A human-inspired model to represent uncertain knowledge in the Semantic Web

Salvatore F. Pileggi

School of Systems, Management and Leadership
University of Technology Sydney
Australia
SalvatoreFlavio.Pileggi@uts.edu.au

Abstract. One of the most evident and well-known limitations of the Semantic Web technology is its lack of capability to deal with uncertain knowledge. As uncertainty is often part of the knowledge itself or can be inducted by external factors, such a limitation may be a serious barrier for some practical applications. A number of approaches have been proposed to extend the capabilities in terms of uncertainty representation; some of them are just theoretical or not compatible with the current semantic technology; others focus exclusively on data spaces in which uncertainty is or can be quantified. Human-inspired models have been adopted in the context of different disciplines and domains (e.g. robotics and human-machine interaction) and could be a novel, still largely unexplored, pathway to represent uncertain knowledge in the Semantic Web. Human-inspired models are expected to address uncertainties in a way similar to the human one. Within this paper, we (i) briefly point out the limitations of the Semantic Web technology in terms of uncertainty representation, (ii) discuss the potentialities of human-inspired solutions to represent uncertain knowledge in the Semantic Web, (iii) present a human-inspired model and (iv) a reference architecture for implementations in the context of the legacy technology.

1 Introduction

Many systems are experiencing a constant evolution, working on data spaces of an increasing scale and complexity. It leads to a continuous demand for advanced interoperability models, which has pushed the progressive development of the Semantic Web technology [1]. Such a technology, as the name itself suggests, deals with the specification of formal semantics aimed at giving meanings to disparate raw data, information and knowledge. It enables in fact interoperable data spaces suitable to advanced automatic reasoning.

The higher mature level of the Semantic Web infrastructure includes a number of languages (e.g. RDF[2], OWL[3], SWRL[4]) to define ontologies via rich data models capable to specify concepts, relations, as well as the support for automatic reasoning. Furthermore, a number of valuable supporting assets are available from the community, including reasoners (e.g. Jena [5], Pellet [6], HermiT [7]) and query capabilities (e.g. ARQ [8]). Those components enable a pervasive interaction with the semantic structures. Last but not the least, ontology developers are supported by user-friendly editors (e.g. Protege [9]).

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93713-7_21

While the major improvements that have defined the second version of the Web [10] focus on aspects clearly visible to the final users (e.g. the socialization of the Web and enhanced multimedia capabilities), the semantic technology is addressing mainly the data infrastructure of the Web, providing benefits not always directly appreciable from a final user perspective. By enriching the metadata infrastructure, the Semantic Web aims at providing a kind of universal language to define and share knowledge across the Web.

The popularity of the Semantic Web in research is tangible just looking at the massive number of works on the topic currently in literature, as well as at the constant presence of semantic layers or assets in research projects dealing with data, information and knowledge in the context of different domains and disciplines. Although the actual impact in the real world is still to be evaluated, there are significant evidences of application, as recent studies (e.g. [11]) have detected structured or linked data (semantics) over a significant number of websites.

Despite the Semantic Web is unanimously considered a sophisticated environment, it doesn't support the representation of uncertain knowledge, at least considering the ”official” technology. In practice, many systems deal with some uncertainty. Indeed, there is a potentially infinite heterogeneous range of intelligible/unintelligible situations which involve imperfect and/or unknown knowledge. Focusing on computer systems, uncertainty can be part of the knowledge itself, as well as it may be inducted as the consequence of adopting a given model (e.g. simplified) or of applying a certain process, mechanism or solution.

Structure of the paper. The introductory part of the paper continues with two subsections that deal, respectively, with a brief discussion of related work and human-inspired models to represent uncertainty within ontology-based systems. The core part of the paper is composed of 3 sections: first the conceptual framework is described from a theoretical perspective (section 2); then, section 3 focuses on the metrics to measure the uncertainty in the context of the proposed framework; finally, section 4 deals with a reference architecture for implementations compatible with the legacy Semantic Web technology. As usual, the paper ends with a conclusions section.

1.1 Related Work

Several theoretical works aiming at the representation of the uncertainty in the Semantic Web have been proposed in the past years. A comprehensive discussion of such models and their limitations is out of the scope of this paper.

Most works in literature are usual to model the uncertainty according to a numerical or quantitative approach. They rely on different theories including, among others, fuzzy logic (e.g. [12]), rough sets (e.g. [13]) and Bayesian models (e.g. [14]). A pragmatic class of implementations is usual to extend the most common languages with the probability theory [15]: Probabilistic RDF [16] extends RDF [2]; Probabilistic Ontology (e.g. [17]) defines a set of possible extensions for OWL [18]; Probabilistic SWRL [19] provides extensions for SWRL [20].

Generally speaking, a numerical approach to uncertainty representation is effective, completely generic, relatively simple and intrinsically suitable to computational envi-

DOI: 10.1007/978-3-319-93713-7_21

ronments. However, such an approach assumes that all uncertainties are or can be quantified. Ontologies are rich data models that implement the conceptualisation of some complex reality. Within knowledge-based systems adopting ontologies, uncertainties may depend on underpinning systems and are not always quantified. The representation of uncertainties that cannot be explicitly quantified is one of the key challenges to address.

1.2 Why human-inspired models?

A more qualitative representation of the uncertainty in line with the intrinsic conceptual character of ontological models is the main object of this paper. As discussed in the previous sub-section, considering the uncertainty in a completely generic way is definitely a point. However, in terms of semantics it could represent an issue as, depending on the considered context, different types of uncertainty can be identified. Assuming different categories of uncertainty results in higher flexibility, meaning that different ways to represent uncertainties can be provided for the different categories of uncertainty. At the same time, such an approach implies specific solutions aimed at the representation of uncertainties with certain characteristics, rather than a generic model for uncertainty representation.

As discussed later on in the paper, the proposed framework defines different categories of uncertainty and a number of concepts belonging to each category. The framework is human-inspired as it aims at reproducing the different human understandings of uncertainty in a computational context. The underlying idea is the representation of uncertainties according to a model as similar as possible to the human one.

Human-inspired technology is not a novelty and it is largely adoped in many fields including, among others, robotics [21], human-machine interaction [22], control logic [23], granular computing [24], situation understanding [25], trust modeling [26], computational fairness [27] and evolving systems [28].

2 Towards human-inspired models: a conceptual framework

The conceptual framework object of this work (fig. 1) is a simplification of a possible human perception of uncertain knowledge. It distinguishes three main categories of uncertainty. The first category, referred to as *numerical or quantitative uncertainty*, includes the uncertainties that are or can be quantified; this is intrinsically suitable to a computational context and, indeed, has been object of an extensive research, as briefly discussed in the previous sections. The second category, called *qualitative uncertainty*, includes those uncertainties that cannot be quantified. While a quantitative understanding of uncertainty is typical of human reasoning, it is not obvious within computational environments. The third category considered in the framework, referred to as *indirect uncertainty*, is related to factors external to the knowledge represented by the considered ontology. For example, an ontology could be populated by using the information from different datasets; those datasets could have a different quality, as well as the different providers could be associated with a different degree of reliability. It leads indirectly to an uncertainty. The different categories of uncertainty will be discussed separately in the next sub-sections.

DOI: 10.1007/978-3-319-93713-7_21

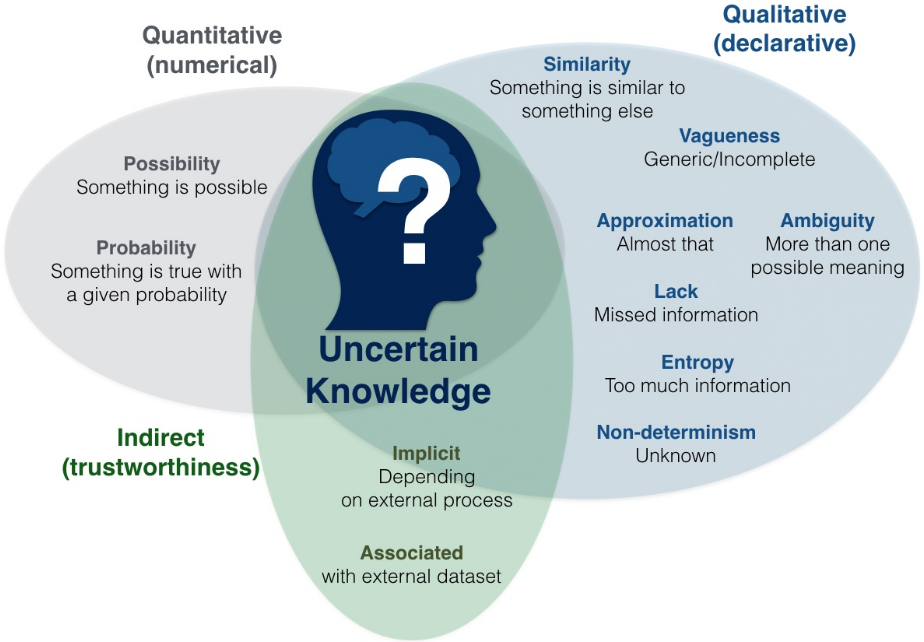


Fig. 1: Conceptual framework.

2.1 Numerical or quantitative uncertainty

The quantitative approach relies on the representation of the uncertainty as a numerical value. In a semantic context, it is normally associated with assertions or axioms. As pointed out in the previous sections, such an approach is generic and, in general, effective in a computational context.

On the other hand, it can be successfully adopted within environments in which all uncertainties are or can be quantified. A further (minor) limitation is represented by the interpretation of the ”number” associated with the uncertainty that, in certain specific cases, may result ambiguous.

Most numerical approaches are based on the classic concepts of *possibility* or *probability*, as in their common meaning. As quantitative models have been object of several studies and implementations, a comprehensive discussion of those solutions is considered out of the scope of this paper, which rather focuses on qualitative models and indirect uncertainty.

2.2 Qualitative uncertainty

A non-numerical approach is a major step towards human-inspired conceptualizations of uncertainty. Within the qualitative category, we define a number of concepts, including *similarity*, *approximation*, *ambiguity*, *vagueness*, *lack of information*, *entropy* and *non-determinism*. The formal specification of the conceptual framework assumes OWL technology. A simplified abstracted view of a conventional OWL ontology is defined by the following components:

- a set *C* of classes
- a set *I* of instances (referred to as individuals) of *C*
- a set of assertions or axioms *S* involving *C* and *I*

Representing uncertainty implies the extension of the model as follows:

- a set of assertions S^* to represent uncertainty on some knowledge
- a set of assertions S^u which represents lacks of knowledge

The union of S^* and S^u defines the uncertainty in a given system. However, the two sets are conceptually different; indeed, the former set includes those assertions that involve some kind of uncertainty on an existing knowledge; the latter models the awareness of some lack of knowledge.

Similarity. Similarity is a very popular concept within computer systems. It is extensively adopted in a wide range of application domains, in which the elements of a given system can be related to each other, according to some similarity metric. Similarity is a well-know concept also in the specific context of the Semantic Web technology, where *semantic similarity* is normally established on the base of the semantic structures (e.g. [29]).

Unlike most current understandings, where the similarity is associated somehow with knowledge, modeling the uncertainty as a similarity means focusing on the unknown aspect of such a concept [30]. Indeed, the similarity among two concept is not an equivalence.

Given two individuals *i* and *j*, and the set of assertion *S* involving *i*, a *full similarity* between *i* and *j* implies the duplication of all the statements involving *i*, replacing *i* with *j* (eq. 1a). The duplicated set of axioms is considered an uncertainty within the system.

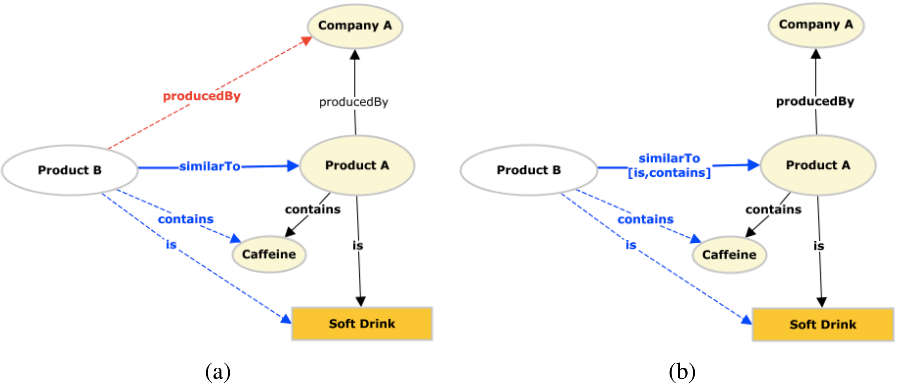
$$sim(i \in I, j \in I) \Rightarrow \forall s(i, \_) \in S \rightarrow \exists s^*(j, \_) \qquad s^* \in S^* \tag{1a}$$

$$sim(i \in I, j \in I, S_k \subset S) \Rightarrow \forall s(i, \_) \in S_k \rightarrow \exists s^*(j, \_) \qquad s^* \in S^* \tag{1b}$$

A full similarity as defined in eq. 1a is a strong relation, meaning it is conceptually close to a semantic equivalence. The key difference between a semantic equivalence and a full similarity relies in the different understanding of j which, in the latter case, is characterized but also explicitly stated as an uncertainty.

An example of full similarity is depicted in fig. 2a: a product *B* is stated similar to a product *A* in a given context; the two products are produced by different companies; because of the similarity relation, exactly as *A*, *B* is recognized to be a soft drink containing caffeine; on the other hand, according to the full similarity model, *B* is also considered as produced by the same company that produces *A*, which is wrong in this case.

Partial similarity is a more accurate relation which restricts the similarity and, therefore, the replication of the statements to a subset S_k of specific interest (eq. 1b). By adopting partial similarity, the previous example may be correctly represented and processed (fig. 2b): as the similarity is limited at the relations *is* and *contains*, the relation *producedBy* is not affected. *Web of Similarity* [30] provides an implementation of both full and partial similarity. It will be discussed later on in the paper.



(a)

(b)

Fig. 2: Example of full (a) and partial (b) similarity.

Approximation. Approximation is the natural extension of similarity. Unlike similarity, which applies to instances of classes, the concept of approximation is established among classes (eq. 2a): a class *i* which approximates a class *j* implies each statement involving the members e_i of the class *i*, replicated to the members e_j of the class *j*.

$$aprx(i \in C, j \in C) \quad \Rightarrow \quad \forall s(e_i \in i, \_) \rightarrow \exists s^*(e_j \in j, \_) \qquad s^* \in S^* \tag{2a}$$

$$aprx(i \in C, j \in C, S_k \subset S) \quad \Rightarrow \quad s(e_i \in i, \_) \in S_k \rightarrow \exists s^*(e_j \in j, \_) \qquad s^* \in S^* \tag{2b}$$

$$aprx(i \in C, j \in C) \quad \Rightarrow \quad \forall e_i \in i \rightarrow e_i \in j \qquad (e_i \in j) \in S^* \tag{2c}$$

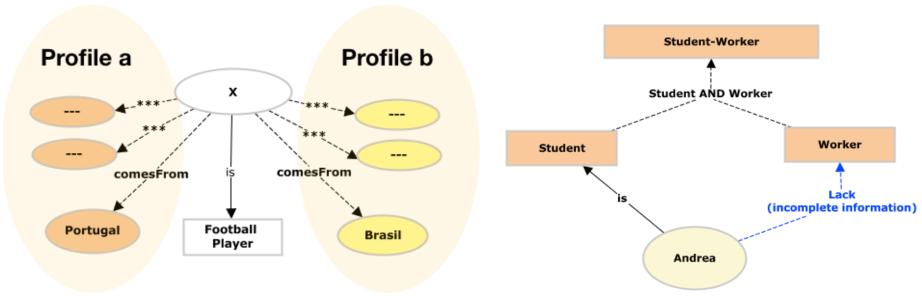
As for similarity, we distinguish between a *full approximation* (as in eq. 2a) and a more accurate variant of the concept (*partial approximation*), for whom the approximation is limited to a set S_k of assertions (eq. 2b). The semantic of approximation as defined in eq. 2a and 2b can be considered to be just theoretical, as it is not easy to apply in real contexts. It leads to a much simpler and more pragmatic specification, referred to as *light approximation*, which is defined by eq. 2c: if the class i approximates the class j, than a member e_i of the class i is also member of the class j.

Ambiguity. According to an intuitive common definition, an ambiguous concept may assume more than one meaning and, therefore, more than one semantic specification. The different possible semantic specifications are considered mutual exclusive. That leads to an uncertainty.

Such a definition of ambiguity can be formalised as an exclusive disjunction (XOR): considering an individual *k* and two possible semantic specifications of it, S_i and S_j, $ambiguity(k, S_i, S_j)$ denotes that S_i and S_j cannot be simultaneously valid (eq. 3).

$$\begin{aligned}
&ambiguity(k \in I, S_i \in S^*, S_j \in S^*) \Rightarrow \\
&\qquad [S_i(k, \_) = null \quad AND \quad S_j(k, \_) \neq null \in S] \qquad XOR \\
&\qquad\qquad [S_j(k, \_) = null \quad AND \quad S_i(k, \_) \neq null \in S] \qquad (3)
\end{aligned}$$

Ambiguities are very common. As well as disambiguation is not always possible, especially in presence of unsupervised automatic processes. For instance, ambiguity is usual to rise processing natural language statements. A very simple example of ambiguity is depicted in fig. 3a: the ambiguity is generated by the homonymy between two football players; each player has his own profile within the system; the link between the considered individual and the profile defines an uncertainty.



(a)

(b)

Fig. 3: Example of ambiguity (a) and lack of information (b).

The definition of ambiguity as in eq. 3 allows the correct co-existence of multiple and mutual-exclusive representations.

Lack of information. A lack of information on a given individual *i* models those situations in which only a part of the knowledge related to an individual is known. The uncertainty comes from the awareness of a lack of information (eq. 4). The formal specification assumes the existence of a number of unknown assertions (s^u) to integrate the information available S_i.

$$lack(i \in I, S, S^u) \qquad \Rightarrow \qquad \exists s(i, \_) \in S, \quad \exists s^u(i, \_) \in S^u \tag{4}$$

An example of lack of information is represented in fig. 3b: considering a database in which people are classified as students or workers, while the information about students is complete, the information about workers presents lacks; it means that a person appearing in the database is definitely recognized as a student if he/she is a student but a person is not always recognized as a worker even if he/she is a worker. That situation produces an evident uncertainty on the main classification; moreover, such an uncertainty is propagated throughout the whole model as it affects the resolution of inference rules (e.g. *student-worker* in the example).

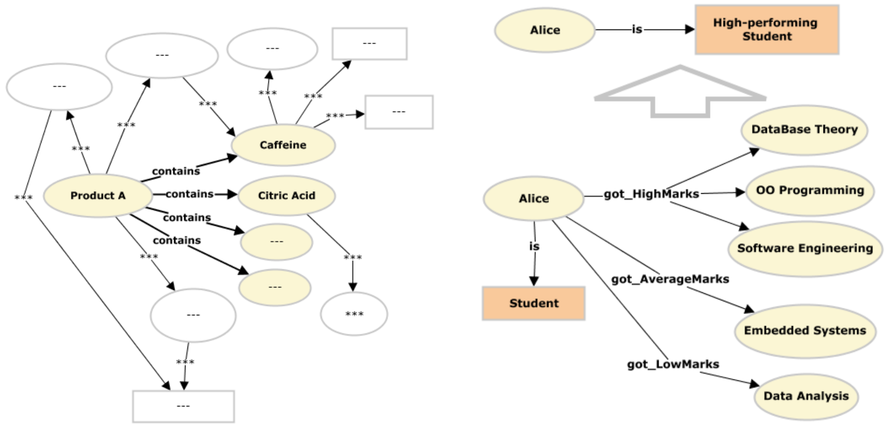
Entropy. Huge amount of complex data may lead to a situation of entropy, meaning a lot of information available but just a small part of it relevant in terms of knowledge in a given context for a certain purpose. Entropic environments are normally not very intelligible and, generally speaking, require important amount of resources to be computed correctly. Within the proposed model, the entropy is a kind of filter for raw data that consider only a subset of statement S_k (eq. 5).

$$entropy(S, S_k) \quad \Rightarrow \quad S^* = S - S_k \tag{5}$$

In practice, S_k may be defined by considering a restricted number of classes, individuals or relations. An example is shown in fig. 4a, where a subset of the data space is identified on the base of the relation *contains*. The definition provided is not necessarily addressing an uncertainty. However, in presence of rich data models (ontologies) that include inference rules, the result of a given reasoning might be not completely correct as inference rules apply just to a part of the information. Those situations lead to uncertainties.

Vagueness. Vagueness is a strongly contextual concept. Within the model, we consider the case in which there is a gap in terms of abstraction between the semantic structure adopted and the reality object of the representation. That kind of circumstance defines an uncertainty as a vagueness.

In order to successfully model vagueness, we first define a supporting operator, *abst*, that measures the abstraction, namely the level of detail, of a given ontological subset. We use this operator to rank sets of assertions as the function of their level of detail. Therefore, $abst(a) > abst(b)$ indicates that the set *a* and *b* are addressing a similar target knowledge, with *b* more detailed (expressive) than *a*. According to such a definition, *a* and *b* are not representing exactly the same knowledge, as a is missing details. Furthermore, we assume there is not an internal set of inference rules that relates *a* and *b*.



(a)

(b)

Fig. 4: Example of entropy (a) and vagueness (b).

The formal specification of vagueness (eq. 6) assumes the representation of a given individual i by using a more abstracted model S_i^a than the required one S_i to address all details.

$$vag(i \in I, S_i, S_i^a) \Rightarrow s(i,\_) \in S_i^a \rightarrow s(i,\_) \in S^*, \quad abst(S_i^a) > abst(S_i) \qquad (6)$$

An example is depicted in fig. 4b: a qualitative evaluation of students' performance assumes their marks belonging to 3 main categories (low, medium and high); a process external to the ontology generates an overall evaluation on the base of the breakdown: in the example Alice got one low score, one average score, as well as three high scores; according to the adopted process, Alice is just classified as an high-performing student without any representation of the data underpinning that statement, as well as of the process adopted; that same statement could reflect multiple configurations providing, therefore, a vague definition due to the lack of detail.

2.3 Indirect uncertainty

Moving from a quantitative model to a more flexible approach, which integrates qualitative features, is a clear step forward towards richer semantics capable to address uncertain knowledge in the Semantic Web. However, one of the main reasons for the huge gap existing between the human understanding of uncertainty and its representation in computer systems is often the ”indirect” and contextual character of the uncertainty. The indirect nature of the uncertainty is an intuitive concept: sometimes an uncertainty is not directly related to the knowledge object of representation, but it can be indirectly associated with some external factor, such as methods, processes and underpinning data.

It leads to a third class of uncertainty, referred to as *indirect uncertainty*, which deals with the uncertainty introduced by external factors and, therefore, with a more holistic understanding of uncertainty.

Associated uncertainty. As previously mentioned, the uncertainty can have a strongly contextual meaning and could be associated with some external factor or concept. Because of its associative character, this kind of uncertainty is referred to as *associated uncertainty*. A possible formalization of the model is proposed in eq. 7: an uncertainty related to a certain element u of the knowledge space is associated to an external concept c that is not part of the ontological representation.

$$uncertainty(u, s(u, \_) \in S) \quad \Rightarrow \quad u \leftarrow c \qquad\qquad \nexists \quad s(c, \_) \in S \tag{7}$$

As an example, we consider the population of an ontology with data from a number of different data sets. The considered data sets are provided by different providers, each one associated with a different degree of reliability. In such a scenario, the reliability of the whole knowledge environment depends on the reliability of the underpinning data. However, a potential uncertainty is introduced in the system because the information is represented indistinctly, although it comes from different data sets associated with a different degree of reliability.

Implicit uncertainty. *Implicit uncertainty* is a concept similar to the previous one. However, it is not related to underpinning data. It rather focus on the way in which such data is processed to generate and populate the ontological structure (eq. 8).

$$uncertainty(u, s(u, \_) \in S) \Rightarrow u = f(c) \qquad\qquad \nexists \quad s(c, \_) \in S \tag{8}$$

The function $f(c)$ in eq. 8 represents a generic method, process or formalization operating on external information c. Assuming n processes, implicit uncertainty reflects the uncertainty inducted by simplifications, approximations and assumptions in external methods underlying the knowledge base.

3 Measuring and understanding uncertainty

Uncertain knowledge assumes the coexistence of knowledge and uncertainty. Despite the qualitative approach allows a non-numerical representation of uncertainty, measuring the uncertainty or, better, the relation between knowledge and uncertainty is a key issue, at both a theoretical and an application level.

We consider two different correlated classes of metrics to quantify and understand the uncertainty in a given context:

- *Global Uncertainty Metric (GUM)* refers to the whole knowledge space and, therefore, is associated with all the knowledge $S \cup S^*$ available (eq. 9) .

$$GUM(S, S^*) = f(s \in \{S \cup S^*\}) \tag{9}$$

DOI: 10.1007/978-3-319-93713-7_21

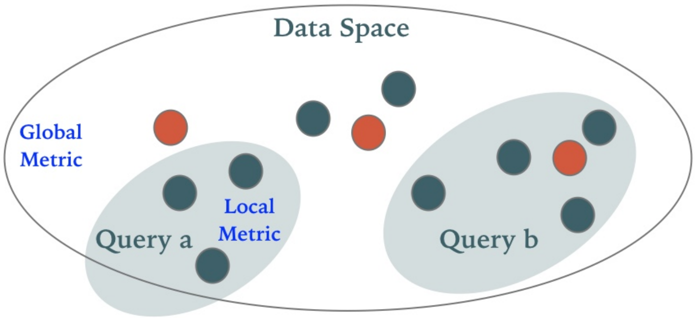


Fig. 5: Global metrics versus local metrics.

- *Local Uncertainty Metric (LUM)* is a query-specific metric related only to that part of information S_q which is object of the considered query q (eq. 10).

$$LUM(S, S^*, q \in Q) = f(s \in S_q \subseteq \{S \cup S^*\}) \tag{10}$$

While *GUM* provides a generic holistic understanding of the uncertainty existing in the knowledge space at a given time, *LUM* is a strongly query-specific concept that provides a local understanding of the uncertainty. An effective model of analysis should consider both those types of metrics.

4 Reference Architecture

The reference architecture (fig. 6) aims at implementations of the proposed framework compatible with the legacy semantic technology. As shown in the figure, we consider the core part of a semantic engine composed of four different functional layers:

- *Ontology layer*. It addresses, as in common meanings, the language to define ontologies; we are implicitly considering ontologies built upon OWL technology. Within the proposed architecture, uncertainties are described according to an ontological approach. Therefore, the key extension of this layer with respect to conventional architectures consists of an ontological support to represent uncertainty.
- *Reasoning layer*. It provides all the functionalities to interact with ontologies, both with reasoning capabilities. The implementation of the framework implies the specification of built-in properties to represent uncertainties, both with the reasoning support to process them. Most existing OWL reasoners and supporting APIs are developed in Java.
- *Query layer*. The conventional query layer, that is usual to provide an interface for an effective interaction with ontologies through a standard query language (e.g. SPARQL [31]), is enriched by capabilities for uncertainty filtering. Indeed, the ideal query interface is expected to retrieve results by considering all uncertainties, considering just a part of them filtered according to some criteria, or, even, considering no uncertainties.

DOI: 10.1007/978-3-319-93713-7_21

- *Application layer*. The main asset provided at an application level is the set of metrics to measure the uncertainty (as defined in section 3).

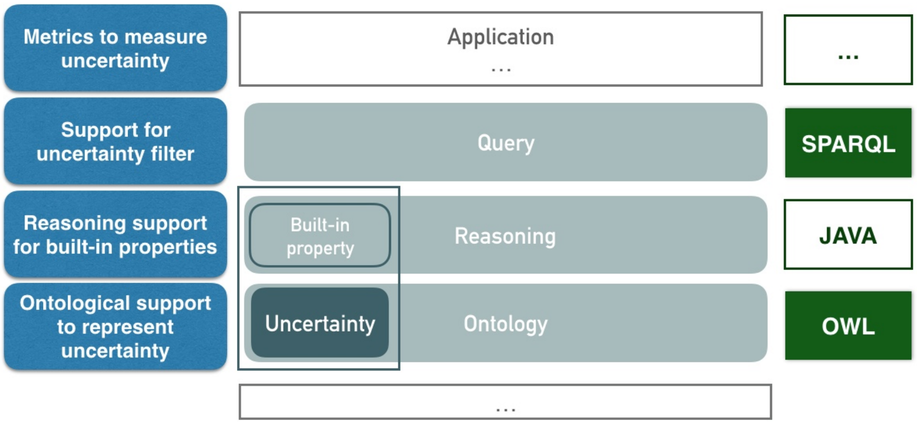

Metrics to measure uncertainty
Application
...
...
Support for uncertainty filter
Query
SPARQL
Reasoning support for built-in properties
Built-in property
Reasoning
JAVA
Ontological support to represent uncertainty
Uncertainty
Ontology
OWL
...

Fig. 6: Reference Architecture.

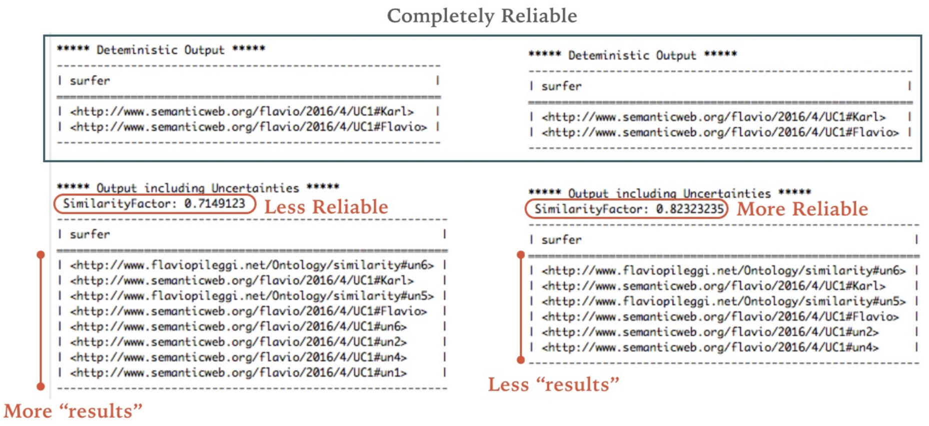
4.1 Current implementation

The implementation of the framework is currently limited to the concept of uncertainty as a similarity. It is implemented by the *Web of Similarity (WoS)* [30]. WoS matches the reference architecture previously discussed as follows:

- Uncertainty is specified according to an ontological model in OWL.
- The relation of similarity is expressed through the built-in property *similarTo*; WoS implements related reasoning capability.
- The query layer provides filtering capabilities for uncertainty. An example of output is shown in fig. 7: the output of a query includes two different result sets obtained by reasoning with and without uncertainties; uncertainty is quantified by a global metric.
- The current implementation only includes global metrics to measure uncertainty.

5 Conclusions

This paper proposes a conceptual framework aimed at the representation of uncertain knowledge in the Semantic Web and, more in general, within systems adopting Semantic Web technology and ontological approach. The framework is human-inspired and is expected to address uncertainty according to a model similar to the human one. On



```
***** Deteministic Output *****
----------------------------------------------------------
| surfer                                                 |
==========================================================
| <http://www.semanticweb.org/flavio/2016/4/UC1#Karl>    |
| <http://www.semanticweb.org/flavio/2016/4/UC1#Flavio>  |
----------------------------------------------------------

***** Output including Uncertainties *****
SimilarityFactor: 0.7149123
-----------------------------------------------------------
| surfer                                                  |
===========================================================
| <http://www.flaviopileggi.net/Ontology/similarity#un6>  |
| <http://www.semanticweb.org/flavio/2016/4/UC1#Karl>     |
| <http://www.flaviopileggi.net/Ontology/similarity#un5>  |
| <http://www.semanticweb.org/flavio/2016/4/UC1#Flavio>   |
| <http://www.semanticweb.org/flavio/2016/4/UC1#un6>      |
| <http://www.semanticweb.org/flavio/2016/4/UC1#un2>      |
| <http://www.semanticweb.org/flavio/2016/4/UC1#un4>      |
| <http://www.semanticweb.org/flavio/2016/4/UC1#un1>      |
-----------------------------------------------------------
```

```
***** Deteministic Output *****
-----------------------------------------------------------
| surfer                                                  |
===========================================================
| <http://www.semanticweb.org/flavio/2016/4/UC1#Karl>     |
| <http://www.semanticweb.org/flavio/2016/4/UC1#Flavio>   |
-----------------------------------------------------------

***** Output including Uncertainties *****
SimilarityFactor: 0.82323235
-----------------------------------------------------------
| surfer                                                  |
===========================================================
| <http://www.flaviopileggi.net/Ontology/similarity#un6>  |
| <http://www.semanticweb.org/flavio/2016/4/UC1#Karl>     |
| <http://www.flaviopileggi.net/Ontology/similarity#un5>  |
| <http://www.semanticweb.org/flavio/2016/4/UC1#Flavio>   |
| <http://www.semanticweb.org/flavio/2016/4/UC1#un2>      |
| <http://www.semanticweb.org/flavio/2016/4/UC1#un4>      |
-----------------------------------------------------------
```

Fig. 7: An example of query result from Web of Similarity.

one hand, such an approach allows uncertainty representation beyond the common numerical or quantitative approach; on the other hand, it assumes the classification of the different kinds of uncertainty and, therefore, it can miss genericness.

The framework establishes different categories of uncertainty and a set of concepts associated with those categories. Such concepts can be considered either as stand-alone concepts or as part of an unique integrated semantic framework. The reference architecture proposed in the paper aims at implementations of the framework fully compatible with the legacy Semantic Web technology. The current implementation of the framework is limited to the concept of similarity.

References

1. Berners-Lee, T., Hendler, J., Lassila, O., et al.: The semantic web. Scientific american **284**(5) (2001) 28–37
2. Hayes, P., McBride, B.: Rdf semantics. W3C recommendation **10** (2004)
3. McGuinness, D.L., Van Harmelen, F., et al.: Owl web ontology language overview. W3C recommendation **10**(10) (2004) 2004
4. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M., et al.: Swrl: A semantic web rule language combining owl and ruleml. W3C Member submission **21** (2004) 79
5. Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K.: Jena: implementing the semantic web recommendations. In: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, ACM (2004) 74–83
6. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical owl-dl reasoner. Web Semantics: science, services and agents on the World Wide Web **5**(2) (2007) 51–53
7. Shearer, R., Motik, B., Horrocks, I.: Hermit: A highly-efficient owl reasoner. In: OWLED. Volume 432. (2008) 91
8. : Arq - a sparql processor for jena. http://jena.apache.org/documentation/query/index.html Accessed: 2018-02-15.

DOI: 10.1007/978-3-319-93713-7_21

9. Gennari, J.H., Musen, M.A., Fergerson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F., Tu, S.W.: The evolution of protégé: an environment for knowledge-based systems development. International Journal of Human-computer studies **58**(1) (2003) 89–123
10. Murugesan, S.: Understanding web 2.0. IT professional **9**(4) (2007)
11. Guha, R.V., Brickley, D., Macbeth, S.: Schema. org: Evolution of structured data on the web. Communications of the ACM **59**(2) (2016) 44–51
12. Straccia, U.: A fuzzy description logic for the semantic web. Capturing Intelligence **1** (2006) 73–90
13. Kana, D., Armand, F., Akinkunmi, B.O.: Modeling uncertainty in ontologies using rough set. International Journal of Intelligent Systems & Applications **8**(4) (2016)
14. Da Costa, P.C.G., Laskey, K.B., Laskey, K.J.: Pr-owl: A bayesian ontology language for the semantic web. In: Uncertainty Reasoning for the Semantic Web I. Springer (2008) 88–107
15. Pileggi, S.F.: Probabilistic semantics. Procedia Computer Science **80** (2016) 1834–1845
16. Udrea, O., Subrahmanian, V., Majkic, Z.: Probabilistic rdf. In: 2006 IEEE International Conference on Information Reuse & Integration, IEEE (2006) 172–177
17. Ding, Z., Peng, Y.: A probabilistic extension to ontology language owl. In: System Sciences, 2004. Proceedings of the 37th Annual Hawaii international conference on, IEEE (2004) 10–pp
18. Bechhofer, S.: Owl: Web ontology language. In: Encyclopedia of Database Systems. Springer (2009) 2008–2009
19. Pan, J.Z., Stoilos, G., Stamou, G., Tzouvaras, V., Horrocks, I.: f-swrl: a fuzzy extension of swrl. In: Journal on Data Semantics VI. Springer (2006) 28–46
20. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M., et al.: Swrl: A semantic web rule language combining owl and ruleml. W3C Member submission **21** (2004) 79
21. Coradeschi, S., Ishiguro, H., Asada, M., Shapiro, S.C., Thielscher, M., Breazeal, C., Mataric, M.J., Ishida, H.: Human-inspired robots. IEEE Intelligent Systems **21**(4) (2006) 74–85
22. Moore, R.: Presence: A human-inspired architecture for speech-based human-machine interaction. IEEE Transactions on Computers **56**(9) (2007) 1176–1188
23. Gavrilets, V., Mettler, B., Feron, E.: Human-inspired control logic for automated maneuvering of miniature helicopter. Journal of Guidance, Control, and Dynamics **27**(5) (2004) 752–759
24. Yao, Y.: Human-inspired granular computing. Novel developments in granular computing: applications for advanced human reasoning and soft computation (2010) 1–15
25. Liang, Q.: Situation understanding based on heterogeneous sensor networks and human-inspired favor weak fuzzy logic system. IEEE Systems Journal **5**(2) (2011) 156–163
26. Velloso, P.B., Laufer, R.P., Duarte, O.C., Pujolle, G.: Hit: A human-inspired trust model. In: Mobile and Wireless Communication Networks. Springer (2006) 35–46
27. de Jong, S., Tuyls, K.: Human-inspired computational fairness. Autonomous Agents and Multi-Agent Systems **22**(1) (2011) 103–126
28. Lughofer, E.: Human-inspired evolving machinesthe next generation of evolving intelligent systems. IEEE SMC newsletter **36** (2011)
29. Pesquita, C., Faria, D., Falcao, A.O., Lord, P., Couto, F.M.: Semantic similarity in biomedical ontologies. PLoS comput biol **5**(7) (2009) e1000443
30. Pileggi, S.F.: Web of similarity. Journal of Computational Science (2016)
31. Quilitz, B., Leser, U.: Querying distributed rdf data sources with sparql. In: European Semantic Web Conference, Springer (2008) 524–538