

# Ontology-Based Representation and Query of Colour Descriptions from Botanical Documents

Shenghui Wang and Jeff Z. Pan<sup>(\*)</sup>

School of Computer Science, University of Manchester, UK  
{wangs, pan}@cs.man.ac.uk

**Abstract.** A proper representation of the semantics of colour descriptions is necessary when colour information needs to be processed semantically, such as in the area of information integration. Semantics-based methods can help information from different sources to be integrated more easily and make final results more accurate and understandable among different sources. This paper introduces an ontology-based representation of the semantics of colour descriptions. By using a quantitative colour model and a Description Logic (DL) with datatype support, the semantics of a single colour term can be represented. Multiple such terms are then combined to generate the semantics of a complex colour description, which is interpreted by using morpho-syntactic rules derived from the analysis of colour descriptions in botanical documents. A colour reasoner, interacting with the FaCT-DG DL reasoner, can support colour-related queries and give domain-oriented results.

## 1 Introduction

Ontologies are currently perceived as an appropriate modelling structure for representing such knowledge as taxonomic knowledge in botany or zoology. Ideally, an ontology captures a shared understanding of certain aspects of a domain. More specifically, an ontology provides a common *vocabulary*, including important concepts, properties and their definitions, and *constraints* regarding the intended meaning of the vocabulary, sometimes referred to as background assumptions. In this paper, we use a multi-parameter ontology to capture the semantics of colour descriptions from botanical documents and to represent them in a plant ontology using the OWL-Eu [1] ontology language. OWL-Eu is an ontology language which extends the W3C standard OWL DL [2] with customised datatypes. A colour reasoner is proposed that interacts with the FaCT-DG DL reasoner [3] to answer colour-related queries which are useful in botanical practice.

Colours play an important role in the identification of plant species. A complete list of species containing those plants that have flowers of the requested colour, can be very helpful to botanists in identifying a plant sample in nature. Colour descriptions of the same species are found in many different *floras*,<sup>1</sup> and

<sup>(\*)</sup> This work is partially supported by the FP6 Network of Excellence EU project Knowledge Web (IST-2004-507842).

<sup>1</sup> A flora is a treatise describing the plants of a region or time.

are therefore treated as parallel sources. For instance, the species *Origanum vulgare* (Marjoram) has at least four colour descriptions of its flowers from four floras:

- “violet–purple”, in *Flora of the British Isles* [4],
- “white or purplish–red”, in *Flora Europaea* [5],
- “purple–red to pale pink”, in *Gray’s Manual of Botany* [6],
- “reddish–purple, rarely white”, in *New Flora of the British Isles* [7].

It has been demonstrated in [8] that extracting and integrating parallel information from different sources can produce more accurate and complete results. Some current projects [9, 10] attempt to store knowledge extracted from natural language documents in electronic forms. These projects generally allow keyword-based queries but do not support a formal representation of the semantics.

In this paper, we present an ontology-based approach [11, 12] to tackle this problem. Information extraction techniques are used to get proper colour descriptions from botanical documents. In order to decompose the semantics of colour descriptions, we propose a quantitative model based on the HSL (Hue Saturation Lightness) colour model. By using a parser based on our BNF syntax, we can quantify complex colour descriptions more precisely; for instance, we support adjective modifiers, ranges, conjunction or disjunction relations indicated by natural language constructions. Based on the semantics of colour descriptions, we can generate an ontology to model such complex colour information in our project. Such an ontology provides a foundation for information integration and domain-oriented query answering.

The semantics of information are important for both extraction and integration. However, existing information integration systems [13] do not process information directly based on their semantics. One obvious reason for this is that there is a deep gap between linguistic and logical semantics [14]. As we will show later in the paper, customised datatypes are crucial to capture the semantics of the quantitative model. This suggests that we can not use the Semantic Web standard ontology language OWL DL for our purpose, since OWL DL does not support customised datatypes. Instead, we use the OWL-Eu ontology language, which is a datatype extension of OWL DL. In our ontology, we can represent complex colour descriptions as OWL-Eu class descriptions. Therefore, we can make use of the subsumption checking reasoning service provided by the FaCT-DG DL reasoner to check if a colour description is more general than another one. A colour reasoner, interacting with the FaCT-DG reasoner, is implemented to answer colour-related species identification queries and return more useful results for practical botanical purposes.

The rest of the paper is structured as follows. Section 2 introduces some technical background knowledge of multi-parameter colour models and the OWL-Eu ontology language. Section 3 provides the morpho-syntactic rules of building complex colour descriptions and their influences in generating final semantics. Section 4 describes how the semantics of colour descriptions are represented in the OWL-Eu language. Section 5 introduces a domain-oriented usage of such a

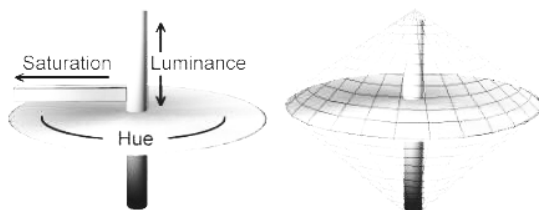
multi-parameter representation, i.e. colour-related species query. Section 6 gives primary experimental results of representation and several queries based on it. Finally, Section 8 concludes this paper and discusses some of our future work.

## 2 Technical Background

### 2.1 The Colour Model

Several colour representations using a multi-parameter colour space (CIE XYZ,  $L^*a^*b^*$ ,  $L^*u^*v^*$ , RGB, CMYK, YIQ, HSV, HSL, etc.) are used in computer graphics and image processing. Colours are quantified as points (or regions) in those spaces. Linguistically naming the physically represented colours has been thoroughly investigated [15].

The psychologically based HSL (Hue Saturation Lightness) model is more accurate than machine-oriented colour models, such as the RGB (Red Green Blue) model, in colour notation, and is second only to natural language [16]. The HSL model was therefore chosen to model basic colour terms parametrically. Its colour space is a double cone, as shown in Figure 1.



**Fig. 1.** HSL Colour Model

In the HSL model, a colour is represented by the following three parameters:

- *Hue* is a measure of the colour tint. In fact, it is a circle ranging from 0 (red) to 100 (red again), passing through 16 (yellow), 33 (green), 50 (cyan), 66 (blue) and 83 (magenta).
- *Saturation* is a measure of the amount of colour present. A saturation of 0 is a total absence of colour (i.e. black, grey or white), a saturation of 100 is a pure colour tint.
- *Lightness* (also Luminance or Luminosity) is the brightness of a colour. A lightness of 0 is black, and 100 is white, between 0 and 100 are shades of grey. A lightness of 50 is used for generating a pure colour.

Each basic colour term corresponds to a small space in the double cone whose centre is the particular point representing its HSL value, that is, instead of a point, a colour term is represented by a cuboid space, defined by a range

triplet (hueRange, saturationRange, lightnessRange). For instance, “purple” is normally defined as the HSL point (83, 50, 25), but is represented in our ontology as the region (78–88, 45–55, 20–30), adding a certain range to each parameter.<sup>2</sup>

## 2.2 OWL DL and Its Datatype Extension OWL-Eu

The OWL Web Ontology Language [2] is a W3C recommendation for expressing ontologies in the Semantic Web. OWL DL is a key sub-language of OWL. Datatype support [18, 19] is one of the most useful features that OWL is expected to provide, and has brought extensive discussions in the RDF–Logic mailing list [20] and Semantic Web Best Practices mailing list [21]. Although OWL provides considerable expressive power to the Semantic Web, the OWL datatype formalism (or simply *OWL datatyping*) is much too weak for many applications. In particular, OWL datatyping does not provide a general framework for customised datatypes, such as XML Schema user-defined datatypes.

To solve the problem, Pan and Horrocks [1] proposed OWL-Eu, a small but necessary extension to OWL DL. OWL-Eu supports customised datatypes through unary datatype expressions (or simply datatype expressions) based on unary datatype groups. OWL-Eu extends OWL DL by extending datatype expressions with OWL data ranges.<sup>3</sup> Let  $\mathcal{G}$  be a unary datatype group. The set of  $\mathcal{G}$ -datatype expressions,  $\mathbf{Dexp}(\mathcal{G})$ , is inductively defined in abstract syntax as follows [1]:

1. *atomic expressions*: if  $u$  is a datatype URIref, then  $u \in \mathbf{Dexp}(\mathcal{G})$ ;
2. *relativised negated expressions*: if  $u$  is a datatype URIref, then  $\mathbf{not}(u) \in \mathbf{Dexp}(\mathcal{G})$ ;
3. *enumerated datatypes*: if  $l_1, \dots, l_n$  are literals, then  $\mathbf{oneOf}(l_1, \dots, l_n) \in \mathbf{Dexp}(\mathcal{G})$ ;
4. *conjunctive expressions*: if  $\{E_1, \dots, E_n\} \subseteq \mathbf{Dexp}(\mathcal{G})$ , then  $\mathbf{and}(E_1, \dots, E_n) \in \mathbf{Dexp}(\mathcal{G})$ ;
5. *disjunctive expressions*: if  $\{E_1, \dots, E_n\} \subseteq \mathbf{Dexp}(\mathcal{G})$ , then  $\mathbf{or}(E_1, \dots, E_n) \in \mathbf{Dexp}(\mathcal{G})$ .

*Uniform Resource Identifiers* (URIs) are short strings that identify Web resources [22]. A *URI reference* (or URIref) is a URI, together with an optional fragment identifier at the end. In OWL, URIrefs are used as symbols for classes, properties and datatypes, etc.

For example, the following XML Schema user-defined datatype

```
<simpleType name = "HueRange">
  <restriction base = "xsd:integer">
    <minInclusive value = "0"/>
    <maxInclusive value = "100"/>
  </restriction>
</simpleType>
```

can be represented by the following conjunctive datatype expression:

$\mathbf{and}(\mathbf{xsd:nonNegativeInteger}, \mathbf{xsd:integerLessThanOrEqualTo100})$ ,

<sup>2</sup> Referring to the NBS/ISCC Color System [17], giving a 100-point hue scale, each major hue places at the middle of its 10-point spread, or at division 5.

<sup>3</sup> This is the *only* extension OWL-Eu brings to OWL DL.

**Table 1.** Colour description patterns and their relative frequencies of occurrence, where X, Y and Z each represent a single colour term or a simpler colour phrase, A is a degree adjective and P is a probability adverb

Colour description pattern	Frequency of occurrence	Example
X	25.5%	“orange”
A X	36.5%	“pale blue”
X to Y (to Z...)	25.9%	“white to pink to red to purple”
X–Y	19.9%	“rose–pink”
X+ish(–)Y	13.2%	“reddish–purple”
X(, Y) or Z	6.5%	“white or violet”
X(, Y), P Z	6.4%	“reddish–purple, rarely white”
X/Y	4.6%	“pink/white”
X, Y	2.8%	“lavender, white–pink”
X(, Y), and Z	2.3%	“white and green”

where `xsd:integerLessThanOrEqualTo100` is the URIrefs for the user-defined datatype  $\leq_{100}$ . Readers are referred to [23] for more details of OWL abstract syntax and semantics.

Similarly to an OWL DL ontology, an OWL-Eu ontology typically contains a set of class axioms, property axioms and individual axioms. FaCT-DG, a datatype group extension of the FaCT DL reasoner, supports TBox reasoning on OWL-Eu ontologies without nominals.

### 3 NL Processing

From a closer observation of the real data in floras, we find that colour descriptions are mostly compound phrases so that they can cover the variations of plant individuals in the field, such as the example shown in Section 1. Complex colour descriptions are built from multiple basic colour terms by certain morpho-syntactic rules. In order to be represented correctly, a complex colour description has to be analysed using the same rules.

We carry out a morpho-syntactic analysis on 227 colour descriptions of 170 species from five floras.<sup>4</sup> Different types of phrases and their relative frequencies of occurrence in the data set are summarised in Table 1 (page 1283). Table 2 (page 1284) gives the corresponding BNF syntax of these phrases for colour description. As shown in Table 1, most patterns describe colour ranges that are built from several atomic colour phrases, such as “blue”, “blue–purple” or “bright yellow”.

There are two steps in our text processing. Firstly, we construct the following atomic colour phrases as basic colour spaces.

<sup>4</sup> They are *Flora of the British Isles* [4], *Flora Europaea* [5], *The New Britton and Brown Illustrated Flora of the Northeastern United States and Adjacent Canada* [6], *New Flora of the British Isles* [7] and *Gray’s Manual of Botany* [24].

**Table 2.** BNF syntax of colour descriptions

$\langle Cterm \rangle ::= red yellow green  \dots$	
$\langle Dmodifier \rangle ::= strong pale bright deep dull light dark  \dots$	
$\langle Pmodifier \rangle ::= usually often sometimes occasionally rarely never  \dots$	
$\langle Cphrase \rangle ::= \langle Cterm \rangle$	
	$  \langle Cterm \rangle [ish][ - ] \langle Cterm \rangle$
	$  \langle Cphrase \rangle - \langle Cphrase \rangle$
	$  \langle Dmodifier \rangle \langle Cterm \rangle$
$\langle Cdescription \rangle ::= \langle Cphrase \rangle$	
	$  \langle Cphrase \rangle \{ to \langle Cphrase \rangle \}$
	$  \langle Cphrase \rangle , \langle Cphrase \rangle$
	$  \langle Cphrase \rangle / \langle Cphrase \rangle$
	$  \langle Cphrase \rangle \{ , \langle Cphrase \rangle \} or \langle Cphrase \rangle$
	$  \langle Cphrase \rangle \{ , \langle Cphrase \rangle \} and \langle Cphrase \rangle$
	$  \langle Cphrase \rangle \{ , \langle Cphrase \rangle \} , \langle Pmodifier \rangle \langle Cphrase \rangle$

**Table 3.** Meanings of adjective modifiers and their corresponding operations on a colour space

Adjective Meaning <sup>a</sup>		Operation <sup>b</sup>
strong	high in chroma	satRange + 20
pale	deficient in chroma	satRange - 20, ligRange + 20
bright	of high saturation or brilliance	satRange + 20, ligRange + 20
deep	high in saturation and low in lightness	satRange + 20, ligRange - 20
dull	low in saturation and low in lightness	satRange - 20, ligRange - 20
light	medium in saturation and high in lightness	satRange - 20, ligRange + 20
dark	of low or very low lightness	ligRange - 20

<sup>a</sup> referring to Merriam–Webster online dictionary.

<sup>b</sup> Referring to the specifications from Colour Naming System (CNS) [25], saturation and lightness are each divided into 5 levels, which causes a range/ranges to change by 20 (100/5).

- X** : This is a single colour space, i.e. (hueRange, satRange, ligRange).
- A X** : We need to modify the space of X according to the meaning of A, as shown in Table 3 (page 1284). For example, “light blue” is represented as (61–71, 70–80, 65–75) where “blue” is (61–71, 90–100, 45–55).
- X–Y** : This represents an intermediate colour between the two colours X and Y [25]. For example, “blue–purple” is generated from the halfway colour between “blue” (66, 100, 50) and “purple” (83, 50, 25), that is, the colour with HSL value of (75, 75, 38), the hue, saturation and lightness of which are calculated by the following formulae (similar calculation for saturation and lightness) and finally represented by the range triple (70–80, 70–80, 33–43).

$$Hue_{X-Y} = \frac{Hue_X + Hue_Y}{2}$$

**Xish–Y** : This denotes a quarterway value between the two colours [25], closer to the latter colour term. For instance, “reddish–purple” means it is basically purple (83, 50, 25) but reflecting a quarterway deviation to red (100, 100, 50), so the final hue range for “reddish–purple” is (87, 63, 34) calculated by the following formula (similar formulas for saturation and lightness), which is finally (82–92, 58–68, 29–39).

$$Hue_{X_{ish}-Y} = Hue_Y + \frac{Hue_X - Hue_Y}{4}$$

Secondly, we build up combined colour spaces based on basic ones. Specifically, combined colour spaces are built up by a colour reasoner, according to the following morpho-syntactic rules:

1. If basic colour terms are connected by one or more “to”s, the final colour space should be the whole range from the first colour to the last one. For instance, if light blue is (66, 100, 70) and purple is (83, 50, 25), “light blue to purple” should be the whole range (66–83, 50–100, 25–70), which contains any colour in between.

Note that special care is needed for ranges starting or ending with a grey colour, such as “white to purple”. In the HSL model, colours ranging from white, through different levels of grey, to black have no hue and saturation values. For instance, the HSL value of “white” is (0, 0, 100), while “red” also has a hue value of 0. A special rule for building such kind of ranges has to be followed; that is, a range from colour A (0, 0,  $l_a$ ) to colour B ( $h_b$ ,  $s_b$ ,  $l_b$ ) should be ( $\overline{h_b - 5} - \overline{h_b + 5}$ ,  $0 - s_b$ ,  $l_a - l_b$ ). For example, “white to purple” should be represented by the triple (78–88, 0–50, 25–100).

2. If basic colour terms are connected by any of these symbols: “or”, “and”, comma (“,”) or slash (“/”), they are treated as separate colour spaces; that is, they are disjoint from each other. For instance, “white, lilac or yellow” means that the colour of this flower could be either white or lilac or yellow, not a colour in between.

Notice that “and” is treated as a disjunction symbol because, in floras, it normally means several colours can be found in the same species, instead of indicating a range by normal logical conjunction. For instance, flowers of species *Rumex crispus* (Curled Dock) are described as “red and green”, which means flowers that either red or green may occur in the same plant, but it does not mean that one flower is both red and green.

By using a parser based on our BNF syntax, we can generate an OWL-Eu ontology to model complex colour information.

## 4 Representation of Colour Descriptions in OWL–Eu

Based on the morpho-syntactic rules introduced in Section 3, we can decompose the semantics of colour descriptions into several quantifiable components, which

can be represented as DL datatype expressions. In this section, we will show how to use the OWL-Eu ontology language to represent the semantics of a colour description.

The fragment of our plant ontology  $\mathcal{O}_C$  contains Colour as a primitive class. Important primitive classes in  $\mathcal{O}_C$  include

```
Class(Species), Class(Flower), Class(Colour);
```

important object properties in  $\mathcal{O}_C$  include

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

important datatype properties in  $\mathcal{O}_C$  include

```
DatatypeProperty(hasHue Functional
  range(and(xsd:nonNegativeInteger, xsdx:integerLessThanOrEqualTo100))),
DatatypeProperty(hasSaturation Functional
  range(and(xsd:nonNegativeInteger, xsdx:integerLessThanOrEqualTo100))),
DatatypeProperty(hasLightness Functional
  range(and(xsd:nonNegativeInteger, xsdx:integerLessThanOrEqualTo100))),
```

which are all functional properties. A *functional* datatype property relates an object with at most one data value. Note that the datatype expression

```
and(xsd:nonNegativeInteger, xsdx:integerLessThanOrEqualTo100)
```

is used as the range of the above datatype properties.

Based on the above primitive classes and properties, we can define specific colours, such as Purple, as OWL-Eu defined classes (indicated by the keyword “complete”).

```
Class(Purple complete Colour
  restriction(hasHue someValuesFrom
    (and(xsdx:integerGreaterThanOrEqualTo78,
        xsdx:integerLessThanOrEqualTo88)))
  restriction(hasSaturation someValuesFrom
    (and(xsdx:integerGreaterThanOrEqualTo47,
        xsdx:integerLessThanOrEqualTo52)))
  restriction(hasLightness someValuesFrom
    (and(xsdx:integerGreaterThanOrEqualTo20,
        xsdx:integerLessThanOrEqualTo30))))
```

In the above class definition, datatype expressions are used to restrict the values of the datatype properties *hasHue*, *hasSaturation* and *hasLightness*. Note that not only colour terms but also complex colour descriptions can be represented in OWL-Eu classes, as long as they can be transformed into proper colour subspaces with constraints on their hue, saturation and lightness.

As colour descriptions are represented by OWL-Eu classes, we can use the subsumption checking service provided by the FaCT-DG reasoner to check if one colour description is more general than another. Namely, if ColourA is subsumed

by ColourB, we say that ColourB is more general than ColourA. The formal representation of colour descriptions makes it possible to express a query about a range of colours, such as to retrieve all species which have “bright rose–pink” or “light blue to purple” flowers, with the help of the FaCT-DG DL reasoner.

## 5 Domain-Oriented Queries

The flower colour of an individual plant is an important distinguishing feature for identifying which species it belongs to. The species identification that botanists are interested in can be written as a query: “Given a certain colour, tell me all the possible species whose flowers have such a colour.” We would like to point out that, from a botanical point of view, one has to take the variations between individuals in nature into account. In other words, botanists rarely use colour as a strict criterion. It is more appropriate to answer such species identification queries in a fuzzy manner, that is, returning a list which contains all species that could match the query. This kind of query, which is particularly suitable for domain interests, is called domain-oriented queries.

We can answer species identification queries based on subsumption queries that are supported by the FaCT-DG DL reasoner. For example, if our plant ontology contains the following class axioms:

```
Class(SpeciesA restriction(hasPart someValueFrom(FlowerA)))
Class(FlowerA restriction(hasColour someValueFrom(ColourA)))
Class(SpeciesB restriction(hasPart someValueFrom(FlowerB)))
Class(FlowerB restriction(hasColour someValueFrom(ColourB)))
```

and if from the definitions of ColourA and ColourB we can conclude that ColourA is subsumed by ColourB, when we ask our DL reasoner whether the above ontology entails that SpeciesA is subsumed by SpeciesB, the reasoner will return “yes”. By using this kind of subsumption query, we can, for example, conclude that a species having “golden–yellow” flowers is subsumed by a more general species which has “yellow” flowers, which again is subsumed by another species which has “red to orange to yellow” flowers. Therefore, if one asks “Which species might have yellow flowers?”, our colour reasoner will return all these three species.

For species identification, this hierarchical subsumption matching is very useful for shortening the possible species list. After classification reasoning, we have already had three different levels of matchings: **Exact** matching ( $\text{Class}_{\text{RealSpecies}} \equiv \text{Class}_{\text{QuerySpecies}}$ ), **PlugIn** matching ( $\text{Class}_{\text{RealSpecies}} \sqsubseteq \text{Class}_{\text{QuerySpecies}}$ ) and **Subsume** matching ( $\text{Class}_{\text{RealSpecies}} \sqsupseteq \text{Class}_{\text{QuerySpecies}}$ ) [26, 3]. Actually there is another possible species list, which is not covered by the above three kinds of matchings, that is, **Intersecting** matching ( $\neg(\text{Class}_{\text{RealSpecies}} \sqcap \text{Class}_{\text{QuerySpecies}} \sqsubseteq \perp)$ ). For example, if a species has “greenish–yellow” flowers, it would also be possible to find in the field an individual which has “yellow” flowers. Although this latter list has a lower probability to contain the correct answers, it is still helpful from botanical point of view.

We have implemented a colour reasoner to reduce our domain problems into standard Description Logics reasoning problems. In fact, it interacts with the

**Table 4.** Query results of species having “yellow” flowers (partial)

Species	Flower colour	Matching type
<i>Amsinckia menziessi</i>	yellow	Exact matching
<i>Ranunculus acris</i>	golden–yellow	PlugIn matching
<i>Barbarea vulgaris</i>	yellow to pale yellow	Subsume matching
<i>Eucalyptus globulus</i>	creamy–white to yellow	Subsume matching
<i>Anaphalis margaritacea</i>	white, yellow to red	Subsume matching
<i>Castilleja wightii</i>	yellow–orange–apricot–red	Subsume matching
<i>Tropaeolum majus</i>	yellow to orange to red	Subsume matching
<i>Myrica californica</i>	green to red to brown	Subsume matching
<i>Trillium chloropetalum</i>	dark red to greenish–white	Subsume matching
<i>Artemisia californica</i>	whiteish–yellow	Intersection matching
<i>Rumex acetosella</i>	reddish–yellow	Intersection matching
<i>Rhodiola sherriffii</i>	greenish–yellow	Intersection matching
<i>Lasthenia californica</i>	yellow–orange	Intersection matching
<i>Mimulus aurantiacus</i>	orange	Intersection matching
<i>Artemisia douglasiana</i>	whiteish–green to whiteish–yellow	Intersection matching
<i>Eschscholzia californica</i>	deep orange to pale yellow	Intersection matching

FaCT-DG reasoner, in order to answer domain-oriented queries. First of all, the colour in a query is represented by an OWL-Eu class  $Q$  with datatype constraints about its hue, saturation and lightness. Secondly, the colour reasoner calculates the complete set of colours  $complete_Q$  which satisfies the above four levels of matching. Specifically,  $complete_Q$  consists of the following four sets.

- $equiv_Q$ : all elements are equivalent to the class  $Q$ , such as “yellow”;
- $sub_Q$ : all elements are subsumed by the class  $Q$ , such as “golden–yellow”;
- $super_Q$ : all elements subsume the class  $Q$ , such as “yellow to orange to red”;
- $intersection_Q$ : all elements intersect with the class  $Q$ , such as “greenish–yellow”.

Note that the first two contain answers with 100% confidence, while the latter two contain those with less confidence. Thirdly, in order to find all species that have flowers whose colour satisfies the query, the colour reasoner interacts with the Fact-DG reasoner to return those species which have flowers whose colour is contained in  $complete_Q$  set.

## 6 Experiments and Discussions

In this section, we will present some of the experiments, in terms of species identification queries, we did with our plant ontology.

We chose 100 colour terms, which are commonly used in floras, as basic colour terms. For each basic term, we obtained its RGB value by referring to the X11 Colour Names,<sup>5</sup> converted it into the corresponding HSL value and finally defined it as ranges in hue, saturation and lightness (as described in Section 4).

<sup>5</sup> [http://en.wikipedia.org/wiki/X11\\_Color\\_Names](http://en.wikipedia.org/wiki/X11_Color_Names)

**Table 5.** Query results of species having “light blue” flowers (partial)

Species	Flower colour	Matching type
<i>Aster chilensis</i>	light blue	Exact matching
<i>Ceanothus thyrsiflorus</i>	pale blue to bright blue	Subsume matching
<i>Heliotropium curassavicum</i>	white to bluish	Subsume matching
<i>Linum bienne</i>	pale blue to lavender	Subsume matching
<i>Phacelia ramosissima</i>	white, sometimes pale blue	Subsume matching
<i>Viola adunca</i>	light blue to purple	Subsume matching
<i>Triteleia laxa</i>	blue to violet	Intersection matching
<i>Vinca major</i>	light-to-dark blue-violet	Intersection matching
<i>Dichelostemma congestum</i>	pink to blue	Intersection matching

By using the method described in Section 3 on the same data set we used for linguistic analysis, each colour description was transformed into a small colour space. A simple plant ontology, mentioned in Section 4, was constructed using the OWL-Eu language. This ontology contains 170 species classes, each species has a flower part which has a colour property. The colour property is represented by a datatype expression. For example, species *Viola adunca* has “light blue to purple” flowers.

```

Class(Viola_adunca complete Species
  restriction(hasPart someValuesFrom(Viola_adunca_flower))),
Class(Viola_adunca_flower complete Flower
  restriction(hasColour someValuesFrom(Viola_adunca_flower_colour))),
Class(Viola_adunca_flower_colour complete Colour
  restriction(hasHue someValuesFrom
    (and(xsd:integerGreaterThanOrEqualTo66,
        xsd:integerLessThanOrEqualTo83)))
  restriction(hasSaturation someValuesFrom
    (and(xsd:integerGreaterThanOrEqualTo50,
        xsd:integerLessThanOrEqualTo100)))
  restriction(hasLightness someValuesFrom
    (and(xsd:integerGreaterThanOrEqualTo25,
        xsd:integerLessThanOrEqualTo70))))

```

In the species identification queries, we used colour descriptions with different levels of complexity (as shown in Table 1 on page 1283). Some of the results are presented in Tables 4, 5 and 6, in the order of complexity of colours: “yellow”, “light blue”, “light blue to purple”.

From Table 4, all species which could have “yellow” flowers have been queried out. In the result, we can find the species whose flowers are explicitly described as “yellow” or some variation thereof. We also find that those species which have “orange”, “green to red to brown” or “dark red to greenish–white” flowers are also returned by our colour reasoner, as the results of subsumption or intersection

**Table 6.** Query results of species having “light blue to purple” flowers (partial)

Species	Flower colour	Matching type
<i>Viola adunca</i>	light blue to purple	Exact matching
<i>Aster chilensis</i>	pale blue	PlugIn matching
<i>Scoliopus bigelovii</i>	purple	PlugIn matching
<i>Linum bienne</i>	pale blue to lavender	PlugIn matching
<i>Myosotis latifolia</i>	blue	PlugIn matching
<i>Eriodictyon californicum</i>	light purple/lavender	PlugIn matching
<i>Ceanothus thyrsoiflorus</i>	pale blue to bright blue	PlugIn matching
<i>Iris douglasiana</i>	pale lavender to blue to purple	PlugIn matching
<i>Delphinium variegatum</i>	royal blue to purple	PlugIn matching
<i>Verbena lasiostachys</i>	blue–purple	PlugIn matching
<i>Cirsium occidentale</i>	red–purple	Intersection matching
<i>Cirsium vulgare</i>	reddish–purple	Intersection matching
<i>Epilobium cilatum</i>	white to pink to red to purple	Intersection matching
<i>Heliotropium curassavicum</i>	white to bluish	Intersection matching
<i>Lupinus eximus</i>	blue to purple, sometimes lavender	Intersection matching
<i>Polygala californica</i>	rose–pink/purple	Intersection matching
<i>Solanum</i>	white to white–lavender to pink/blue	Intersection matching
<i>Vinca major</i>	light–to–dark blue–violet	Intersection matching
<i>Dichelostemma congestum</i>	pink to blue	Intersection matching
<i>Sisyrinchium bellum</i>	strong blue–purple	Intersection matching
<i>Stachys bullata</i>	light purple to pink to white	Intersection matching
<i>Triteleia laxa</i>	blue to violet	Intersection matching

matching. This is consistent with their real meanings, because in our colour model “yellow” is contained by or intersects with them.

We can query in a specific manner, such as to find species which have “light blue” flowers but excluding those with “dark blue” flowers (see Table 5); or in a more general style, such as to query all species which could have flowers ranging from “light blue to purple” (see Table 6). All of these owe to our quantitative model which makes it possible to compare and reason with classes at a semantic level.

As stated in Section 5, the resulting list is from four different levels of matching, which gives a complete list for species identification. We can also specify to stop at certain levels of matching to get results with different confidences, such as only return those species which fully satisfy the query.

The semantics of a colour term or a complex colour description is decomposed and represented by a group of ranges in multiple numerical parameters, which is a small subspace in a multi–dimensional space. Numerical representation makes it easy to build ranges between colours, but a further observation shows that this is not as obvious as we thought. For example, there could be different ways of interpreting the meaning of “light blue to purple” (see Fig. 2):

- light blue to purple directly (area B),
- light blue to blue then to purple,
- light blue to light purple then to purple,
- the whole rectangle.

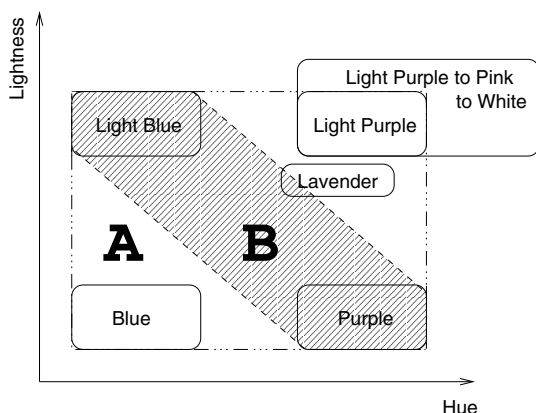


Fig. 2. Range between “light blue” and “purple”

In our experiment (see Table 6), we used the last option (the whole rectangle). We plan to extend our work and to allow the users to pick up one of the above options when they query with the keyword “to”.

## 7 Related Work

Automatically integrating information from a variety of sources has become a necessary feature for many information systems [13]. Compared to structured or semi-structured data sources, information in natural language documents is more cumbersome to access [27]. Our work focuses mainly on parallel information extraction and integration from homogeneous monolingual (English) documents.

Information Extraction (IE) [28] is a common Natural Language Processing (NLP) technique which can extract information or knowledge from documents. Ontologies, containing various semantical expressions of domain knowledge, have recently been adopted in many IE systems [29, 30, 31]. Semantics embedded in ontologies can boost the performance of IE in terms of precision and recall [32]. Since they can be shared by different sources, ontologies also play an important role in the area of information integration [13, 33, 27]. Logic reasoning, as supported by ontology languages, is also introduced naturally into the extraction and integration process [34, 35, 32].

The work in this paper is similar to some research in computational linguistics, such as Lexical Decomposition [36], which attempts to break the meanings of words down to more basic categories. Instead of certain qualitative criteria, our method attempts to give precise semantics to the classes in an ontology by using multiple basic and quantifiable classes. This makes comparing and reasoning with classes easier.

The quantitative semantic model can produce more useful results for real domain purposes. Specifically, in botanical domain, many current plant databases

can only support keyword-based query, such as the ActKey [9] provided by the floras project,<sup>6</sup> ePIC project [10], etc. They rely heavily on the occurrence of keywords, which are normally very general and therefore less robust. As demonstrated in Section 6, our method can compute on their real semantics instead of pure keyword matching, which supports more flexible-styled queries and gets more useful results.

## 8 Conclusion and Outlook

This paper introduces an ontology-based approach to capture the semantics of colour descriptions and to support colour-related species identification queries. It turns out that, even in a limited domain, representing the semantics of colour descriptions is not a trivial problem. Based on a multi-parameter semantic model for colour descriptions and certain morpho-syntactic rules, we have implemented an NLP parser which translates complex colour descriptions into quantifiable logic representations. More importantly, a colour reasoner is implemented to carry out queries for real botanical applications by interacting with the FaCT-DG DL reasoner.

We have shown that our approach outperforms text-based approaches by providing more precise integration results. Firstly, our quantifiable model makes it possible to reason and query on a semantic level. Relations between colour descriptions are more tractable. For example, yellow is between red and green in terms of hue, lilac is lighter than purple although they have the same hue. Furthermore, based on the support of adjective modifiers and ranges, we can query in a detailed manner, such as “light blue”, which excludes pure blue and dark blue. We can also query on a fuzzy manner, such as “light blue to purple”, as required for particular domain purposes.

Interestingly, our work also provides a use case for the OWL-Eu ontology language. OWL-Eu extends OWL DL with user-defined datatypes, which are needed to represent the ranges for hue, saturation and lightness used in colour descriptions, not to mention the degree adjective described in Table 3.

Encouraged by the existing results, we plan to further extend our work on ontology-based species identification queries. Firstly, as suggested in Section 6, a future version of our colour reasoner should provide several options so as to allow users to decide their intended meaning of the ‘to’ keyword. Technically, this requires the use of not only unary but also n-ary datatype expressions as constraints on datatype properties *hasHue*, *hasSaturation* and *hasLightness*. To capture these constraints, we need to use the OWL-E [37, 3] ontology language, which is the n-ary extension of OWL-Eu.

Taking advantage of the quantitative representation, similarities between different descriptions from different authors can be measured as the distances between their corresponding colour subspaces. As long as an appropriate distance measurement is chosen, such distances can easily tell us how different two descriptions are and to what extent they overlap with each other. How to define

---

<sup>6</sup> <http://www.efloras.org/>

the distance and how to use it for integration is an interesting work to explore in the future.

Another future work is to represent the probabilistic information in the ontology. There are many descriptions with adverbs of quantification, such as “sometimes”, “rarely”, “often”, etc., which also indicate the probability of certain colours. Because current ontology languages do not support the annotation of classes with probabilities, the probability aspect is ignored in the text processing which affects the precision of integration. However, there are several attempts to extend DL languages with fuzzy expressions [38, 39, 40], which, in the future, may be used to enable our logic representation to capture more of the original semantics implied by natural language.

Furthermore, our approach is applicable in other similar areas, such as the representation of leaf shapes, which is another key feature of identifying species. We have started to experiment with a quantitative model generated by a Super-Shape formula [41]. We expect that our method can also produce similar results in this case.

## References

1. Pan, J.Z., Horrocks, I.: OWL-Eu: Adding Customised Datatypes into OWL. In: Proc. of Second European Semantic Web Conference (ESWC 2005). (2005)
2. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., eds., L.A.S.: OWL Web Ontology Language Reference. <http://www.w3.org/TR/owl-ref/> (2004)
3. Pan, J.Z.: Description Logics: Reasoning Support for the Semantic Web. PhD thesis, School of Computer Science, The University of Manchester (2004)
4. Clapham, A., Tutin, T., Moore, D.: Flora of the British Isles. Cambridge University Press (1987)
5. Tutin, T.G., Heywood, V.H., Burges, N.A., Valentine, D.H., Moore(eds), D.M.: Flora Europaea. Cambridge University Press (1993)
6. Gleason, H.: The New Britton and Brown Illustrated Flora of the Northeastern United States and Adjacent Canada. Hafner Publishing Company, New York (1963)
7. Stace, C.: New Flora of the British Isles. Cambridge University Press (1997)
8. Wood, M.M., Lydon, S.J., Tablan, V., Maynard, D., Cunningham, H.: Using parallel texts to improve recall in ie. In: Proceedings of Recent Advances in Natural Language Processing (RANLP-2003), Borovetz, Bulgaria (2003) 505–512
9. : Actkey. (Web Page <http://flora.huh.harvard.edu:8080/actkey>)
10. Royal Botanic Gardens, K.: electronic plant information centre. (Published on the Internet <http://www.kew.org/epic/>)
11. Wood, M., Lydon, S., Tablan, V., Maynard, D., Cunningham, H.: Populating a database from parallel texts using ontology-based information extraction. In: Meziane, F., Métails, E., eds.: Proceedings of Natural Language Processing and Information Systems, 9th International Conference on Applications of Natural Languages to Information Systems, Springer (2004) 254–264
12. Wood, M., Wang, S.: Motivation for “ontology” in parallel-text information extraction. In: Proceedings of ECAI-2004 Workshop on Ontology Learning and Population (ECAI-OLP), Poster, Valencia, Spain (2004)

13. Wache, H., Voegele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Huebner, S.: Ontology-based integration of information - a survey of existing approaches. In: Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA (2001) 108–117
14. Dik, S.C.: Coordination: Its implications for the theory of general linguistics. North-Holland, Amsterdam (1968)
15. Lammens, J.M.: A computational model of color perception and color naming. Ph.D. thesis, State University of New York (1994)
16. Berk, T., Brownston, L., Kaufman, A.: A human factors study of color notation systems for computer graphics. *Communications of the ACM* **25** (1982) 547–550
17. U.S. Department of Commerce, National Bureau of Standards: Color: Universal Language and Dictionary of Names. NBS Special Publication 440. U.S. Government Printing Office, Washington D.C. (1976) (S.D. Catalog No. C13.10:440).
18. Pan, J.Z., Horrocks, I.: Extending Datatype Support in Web Ontology Reasoning. In: Proc. of the 2002 Int. Conference on Ontologies, Databases and Applications of SEMantics (ODBASE 2002). (2002) 1067–1081
19. Pan, J.Z., Horrocks, I.: Web Ontology Reasoning with Datatype Groups. In: Proc. of the 2nd International Semantic Web Conference (ISWC2003). (2003)
20. : <http://lists.w3.org/archives/public/www-rdf-logic/>. W3C Mailing List (starts from 2001)
21. : <http://lists.w3.org/archives/public/public-swbp-wg/>. W3C Mailing List (starts from 2004)
22. Group, J.W.U.P.I.: URIs, URLs, and URNs: Clarifications and Recommendations 1.0. URL <http://www.w3.org/TR/uri-clarification/> (2001) W3C Note.
23. Patel-Schneider, P.F., Hayes, P., Horrocks, I.: OWL Web Ontology Language Semantics and Abstract Syntax. Technical report, W3C (2004) W3C Recommendation.
24. Fernald, M.: Gray's Manual of Botany. American Book Company, New York (1950)
25. Berk, T., Brownston, L., Kaufman, A.: A new color-naming system for graphics languages. *IEEE Computer Graphics and Applications* **2** (1982) 37–44
26. Li, L., Horrocks, I.: A Software Framework For Matchmaking Based on Semantic Web Technology. In: Proc. of the Twelfth International World Wide Web Conference (WWW 2003), ACM (2003) 331–339
27. Williams, D., Poulouvassilis, A.: Combining data integration with natural language technology for the semantic web. In: Proc. Workshop on Human Language Technology for the Semantic Web and Web Services, at ISWC'03. (2003)
28. Gaizauskas, R., Wilks, Y.: Information Extraction: Beyond Document Retrieval. In: Computational Linguistics and Chinese Language Processing, Number 2 (1998) 17–60
29. Embley, D., Campbell, D., Liddle, S., Smith, R.: Ontology-based extraction and structuring of information from data-rich unstructured documents. In: Proceedings of International Conference On Information And Knowledge Management, 7, Bethesda, Maryland, USA. (1998)
30. Maedche, A., Neumann, G., Staab, S.: Bootstrapping an ontology-based information extraction system. studies in fuzziness and soft computing. In Szczepaniak, P., Segovia, J., Kacprzyk, J., Zadeh, L.A., eds.: *Intelligent Exploration of the Web*. Springer, Berlin (2002)
31. Alani, H., Kim, S., Millard, D.E., Weal, M.J., Hall, W., Lewis, P.H., Shadbolt, N.R.: Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems* (2003) 14–21

32. Ferrucci, D., Lally, A.: UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Journal of Natural Language Engineering* **10** (2004) 327–348
33. Goble, C., Stevens, R., Ng, G., Bechhofer, S., Paton, N., Baker, P., Peim, M., Brass, A.: Transparent access to multiple bioinformatics information sources. *IBM Systems Journal Special issue on deep computing for the life sciences* **40** (2001) 532 – 552
34. Calvanese, D., Giuseppe, D.G., Lenzerini, M.: Description logics for information integration. In Kakas, A., Sadri, F., eds.: *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski*. Volume 2408 of *Lecture Notes in Computer Science*. Springer (2002) 41–60
35. Maier, A., Schnurr, H.P., Sure, Y.: Ontology-based information integration in the automotive industry. In: *Proceedings of the 2nd International Semantic Web Conference (ISWC2003)*, 20-23 October 2003, Sundial Resort, Sanibel Island, Florida, USA, Springer (2003) 897–912 None.
36. Dowty, D.R.: *Word Meaning and Montague Grammar*. D. Reidel Publishing Co., Dordrecht, Holland (1979)
37. Pan, J.Z.: Reasoning Support for OWL-E (Extended Abstract). In: *Proc. of Doctoral Programme in the 2004 International Joint Conference of Automated Reasoning (IJCAR2004)*. (2004)
38. Tresp, C., Molitor, R.: A description logic for vague knowledge. In: *Proceedings of the 13th biennial European Conference on Artificial Intelligence (ECAI'98)*, Brighton, UK, J. Wiley and Sons (1998) 361–365
39. Straccia, U.: Transforming fuzzy description logics into classical description logics. In: *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA-04)*. Number 3229 in *Lecture Notes in Computer Science*, Lisbon, Portugal, Springer Verlag (2004) 385–399
40. Giorgos Stoilos, Giorgos Stamou, V.T.J.Z.P., Horrocks, I.: A Fuzzy Description Logic for Multimedia Knowledge Representation. In: *Proc. of the International Workshop on Multimedia and the Semantic Web*. (2005) To appear.
41. Gielis, J.: A generic geometric transformation that unifies a wide range of natural and abstract shapes. *American Journal of Botany* **90** (2003) 333–338