

Improving the efficiency of ontology engineering by introducing prototypicality

Xavier Aimé¹ and Frédéric Fürst² and Pascale Kuntz³ and Francky Trichet⁴

Abstract. This paper introduces the notion of prototypicality in Ontology Engineering. Three kinds of prototypicality are considered: a concept can be more or less representative of its super-concept (conceptual prototypicality); a term can be more or less associated to a concept (lexical prototypicality); an instance can be more or less representative of its concept (instance prototypicality). Prototypicalities are modeled as order relations which allow to modulate links between concepts, terms and instances within an ontology. To calculate prototypicality gradients used to quantify these orders, we advocate a specific method which is based on all the components of an ontology (i.e. concepts, properties and instances) and a corpus. This paper also underlines the relevance of prototypicality for improving the efficiency of Ontology Engineering processes, in particular Ontology Personalization and Semantic-based Information Retrieval.

1 Introduction

Currently, a lot of Information Systems exploit ontologies which are defined as conceptual representations of knowledge related to a given field and are built according to a consensus shared by the members of an ingroup⁵. Classically, an ontology is composed of ordered sets of concepts and properties⁶. But such an ontology does not capture all the knowledge the members of the ingroup have on the domain. In particular, it does not specify how a concept is representative of its upper concept. Nevertheless, this notion, called *prototypicality* in cognitive psychology, underlies all conceptual categorisation [10, 4]. For example, parrots, sparrows and chicken are all considered as birds, but the concept of sparrow is more close to the concept of bird than those of parrot and chicken (at least for european people). In other words, thinking to a bird leads more to think to a sparrow than to a parrot or a chicken.

We propose to introduce this notion in ontology, as a supplementary knowledge layer above existing models. We introduce prototypicalities between two concepts hierarchically linked (directly or not), but also between concepts and their terms, and between concepts and their instances. In practice, prototypicalities are represented by weights added on the links defined between the considered elements (*isa* links between concepts, *instanciation* links between concepts and instances, *namming* links between concepts and

terms). To perform the calculation of these weights, we have defined semi-automatic and automatic methods which are based on (1) the semantics expressed in the ontology, (2) a textual corpus of the domain which is considered and (3) a population of concept instances. Our work falls within the scope of the semiotic approach of Knowledge Engineering, which takes the three dimensions of knowledge [6, 7] (i.e. intensional, extensional and expressional dimension) into account.

Various applications of prototypicality can be considered in Ontology Engineering and Knowledge-Based Systems (KBS). The main issues we focus on are KBS personalization and information retrieval, mainly in the context of industrial projects.

The rest of this paper is structured as follows. Section 2 introduces the model we advocate to equip ontology with prototypicality. Each type of prototypicality (i.e. conceptual, lexical and instance prototypicality) is associated with a specific gradient and the method used to compute it. This section also presents how an emotional state can be used to modulate the gradients. Section 3 describes how prototypicality can be used to refine information retrieval and KBS personalization. Section 4 shows the results of two experiments respectively dedicated to environment legal intelligence and environment policy.

2 Introducing prototypicality in ontology

To formalize the notion of prototypicality, we consider an ontology O (related to a domain D and an ingroup G) as a tuple:

$$O_{(D,G)} = \{\mathcal{C}, \mathcal{P}, \mathcal{I}, \leq^C, \leq^P, dom, codom, \sigma, L\} \text{ where}$$

- $\mathcal{C}, \mathcal{P}$ et $\mathcal{I}$ are respectively the sets of concepts, properties and instances;
- $\leq^C: \mathcal{C} \times \mathcal{C}$ et $\leq^P: \mathcal{P} \times \mathcal{P}$ are partial orders which organize the hierarchies of concepts and properties⁷;
- $dom: \mathcal{P} \rightarrow \mathcal{C}$ and $codom: \mathcal{P} \rightarrow (\mathcal{C} \cup Datatypes)$ associates to each property its domain and co-domain;
- $\sigma: \mathcal{C} \rightarrow \wp(\mathcal{I})$ associates to each concept its instances;
- $L = \{L_C \cup L_P \cup L_I, term_c, term_p, term_i\}$ is the vocabulary of the language used by G to speak about the domain D , where L_C, L_P and L_I are the sets of terms associated to $\mathcal{C}, \mathcal{P}$ and $\mathcal{I}$, and where $term_c: \mathcal{C} \rightarrow \wp(L_C)$, $term_p: \mathcal{P} \rightarrow \wp(L_P)$ and $term_i: \mathcal{I} \rightarrow \wp(L_I)$ are the fonctions which associate to each concept, property or instance the terms used to denote them.

We introduce the prototypicality by weighting the different kinds of links of the ontology. These weights form numerical gradients which express three kinds of prototypicality:

¹ LINA, University of Nantes & Tennaxia, France, xaime@tennaxia.fr

² MIS, University of Picardie, France, frederic.furst@u-picardie.fr

³ LINA, University of Nantes, France, pascale.kuntz-cosperec@univ-nantes.fr

⁴ LINA, University of Nantes, France, francky.trichet@univ-nantes.fr

⁵ The term *ingroup* comes from sociology. We use it to denote the set of individuals who share the conceptualisation expressed in the ontology O , and not only the set of individuals who have built O .

⁶ We call property both concept attributes and binary relations between concepts.

⁷ $c_1 \leq^C c_2$ means that the concept c_2 is subsuming the concept c_1 .

- **conceptual prototypicality** defined on links $\leq^C$: two concepts, with one inherits from the other, can be more or less close together. More precisely, in a sibship of concepts, some of them are more prototypical of their father than the others. For example, among all plane types, the most representative, about which we tend to think when we think to the concept of plane, is rather a modern and commercial plane, and not the first biplane or a plane propelled by muscular strength.
- **lexical prototypicality** defined on links $term_c$ and $term_p$: for a given concept (or a property) with several terms used to name it, some terms are most often used than the others. For example, *plane* is rather used than *aeroplane* or *heavier-than-air*.
- **instance prototypicality** defined on links σ : among the instances of a given concept, some instances are more representative than the others. For example, when thinking to the concept of *plane-which-flew-over-the-Atlantic*, we think rather to the *Spirit-of-St.-Louis* than to one of the current planes that crosses the Atlantic everyday.

2.1 Conceptual prototypicality gradient

From a semiotic perspective, the conceptual prototypicality gradient, that we call *Semiotic-Based Conceptual Prototypicality Gradient* (SPG), is composed of three components: (1) an **intensional component**, based on the comparison between the concept intensions (*i.e.* the properties of concepts), (2) an **extensional component**, based on the comparison between concept instances and (3) an **expressional component**, based on the comparison between concept terms which appears in a corpus. The calculation of the extensional component requires a set of concept instances, which must be representative of the cognitive universe of the ingroup. The calculation of the expressional component requires a corpus of documents representative of the cognitive universe of the ingroup. And the calculation of the intensional component requires that properties exist in the ontology.

Moreover, the intensional component can be refined by weighting links between properties and concepts: these weights express, for each concept and each property, the significance of the property in the definition of the concept. At present, these weights have to be specified manually. For each property, all the concepts which own the property have to be ordered on the interval $[0,1]$. For example, for the property *has_author*, the concept *Scientific article* will be placed first (close to 1), the concept *Press article* will be placed after (because to have an author is less important for this concept) and the concept *Instructions leaflet* will be placed close to 0.

Of course, all these resources (instances, corpus, property weights) are not always available, but the method we propose to calculate prototypicalities remains valid whatever resources we have. For example, if there is no instance, only the intensional and expressional components are calculated. If the ontology is the only available resource, with no instance, no weight on properties and no corpus, only the intensional component is calculated, with all the weights equal to 1.

Then, the model we advocate allows to weight each component of the SPG, in order to take the domination of the intensional, extensional and/or expressional aspects in the conceptualisation of the domain into account. These dominations are conditioned by the domain, the ingroup and the applicative context. For example, in the field of mathematics, the intensional aspect prevails. In the domain of biological species, a biologist builds its conceptualisation on biological properties (the intensional aspect prevails), but most of people usually use extensional conceptualisations, based on the animals encountered during their lives.

Formally speaking, the SPG is a function $spg : \mathcal{C} \times \mathcal{C} \rightarrow [0, 1]$ that, to each pair of concepts $(c, c_f) \in \mathcal{C} \times \mathcal{C}$ such that $c \leq^C c_f$, associates the value:

$$spg(c, c_f) = \alpha.int(c, c_f) + \beta.ext(c, c_f) + \gamma.exp(c, c_f)$$

The *int*, *ext* and *exp* functions are respectively presented in the following sections. α , β and γ are positive coefficients which weight each component of the SPG. To normalize the values, we impose the SPG and each of its component to be between 0 (minimum representativeness) and 1 (maximum representativeness), and $\alpha + \beta + \gamma = 1$. The values of these 3 coefficients can be set arbitrarily or experimentally; we also provide a method to set them automatically.

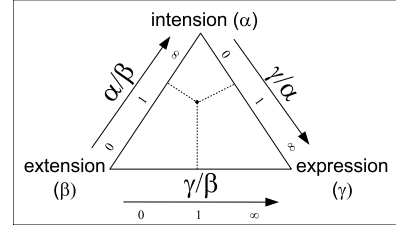


Figure 1. The semiotic triangle with SPG coefficients. γ/α close to 0 means that the conceptualization of the domain is more intensional than expressional; this ratio close to ∞ means the opposite and this ratio equal to 1 means that the conceptualization is as intensional as expressional. A similar approach is adopted for the other ratios.

Because the ratios between α , β and γ express the cognitive coordinates of the ingroup in the semiotic triangle (cf. figure 1), it is impossible to set independently the 3 values. We chose to calculate the ratios γ/α and γ/β , then α , β and γ are inferred from these ratios and the above constraints.

γ/α represents the ratio between what is conceptualized by the ingroup and what is expressed in the corpus, *i.e.* the ratio between what is purely intensional (concepts of the ontology which are not expressed in the corpus) and what is purely expressional (terms which appear in the corpus but which name concepts missing from the ontology). But we suppose that the ontology covers all the corpus, *i.e.* each term of the corpus denotes at least a concept, a property or an instance of the ontology. Then, γ/α is estimated by the cover rate of concepts in the corpus, that is the number of concepts which at least one term appears in the corpus, divided by the total number of concepts. Similarly, γ/β is estimated by the cover rate of instances in the corpus, that is the number of instances which at least one term appears in the corpus divided by the total number of instances.

When no corpus is available, only α/β is calculated. A concept that owns very significant properties, but with few instances (for example *Dragon*) is conceptualized more in an intensional way. Conversely, a concept with ordinary properties but which owns a lot of instances (for example *Car*) is conceptualized in an extensional way. So, α/β is estimated by the average ratio, for all concepts, between the significance of properties owned by a concept and the number of instances of this concept.

2.1.1 Intensional component

The intensional component of the SPG assesses how much a concept c is representative of its father c_f , by comparing properties of the two concepts. Like in [1], concepts are represented by vectors in the vectorial space of properties. But [1] uses fuzzy truth values

as vector coordinates, whereas our coordinates are values which assesses the significance of properties in the definitions of concepts. Formally speaking, to each concept $c \in \mathcal{C}$ is associated the vector $\vec{v}_c = (v_{c1}, v_{c2}, \dots, v_{cn})$ with $n = |\mathcal{P}|$ and $v_{ci} \in [0, 1], \forall i \in [1, n]$. v_{ci} is the weight fixed for the concept c with respect to the property i ($v_{ci} = 1$ if the weight is not fixed).

The prototype of the upper concept c_f is calculated as a vector $\vec{p}_{c_f}$, and the intensional prototypicality between c and c_f is the euclidian distance between $\vec{p}_{c_f}$ and $\vec{v}_c$, the vector that represents c .

The prototype of a concept c_f has been introduced in [1] as the average of the vectors which represents the children of c_f . But [1] only considers the concepts which directly inherit from c_f , whereas we extend the calculation to all the descendants of c_f , because properties which appear only in an indirect descendant of c_f can nevertheless appear in the prototype of c_f . For example, to wear a white coat is not a property of the concept *Researcher*, but the prototype of *Researcher* can have this property. The prototype vector $\vec{p}_{c_f}$ is a vector in the vectorial space of properties, with the coordinate for the property i is the average of all the coordinates of the c_f descendants for this property. With $i \in \mathcal{P}$ and $S_i(c) = \{c_j \leq^C c, c_j \in \text{dom}(i)\}$:

$$\vec{p}_{c_f}[i] = \frac{\sum_{c_j \in S_i(c_f)} \vec{v}_{c_j}[i]}{|S_i(c_f)|}$$

The intensional component is $\text{int}(c, c_f) = 1 - d(\vec{v}_c, \vec{p}_{c_f})$ with d the euclidian distance in the space of properties.

2.1.2 Extensional component

The extensional component of the SPG assesses how much a concept c is representative of its father c_f by estimating the space occupied by the instances of c in the extension of c_f . More this space is wide, more c is prototypical of c_f . For example, someone who owns a dozen of cats considers a cat more prototypical with respect to the concepts *Animal* than someone who owns a goldfish. Calculating this component supposes that instances are fixed in the ontology. We use a logarithmic function in order to obtain a behavior of the extensional component close to human evaluation (prototypicalities of concepts with very few instances are not too close to 0). The extensional component is:

$$\text{ext}(c, c_f) = 1 / \left(1 - \log \left(\frac{|\sigma(c)|}{|\sigma(c_f)|} \right) \right)$$

2.1.3 Expressional component

The expressional component of the SPG assesses how much a concept c is representative of its father c_f by comparing the way they are expressed. More c is expressed, more it is representative of c_f . A first estimation of the expressional prototypicality of c is based on the number of terms: more c has terms, more the place that c occupies in the cognitive universe of the ingroup is wide. For example, the concept *Horse*, that has a lot of synonyms (e.g. *cob*, *nag*, *gee-gee* or *jade*) is more prototypical of the concept *Animal* than the concept *Raccoon* that has only one term. This first estimation is given by the ratio between the number of terms that name c and the maximum number of terms that name each direct child of c_f .

$$\text{exp}_{syn}(c, c_f) = \frac{|\text{term}_c(c)|}{\max_{c_i \leq^C c_f, \#c_j, c_i \leq^C c_j \leq^C c_f} (|\text{term}_c(c_i)|)}$$

This estimation is based on terms which appear in the ontology. If a corpus considered as representative of the domain is available, it can be used to refine the expressional component: the more the terms of c or its descendants are present in the corpus, the more c is expressed in the cognitive universe of the ingroup, and the more c is prototypical of c_f . The occurrences of terms are weighted according to structures of the documents. For example, an occurrence that appear in a title or in a keyword list counts more than an occurrence that appears in a paragraph. We also want to take into account the distribution of occurrences between documents, because a term that appears often but in few documents should have smaller weight than one which appears not very often but in most documents.

We measure the pregnancy of a term t in the corpus with the function $\text{pregnancy}_t(t) : L_C \rightarrow [0, 1]$ defined as follows:

$$\text{pregnancy}_t(t) = \frac{\text{count}_{occ}(t)}{N_{occ}} * \frac{\text{count}_{doc}(t)}{N_{doc}}$$

with $\text{count}_{occ}(t)$ is the weighted occurrence number of t in the documents, $\text{count}_{doc}(t)$ the number of documents where t appears, N_{occ} the sum of all weighted occurrences of all terms in the corpus and N_{doc} the number of documents in the corpus.

The function $\text{pregnancy}_c(c)$ represents the pregnancy of concepts and is defined as follows ($S_{term}(c)$ is the set of terms that name c or one of its descendant):

$$\text{pregnancy}_c(c) = \sum_{t \in S_{term}(c)} \text{pregnancy}_t(t)$$

The expressional component is $\text{exp}(c, c_f) = \text{exp}_{syn}(c, c_f) \times \frac{\text{pregnancy}_c(c)}{\text{pregnancy}_c(c_f)}$ (or $\text{exp}(c, c_f) = \text{exp}_{syn}(c, c_f)$ if no corpus is available).

2.2 Lexical prototypicality gradient

The Lexical Prototypicality Gradient (LPG) estimates, for a given concept C and a term T used to name C , the representativeness of T to name C . The calculation of the LPG requires a corpus. The principle is that the more the ratio between occurrence number of the term and occurrence number of one of the terms used to name the concept is close to 1, the more the term is prototypical of this concept. Occurrences of terms are weighted according to the structure of corpus and their distributions in the documents. The function $\text{lpg}(t, c) : L_C \times \mathcal{C} \rightarrow [0, 1]$ is defined as follows, for each couple (t, c) with c the concept named by t :

$$\text{lpg}(t, c) = 1 / \left(1 - \log \left(\frac{\text{pregnancy}_t(t)}{\sum_{m \in \text{term}_c(c)} \text{pregnancy}_t(m)} \right) \right)$$

2.3 Instance prototypicality gradient

As for terms, instances of concepts can be more or less representative of the concept to which they belong. For example, all cats are instance of the concept *Cat*, but someone who owns a cat would consider his cat as more representative than the others. The estimation of this Instance Prototypicality Gradient (IPG) can only be done by using the corpus, as for LPG. The function pregnancy_i , similar to pregnancy_t , estimate the pregnancy of an instance i by the ratio between the occurrence number of terms that name i and the occurrence number of all terms in the corpus, considering structures of the documents. The function $\text{ipg}(i, c) : \mathcal{I} \times \mathcal{C} \rightarrow [0, 1]$ is defined as follows, for each couple (i, c) with i an instance of the concept c :

$$\text{ipg}(i, c) = 1 / \left(1 - \log \left(\frac{\text{pregnancy}_i(i)}{\sum_{j \in \sigma(c)} \text{pregnancy}_i(j)} \right) \right)$$

2.4 Emotional factor

Works in cognitive psychology show that the emotional state of a person influences his perception of categories: more the person is stressed, more his mind is concentrated on objects cognitively close to those which concern him [5, 8]. Then, we introduce an emotional factor, represented by a coefficient δ , that can vary from 0 to 1, for an open state of mind, and from 1 to ∞ , for a closed state of mind. SPG, LPG and IPG are elevated to the power $1/\delta$, in order to modulate the gradients according to a state of mind: for an open state of mind, weak values of prototypicality are increased and for a closed state of mind, they are decreased.

3 Applications of prototypicality

We introduce the notion of prototypicality as an additional cognitive element above existing ontology models. Thus, as the basics elements of a domain ontology (e.g. concepts, properties or axioms), prototypicalities can be applied (at the operational level) to various tasks.

3.1 Application to ontology engineering

Prototypicality gradients can be used as a tool to facilitate *ontology building*. Indeed, the expressional component of the SPG can be calculated from a corpus for all couples of concepts of an ontology, and subsumption links automatically added between each couple of concepts c_1 and c_2 such that $exp(c_1, c_2)$ is higher than a given threshold. This can help to automatically extract concept hierarchies from text, as long as the concepts have been extracted using text-mining techniques. If properties of concepts are also available, the intensional component can also be used to complete the estimation of the SPG, in order to refine the automatic building of the hierarchy.

Prototypicality can also be used for *ontology validation* and *ontology evolution*. Given an ontology and a corpus representative of the domain, SPG, LPG and IPG can be calculated. If some values of SPG are very low, the corresponding *isa* links should be considered as inadequate. Perhaps the child concept could inherit from another concept, or an additional concept could be inserted between the two initial ones, in order to smooth the distribution of SPG within the ontology. Similarly, if the value of a LPG (resp. IPG) is very low, the term (resp. instance) would be allocated to another concept.

3.2 Application to information retrieval

The first application of SPG to ontology-based information retrieval is the improvement of request expansion. Given a concept c in a request, the request can be expanded to all subconcepts of c such that $spg(c, c_f)$ is higher than a given value. Request expansion can then be modulated according to prototypicalities. For example, a request about *Bird* can be expanded to *Sparrow*, *Partridge* and *Duck*, but not to *Ostrich*. An horizontal expansion can also be considered: given a concept c in a request, the request can be expanded to each concept c' that inherits from the same concept c_f than c and such that $spg(c, c_f)$ and $spg(c', c_f)$ are close enough. For example, a request about *Cat* can be expanded to *Dog*, which is as representative of *Domestic animal* as *Cat*, but not to *Crocodile*. Moreover, prototypicality can be used to sort results of a expanded request c . Of course, results can be graded from more to less prototypical one, according to how the concept used to annotate the result is prototypical of c .

Information retrieval is a good application to exploit the emotional factor. The coefficient δ is close to 0 if the user of an ontology-based

information retrieval system wants a broad search, and then the request expansion will be large. If the user wants to focus on the concept he puts himself in the request, δ is a very high value, and few concepts are added to the request. In an other way, if we suppose that the user stress level is automatically obtained from a device, δ can be fixed as inversely proportional to this level. So if the user is stressed, the system will open the search space, in order to reduce its stress, and inversely, if the user is too relaxed, the system will restrict the results set in order to favour the user concentration.

3.3 Application to KBS personalization

The personalization of an Information System (IS) mainly consists in adapting its behavior to the user profile and the user activity, in order to allow the user accessing to information that are only relevant for its activity. This personalization process is more and more crucial, because of the ceaseless growth in data volume managed by IS. Moreover, IS are more and more open to various users, especially IS that operates on the Web [2]. Personalization is often based on user preferences in the system interface which is enable to modify and expand requests, to filter results and to adapt the presentation of the results. But, in the case of IS that include knowledge representation, we propose to use the ontology as the support of the personalization process: the ontology captures the consensual knowledge that is shared by the end user, and the personalization of the system can be done by adding to the ontology supplementary knowledge that depends on the cognitive universe of the end user.

Ontologies, as sets of concepts, properties and axioms that express the semantics of a domain, are considered as consensual and shared conceptualisations of a domain [3]. We use the term *ingroup* to denote the set of people who share such a conceptualisation. But, within a single ingroup, cognitive elements that are not modeled in the ontology can not be the same from an ingroup member to another one. If concepts are the same in member minds, the instances can be different. For example, in the case of an ontology about trading activity, two marketing men can agree with the ontology, but the instances they know will be different: they do not meet the same customers, they do not sell the same goods and so on.

In a similar idea, documents representative of what ingroup members know about the domain are inevitably not the same for all the members. For example, teachers of mathematics agree with what is Pythagoras' theorem, but each of them can advise to read a particular book to study this theorem. In section 2.1, we have introduced properties weights in order to express the importance of each property in the conceptualization. If ingroup members agree with concepts, properties and links between them that appear in the ontology, they can attribute different weights to properties. For instance, everybody agree that all birds have two feet. But a biologist who studies bird locomotion will judge this property very important, while most of us will consider it of secondary importance.

Thus, resources that we use to calculate prototypicality gradients can vary within the ingroup from which the members agree with the ontology. Of course, to reach a consensus about prototypicality, the ingroup can be reduced to people who agree about concept instances, documents representative of the domain and property weights. This can be done if we want to use the ontology enriched with prototypicality as consensual knowledge resource in a KBS. But the prototypicality can be used to adapt the basic ontology, with gradients that vary from a user to another. Prototypicality is then used to personalize the ontology, and the KBS, according to personal cognitive elements such that instances known by the user, documents the user

usually reads or writes, and importance he attributes to properties in the definition of concepts.

For a given Domain Ontology (DO), gradients of conceptual, lexical and/or instance prototypicality are calculated from instances, corpus and property weights peculiar to a user (or to a group of users). DO enriched with prototypicality is called Personalized Domain Ontology (PDO) and several PDO can be derived from one DO, as shown by figure 2.

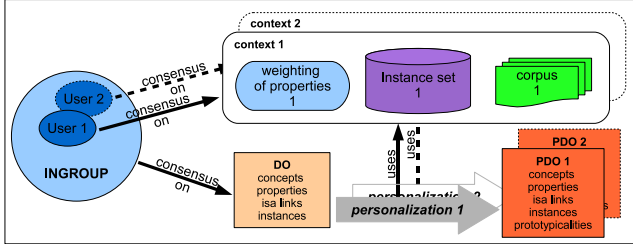


Figure 2. Use of prototypicality for ontology personalization.

Supplementary resources used to calculate prototypicality gradients can be get in different ways. Instances can be specified manually by the user, or discovered in texts he supplies. The corpus can be explicitly supplied by the user, or automatically built from external resources. For example, the corpus can be built from documents produced by the user in Web 2.0 applications such that blogs or wikis. But property weights have to be set manually by the user.

4 Experimental results

Our method is implemented in TOOPRAG (a Tool for Pragmatics in ontology), dedicated to the automatic calculation of our gradients. This tool, written in Java, uses Lucene⁸ and Jena⁹. TooPrag takes as inputs (1) an ontology represented in OWL 1.0, where each concept and relation is associated with a set of terms defined via the primitive *rdfs:label* and (2) a corpus composed of text files. The corpus is first indexed by Lucene. Then the ontology is loaded by Jena and the SPG, LPG and IPG gradients are calculated.

These results are stored in an OWL file whose format extends the current specification of OWL 1.0. As shown in figure 3, a SPG value is represented by a new attribute *xml:spg* associated to the *rdfs:subClassOf* link. For example, the SPG values of the *is-a* links between the concept *working_population_engaged_in_agriculture* and its sub-concepts *agricultural_labour_force* and *farmer* are respectively 0.0074 and 0.9841. And a LPG value is represented by a new attribute *xml:lpg* associated to the *rdfs:label* link. For instance, the LPG values of the terms *grower* and *peasant*, used to denote the concept *agricultural_labour_force*, are respectively 0.375 and 0.

4.1 Experiment in environment legal intelligence

The first evaluation of our work is done in the context in the project REDENE¹⁰ devoted to semantic and personalized information re-

⁸ Lucene is a high-performance, full-featured text search engine distributed as an open-source Java library <http://lucene.apache.org/>.

⁹ Jena is an open-source Java framework for building Semantic Web applications <http://jena.sourceforge.net/>.

¹⁰ REDENE is funded by Tennaxia, a company which offers service and software for legal intelligence in the domain of environment, www.tennaxia.com.

```
<owl:Class rdf:ID="agricultural_labour_force">
  <rdfs:label xml:lang="EN" xml:lpg=0.7>farm worker</rdfs:label>
  <rdfs:label xml:lang="EN" xml:lpg=0.3>agricultural labour
    force</rdfs:label>
  <rdfs:subClassOf rdf:resource="#working_population_engaged_in
    Agriculture" xml:spg=0.0074/>
</owl:Class>

<owl:Class rdf:ID="Farmer">
  <rdfs:label xml:lang="EN" xml:lpg=0.375>grower</rdfs:label>
  <rdfs:label xml:lang="EN" xml:lpg=0.0>peasant</rdfs:label>
  <rdfs:label xml:lang="EN" xml:lpg=0.0>raiser</rdfs:label>
  <rdfs:label xml:lang="EN" xml:lpg=0.625>farmer</rdfs:label>
  <rdfs:subClassOf rdf:resource="#working_population_engaged_in
    agriculture" xml:spg=0.9841/>
</owl:Class>
```

Figure 3. Extract of an OWL file produced by TOOPRAG.

trieval. This project aims at improving the efficiency of an information system dedicated to management of legal texts in the domain of environment. A first ontology of the Health, Security, Safety and Environment domain (HSE) has been built from both expert interviews and legal texts produced by the european and french parliaments. This ontology includes about 10.000 concepts structured in a lattice of depth 12 and maximum width 1500; it also includes 20 properties structured in a lattice of depth 3.

To calculate the SPG gradients (which is the more interesting gradient for this project), a corpus of 1100 legal texts extract from LegiFrance (www.legifrance.gouv.fr) and Eur-Lex (eur-lex.europa.eu) has been used. The average value of SPG is 0.128, 30.2% of these values are not null and 3.34% are equal to 1, with the following distribution:

[0,0.01]	[0.01,0.125]	[0.125,0.25]	[0.25,0.5]	[0.5,0.75]	[0.75,1]
63.23%	15.73%	5.11%	4.97%	3.22%	6.3%

The values of SPG have been validated by the experts, who are consultants in the company which funds the project. This experiment clearly indicates that our model and method is in accordance with the way the experts perceive prototypicality. Our gradients are currently used to expand requests and to grade results of requests in the software proposed by the company.

4.2 Experiment in environment policy

We set-up a second experiment in the context of environment policy, with a group of engineering students as subjects. First, these 25 engineers have built (by hand and from scratch) an ontology from various texts produced by different french political organisations about their choices concerning the domain of environment. About 15 texts per organisation were collected on web sites such that www.lemouvementpopulaire.fr, www.parti-socialiste.fr, www.fondation-nicolas-hulot.org or www.greenpeace.org/france. The ontology contains 130 concepts named by 350 terms, organized in a lattice of depth 3 and maximum width 9. All the students agree with the hierarchy, although they do not attach the same importance to each concept.

Then, each student has chosen 10 texts from any kind of sources (not necessarily the web sites used for building the ontology), which he considers to be representative of his own ideas about environment policy. He also manually grades, for each concept of the ontology, all its sub-concepts, according to the representativeness of sub-concepts. Finally, the SPG gradients are calculated for each student, by using their personal corpus of 10 texts. In 89% of the cases, the calculated

sub-concept	SPG	human judgment
Energy	0.93	1st
Water	0.41	2nd
Money	0.01	3rd
Material	0.03	4th

Table 1. SPG and order given by a student for the concept *Resource*

prototypical order and the manually set order clearly coincide (cf. figure 1 for an example).

This experiment confirms the relevance of our model with non-experts. However, our approach is more efficient when the corpus used to calculate the gradients is close to the conceptualisation of the user; for instance, using texts directly written by the end user such that blogs or emails is a relevant way. Moreover, considering instances and properties when calculating the extensional and intensional components of the gradients also improve the results.

This experiment also underlines that the use of prototypicality to personalize a domain ontology seems justified, because even among a little group of person (25 students were involved in the experiment), the SPG gradients vary from a student to another and can be used to clearly personalize the ontology and adapt information retrieval. However, we should experiment prototypicality-based personalization in the context of a more consensual knowledge domain. For this purpose, we plan to lead the same experiment with the experts of legal intelligence.

5 Conclusion

The main objective of our work is to introduce prototypicality in the current ontology models, in order to refine knowledge description. Prototypicality is used to weigh *isa* links between concepts but also links between concepts and terms that name them, and between concepts and their instances. We propose a method to calculate these 3 kinds of prototypicality. This method is based on the use of three kinds of resource: the ontology for which the prototypicality values are calculated, a corpus and a set of instances. The method is still applicable if only one of these resources is available.

Prototypicality has already been studied in Ontology Engineering. In particular, [1] proposes to calculate prototypicality values only from properties of concepts. But the weights they use are fuzzy truth values. This approach is not coherent with the formal semantics underlying an ontology, where all the properties of a concept are fully shared by all the instances of the concept. The weights we propose to use express the importance of the properties in the concept definition. This approach respects the formal semantics and is more coherent with the prototypicality notion. Moreover, to calculate the intensional component of our gradients, we extend the method proposed by [1] by considering the properties of all the concepts belonging to the descendance of the considered concept, and not only those of the direct sub-concepts. Our approach is closer to cognitive process.

The calculation of the expressional component of our gradients is quite similar to the Resnik's semantic measure which uses information content [9]. But Resnik considers the corpus as a whole, whereas we propose to exploit the granularity of the corpus by taking the number of documents where a given term appears into account. We also consider that a concept is more present in the corpus if the term that names the concept is few present but in a lot of documents, than if a lot of occurrence of the term appear in few texts. Comparing to Resnik, we also consider the structure of documents to weight the

occurrences, according to they appears in a title, an abstract or other kind of text element.

The main applications we have studied concern information retrieval. Prototypicality can be used to extend requests, to grade and filter results. Calculating prototypicality gradients from peculiar resources of each user allows us to personalize the ontology and the system that exploits it. The experiments presented in this paper clearly demonstrate that our gradients are close to expert and non-expert human judgements.

But prototypicality can also be applied to ontology engineering for ontology validation or evolution. A very low prototypicality value between two concepts could indicate that the *isa* link between them is not correct, or that a third concept should be inserted. Calculating the gradients between two concepts whose one inherits from the other without being one of its direct sub-concepts is also possible. If the obtained value is high, it means that the child concept should be placed higher in the hierarchy.

Prototypicality can also be calculated between two concepts that are not linked by an *isa* link. The intensional component is the same, but extensional and expressional components of the gradient must be modified. Using this extension of our method for ontology validation could reveal missing *isa* links in ontology. Generally speaking, the prototypicality gradients represent a semantic measure of concepts parentage within an ontology. We work on the definition of such a measure which combines the semiotic approach and our model of prototypicality.

REFERENCES

- [1] C. M. Au Yeung and H. F. Leung, 'Ontology with likeliness and typicality of objects in concepts', in *Proceedings of the 25th International Conference on Conceptual Modeling (ER'2006)*, volume 4215, pp. 98–111. Springer - LNCS, (2006).
- [2] P. Brusilovsky and A. Kobsa, *The Adaptive Web: Methods and Strategies of Web Personalization*, Springer, 2007.
- [3] Thomas Gruber, 'A translation approach to portable ontology specifications', *Knowledge Acquisition*, 5 (2), 199–220, (1993).
- [4] Stevan Harnad, 'Categorical perception', *Encyclopedia of Cognitive Science*, LXVII(4), (2003).
- [5] M. Mikulincer, P. Kedem, and D. Paz, 'The impact of trait anxiety and situational stress on the categorization of natural objects', *Anxiety Research*, 2, 85–101, (1990).
- [6] C.W. Morris, *Foundations of the Theory of Signs*, Chicago University Press, 1938.
- [7] C. K. Ogden and L.A. Richards, *The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism*, Harcourt, 1989.
- [8] J. Park and M. Nanaji, 'Mood and heuristics: The influence of happy and sad states on sensitivity and bias in stereotyping', *Journal of Personality and Social Psychology*, (78), 1005–1023, (2000).
- [9] Philip Resnik, 'Using information content to evaluate semantic similarity in a taxonomy', in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, volume 1, pp. 448–453, (1995).
- [10] E. Rosch, 'Cognitive reference points', *Cognitive Psychology*, (7), 532–547, (1975).