [a car dealer]. → relative clause *which*

(35) a. $\{x_1 | car(x_1), exterior\_color(x_1, x_{1,1}), color(x_{1,1})\}$
→ Noun
- ?[The car] has [`color`] exterior. → present verb form *have* + adj. phrase
- ?[The car] should have [`color`] exterior. → should + Infinitive verb form
-[The car] should be [`color`]. → should be + adj. phrase
- ?[A car] which has [`color`] exterior. → relative clause *which*
-[A car] which is [`color`]. → relative clause *which*
-[A [`color`] car]. → adjective
When talking about the *exterior color* of a car English the *to be* copula construction linking the color with the car is used; i.e., the natural expression used in English is *The car is red* which implies that the car has an exterior color and that that color is red.

(36) a. $\{x_1 | car(x_1), has\_car\_body(x_1, x_{1,1}), car\_body(x_{1,1})\}$
→ *have* + Noun
-[The car] should have body [`car_body`]. → should + Infinitive verb form
-[The car] should be with body [`car_body`]. → should be + adj. phrase
-[A car] which has body [`car_body`]. → relative clause *which*

(37) a. $\{x_1 | car\_dealer(x_1), located\_in\_city(x_1, x_{1,1}), city(x_{1,1})\}$
→ past participle
-[A car dealer] located in [`city`]. → past participle
-[A car dealer] located in [a city]. → past participle
-[The car dealer] should be located in [`city`]. → should be + adj. phrase (idem should + Infinitive verb form)
-[The car dealer] should be located in [a city]. → should be + adj. phrase (idem should + Infinitive verb form)

-[A car dealer] who is located in [`city`]. → relative clause *who*
-[A car dealer] who is located in [a city]. → relative clause *who*

(38) a. $\{x_1 | thing(x_1), located\_in\_country(x_1, x_{1,1}), country(x_{1,1})\}$
→ past participle
-[something] located in [`country`]. → past participle
-[something] located in [a country]. → past participle
-[something] should be located in [`country`]. → should be + adj. phrase (idem should + Infinitive verb form)
-[something] should be located in [a country]. → should be + adj. phrase (idem should + Infinitive verb form)
-[something] that is located in [`country`]. → relative clause *that*
-[something] that is located in [a country]. → relative clause *that*

(39) a. $\{x_1 | city(x_1), located\_in\_state\_province(x_1, x_{1,1}), state\_Province(x_{1,1})\}$
→ past participle used as adjective.
-[A city] located in [`state_Province`]. → past participle
The participial phrase functions as an adjective modifying city
located (in) (participle)
state_Province (direct object of the state expressed by the participle).
-[The city] should be located in [`state_Province`]. → should be + adj. phrase (idem should + Infinitive verb form)
-[A city] which is located in [state_Province]. → relative clause *which*

(40) a. $\{x_1 | thing(x_1), produce\_model(x_1, x_{1,1}), thing(x_{1,1})\}$
→ simple present (Verb + Noun)
-[somebody/something] producing [some model]. → gerund
-[somebody/something] should produce [some model]. → should + Infinitive verb form
-[somebody/something] who/which produces [some model]. → relative clause *who/which*

b. inverse: produced_by
→ past participle
-[something] produced by [somebody/something]. → past participle
-[something] should be produced by [somebody/something]. → should be + adj. phrase (idem should + Infinitive verb form)
-[something] which is produced by [somebody/something]. → relative clause *which*

(41) a. $\{x_1 | car(x_1), run\_on(x_1, x_{1,1}), fuel(x_{1,1})\}$
→ simple present (verb)
-[A car] running on Fuel. → gerund
-[The car] should be running on Fuel. → should be + adj. phrase
-[The car] should run on Fuel. → should + Infinitive verb form
-[A car] which runs on Fuel. → relative clause *which*

4.2.2 Datatype Properties

(42) $\{x_1 | car\_dealer(x_1), address(x_1, x_{1,1}), x_{1,1} in \{String\}\}$

→ Noun
-[A car dealer] with address [`address`]. → prepositional phrase
-[A car dealer] having address [`address`]. → gerund
-[The car dealer] should have address [`address`]. → should + Infinitive verb form
-[The car dealer] should have address [`address`]. → should be + adj. phrase
-[A car dealer] who has address [`address`]. → relative clause *who*

(43) $\{x_1 | city(x_1), city\_name(x_1, x_{1,1}), x_{1,1} in \{String\}\}$
→ Noun
-[A city] with the name [`city-name`]. → prepositional phrase
-[A city] named [`city-name`]. → past participle
-[A city] called [`city-name`]. → past participle
-[The city] should have name [`city-name`]. → should + Infinitive verb form
-[The city] should be named [`city-name`]. → should be + adj. phrase
-[A city] which has named [`city-name`]. → relative clause *which*
-[A city] that is called [`city-name`]. → relative clause *that*

(44) $\{x_1 | car(x_1), number\_of\_doors(x_1, x_{1,1}), x_{1,1} in \{String\}\}$
→ Noun
-[A car] with number of doors equal to [`number-of-doors`]. → prepositional phrase
-[A car] with [`number-of-doors'`] number of doors. → prepositional phrase
-[A car] having [`number-of-doors`] number of doors. → gerund
-[The car] should have [`number-of-doors`] number of doors. → should + Infinitive verb form
-[A car] which has [`number-of-doors`] number of doors. → relative clause *which*

(45) $\{x_1 | car(x_1), kilometers(x_1, x_{1,1}), x_{1,1} in \{int\}\}$
→ Noun
-[A car] with number of kilometers equal to [`kilometers`]. → prepositional phrase
-[A car] with [`kilometers`] kilometers. → prepositional phrase
-[A car] having [`kilometers`] kilometers. → gerund
-[The car] should have [`kilometers`] kilometers. → should + Infinitive verb form
-[A car] which has [`kilometers`] kilometers. → relative clause *which*

4.2.3 Individuals

Each individual in an ontology has the instance's name :NAME. As its verbalization it should be possible to use this value such as in example (47). Or any other datatype property value like in (46) where it could be possible to use the value of the 'name' datatype property.

(46) $< Car\_dealerrdf : about = "\#Auto\_Center\_Wetzikon" >$
- Auto Center Wetzikon

(47) $< VintageYearrdf : ID = "Year1998" >$
- Year 1998

To verbalize the values of datatype properties just the value itself should be taken, for instance, in the cars domain the datatype property value in (45) should be the value associated with this property in the given instance, in this case an integer value.

4.2.4 Classes

In previous section, we gave some examples when describing possible verbalizations for properties, following we give two more examples. As discussed in Section 2.1, noun phrases are taken to be the NL counterparts of classes. The possible noun phrases would be definite or indefinite ones. The noun could be the same as the name of the unary concept or another given by the lexicalization.

(48) $\{x_1 | Air\_bag\_system(x_1)\}$
-An Air bag system
-The Air bag system

(49) $\{x_1 | equipment(x_1)\}$
-Some equipment

5 Appendix: Definitions

Non-finite The verb form of a verb is not limited by a subject and, more generally, is not fully inflected by categories that are marked inflectionally in language, such as tense, aspect, mood, number, gender, and person. As a result, a non-finite verb cannot generally serve as the main verb in an independent clause; rather, it heads a non-finite clause.

Participles A participle is a verbal that is used as an adjective and most often ends in -ing or -ed. The term verbal indicates that a participle, like the other two kinds of verbals, is based on a verb and therefore expresses action or a state of being. However, since they function as adjectives, participles modify nouns or pronouns. There are two types of participles: present participles and past participles. Present participles end in -ing. Past participles end in -ed, -en, -d, -t, or -n, as in the words asked, eaten, saved, dealt, and seen.

* The crying baby had a wet diaper.
Shaken, he walked away from the wrecked car.
Smiling, she hugged the panting dog.

A participial phrase is a group of words consisting of a participle and the modifier(s) and/or (pro)noun(s) or noun phrase(s) that function as the direct object(s), indirect object(s), or complement(s) of the action or state expressed in the participle, such as:

Delores noticed her cousin walking along the shoreline.

> Here, it is an adjective clause where the relativizer is omitted. The participial phrase functions as an adjective modifying cousin. walking (participle) along the shoreline (prepositional phrase as adverb)

Prepositional Phrase .

In simplest terms, prepositional phrases consist of a preposition and an object of a preposition. Prepositions are indeclinable words that introduce the object of a prepositional phrase. Indeclinable words are words that have only one possible form. For example, below is a preposition, but belows or belowing are not possible forms of below.

The noun phrase or pronoun that follows the preposition is called the object of the preposition. For example, *behind the couch* is a prepositional phrase where *behind* is the preposition and the noun phrase *the couch* acts as the object of the preposition. Sometimes adjectives are used to further modify the object of the preposition, as in *behind the big old smelly green couch*.

Formal Functions of Prepositions. Prepositions perform three formal functions in sentences. They can act as an adjective modifying a noun, as an adverb modifying a verb, or as a nominal when used in conjunction with the verb form to be.

In our query verbalisation language we focus on prepositions functioning as adjectives. For instance, in the following sentences, prepositional phrases perform the function of modifying the nouns boat and car:

> *Look at the boat with the blue sail.*
> *Park the car beside the fence.*