

Ontology-based Integration and Retrieval over Multiple Quantities — What if “Ovate leaves and often blue to purple flowers”

Shenghui Wang
Department of Computer Science
Vrije Universiteit Amsterdam
The Netherlands

Jeff Z. Pan
Department of Computing Science
University of Aberdeen
United Kingdom

Abstract

Information integration and retrieval have been important problems for many information systems — it is hard to combine multidimensional and parallel information and make them available for application queries. In our previous work [12], we have shown how to use ontologies to facilitate integrating and querying parallel but single dimensional information. In this paper, we further investigate how to take advantage of ontologies to facilitate integrating parallel information and querying over multiple quantities.

1 Introduction

Information integration and retrieval have been important problems for many information systems, including those based on the Web [9] — it is hard to combine information from different sources and make them available for application queries. In this paper, we focus on *descriptive* domains, where most information is mostly available in natural language (NL) form and comes *parallel*, *i.e.*, the same objects or phenomena are described in multiple free-styled documents [3]. To some extent, the Web itself is a huge source of parallel descriptions. It has been argued in [13] that NLS are not adept at describing these continuous quantities precisely. Therefore, automated information processing in descriptive domains suffers from the lack of techniques to capture the semantics of natural language descriptions precisely and represent them properly.

Recently, W3C standardised the OWL Web Ontology Language [1] in its Semantic Web Activity. With ontologies being shared understandings of application domains, ontology-based integration and retrieval [10] is a promising direction. In our previous work [12], we have shown how to use ontologies to facilitate integrating and querying parallel but single quantity information (shape descrip-

tions). More specifically, parallel shape descriptions can be extracted and represented in a uniform ontology, the explicitly written information can be accessed easily and the implicit knowledge can also be deduced naturally by applying reasoning on the whole ontology. In this paper, we further investigate the following issues that are related to how to take advantage of ontology to facilitate integrating parallel information and querying over multiple quantities. Like in [12], we choose botany as our application domain as it is one of the premier descriptive sciences and offers a wealth of material on which to evaluate our approach. In particular, we consider *parallel* colour *and* leaf shape descriptions in our ontology, which is an extension of the ones that we used in [11, 12, 13].

For example, the colour of flowers of species *Paeonia anomala* is described in two floras:

- purple-pink — in *Ornamental Plants From Russia*,
- rose to red, occasionally nearly white — in *Flora of China*;

while its leaf shape is also described differently as

- lanceolate — in *Ornamental Plants From Russia*,
- linear to linear-lanceolate — in *Flora of China*.

Being able to handle each quantity as a separate dimension is simply the first step. With multiple quantities in our ontology, we can ask many interesting questions. For example, one user may ask the plant knowledge base: which species definitely have “linear” leaves and more or less “bluish-purple” flowers, blooming in early spring across the British Isles? English bluebell satisfies this query, but are there any other species having similar morphological features?

The contributions of this paper include solutions to the following issues related to multidimensional integration and querying:

1. We focus on the semantics of natural language descriptions with *frequency information*, such as “sometimes,” “rarely,” *etc.* and its representation in an ontology system.

2. We *extend* the distance function to calculate the semantic distance between such descriptions, (e.g., the distance between “bluish-purple” and “rarely blue”), so that information from different sources becomes comparable. The integration process then uses this to quantify similarity.
3. We *predict missing information* (e.g., some species only have flower information, others might only have leaf shape information) using clustering methods, for further query answering.
4. We provide a strategy of query answering, cooperating standard ontology reasoning and semantic similarity measures, in the presence of *multiple quantities*.
5. We use *preferring modifiers* to express complicated ontology queries, including querying on multiple constraints and different ways of ranking results for better usability of query answering system.

2 Preliminary

2.1 A Region-based Metric Semantics for Descriptions of Continuous Quantities

The semantic processing of natural language descriptions of continuous quantities is based on a representation of the semantics in terms of a metric space. A metric space provides a notion of closeness or proximity of points. For continuous quantities, as they occur in nature or as described in natural languages, collections of points rather than single points are the appropriate denotation for their descriptions. They allow points close to and practically indistinguishable from each other to be collected together in a single denotation. This spread of meaning, arising from the imprecision of language, leads us to the interpretation of words and phrases as *regions* in a metric space. Moreover, some phrases denote ranges of values, either around a point or between two or more points, and the natural interpretation of these is as regions in the space.

In our earlier work [11, 12], we have summarised the common forms of descriptions of continuous quantities which occur in natural languages. The semantics of simple descriptions, including basic terms, qualitative modifiers (such as “pale,” “broadly”), hyphenated expressions, and simple “to” ranges are constructed as the region-based metric semantics. Specifically, each basic term corresponds to a certain region in a space. The semantics of complex expressions are built by applying operations on the regions in the space. For example, “pale blue” shifts “blue” towards less saturation and higher lightness; “violet-blue” is the region centred at the intermediate value between “violet” and “blue;” expressions like “pale blue to violet-blue”

represent a range from the first region to the second one. In Section 3.1, we will introduce how we build the semantics of descriptions with frequency information, such as “sometimes,” “rarely,” *etc.*

2.2 DL and OWL-Eu

As the W3C standard ontology language OWL DL [1] does not support XML Schema user-defined datatypes, we use the OWL-Eu language [6] suggested by a W3C Note [2] from the Semantic Web Best Practice and Deployment Working Group. OWL-Eu supports customised datatypes through unary datatype expressions (or simply datatype expressions) based on unary datatype groups. The support for customised datatypes is just what we need here to capture quantitative information. Like an OWL DL ontology, an OWL-Eu ontology typically contains a set of class axioms, property axioms and individual axioms [6]. Here we use the FaCT-DG ontology reasoner, a Datatype Group extension of the FaCT reasoner, which supports reasoning in OWL-Eu ontologies that do not contain nominals.¹

2.3 Preferring Modifier

SPARQL [7] is a query language (W3C candidate recommendation) for getting information from such RDF graphs. SPARQL provides *solution modifiers* which make it possible to transform the solution list derived from a CQ in several ways. The following solution modifiers are available: Distinct, Order, Limit and Offset. Here is the SPARQL syntax for the last three solution modifiers.

```
SolutionModifier ::= OrderClause? LimitClause?
                  OffsetClause?
OrderClause      ::= 'ORDER' 'BY' OrderCondition+
OrderCondition   ::= ( ( 'ASC' | 'DESC' ) ' (' Expression ')' )
                  | ( FunctionCall | Var | ' (' Expression ')' )
LimitClause      ::= 'LIMIT' INTEGER
OffsetClause     ::= 'OFFSET' INTEGER
```

Siberski et al. [8] argued that preference should be a first-class construct in the query language. Their extended SPARQL syntax is listed below.

```
SolutionModifier ::= PreferringClause? OrderClause?
                  LimitClause? OffsetClause?
PreferringClause ::= 'PREFERRING' MultidimensionalPreference
MultidimensionalPreference ::= CascadedPreference
                             ('AND' CascadedPreference)*
CascadedPreference ::= AtomicPreference
                     ('CASCADE' AtomicPreference)*
AtomicPreference ::= BooleanPreference
                  | HighestPreference | LowestPreference
BooleanPreference ::= Expression
HighestPreference ::= 'HIGHEST (' Expression ')'
LowestPreference  ::= 'LOWEST (' Expression ')'
```

¹Details of the FaCT-DG reasoner as well as its flexible reasoning architecture can be found in [5].

Intuitively, users can specify preferences that do not overwrite each other, by using the Preferring clauses with the definitions independent preference separated by the ‘AND’ construct. In each of these dimensions, atomic preferences can be nested using the ‘CASCADE’ construct. Here, the leftmost part of the preference expression is evaluated first, and only when two solutions are equally preferred with respect to this part, the next atomic preference expression is evaluated.

3 From Parallel Descriptions to an Ontological Knowledge Base

3.1 Semantics of Mixed Forms of Descriptions for Continuous Quantities

Expressions which mix several types of quantity are used in trying to describe the complex variation that could be observed in nature. A typical example is the combination of normal qualitative quantities with frequency information. Examples of this are “violet-blue, rarely pink or white” (combining colour and frequency), or “usually broadly ovate, occasionally deltoid” (combining shape and frequency).

To interpret such phrases, we consider pairs of semantic objects as though they were themselves semantic objects. That is, the interpretation takes place in a *product metric space*. Various products are available reflecting the fact that the metrics for the two quantities can be combined in a variety of ways, which allow us to capture the interaction of quantities in the semantics appropriately. In our work, an expression with a frequency modifier is defined as a region with a confidence value, that is, an indication of the frequency with which this particular morphological feature occurs in nature. Depending on the frequency modifiers, the confidence value is defined as the following:

‘always’	100	‘sometimes’	40
‘usually’	80	‘rarely’	20
‘often’	60	‘never’	0

For example, ‘often pale purple’ is represented as $\langle (78 - 88, 25 - 35, 40 - 50), 60 \rangle$, where the first three ranges corresponding to a hue-saturation-lightness region, while 60 indicates the frequency of this value to be found in nature. If an expression is not modified by any frequency modifier, a confidence value of 100 is given.

3.2 Ontological Representation of Objects in Descriptive Domains

Objects in descriptive domains, such as plants, have structural information, such as “plant has parts of leaf, flower, stem, root, *etc.*” Each part also has its own structure. On the one hand, these objects are related to other ab-

stract concepts, such as “Species A is native to the British Isles” (its habitat) or “English Bluebell always blooms in early spring” (its anthesis). On the other hand, on a concrete level, they are all described in terms of one or several continuous quantities. For instance, colour, shape, texture of flowers, leaves, fruits or stems, temporal expressions for flowering and fruiting time, *etc.* A knowledge base storing information of these objects must be able to represent the information at both the abstract and the concrete level. A OWL DL ontology may well be a promising representation for such purpose, but it requires special expressive power in order to represent the continuous quantities in question. This is the reason why we chose its extension, OWL-Eu, to build a plant ontology.

Our plant ontology \mathcal{O} contains knowledge of plants and their features. The primitive classes that we are concerned with are the following:

```
Class (Species), Class (Flower), Class (Colour),
Class (Habitat), Class (Leaf), Class (LeafShape);
```

important object properties related to them include

```
ObjectProperty (hasPart),
ObjectProperty (hasColour),
ObjectProperty (hasShape),
ObjectProperty (hasHabitat);
```

and important datatype properties include

```
hasHue, hasSaturation, hasLightness,
hasLengthWidthRatio, hasBroadestPosition,
hasApexAngle and hasBaseAngle,
hasConfidenceValue
```

which are all *functional* properties. Each datatype property and its range is also defined, for example,

```
DatatypeProperty (hasBaseAngle Functional
range ( $\geq_0 \sqcap \leq_{180}$ )).
```

Typical relations between classes include:

```
Species  $\sqsubseteq \exists hasPart.Leaf \sqcap \exists hasPart.Flower$ 
 $\sqcap \exists hasHabitat.Habitat$ 
Leaf  $\sqsubseteq \exists hasShape.LeafShape$ 
Flower  $\sqsubseteq \exists hasColour.Colour$ 
```

Concrete colours and leaf shapes are defined based on the above primitive classes and properties, where datatype expressions are used to express semantic regions. For example, the colour ‘purple’ is defined as the following OWL-Eu class:

```
Purple  $\equiv Colour \sqcap \exists hasHue.(\geq_{78} \sqcap \leq_{88}) \sqcap$ 
 $\exists hasSaturation.(\geq_{45} \sqcap \leq_{55}) \sqcap$ 
 $\exists hasLightness.(\geq_{20} \sqcap \leq_{30}) \sqcap$ 
 $\exists hasConfidenceValue.(= 100)$ 
```

Finally, a species with ‘purple’ flowers and ‘ovate’ leaves

will be represented as an OWL class as follows:

```
SpeciesA ⊆ Species ⊓ hasPart.LeafA ⊓ ∃hasPart.FlowerA ⊓
∃hasHabitat.Europe
LeafA ⊆ Leaf ⊓ ∃hasShape.Ovate
FlowerA ⊆ Flower ⊓ ∃hasColour.Purple
```

Similarly, species with complex flower colours and leaf shapes are also defined as OWL classes, where flower colour and leaf shape are represented as OWL-Eu classes.

3.3 Distance-based Integration

The integration of parallel information is based on the semantic distance between descriptions [12]. If two parallel descriptions of the same quantity are “close” or “similar” enough, although they might not be linguistically identical, it is better to combine them into one single “super-description” so that redundancies can be removed. If they are too different, however, it is safer to leave them separate because they are likely to provide complementary information. When a reasonable *threshold* is chosen,² our integration process automatically combines similar descriptions and keep others separate.

We now extend the distance function proposed in [12] to consider the confidence information included in the descriptions. We define

$$D(\langle R_1, conf_1 \rangle, \langle R_2, conf_2 \rangle) = \frac{d(R_1, R_2)}{\min(conf_1, conf_2)}, \quad (1)$$

where the distance between two regions is influenced by their individual confidence values. If one pair of regions has higher confidence value, their distance is shorter than the same pair with lower confidence values. Therefore,

$$D(\text{'blue-purple'}, \text{'rarely violet'}) > D(\text{'blue-purple'}, \text{'often violet'})$$

Using this extended distance function, parallel descriptions with frequency information are integrated and represented in the ontological knowledge base.

3.4 Predicting Missing Information using Clustering Methods

Even after integration, it is not always guaranteed that all the required information is present in the KB for each species. We then apply a clustering-based method to predict missing information. Firstly, a learning set of 6000 species was prepared. The colour of their flowers and the shape of their leaves was recorded in order to find the co-occurrence pattern of colour and shape. This co-occurrence data was clustered using the affinity propagation clustering method [4]. The resulting clustering centres are represented by the most typical shape-colour co-occurring patterns present in the learning set.

²Details of defining the threshold can be found in [12]

Secondly, for each species which only has leaf shape (or flower colour) information, we look for the closest cluster according the known information, *i.e.*, its leaf shape (or flower colour). Once the closest centre is identified, the counterpart of the information, *i.e.*, the flower colour (or leaf shape), is adopted as the most likely values of the missing information. We chose a testing set consisting of another 6000 species which have the information of both quantities. Using their colour information to predict their leaf shape, in total 540 cases were precisely predicted. More tests will need to be performed to assess the practical usefulness of this result. It is encouraging however, as there are more than 4000 different leaf shape descriptions in this test set. Correctly predicting the precise leaf shape for 9% of the species is therefore a real improvement on random guessing, and provides new information to the query. As this information is predicted, the confidence value is set as 50.

We have now populated the ontology-based knowledge base with parallel information from multiple sources as well as the best guess of missing information. We can now perform queries over multiple quantities.

4 Querying over Multiple Quantities

4.1 Query Answering Process

The query answering process is divided into a qualitative and a quantitative part. In the first qualitative part, standard logic reasoning is carried out. A virtual species concept with the queried information (for example, a species with required colour or shape and a specific habitat) is generated. The relation between this virtual species and the existing species in the ontology is evaluated by the standard DL reasoner.

The query on particular values of the continuous quantities is carried out as follows:

- Step 1 The particular value is represented by an OWL-Eu class Q with datatype constraints, for example, hue-saturation-lightness values for colours or four features for leaf shapes [12].
- Step 2 A complete set, $complete_Q$, which satisfies the four levels of matching is returned by the FaCT-DG DL reasoner. Specifically, $complete_Q$ consists of the following four sets.
 - $equiv_Q$: all elements are equivalent to Q ;
 - sub_Q : all elements are subsumed by Q ;
 - $super_Q$: all elements subsume Q ;
 - $intersection_Q$: all elements intersect with Q .

Note that if a species has multiple distinct regions for this quantity, each region is checked with the queried

Q . The species is returned if at least one region satisfies one of above four matchings.

Step 3 The distance between the features of the species in the $complete_Q$ and the queried feature are calculated, using the distance functions introduced in Section 3.3. Again, if a species has multiple regions, the smallest distance is treated as the distance between this species and the query.

Step 4 Perform preference queries (see the next sub-section).

This query answering process benefits from the accurate formation of queries (*i.e.*, in the form of OWL-Eu concepts) and the standard logic reasoning (*i.e.*, subsumption and intersection matching) to recover information from a formal knowledge base. The recovery is based on semantics, which, we will see in the next section, outperforms keyword based matching. Meanwhile, ranking returned results according to different preferences (see the next sub-section) is also a desirable feature for this application.

4.2 Preference Queries

In this section, we present three different kinds of preference queries.

Global Preference This is the most simplified solution for multi-quantity queries. In this approach, users provide a definition for a global distance, combining distances from different quantities. This is best explained with a concrete example. Suppose we would like to ask for species with

“ovate leaves and often blue to purple flowers” (2)

with the global distance being the sum of colour distance and shape distance. In Step 1, we represent the two query concepts,

$Colour_Q \equiv BlueToPurple \sqcap \exists hasConfidenceValue.(= 60)$,
 $Shape_Q \equiv Ovate \sqcap \exists hasConfidenceValue.(= 100)$.

In Step 2, we make use of the FaCT-DG reasoner to come up with the complete set $complete_Q$. In Step 3, the colour distance dc and shape distance ds are calculated for each species S in $complete_Q$. Accordingly, two triples are created for S : $[S \text{ hasColourDistance } dc .]$ and $[S \text{ hasShapeDistance } ds .]$ In Step 4, we can represent the above query in SPARQL as follows, with the global distance being the sum of dc and ds :

```
1 SELECT ?species
2 WHERE {?species hasColourDistance ?dc .
3       ?species hasShapeDistance ?ds . }
4 ORDER BY ASC(?dc+?ds)
```

There are pros and cons to global preferences. On the one hand, it is flexible and users can specify the combined distance definition in the order clause (line 4) of the

SPARQL query, *e.g.*, by adjusting the weights (currently both weights are 1) for $?dc$ and $?ds$. On the other hand, there are concerns on how to set such weights and thus the global distance properly. This motivates the following two kinds of preference queries.

Multidimensional Preferences We use multidimensional preferences in the scenarios when the distances from different quantities are incomparable. Now let us revisit the query (2) under multidimensional preferences. Steps 1–3 remain the same, and in Step 4, we can represent such query in SPARQL as follows:

```
1 SELECT ?species
2 WHERE {?species hasColourDistance ?dc .
3       ?species hasShapeDistance ?ds . }
4 PREFERRING
5     LOWEST (?dc)
6     AND
7     LOWEST (?ds)
```

The PREFERRING keyword on line 4 starts the preference definition. The AND keyword (line 6) is used to separate independent preference dimensions. Only the non-dominated species are included in the answer set; *i.e.*, for any species S in the answer set (with colour distance dc_S and shape distance ds_S), there exist *no* species W such that any of the following options is true: (i) $dc_W < dc_S$ and $ds_W < ds_S$, or (ii) $dc_W = dc_S$ and $ds_W < ds_S$, or (iii) $dc_W < dc_S$ and $ds_W = ds_S$.

Cascaded Preference We use cascaded preferences in scenarios where some quantities are more important than the others. If we revisit the query (2) under the cascaded preference that shape is more important than colour.

Steps 1–3 remain the same, and in Step 4, we can represent such query in SPARQL as follows:

```
1 SELECT ?species
2 WHERE {?species hasColourDistance ?dc .
3       ?species hasShapeDistance ?ds . }
4 PREFERRING
5     LOWEST (?ds)
6     CASCADED
7     LOWEST (?dc)
```

In this case, only species with the smallest shape distance are considered. If there is more than one such species, among those species only the ones with the smallest colour distances are included in the answer set.

It should be noted that the answer sets under the global preference are much larger than the multidimensional preferences and cascaded preferences, because in the latter two cases, only the best group of species are considered. In order to answer *top-k* queries (for the *top k* best solutions), we can (iteratively) calculate the next best solutions by removing the group G_1 of best solutions from the candidate solution set, and then calculate the best solutions for the remaining set. Thus, the smallest n is selected such that $|G_1| + \dots + |G_n| > k$.

4.3 Query Experiments

4.3.1 Comparing with Keyword Matching

We first compared our methods to keyword matching. Here, we consider the query based on flower colour and leaf shape, using the “global preference,” where the distance is taken as the average of the distances in each quantity.

Queries were done on the integrated knowledge base, which consists of 19,800 species. The parallel leaf shape and flower colour descriptions were integrated and added into the plant ontology. If a species has more than one shape and/or colour region which matches the query, only the *best-match* (with the smallest distance to the query) is selected for ranking. As introduced in Section 3.4, if the information of one quantity is missing, we use the predicted information based on shape-colour co-occurrence pattern clustering. In total, there were 66 queries, each of which finished within 15 seconds on a 2G Hz Pentium 4 laptop.

Table 2 shows the species resulting from the query with ‘ovate’ leaves and ‘purple to blue’ flowers. Instead of simple keyword matching, the judgement whether a species matches the query is derived from the underlying semantics of these descriptions.

The responses from semantically based queries were compared to those from keyword matching. For each query Q_i , two lists of species S_i and K_i were produced. List $S_i = \{S_{i1}, \dots, S_{i,m_i}\}$ was returned by our method, where m_i is the number of answers returned to Q_i , and S_{ij} ($j = 1 \dots m_i$) is a semantically matched species. List $K_i = \{K_{i1}, \dots, K_{i,n_i}\}$ was returned by the keyword matching, where n_i is the count of answers, and K_{ik} ($k = 1 \dots n_i$) is a keyword matched species. For each list, the distances of all species to the query were calculated and their average distance was returned as the score of this list. That is, for each query Q_i , we get a pair of scores:

$$s(S_i) = \frac{\sum_{j=1}^{m_i} d(S_{ij}, Q_i)}{m_i}, \quad s(K_i) = \frac{\sum_{k=1}^{n_i} d(K_{ik}, Q_i)}{n_i}$$

Both scores are positive and the list with a smaller score matches the query better, as the smaller score means the smaller average distance to the query.

The paired sample t-Test³ was applied to the 66 pairs of scores. The null hypothesis is that the mean of the difference between the scores is zero. This resulted in a value of -4.0311 for t . The null hypothesis was clearly rejected at a confidence level greater than 99% with 65 degrees of freedom (the critical value is 3.220). That t is negative and far from 0 means that the results returned by our method match the queries significantly better because their distances to the query are significantly smaller.

Table 1 compares the precision/recall performance of

Condition	Semantic matching		Keyword matching	
	Precision	Recall	Precision	Recall
Strict match	97.6%	41.0%	77.83%	63.5%
Relaxed match	90.6%	69.5%		

Table 1. Comparison between different levels of matching

our semantic-based method and keyword matching.⁴ Our method clearly outperforms the keyword-based matching in precision, while due to strict logic matching, some good results were ruled out. Using the *relaxed semantic matching* as described in [12], the recall of the semantic matching approach can be significantly improved while the precision remains high.

4.3.2 Comparing Different Preferences

After the first 2 steps in the query process, the reasoner returned 983 species, examples of which are shown in Table 2. Using different preferences, the answer set is presented differently. Generally, the results which match the query exactly were all recovered no matter which preference was used. In our case, the first group of multidimensional preference (M-1), cascaded preference with shape preferred (Cp-1) and cascaded preference with colour preferred (Cc-1) contain exactly the same 6 species, which were all ranked No. 1 under the global preference.

If choosing the global preference, each species is given a global rank based on its distance to the query, which indicates how well it matches the query, as shown in the first column. Note, 66 species which did not originally have information on the queried quantity, such as the species ranked 274, were also returned according to its predicted information. Although they are globally ranked lower in the list, this is still interesting from the domain point of view.

Results with the other two preferences have both their group number, shown in the square brackets in Table 2 (the smaller group number means the better match) and their global ranks, listed after the group number.

Using the multidimensional preference, species 1 and another 5 species match the query perfectly; therefore, the first best match group overrules all other species. In the second [M-2] (third [M-3]) best group, species 7 and 12 (8 and 65) cannot dominate each other because one matches better in colour while the other better in shape; therefore, they both stay in the answer set.

⁴Here we give our definition of the correctness: if the distance of the answer to the query, in terms of each quantity, is less than the corresponding threshold used for integration, then this answer is regarded as matched with the query. The predicted information is also taken into the calculation of the precision and recall.

³<http://mathworld.wolfram.com/Pairedt-Test.html>

Global Preference	Multidimensional Preference	Cascaded Preference	
		shape preferred	colour preferred
1. <i>Veronica himalensis</i> "ovate" "purple to blue"	[M-1] 1. <i>Veronica himalensis</i> "ovate" "purple to blue"	[Cp-1] 1. <i>Veronica himalensis</i> "ovate" "purple to blue"	[Cc-1] 1. <i>Veronica himalensis</i> "ovate" "purple to blue"
8. <i>Limnophila rugosa</i> "ovate" "purple-red to blue"	[M-2] 7. <i>Adenosma indianum</i> "ovate" "pale purple to dark blue"	[Cp-12] 7. <i>Adenosma indianum</i> "ovate" "pale purple to dark blue"	[Cc-14] 12. <i>Caryopteris trichosphaera</i> "ovate-oblong to broadly ovate" "blue to purple"
248. <i>Anemone berlandieri</i> "ovate to obovate" "blue to violet"	[M-2] 12. <i>Caryopteris trichosphaera</i> "ovate-oblong to broadly ovate" "blue to purple"	[Cp-33] 8. <i>Limnophila rugosa</i> "ovate" "purple-red to blue"	[Cc-47] 7. <i>Adenosma indianum</i> "ovate" "pale purple to dark blue"
274. <i>Rheum sublancoelatum</i> "ovate or broadly lanceolate" (predicted)	[M-3] 8. <i>Limnophila rugosa</i> "ovate" "purple-red to blue"	[Cp-89] 274. <i>Rheum sublancoelatum</i> "ovate or broadly lanceolate" (predicted)	[Cc-65] 8. <i>Limnophila rugosa</i> "ovate" "purple-red to blue"
682. <i>Anemone lyallii</i> "ovate to ovate-lanceolate" "pale yellow, or rarely purple"	[M-3] 65. <i>Veronica stelleri</i> "ovate to ovate-orbicular" "blue to purple"	[Cp-214] 12. <i>Caryopteris trichosphaera</i> "ovate-oblong to broadly ovate" "blue to purple"	[Cc-452] 274. <i>Rheum sublancoelatum</i> "ovate or broadly lanceolate" (predicted)

Table 2. Query results for species with ‘ovate’ leaves and ‘purple to blue’ flowers

As opposed to the multidimensional preference, species are ranked based on the preferred quantity in the cascaded preference situation, as shown in the last two columns. If the values of that quantity are the same, they are ranked by the other quantity. It is interesting to compare the different group ranks of the same species if the quantities are preferred differently. This cascaded view is desirable if one has a strong preference in one of the quantities. With same species ranked differently in order to satisfy different preferences, the usability of the query results is therefore improved.

5 Conclusion

In this paper, we extended our earlier work [11, 12], further investigating the general issue: how to take advantage of ontologies to facilitate integrating parallel natural language descriptions of continuous quantities and querying over multiple quantities.

Querying over multiple quantities is an interesting problem, which naturally involves user preferences. In this paper we provided a way to make standard ontology-reasoning and semantic similarity measures cooperate, as well as an appropriate representation of different preferences and applied those to the botanical domain. The results show the benefits of formal representation of the semantics of natural language information and how to reach a better usability in query answering systems.

References

- [1] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. S. eds. OWL Web Ontology Language Reference. <http://www.w3.org/TR/owl-ref/>, Feb 2004.
- [2] J. J. Carroll and J. Z. Pan. XML Schema Datatypes in RDF and OWL. Technical report, W3C Semantic Web Best Practices and Development Group, Mar 2006. W3C Working Group Note, <http://www.w3.org/TR/swbp-xsch-datatypes/>.
- [3] W. Ceusters, B. Smith, and J. M. Fielding. Linksuite: Formally robust ontology-based data and information integration. In *Proceedings of First International Workshop of Data Integration in the Life Sciences (DILS'04)*, volume 2994 of *Lecture Notes in Computer Science*, pages 124–139. Springer, 2004.
- [4] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, February 16 2007.
- [5] J. Z. Pan. A Flexible Ontology Reasoning Architecture for the Semantic Web. *IEEE Transaction on Knowledge and Data Engineering*, 19(2):246 – 260, 2007.
- [6] J. Z. Pan and I. Horrocks. OWL-Eu: Adding Customised Datatypes into OWL. In *Proceedings of Second European Semantic Web Conference (ESWC 2005)*, 2005. An extended and revised version is published in the *Journal of Web Semantics*, 4(1), 29-39.
- [7] E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF, 2006. W3C Candidate Recommendation, <http://www.w3.org/TR/rdf-sparql-query/>.
- [8] W. Siberski, J. Z. Pan, and U. Thaden. Querying the Semantic Web with Preferences. In *Proc. of the 5th International Semantic Web Conference (ISWC2006)*, pages 612 – 624, 2004.
- [9] H. Stuckenschmidt and F. van Harmelen. *Information Sharing on the Semantic Web*. Springer-Verlag, 2004.
- [10] H. Wache, T. Voegele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Huebner. Ontology-based integration of information - a survey of existing approaches. In *Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing*, pages 108–117, Seattle, WA, 2001.
- [11] S. Wang and J. Z. Pan. Ontology-based representation and query colour descriptions from botanical documents. In *Proceedings of OTM Confederated International Conferences*, volume 3761 of *Lecture Notes in Computer Science*, pages 1279–1295. Springer, 2005.
- [12] S. Wang and J. Z. Pan. Integrating and querying parallel leaf shape descriptions. In *Proceedings of International Semantic Web Conference (ISWC2006)*, pages 668–681, Athens, GA, USA, November 2006. Springer.
- [13] S. Wang, D. Rydeheard, and J. Z. Pan. The semantic processing of continuous quantities for discrete terms in ontologies. *Journal of logic and computation*, 2007. To appear.