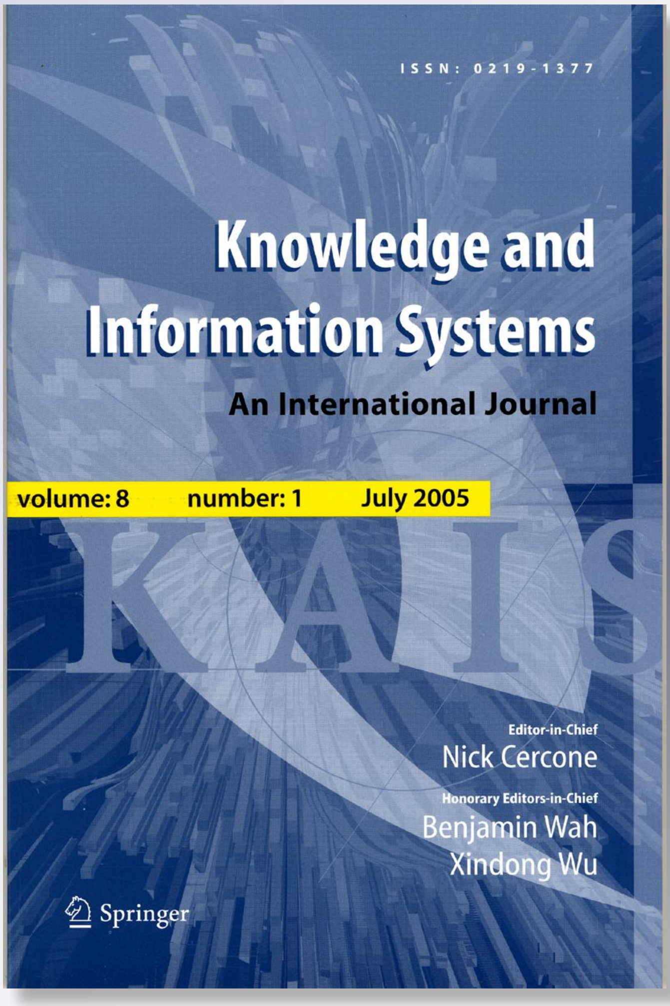
IKARUS-Onto: a methodology to develop fuzzy ontologies from crisp ones

Panos Alexopoulos, Manolis Wallace, Konstantinos Kafentzis & Dimitris Askounis

Knowledge and Information Systems
An International Journal

ISSN 0219-1377
Volume 32
Number 3

Knowl Inf Syst (2012) 32:667-695
DOI 10.1007/s10115-011-0457-6



Springer

Knowl Inf Syst (2012) 32:667–695
DOI 10.1007/s10115-011-0457-6

REGULAR PAPER

IKARUS-Onto: a methodology to develop fuzzy ontologies from crisp ones

Panos Alexopoulos · Manolis Wallace · Konstantinos Kafentzis · Dimitris Askounis

Received: 16 November 2010 / Revised: 11 April 2011 / Accepted: 22 October 2011 / Published online: 12 November 2011

Abstract Fuzzy Ontologies comprise a relatively new knowledge representation paradigm that is being increasingly applied in application scenarios in which the treatment and utilization of vague or imprecise knowledge are important. However, the majority of research in the area has mostly focused on the development of conceptual formalisms for representing (and reasoning with) fuzzy ontologies, while the methodological issues entailed within the development process of such an ontology have been so far neglected. With that in mind, we present in this paper IKARUS-Onto, a comprehensive methodology for developing fuzzy ontologies from existing crisp ones that significantly enhances the effectiveness of the fuzzy ontology development process and the quality, in terms of accuracy, shareability and reusability, of the process's output.

Keywords Ontology engineering · Vagueness · Fuzzy ontologies · Knowledge-based systems · Semantic web · Knowledge management

P. Alexopoulos (✉) · K. Kafentzis
IMC Technologies, Athens, Greece
e-mail: palexopoulos@imc.com.gr

K. Kafentzis
e-mail: kkafentzis@imc.com.gr

M. Wallace
Department of Computer Science and Technology,
University of Peloponnese, Tripolis, Greece
e-mail: wallace@uop.gr

P. Alexopoulos · D. Askounis
School of Electrical and Computer Engineering,
National Technical University of Athens, Athens, Greece

D. Askounis
e-mail: askous@epu.ntua.gr

1 Introduction

Ontologies are formal conceptualizations of domains, describing the meaning of domain aspects in a common, machine-processable form by means of concepts and their interrelations [11]. As such they have been realized as the key technology for modeling and utilizing domain knowledge for purposes like semantic annotation [33], document clustering [16], decision support [8], and knowledge management [20].

Recently, however, classical ontology representation formalisms, such as the Ontology Web Language (OWL) [5], have been deemed as inadequate for the semantic expression of imprecise and vague knowledge which is inherent to several real-world domains [40,44]. Indeed, the phenomenon of vagueness, manifested by terms and concepts like *Tall, Red, Bad, Heap, Child* etc., is quite common in human knowledge and language and it is related to our inability to precisely determine the extensions of such concepts in certain domains and contexts. That is because vague concepts have typically fuzzy, or blurred, boundaries which do not allow for a sharp distinction between the entities that fall within the extension of these concepts and those which do not. Consequently, classical representation formalisms, in which all predicates have well-defined extensions (i.e they cannot have fuzzy boundaries), cannot accommodate the representation of vagueness. Yet, such a representation in ontology-based systems is important, not only because vagueness is present in many domains but also because, in many application scenarios, the consideration and exploitation of vagueness can significantly enhance the systems' effectiveness [3,9].

An emerging result of the above situation has been the notion of **Fuzzy Ontology**, namely an attempt to tackle the problem of representing vague semantic information by borrowing and utilizing concepts and techniques from the area of Fuzzy Set Theory [22]. The latter is a mathematical theory that enables the capture of vague knowledge through the notions of **Fuzzy Set** and **Fuzzy Relation**. Informally, a fuzzy set is a set to which an object may belong to a certain degree (a real number from the interval [0, 1]). In a similar fashion, a fuzzy relation associates two objects to a certain degree [22]. Thus, a fuzzy ontology may be informally defined as an ontology that expresses vague knowledge through the use of these two notions (and those derived from them).

Fuzzy ontologies have been developed and utilized in a number of different systems and application areas including information retrieval [9,29], thematic categorization [43], semantic matchmaking [1], decision support [2], and data mining [13,14]. In all these cases, the common issues that fuzzy ontology developers have to deal with are (i) how to perform the ontology development in the most effective way and (ii) how to ensure that the final product is of good quality. In traditional ontology development, these two issues are typically tackled by means of ontology development methodologies, ontology representation languages, and ontology development tools [17]. In the area of fuzzy ontologies, however, the methodological aspect of the development process has been so far neglected.

More specifically, several conceptual formalisms for representing (and reasoning with) fuzzy ontologies have been proposed in the literature [7,25,27,37,39,45], the most known being those that incorporate fuzzy set theory into description logics [7,27,37]. Also, various software tools have been developed for defining fuzzy ontologies through such formalisms and for performing corresponding reasoning tasks [6,7,35]. However, what the fuzzy ontology community currently lacks is some kind of methodology that could enhance the effectiveness of the fuzzy ontology development process and the quality of its final product by ensuring that:

1. The ontology engineers and the domain experts can easily and correctly identify the knowledge of the domain that needs to be modeled as fuzzy.
2. The ontology engineers can determine the appropriate fuzzy ontology elements for representing this knowledge.
3. The domain experts can intuitively decide what the values of the degrees of the various fuzzy ontology elements should approximately be.
4. The fuzzy degrees of the ontology approximate, as accurately as possible, the vagueness of the domain.
5. The fuzzy ontology is reusable and shareable by having the meaning of its fuzzy elements and their degrees explicitly defined and commonly accepted.

With that in mind, we propose in this paper **IKARUS-Onto** (**I**mprecise **K**nowledge **A**cquisition **R**epresentation and **U**se), a novel methodology for effectively developing reusable and shareable fuzzy ontologies from existing crisp ones. The methodology provides concrete steps and guidelines for (i) correctly identifying vague knowledge within a domain (e.g., by not mixing vagueness with other notions such as uncertainty or ambiguity) and (ii) modeling this knowledge by means of fuzzy ontology elements in an explicit and as much as possible accurate way. Yet, it should not be regarded as a new methodology for developing ontologies but rather as a methodology for the conversion of existing conventional ontologies into fuzzy ones. The provision of support for developing fuzzy ontologies from scratch (i.e., support for developing the crisp part as well) falls outside the scope of our work.

Given the above, the structure of the rest paper is as follows: In the next section, we provide a short overview of the research field of ontology construction and we position our proposed methodology within it. In Sect. 3, we present the ways in which vagueness is present in ontological information. In Sect. 4, we describe the typical elements of a fuzzy ontology while in Sect. 5, we describe the steps of our methodology and the tasks involved within them along with concrete guidelines on how to perform them. In Sect. 6, we practically demonstrate the way to apply the proposed methodology through a concrete application scenario in which we applied our methodology. In Sect. 7, we seek to establish the importance and effectiveness of our approach by conducting an experiment in which we measure the degree of enhancement that the methodology achieves in the effectiveness of the fuzzy ontology development process, as well as the level of easiness at which this happens. Finally, in Sects. 8 and 9, we make a critical discussion of our work, we summarize its key aspects and we discuss the potential directions it could take in the future.

2 Related work on ontology construction

The importance of applying engineering principles in the ontology development process has been recognized for a long time now and several methodologies have been proposed for this purpose including METHONTOLOGY [15], Diligent [42], HCOME [23], and DOGMA [19]. Typically, such a methodology defines a set of activities that need to be performed while developing an ontology and usually suggests or provides methods and techniques for effectively carrying out these activities' tasks.

However, while the introduction of fuzziness into ontologies does not change dramatically the nature of the typical ontology development activities, it definitely induces the need for enriching them with new "fuzzy-related" tasks and relevant methods and techniques that may effectively support them. This kind of enrichment is what IKARUS-Onto provides by defining tasks and techniques for converting conventional ontologies into fuzzy ones while preserving (and requiring) the fundamental activities of classical ontology development. In that way it could be seen as a complementary methodology to ontology construction, particularly appropriate for domains and application scenarios where the handling of vagueness is important.

On the other hand, IKARUS-Onto is not directly related to the research area of automatic ontology construction. This area, also known as **Ontology Learning** [10,48], seeks to discover ontological knowledge from various forms of data [26] automatically or semi-automatically in order to overcome the bottleneck of ontology acquisition in ontology development. For that, it uses techniques from the areas of Machine Learning and Data Mining [4,31,32,38].

Similar research has also been done in the area of fuzzy ontologies with several proposed methods for fuzzy ontology learning [12,24,46,47]. The difference between these methods and IKARUS-Onto is that the latter is a generic methodology for developing fuzzy ontologies rather than a specific method or technique. This means that any of these methods may be used in the process of developing a fuzzy ontology, yet IKARUS-Onto is what will provide the generic context and the required engineering principles for the development.

3 Vagueness

In this section, we present the ways in which vagueness is present in ontological information, we disambiguate the notion from other notions such as uncertainty or inexactness, and we identify basic kinds of vagueness. This kind of background knowledge is an important element of our proposed methodology as it ensures a more accurate and complete treatment of vagueness in the fuzzy ontology development process.

Vagueness as a semantic phenomenon is typically manifested through predicates that admit borderline cases [21,34], namely cases where it is unclear whether or not the predicate applies. For example, some people are borderline tall: not clearly tall and not clearly not tall. In a more formal way, an object a is a borderline case of a predicate P if $P(a)$ is "unsettled", namely if it is not determinately true that $P(a)$ nor is it determinately true that $\neg P(a)$ [34]. The characterization "determinately" for the truth of a predicate means that the thoughts and practices in using the language have established truth conditions for it [28].

Obviously, having borderline cases is related to having fuzzy boundaries. For example, on a scale of heights, there appears to be no sharp boundary between the tall people and the rest. Therefore, two equivalent ways of drawing the distinction between vague and non-vague (or precise) predicates are to say that (i) vague predicates can possibly have borderline cases while precise predicates do not or that (ii) vague predicates lack sharp boundaries.

In the relevant literature, two basic kinds of vagueness are identified: *degree-vagueness* and *combinatory vagueness* [21]. A predicate P has degree-vagueness if the existence of borderline cases stems from the lack (or at least the apparent lack) of precise boundaries between application and non-application of the predicate along some dimension. For example, *Bald* fails to draw any sharp boundaries along the dimension of hair quantity and *Tall* along the dimension of height. Of course it might be that a predicate has degree-vagueness in more than one dimensions (e.g., *Red* can be vague along the dimensions of brightness and saturation).

On the other hand, a predicate P has combinatory vagueness if there is a variety of conditions all of which have something to do with the application of the predicate, yet it is not possible to make any sharp discrimination between those combinations which are sufficient and/or necessary for application and those which are not. A classical example of this type is *Religion* as there are certain features that all religions share (e.g., beliefs in supernatural beings, ritual acts etc.), yet it is not clear which of these features are able to classify something as a religion.

It is important that vagueness is not confused with the notions of inexactness, uncertainty, and ambiguity. For example, stating that someone is between 170 and 180 cm is an inexact statement but it is not vague as its limits of application are precise. Similarly, the truth of an uncertain statement, such as *"Today it might rain"*, cannot be determined due to lack of adequate information about it and not because the phenomenon of rain lacks sharp boundaries. Finally, the truth of a statement might not be determinable due to the ambiguity of some term (e.g., in statement *Yesterday we went to the bank* the term *bank* is ambiguous), yet again this does not make the statement vague.

Finally, vagueness is context dependent as the extension of a vague term may vary depending on the context it is being applied. For example, a person can be tall with respect to the average population height and not tall with respect to professional basketball players. Similarly, a person can be wealthy with respect to its local community but poor with respect to its boss. This does not mean that a term may be vague in one context and non-vague in another but rather that the interpretation of its vagueness may be different.

4 Elements of fuzzy ontologies

Classical ontologies are typically represented by means of concepts, instances, attributes, and relations. A concept represents a set or class of objects within a domain, while the objects that belong to a particular concept are called instances of this concept. An attribute in turn represents some characteristic of a concept(for which instances of this concept have values) and a relation describes the relationship that can be established between concepts (and consequently between their instances). Fuzzy ontologies, on the other hand, are represented by similar to the above elements which, in addition, allow for representation of fuzzy degrees that express the vagueness of the knowledge. Table 1 illustrates the difference between classical ontological statements and fuzzy ones.

In particular, given the various existing formalisms for fuzzy ontologies [7,37,45], the basic elements a fuzzy ontology consists of can be summarized as follows:

– **Fuzzy Concepts:** A fuzzy ontology concept is a concept whose instances may belong to it at certain degrees. Such a degree practically denotes the extent to which a given entity should be considered as being an instance of the concept. As an example, consider the concept *TallPerson* whose instances are meant to be people whose height classifies them as tall. Since *tall* is a vague predicate, the concept is also vague and therefore can be represented as a fuzzy one by allowing the expression of statements such as *Person X is an instance of TallPerson at a degree of* 0.8. In languages based on fuzzy Description Logics, such statements are called *fuzzy concept assertions* [7,37].

Table 1 Crisp ontology versus fuzzy ontology

Crisp ontology statement	Fuzzy ontology statement
Jane is expert at artificial intelligence	Jane is expert at artificial intelligence to a degree of 0.8
The film "Notting Hill" is a comedy	The film "Notting Hill" is a comedy to a degree of 0.6
John is 20 years old	John is young

- **Fuzzy Relations:** Similar to fuzzy concepts, a fuzzy ontology relation links concept instances at certain degrees. Such a degree practically denotes the extent to which the relation between the two instances should be considered as true. For example, the relation *isExpertAt*, which contains the vague predicate *expert*, can be represented as a fuzzy one by allowing the expression of statements like *John is expert at Knowledge Management at a degree of* 0.5. In languages based on fuzzy Description Logics, such statements are called *fuzzy role assertions* [7,37].
- **Fuzzy Attributes:** A fuzzy attribute assigns literal values to concept instances at certain degrees. Such a degree denotes the extent to which the value is applicable to the instance for the given attribute. For example, the attribute *category* of concept *Film*, which takes as values string literals denoting the category a film may belong to (e.g. "science fiction", "horror", "comedy" etc.), can be represented as a fuzzy one by allowing the expression of statements like *The category of film "High Fidelity" is "comedy" at a degree of* 0.7. In practice, fuzzy attributes are like fuzzy relations with the difference that the second relatum is a literal value rather than an instance.
- **Fuzzy Datatypes:** A fuzzy datatype consists of a set of vague terms which may be used within the ontology as attribute values. For example, the attribute *performance*, which normally takes as values integer numbers, may in a given domain, context or application scenario be required to take as values terms like *very poor, poor, mediocre, good*, and *excellent*. What a fuzzy datatype does is to map each term to a fuzzy set that defines its meaning by assigning to each of datatype's potential exact values a fuzzy degree. This degree practically indicates the extent to which the exact value and the vague term express the same. Fuzzy datatypes are also known as fuzzy concrete domains [7] or fuzzy linguistic variables [45].

The above description of fuzzy ontology elements is not exhaustive nor it should be regarded as a proposed conceptual formalism for representing fuzzy ontologies. It is merely a convenient terminology for use within our proposed methodology. As we will see in the rest of the paper, the above descriptions make it easier for both the engineers and the domain experts to identify and express fuzzy information as they are natural extensions of the elements of classical ontologies.

5 The IKARUS-Onto methodology

As suggested in Sect. 1, IKARUS-Onto methodology defines a set of concrete steps and guidelines for representing vague knowledge by means of fuzzy ontology elements. The focus of the methodology is not so much on the structure of the fuzzy ontology but rather on the process followed for its development and the content it ultimately has. The process aims to make it easier for the ontology engineer and the domain experts to identify and model the vagueness of the domain while the content reflects this vagueness as accurately as possible. IKARUS-Onto assumes prior knowledge of classical ontology development from those who use it. Finally, we should clarify that IKARUS-Onto should not be regarded as a new methodology for developing ontologies but rather as a methodology for the conversion of existing conventional ontologies into fuzzy ones.

The lifecycle of IKARUS-Onto and the steps involved in it are depicted in Fig. 1. For each step, we list the purposes it serves, the actions that need to be done in order to fulfill them and who typically performs each action, i.e., ontology engineers (OE) or domain experts (DE).

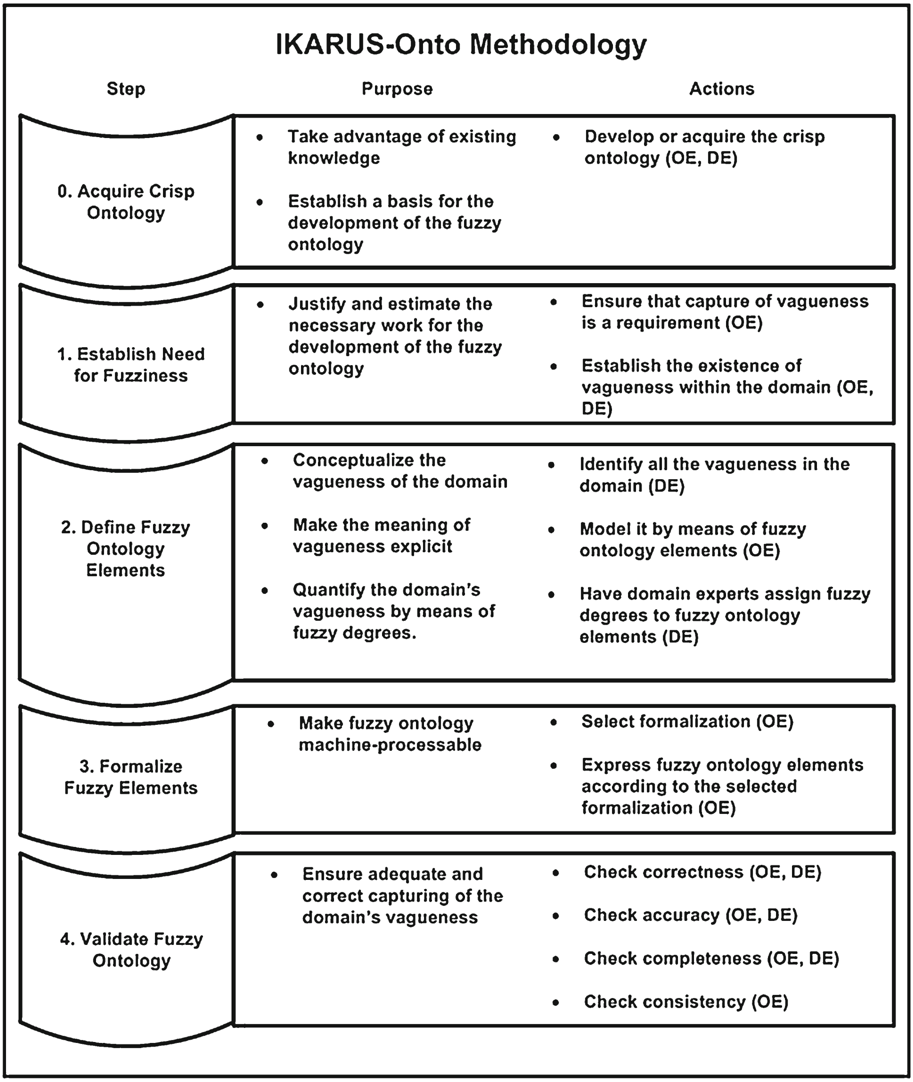


Fig. 1 The IKARUS-Onto methodology

The starting point of the methodology is some domain which is already modeled by means of some crisp ontology. If no such ontology exists, then it needs to be developed (by using some conventional methodology) as it will be the basis upon which the development of the fuzzy ontology will be performed. The main reasons we follow this approach (i.e., not mixing the development of the crisp part of the ontology with the fuzzy one) are because:

- It is easier for the ontology engineers and domain experts to identify vagueness in already conceptualized and structured knowledge.
- In many domains, the crisp part of the fuzzy ontology is already available as a separate ontology.

– This approach enables us to assign the fuzzy part of the ontology development process to experts who are not necessarily familiar with the conventional ontology development methodologies and formalizations, thus partially alleviating the knowledge elicitation problem.

In the following paragraphs, we present each step of the proposed methodology in an more detailed manner.

5.1 Step 1: establishment of the need for fuzziness

Establishing the need for fuzziness practically means determining whether and to what extent is vagueness present in the domain at hand, as well as whether the intended uses of the ontology require the capturing of this vagueness. This is necessary as it justifies and estimates the work that will be required for the development of the fuzzy ontology.

In particular, the execution of this step is performed by first having the ontology engineer ensure that the capturing and modeling of the domain's vagueness is required by the ontology's application scenario. This is usually the case when there is (or there needs to be) some knowledge-based system which handles and utilizes vague knowledge, by means of fuzzy ontologies, for purposes such as information retrieval, thematic categorization, knowledge management, etc. In rarer cases, there might not be such a system but merely the requirement to represent the domain's vagueness in an explicit way.

Given the requirement for vagueness, the engineer then establishes whether the latter is actually present within the domain. This is done by identifying elements of the (acquired or developed) crisp domain ontology whose meaning may be interpreted as vague in the given domain and/or application scenario. In particular, given the definitions about vagueness of Sect. 3 and the elements of Sect. 4, this identification is performed as follows:

– **Identification of vague concepts:** A concept is vague if, in the given domain, context or application scenario, it admits borderline cases, namely if there are (or could be) individuals for which it is indeterminate whether they instantiate the concept. Primary candidates for being vague are concepts that denote some phase or state (e.g Adult, Child) as well as attributions, namely concepts that reflect qualitative states of entities (e.g., Red, Big, Broken etc.).
– **Identification of vague relations and attributes:** A relation is vague if, in the given domain, context or application scenario, it admits borderline cases, namely if there are (or could be) pairs of individuals for which it is indeterminate whether they stand in the relation. The same applies for attributes and pairs of individuals and literal values.
– **Identification of vague attribute value terms:** Such terms are identified by considering the ontology's attributes and assessing whether their potential values can be expressed through vague terms. Primary candidates for generating such terms are gradable attributes such as size or height which give rise to terms such as *large, tall, short*, etc.

Obviously, during this step, the identification of vagueness needs not be exhaustive but merely sufficient for establishing the need for the development of the fuzzy ontology. What is important, however, is making sure that the identified as vague elements do actually convey a vague meaning and not something else (e.g., uncertain or inexact meaning). To that end, the above guidelines of our methodology are important.

5.2 Step 2: definition of fuzzy ontology elements

The second step of IKARUS-Onto involves the comprehensive identification of the vague knowledge of the domain and its explicit description by means of the fuzzy ontology elements of Sect. 4. The goal of this description is to ensure that the defined fuzzy elements (i) have a clear and specific vague meaning which makes them shareable and reusable and (ii) approximate, through their fuzzy degrees, this vagueness as accurately as possible. For that, IKARUS-Onto proposes a specific procedure and description template for describing each type of fuzzy ontology elements.

In particular, step 2 starts by identifying and describing the ontology's fuzzy relations and fuzzy attributes. As suggested in the previous sections, these two elements are quite similar differing only in that fuzzy relations link instances to each other while fuzzy attributes instances to literal values. Therefore, the procedure for defining these two types of elements is the same and comprises the following tasks:

1. The identification in the crisp ontology of <u>all</u> the relations/attributes that convey vague meaning using the guidelines described in step 1.
2. The determination for each relation/attribute of the type of its vagueness (combinatory or degree-vagueness). If the element has degree-vagueness, then the dimensions along which it is vague need to be identified.
3. The usage of the above information for defining for each element the exact meaning of its vagueness. In case the element has degree-vagueness along multiple dimensions then the distinction between the dimensions might or might not be important. In case it is then it is necessary to define a distinct fuzzy element for each dimension.
4. The definition of the expected interpretation of each element's fuzzy degrees. If fuzziness is due to degree-vagueness then the fuzzy degree of a related pair of instances (or instances and literal values) practically approximates the extent to which the pair's value for the given dimension places it within the elements's application boundaries. If, on the other hand, fuzziness is due to combinatory vagueness, then the fuzzy degree practically approximates the extent to which the pair's set of satisfied application conditions of the relation/attribute is deemed sufficient for the relation/attribute to apply.
5. The assignment of specific fuzzy degrees to pairs of instances (or instances and literal values) that instantiate each element. These degrees should approximate as accurately as possible the already defined element's degree interpretation for the given pair.

In the end of the above procedure, a set of element descriptions similar to those of Table 2 shall be produced. Through this template, the nature of the fuzzy element's vagueness and the expected interpretation of its fuzzy degrees are made explicit. This ensures not only the common understanding of the elements' vague meaning by all the fuzzy ontology users but it also helps the domain experts to assign fuzzy degrees to the instances of these elements in a more intuitive and accurate way.

After the specification of all the ontology's fuzzy relations and attributes, the specification of fuzzy datatypes takes place. The procedure for this involves:

1. The identification of the ontology attributes whose values may be expressed by means of vague terms.
2. The identification of these terms and their grouping into fuzzy datatypes (usually one datatype for each attribute). It should be noticed that the same term may belong to more than one datatypes.

Table 2 Sample description template for fuzzy ontology relations and attributes

Relation or attribute	Vagueness nature	Degree interpretation
isNearTo	Degree-vagueness along the dimension of distance	The extent to which the distance between the related instances classifies them as being near to each other
isFunctionalPartOf	Degree-vagueness along the dimension of the part's contribution to the functionality of the whole	The extent to which the part's contribution to the functionality of the whole classifies the part as functional
isCompetitorOf	Degree-vagueness along the dimension of the competitor's business areas and the dimension of the competitor's target markets	The extent to which the relation subject's business areas and/or target markets classifies it as a competitor of the object
belongsToCategory	Combinatory vagueness due to the lack of sharp discrimination between those conditions that are necessary for something to belong to a given category	The extent to which the subject's set of satisfied category's conditions classifies it as belonging to this category
isExpertAt	Degree-vagueness along the dimension of the level of knowledge on a subject	The extent to which the level of someone's knowledge on a subject classifies him as expert on it

3. The definition for each vague term within a fuzzy datatype of a fuzzy set that defines its meaning. Again its important to note that the same term may be associated with different fuzzy sets in different datatypes.

Figure 2 shows a sample fuzzy datatype for the attribute *Performance* which comprises the vague terms *very poor, poor, mediocre, good*, and *excellent*. As shown in the figure, each term corresponds to a fuzzy set that defines its meaning. Thus for example, a performance of 45 is considered *mediocre* at a degree of 1.0, while a performance of 38.75 is considered *poor* at a degree of 0.5 and *mediocre* at the same degree.

The final type of fuzzy ontology elements to be defined within this step is that of fuzzy concepts. The process followed for this definition is similar to the one for fuzzy relations and attributes with, nevertheless, an important difference. In many cases, vague concepts "owe" their vagueness to some vague relation, attribute or term which has been already defined by means of a corresponding fuzzy ontology element. If that is the case, then the definition of the concept's vagueness can be directly derived from the one of its causal element. Thus, the definition process for fuzzy concepts includes:

1. The identification of all the ontology concepts that convey vague meaning using the guidelines described in step 1.
2. The determination for each vague concept of whether its vagueness is caused by one or more already defined fuzzy elements (fuzzy relation, fuzzy attribute, fuzzy datatype). If this is the case, then the meaning of the concept's vagueness as well as the interpretation of the fuzzy degrees of its instances are directly derived from the ones of the causal element.
3. The definition, otherwise, of each concept in the same way as in the case of fuzzy relations and fuzzy attributes (including the assignment of fuzzy degrees to pairs of instances and concepts).

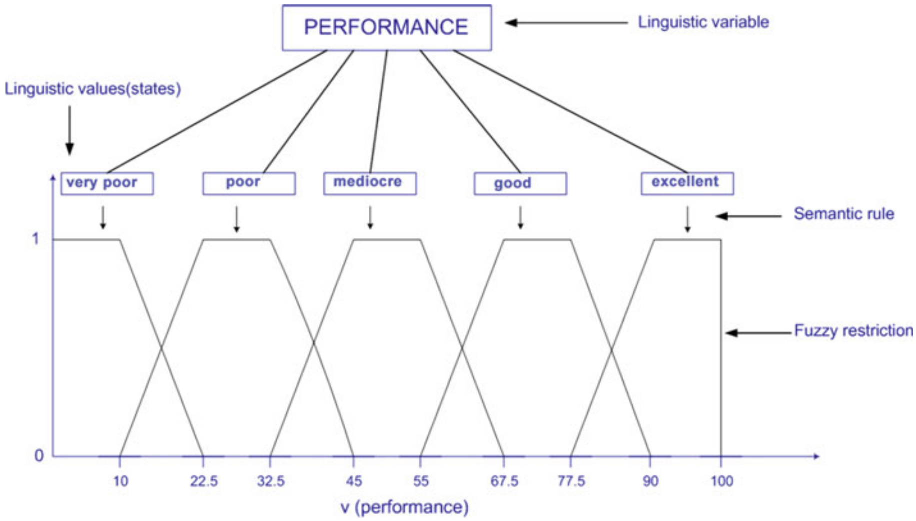


Fig. 2 Fuzzy datatype example

Table 3 Sample description template for fuzzy ontology concepts

Concept	Vagueness nature	Degree interpretation
BaldPerson	Degree-vagueness along the dimension of hair quantity	The extent to which the person's hair quantity classifies it as bald
MediocrePerformer	Degree-vagueness derived from the *Performance* fuzzy datatype	The same as in the term *mediocre* of the *Performance* datatype
Religion	Combinatory vagueness due to the lack of sharp discrimination between those conditions that are necessary for something to be a religion	The extent to which the instance's set of satisfied religion-related conditions classifies it as a religion
ScienceFictionFilm	Combinatory vagueness derived from the *belongsToCategory* fuzzy ontology relation	The same as in the *belongsToCategory* relation

Table 3 shows a sample output of the above process. Again, the benefit of such a description is related to the explicitness of the concepts' vague meaning and the ease and accuracy of the assignment of fuzzy degrees to their instances.

5.3 Step 3: formalization of fuzzy ontology elements

The formalization step involves the transformation of the defined fuzzy ontology elements into a formal machine-interpretable form through some corresponding fuzzy ontology language. As with conventional ontology languages, fuzzy ontology languages typically vary in terms of the representation and reasoning capabilities they provide. Therefore, the ontology engineer needs to consider the particular characteristics of each language and the capabilities it provides for representing (and reasoning with) vague knowledge. In particular, important parameters of a fuzzy ontology language include:

- **The range of fuzzy ontology elements it supports:** Not all languages support the whole range of Sect. 4 fuzzy ontology elements and in the same way. For example, the fuzzy description logic f-SHIN [36] does not support fuzzy datatypes and the language in [7] supports fuzzy datatypes by means of fuzzy concrete domains while the one in [45] by means of fuzzy linguistic variables.
- **The range of fuzzy reasoning capabilities it has support for:** Certain reasoning services provided by crisp ontology reasoners require adaptation when the ontology is fuzzy. An example is entailment which, in the fuzzy case, requires from the reasoner to be able to determine whether an individual belongs to a concept in a specific degree. Some, but not all, fuzzy ontology languages are accompanied by corresponding reasoners [6,7,35] which typically vary in the range of reasoning capabilities they provide.

The above practically means that, depending on the application scenario within which the fuzzy ontology is utilized, a certain language might be more appropriate than another. A more detailed description of how a fuzzy ontology is actually formalized through such a language is a language-specific task and falls outside the scope of this work.

5.4 Step 4: validation of fuzzy ontology

After the fuzzy ontology has been built, a validation process needs to take place in order to ensure that the developed artifact captures and represents the vagueness of the domain in an adequate and correct way. This is practically translated into evaluating the following four properties of the fuzzy ontology:

- **Correctness:** A fuzzy ontology is correct when all its fuzzy elements convey a meaning which is indeed vague in the given domain, context or application scenario. This means that the fuzziness of each element is actually caused by the potential existence of borderline cases and not because of some other reason (e.g., uncertainty or ambiguity).
- **Accuracy:** A fuzzy ontology is accurate when the degrees of its fuzzy elements approximate the latter's vagueness in an intuitively accurate way for the given domain, context, or application scenario. This does not mean that the fuzzy degrees should have specific values but merely that these values should be perceived as natural by those who use the ontology. For example, the fuzzy statement *Obama is a BlackPerson at a degree of 0.2* is highly unintuitive (and therefore inaccurate) while the statement *Sergey Brin is a RichPerson at a degree of 0.8* makes much more sense.
- **Completeness:** A fuzzy ontology is complete when all the vagueness of the domain has been represented within the ontology. This primarily means that the ontology should not contain crisp elements which in the given domain, context, or application scenario convey vague meaning. It also means that for each vague element all the (required) dimensions of its vagueness should have been identified and modeled.
- **Consistency:** A fuzzy ontology is consistent when it does not contain controversial information about the domain's vagueness as this is expressed by fuzzy degrees. For example, the fuzzy statements *Obama is a BlackPerson at a degree of 0.9* and *Obama hasSkinColor black at a degree of 0.4* are controversial.

The first three properties are typically evaluated by humans (ontology users and domain experts) while the fourth may be checked by means of some fuzzy ontology reasoner. Also, it should be noted that the evaluation of completeness is performed with respect to the domain as this has been captured by the initial crisp ontology. This means that vague knowledge that was not originally contained as crisp within the initial domain ontology will not be contained in the final fuzzy ontology either and, therefore, should not be included in the evaluation process.

Finally, the validation process does not include the task of checking the applicability of the developed ontology because (i) the applicability of the utilized crisp ontology has already been checked in step 0 of the methodology as part of the task of deciding what ontologies to reuse and (ii) fuzziness does not really affect the applicability of the original ontology, it merely makes the vagueness of the domain explicit. Such a check would be important if someone considered reusing an existing fuzzy ontology but this task is out of the scope of this paper.

6 Practical example: developing a fuzzy enterprise ontology for a consulting firm

To illustrate the way IKARUS-Onto can be practically applied to the construction of fuzzy ontologies, we present in this section an indicative portion of a real life application scenario that involves the development of a fuzzy enterprise ontology for a consulting firm. According to the scenario, the ontology needs to model knowledge about the firm's operation which is to be utilized by a knowledge-based decision support system the consulting firm has. The system is described in more detail in [2], and it is capable of handling and exploiting vague knowledge in the form of fuzzy ontologies for helping the firm to decide whether it should write a proposal for a tender call (Fig. 3).

In the following paragraphs, we describe how each step of IKARUS-Onto is executed in order to produce the aforementioned ontology in such a way so as to effectively capture and express the vagueness of its domain. We omit from the example step 3 (formalization) as this would require the analytical description of some specific fuzzy ontology language, which falls outside the scope of this work.

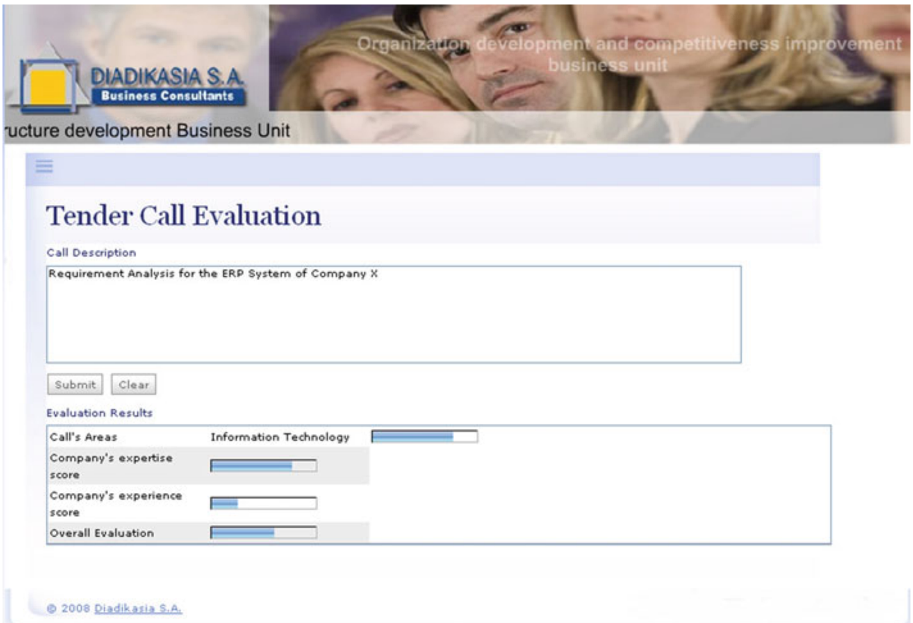


Fig. 3 Decision support system for tender call evaluation using fuzzy ontology

6.1 Applying step 0: acquiring the crisp ontology

As suggested in Sect. 5, the starting point of the fuzzy ontology development is the acquisition of a crisp ontology that models the target domain. For the needs of this demonstration and given the publication's length and clarity considerations, we limit the ontology to the following elements:

Concept taxonomy

- Company
  - Competitor
- Employee
  - HighPotentialEmployee
- ConsultingArea
  - CompanyCoreConsultingArea: Refers to those consulting areas that constitute the firm's core competencies.
- Project
  - IT_Project
  - HighBudgetProject

Relations

- isExpertAtConsultingArea: Relates employees to the consulting areas they are expert at.
- regardsConsultingArea: Relates projects to the consulting areas they fall into.

Attributes

- projectBudget
- employeeExperience

Concept instances

- Companies: Accenture, McKinsey, Bain, Boston Consulting Group
- Competitors: Accenture, McKinsey
- Employees: Jane, John, Karen, Ian
- High Potential Employees: Jane, Ian
- Consulting Areas: Information Technology, Human Resources, Strategy, Marketing
- Company Core Consulting Areas: Information Technology, Strategy
- Projects: P1, P2, P3
- IT Projects: P1, P2

Relation instance pairs

- *isExpertAtConsultingArea*(Jane, Information Technology)
- *isExpertAtConsultingArea*(Jane, Strategy)
- *isExpertAtConsultingArea*(Ian, Strategy)
- *isExpertAtConsultingArea*(Ian, Marketing)
- *isExpertAtConsultingArea*(John, Marketing)
- *isExpertAtConsultingArea*(Karen, Strategy)
- *regardsConsultingArea*(P1, Information Technology)
- *regardsConsultingArea*(P1, Human Resources)
- *regardsConsultingArea*(P2, Information Technology)
- *regardsConsultingArea*(P3, Strategy)

6.2 Applying step 1: establishing the need for fuzziness

The first premise for fuzziness, namely the existence of some knowledge-based system which handles and utilizes fuzzy ontologies, is satisfied within the scenario through the firm's intelligent decision support system. For the second premise, namely the existence of vagueness within the domain, the ontology engineer tries, in collaboration with the firm's consultants who play the role of the domain expert, to detect borderline cases (for relations, attributes and concepts) and potential vague terms (for attributes) in the above ontology. The result of this process is the following vague elements:

- **Vague concepts:** Competitor (because some companies are borderline competitors), HighPotentialEmployee (because some employees are borderline high potential), CompanyCoreCompetenceArea (because some areas are borderline core), IT_Project (because some projects are borderline IT), and HighBudgetProject (because some projects are borderline high valued).
- **Vague relations:** *isExpertAtConsultingArea* (because some employees are borderline expert at some areas) and *regardsConsultingArea* (because some projects are borderline relevant to some areas).
- **Vague attribute value terms:** {low, average, high} for the attribute projectBudget and {junior, senior, veteran} for the attribute employeeExperience.

6.3 Applying step 2: defining the fuzzy ontology elements

Given the vague ontology elements of step 1, step 2 proceeds with the definition of each of these elements as fuzzy ones by following the corresponding procedures and templates of paragraph 5.2. This means that the ontology engineer, in collaboration with the domain experts, produces the following:

- The descriptions of the two fuzzy ontology relations (Table 4) and their corresponding fuzzily related instance pairs (Table 5). The fuzzy degrees are assigned by the domain experts based on the degree interpretations that they have themselves assigned to the fuzzy ontology relations (second column of Table 4).
- Two fuzzy datatypes (Figs. 4 and 5), each defining the meaning of the vague value terms of the attributes projectBudget and employeeExperience. The definition of the terms' fuzzy sets for each fuzzy datatype is performed by the domain experts based on their own understanding about when, for example, a budget is considered high or a consultant veteran.

Table 4 Fuzzy relations for the consulting ontology example

Relation	Vagueness nature	Degree interpretation
isExpertAtConsultingArea	Degree-vagueness along the dimension of the level of knowledge on a consulting area	The extent to which someone's knowledge on a consulting area classifies him as expert on it
regardsConsultingArea	Combinatory vagueness due to the lack of sharp discrimination between those conditions that are necessary for a project to belong to a given consulting area	The extent to which the project's set of satisfied consulting area's conditions classifies it as belonging to this area

Table 5 Fuzzy relations statements for the consulting ontology example

Relation statement	Fuzzy degree
isExpertAtConsultingArea(Jane, information technology)	0.8
isExpertAtConsultingArea(Jane, strategy)	0.9
isExpertAtConsultingArea(Ian, strategy)	0.6
isExpertAtConsultingArea(Ian, marketing)	0.4
isExpertAtConsultingArea(John, marketing)	0.75
isExpertAtConsultingArea(Karen, strategy)	0.8
regardsConsultingArea(P1, information technology)	0.9
regardsConsultingArea(P2, human resources)	0.5
regardsConsultingArea(P2, information technology)	0.5
regardsConsultingArea(P3, strategy)	0.8

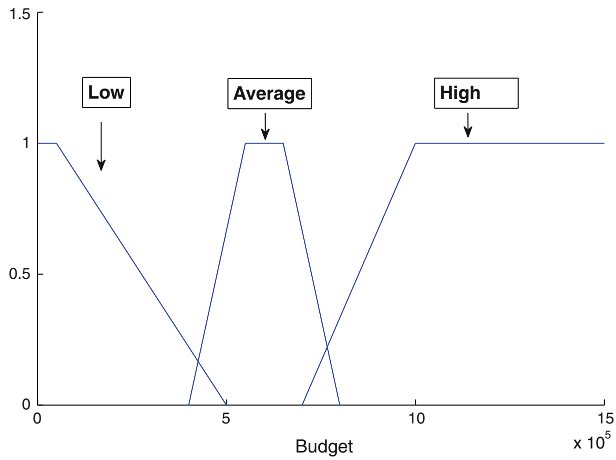


Fig. 4 Fuzzy datatype for project budget attribute

– The descriptions of the fuzzy ontology concepts (Table 3) and their corresponding fuzzily assigned instances (Table 7). For those fuzzy concepts that are defined through some fuzzy relation or fuzzy datatype (IT_Project and HighBudgetProject respectively), the fuzzy degrees of their instances are left to be determined by the system's reasoner. The others are determined by the domain experts again by considering the degree interpretations that they have themselves assigned to the fuzzy concepts (second column of Table 3).

6.4 Applying step 4: validating the fuzzy ontology

Validating the above fuzzy ontology in the specific scenario means ensuring that:

– The fuzzy ontology is correct, namely all its fuzzy elements convey a meaning which is indeed vague in the given domain. This is done by having another team of domain experts verify that the definitions of Tables 4 and 6 are correct.

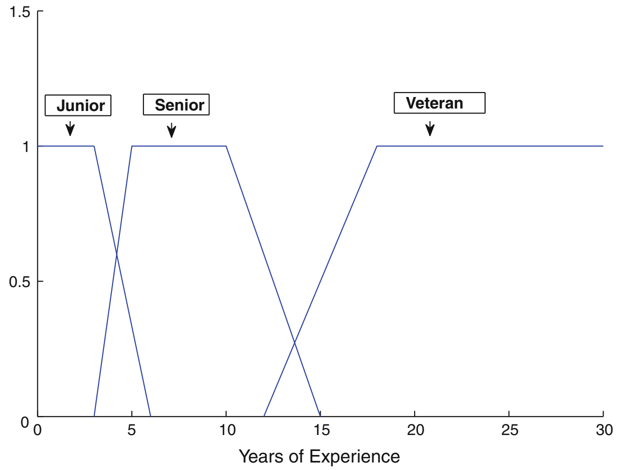


Fig. 5 Fuzzy datatype for employee experience attribute

Table 6 Fuzzy concepts for the consulting ontology example

Concept	Vagueness nature	Degree interpretation
Competitor	Degree-vagueness along the dimension of the competitor's business areas and the dimension of the competitor's target markets	The extent to which the instance's business areas and target markets classifies it as a competitor of the company
CompanyCoreConsultingArea	Degree-vagueness along the dimensions of the company's expertise and experience in the given area	The extent to which the company's expertise and experience in the given area classify the latter as core
HighPotentialEmployee	Combinatory vagueness due to the lack of sharp discrimination between those characteristics that are necessary for an employee to be considered as high potential	The extent to which the employees's characteristics classify him/her as high potential
IT_Project	Combinatory vagueness derived from the *regardsConsultingArea* vague relation	The same as in the *regardsConsultingArea* relation
HighBudgetProject	Degree-vagueness derived from the *projectBudget* fuzzy datatype	The same as in the term *high* of the *projectBudget* datatype

– The fuzzy ontology is accurate, namely the degrees of its fuzzy elements approximate the latter's vagueness in an intuitively accurate way. This is done by having another team of domain experts assess that the degrees of Tables 5 and 7 are intuitively correct.
– The fuzzy ontology is complete, namely all the vagueness of the domain has been represented within the ontology. This is done by having another team of domain experts check that the non-fuzzy parts of the ontology do not convey vague meaning.

Table 7 Fuzzy concept instances for the consulting ontology example

Concept instance	Fuzzy degree
Competitor(Accenture)	0.8
Competitor(McKinsey)	0.5
CompanyCoreConsultingArea(Information technology)	1.0
CompanyCoreConsultingArea(Strategy)	0.6
HighPotentialEmployee(Jane)	0.9
HighPotentialEmployee(Ian)	0.5

– The fuzzy ontology is consistent, namely it does not contain controversial information about the domain's vagueness. This is done by using some fuzzy ontology reasoner that is compatible to the language the ontology has been formalized with.

7 IKARUS-Onto evaluation

7.1 Evaluation approach

A methodology is in principle an abstract description of tasks to be performed by humans and thus inherently subjective as to how it may actually be executed. As a result, although knowledge engineering is clearly an engineering discipline, it lacks the quantitative analysis and evaluation procedures that we so much enjoy as engineers. This means that whereas we would like to test a new methodology against ground truth and in comparison to previous works and calculate the exact percentage of enhancement that it achieves, none of that is possible. This is the reason that traditionally methodologies related to knowledge engineering tasks are left unevaluated and are simply accepted as is.

Looking at the relevant literature [30], we find: METHONTOLOGY, which is probably the most widely used and acknowledged ontology development methodology currently available, is presented in [15] where there is not any sort of evaluation. Diligent is another well known and widely used methodology. It is presented in [42], where some examples of the application of its steps are given, but again without any sort of evaluation. Similarly, the presentation of DOGMA in [19] is accompanied by an example but again no reference to any evaluation. In [23] where HCOME is presented, we find in the text a statement that the method has been evaluated from a group of students but no actual details are presented, including the way the evaluation was set up, the group's organization, the students' role in the process, the questionnaires used, the answers obtained or even the conclusions drawn from the evaluation. In short, although there is a claim that the methodology has been evaluated, no such evaluation is given. Also the methodology of Gruninger and Fox, presented in [18], is given as is, without any sort of evaluation. Finally, the methodology in [41] is accompanied by an abstract case study, for which no actual parts of the ontology or specific steps of the process are presented. Again, there is no reference to any evaluation.

As seen from the above, it has generally been accepted that due to the subjective nature of the field, ontology development methodologies cannot really be evaluated rigorously. On the other hand, this does not alleviate our desire to have some sort of quantifiable measure of our method's performance, alone or with respect to previously proposed approaches. For this, we turn our attention to another field where subjectivity hinders evaluation efforts: that of personalization. Profile extraction systems, personalized interfaces, recommender systems and so on, although being branches of the broader field of personalization, are typically found in publications that include (shorter in some cases, extensive in others) evaluation sections. The approach followed there in order to overcome the problem of subjectivity is the use of large numbers of humans, so that the bias entered in the process by each one is overall statistically ignored.

Following a similar approach, we have designed a process that, via the involvement of relevant to the fuzzy ontology development process people, has allowed us to evaluate our proposed methodology. Of course, ours being, to the best of our knowledge, the first approach toward the fuzzification of an existing conventional ontology, there is no previous work to compare against. Instead, we compare against the previous best practice, which asks the knowledge engineer to perform the task based on intuition alone and not following some specific methodology. It is of course expected that the use of a methodology will produce either better or at least the same results, so the critical question in our evaluation is whether the enhancement provided by our methodology is important enough to justify the overhead of learning and using it.

7.2 Evaluation process

The evaluation of the IKARUS-Onto methodology took place as an in-house experiment at IMC Technologies involving the development of a fuzzy enterprise ontology, similar to the one of Sect. 6. The overall process followed is summarized in Fig. 6.

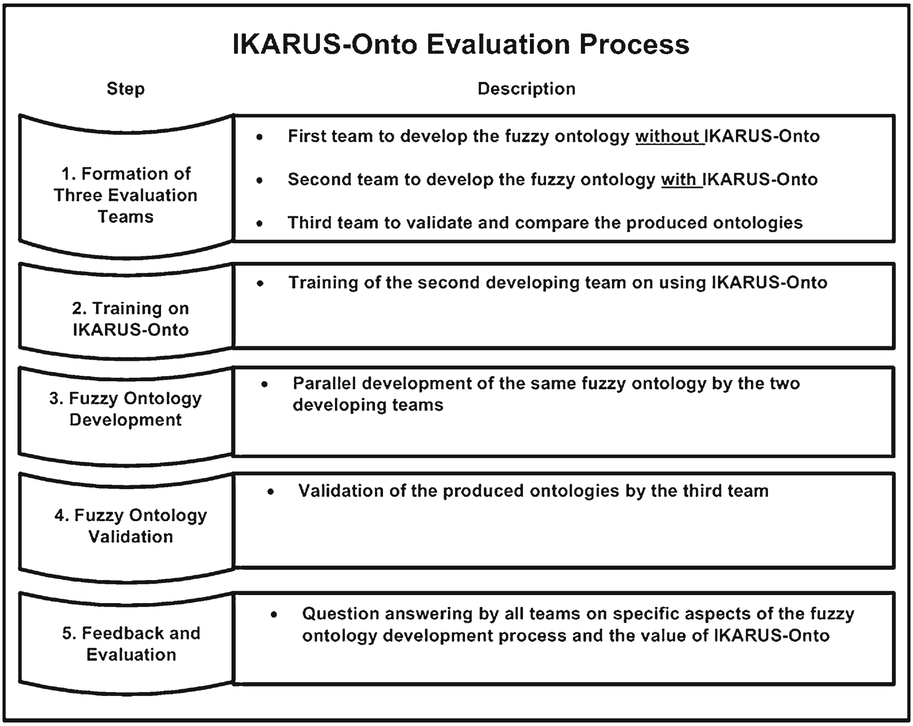


Fig. 6 IKARUS-Onto evaluation process

According to it, three development teams were formed, each with a different task:

- The first team had to develop the fuzzy ontology by using the IKARUS-Onto methodology.
- The second team had to develop the ontology without IKARUS-Onto, merely based on the existing literature about the subject (i.e., fuzzy ontology models and languages).
- The third team had to compare and evaluate the fuzzified ontologies produced by the former teams.

Each team comprised one knowledge engineer and five company employees who had the role of the domain expert. Prior to the execution of their tasks, a training session took place in which the members of all teams were acquainted with the notion of fuzzy ontology and the main elements this might consist of, while the members of the first team were additionally trained in using the IKARUS-Onto methodology. After the training the two teams went on with the development of their ontologies and, when they had finished, the third team performed an intuitive evaluation of the two ontologies. Subsequently, the third team was asked to perform two more evaluations, one after having been acquainted with the validation process of the IKARUS-Onto methodology and one after having been trained in the whole methodology. In the end, the members of the three teams were asked to answer the following questions:

- Knowledge engineers and domain experts of the teams that were trained in the methodology:
    - How easy did you find the task of becoming familiar with the whole process and applying it in practice?
- Domain experts of the two developing teams:
    - How easy was it for you to identify vague knowledge within the domain?
    - How easy was it for you to assign fuzzy degrees to the defined fuzzy elements?
- Knowledge engineers of the two developing teams:
    - How easy it was for you to guide the domain experts in their tasks (identification of vague knowledge and assignment of fuzzy degrees)?
- Knowledge engineer of the evaluation team:
    - How easy was for you to determine the criteria of the validation and evaluation process when you weren't aware of IKARUS-Onto?
- Domain experts of the evaluation team:
    - Given the validation criteria of IKARUS-Onto, but not the rest of the methodology, how easy was it for you to perform the validation? How easy was it after knowing the whole IKARUS-Onto?
    - Which ontology was easier to validate and how did each ontology perform in terms of completeness, correctness, and accuracy?

For all the "easiness" assessment questions, people were asked to select between three possible answers, namely "low", "average", and "high". For the last two questions that regarded the comparative evaluation of the two ontologies, people were asked to grade the performance for each evaluation criterion (easiness of validation, correctness, completeness, and accuracy) at a scale from 1 to 10.

Table 8 Timeline of the IKARUS-Onto evaluation experiment

<table>
<tr><th>Time</th><th>Control group</th><th>IKARUS-Onto group</th></tr>
<tr><td>0:30</td><td rowspan="2">Introduction to the scope of the work</td><td rowspan="2">Introduction to the scope of the work</td></tr>
<tr><td>1:00</td></tr>
<tr><td>1:30</td><td rowspan="8">Development of first draft</td><td rowspan="4">Training on IKARUS-Onto</td></tr>
<tr><td>2:00</td></tr>
<tr><td>2:30</td></tr>
<tr><td>3:00</td></tr>
<tr><td>3:30</td><td rowspan="6">Development of the ontology</td></tr>
<tr><td>4:00</td></tr>
<tr><td>4:30</td></tr>
<tr><td>5:00</td></tr>
<tr><td>5:30</td><td rowspan="4">Enhancement of the ontology</td></tr>
<tr><td>6:00</td></tr>
<tr><td>6:30</td><td></td></tr>
<tr><td>7:00</td><td></td></tr>
<tr><td>7:30</td><td rowspan="2">Gathering feedback regarding the process</td><td rowspan="2">Gathering feedback regarding the process</td></tr>
<tr><td>8:00</td></tr>
</table>

In terms of time, the first part of the experiment, i.e., the development of the fuzzy ontologies, was performed over a single 8-hour work day, while the evaluation of the produced ontologies took part on the second day. On the first day, one hour was dedicated to the "generic" training of all participants of the experiment. This includes a typical introduction to what an ontology is, what a fuzzy ontology is, which is the scope of the project in hand, and which will be the use of the developed ontology. This is a standard training that is always performed when the company is asked to develop a new ontology, be it conventional or fuzzy, as it facilitates the communication between the knowledge engineer that leads the effort and the domain experts. The first team was then given 6 h to develop the fuzzy ontology.

At the same time, the second team spent two additional hours working on the IKARUS-Onto methodology. The first 30 min approximately were dedicated to the presentation of the methodology to the knowledge engineer leading the team, while the remaining hour and a half was dedicated to introducing the methodology to all the members of the field. This introduction was not limited to a theoretical presentation but also included practical examples, similar to the ones listed in this paper, in order to facilitate understanding. The second team was then given 4 h to develop the ontology.

The second team concluded their work with an hour to spare, i.e., within 3 h. The first team was also ready to turn in their ontology at that time, but their leader clarified that if more time was available they would enhance their ontology (as in fact they did until the expiration of the time they were given) while if less time were available they could have submitted their ontology after just 4 h of work. As far as the overall time itself is concerned, the time-line of the experiment is summarized in Table 8.

7.3 Evaluation results

Figures 7, 8, 9 illustrate the responses to the questions described in the previous paragraph. Starting with the first question, namely the easiness of learning and applying the methodology, Fig. 7 shows that the answers provided by the members of the two teams trained in the methodology (the second developing team and the validation one) were either "average" or "high" while only one person (a domain expert) found the methodology difficult to grasp and

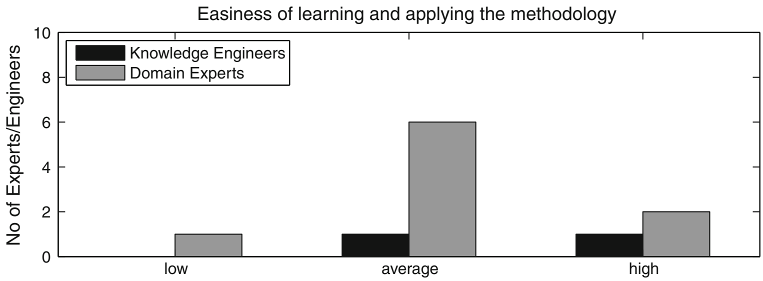


Fig. 7 Evaluation of the IKARUS-Onto learning & applying process

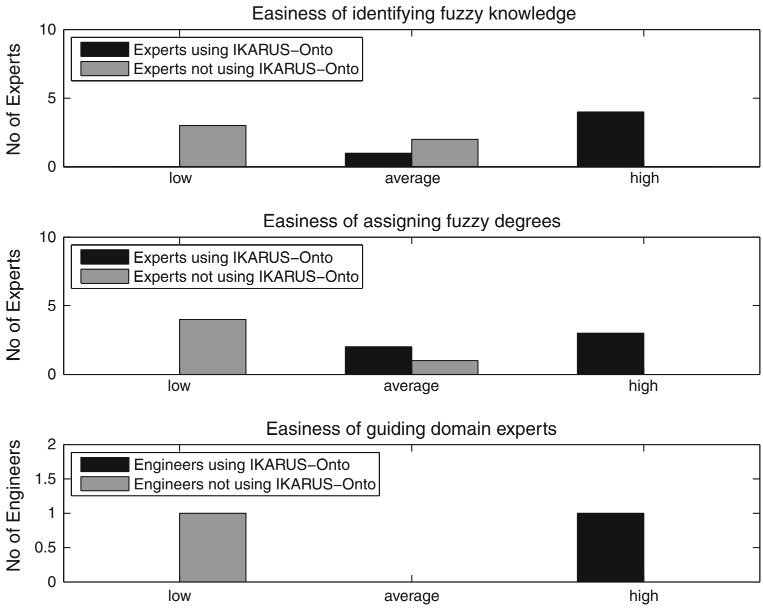


Fig. 8 Evaluation of IKARUS-Onto for the fuzzy ontology modeling process

apply. From the feedback we received from the people, two seemed to be the reasons why this has happened:

1. Because the methodology defines a pathway similar to the ones of crisp ontology development methods to which the engineers and some of the domain experts were already familiar.
2. Because the methodology focuses on the phenomenon of vagueness and its conceptual modeling rather than the complicated formalisms typically used for representing fuzzy ontologies.

Going deeper into the fuzzy ontology development process, Fig. 8 illustrates the two developing teams' responses to the questions regarding their experience with the identification and conceptual modeling of fuzzy ontology elements. As one may easily see, the members of the team that did not use IKARUS-Onto faced significantly more difficulties in coping with vagueness and fuzzy degrees than the team that did use our methodology. That is because, as the domain experts of the first team suggested, people disagreed too often on whether an

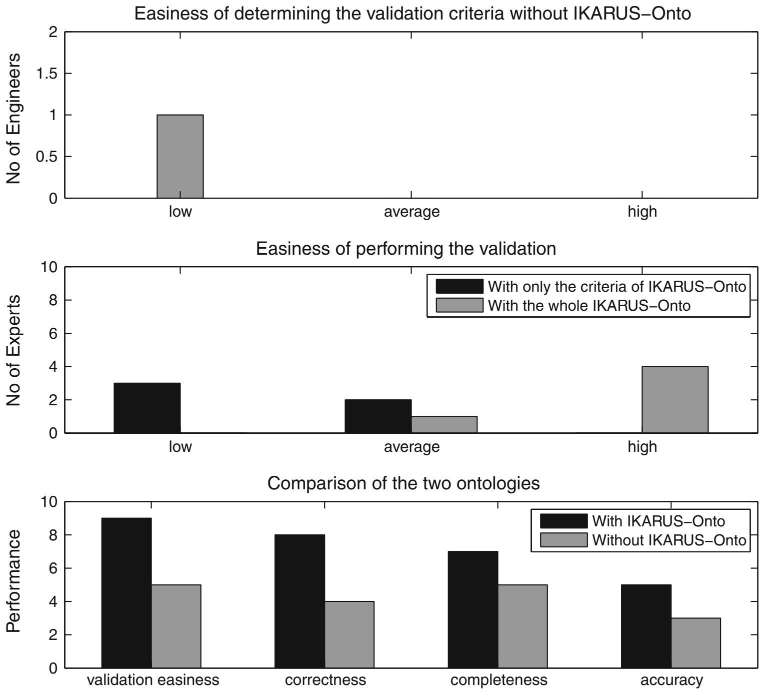


Fig. 9 Evaluation of IKARUS-Onto for the fuzzy ontology validation process and the quality of the final ontology

ontology element did actually convey vague meaning (especially for relations) and, moreover, they complained that it was quite difficult for them to assign appropriate degrees to fuzzy statements. Also, the knowledge engineer of this team suggested that it was quite difficult for him to orchestrate the whole modeling process and especially to explain the notion of fuzziness to the domain experts. On the other hand, the team that did use IKARUS-Onto faced also some of these difficulties but in a much lesser degree.

Figure 9 in turn shows the answers of the validation team regarding the efficiency of the validation process. As the first chart indicates, the knowledge engineer of the team had difficulty in determining the criteria he should use for performing the validation. As he said, the only criterion that he could naturally think of was the accuracy of the fuzzy degrees. Furthermore, even when the team learned the validation criteria of IKARUS-Onto, namely correctness, completeness, and accuracy, but not the rest of the methodology, they faced the same problems as the first developing team regarding the identification of vague meaning and the judgment of fuzzy degrees. The second chart of Fig. 9 shows the improvement that the learning and applying of the whole IKARUS-Onto methodology made to the validation process.

Finally, regarding the last two questions, the third chart in Fig. 9 shows that the ontology developed with IKARUS-Onto was easier to validate and performed better in all three validation criteria. The reason for the former, namely the easiness of the validation, was the fact that the elements of that ontology were accompanied by an explicit description of their vagueness meaning and of the interpretation of their fuzzy degrees. The reason for the latter was that the vagueness identification guidelines of IKARUS-Onto enabled the domain experts of the team to judge in a more effective way the correctness and completeness of the fuzzy ontology elements as well as the accuracy of their fuzzy degrees.

In overall, the evaluation results clearly indicate that, in the specific experiment, our proposed methodology succeeds in enhancing the effectiveness of the fuzzy ontology development process compared with the previous best practice in the area, namely not using any methodology but merely fuzzy ontology languages. Moreover, as the reasons for this enhancement have been well documented in the above, it is justified to expect that the methodology will have similar gains in other domains and application scenarios as well. Finally, given the answers of Fig. 7 about the easiness of learning and applying the methodology, it can be said that the enhancement the latter provides is important enough to justify the overhead of learning and applying.

Clearly, this has not been an extensive field study on the efficiency of the proposed methodological approach. Still, given the difficulty of the task (large groups of knowledge engineers with similar backgrounds who work on the same topic in parallel are difficult to find/construct) and the trend of the field (such evaluations are typically not performed at all), we are particularly content with the study reported above. Moreover, the answers provided by the knowledge engineers and domain experts of this in-house experiment have indicated a clear margin between those using the proposed methodology and those not using it. As a result, we can safely conclude that the utilization of the proposed approach offers a considerable advantage to the knowledge engineer who attempts to construct a fuzzy ontology.

7.4 Discussion

As we have already mentioned, one could of course argue that in our evaluation, we have compared the use of a methodological tool against the use of no tool whatsoever, so it is only natural that our approach is found to have outperformed the presented alternative. The true question is whether the overhead added by the proposed methodology is justified by an corresponding benefit or whether the benefit provided is trivial. In this sense as "cost" we can use the time and effort that is put into the process, while as benefit the quality of the produced fuzzy ontology. Therefore, in order to estimate the "cost" of the process, we need to take a closer look to the way in which it was implemented.

This way has already been presented in Sect. 7.2 where the details and the timeline of the experiment have been described. At a first glance, we can see that the overall time required to apply the conventional approach was 7 h, while the proposed approach needed 6 h. On the other hand, the minimum overall time required to apply the conventional approach was 5 h, which is one hour less than the proposed approach. We are convinced that the improvement of the quality of the resulting ontology justifies this overhead in time, but this is a qualitative and therefore subjective claim that is open to evaluation and discussion.

What is worth noting is that this is only true for the first time that our approach is applied by a group of knowledge engineers and domain experts. If the same group of people are asked to apply the same methodology again the 2 h of IKARUS-Onto training will not be required, which makes the proposed approach faster in any case. Even if a new topic is modeled, and therefore IKARUS-Onto training is required, using an experienced knowledge engineer reduces the training by half an hour which makes the difference between the proposed and the conventional approach negligible.

More importantly, with the proposed approach, we are sure at the end that the process has been completed and nothing remains to be done, while in the conventional approach if even more time was required it would be used. This was an anticipated advantage of our approach that has some very desired implications for organizations that develop ontologies professionally, such as being able to schedule and budget the relevant tasks.

8 Discussion

In the previous section, we demonstrated through an experiment that IKARUS-Onto manages to enhance the effectiveness of the fuzzy ontology development process and the quality of its output. The main reasons why IKARUS-Onto has achieved such an enhancement are the following:

- It focuses more on the knowledge that should be represented as fuzzy within the ontology rather than the exact way it should be represented. That enables developers to focus on the ontology content without worrying much about the formalism they are going to use in the end.
- It treats vagueness as a first-class citizen by describing its nature and the way it is typically manifested in a domain thus making sure that everybody involved within the ontology development process can identify vague knowledge in an easier, more complete and more accurate way.
- It requires from the developers to state (and record) explicitly the exact meaning of the fuzzy ontology elements they define. This has two benefits: firstly the communication between those involved in the development process becomes more effective and secondly the ontology becomes more shareable and reusable as people that haven't been involved in the development process may easily understand through the ontology's documentation the meaning of its fuzzy elements.
- It does not dismiss the practices and methods of crisp ontology engineering but it builds upon them by covering and tackling the issues that arise from the existence of vagueness. This enables people who are already familiar with ontology engineering to move on to the development of fuzzy ontologies in a quicker and smoother way.
- It uses as a starting point already developed crisp ontologies. This enables the reuse of knowledge (as in many domains there are already developed crisp ontologies) and makes more effective the modeling process (as it is easier for people to work on already structured knowledge rather than unstructured one).

9 Conclusions and future work

Fuzzy Ontologies comprise a relatively new knowledge representation paradigm that is being increasingly applied in application scenarios where the treatment and utilization of vague knowledge are important. Such scenarios appear in a number of application areas, such as information retrieval or decision support, and their successful tackling by respective knowledge systems is highly dependent on the quality of the fuzzy ontology that models the domain's vague knowledge. In order to ensure that this quality is as high as possible we proposed in this paper IKARUS-Onto, a novel fuzzy ontology development methodology that manages to significantly enhance the effectiveness of the development process and the quality of the latter's output, namely the fuzzy domain ontology.

To establish the importance and effectiveness of the above features, we conducted an experiment in which we tried to qualitatively measure the degree of enhancement that the methodology achieves in the effectiveness of the fuzzy ontology development process. The results of this experiment (Sect. 7) verified our initial claims.

The existence of a detailed and specific methodology that facilitates and standardizes the development of fuzzy ontologies has great implications for the field. First of all, the structure of the methodology is such that allows for the exploitation of existing conventional ontologies in the development of their fuzzy counterparts. Secondly, the standardization of the process makes the produced ontologies easily reusable, which in turn allows for the incremental development of larger fuzzy ontology bases that may serve as generic reference works. Finally, the detachment of the process from the formalization technicalities allows for the true domain experts to be more directly involved in the process, which allows for better ontologies to be developed and for a larger range of topics, as until now detailed ontologies could be developed only for topics in which the knowledge engineers are in part also the domain experts.

In our future work we intend to continue working on the direction of making the development of fuzzy ontologies an increasingly easier task in two ways:

- By developing a graphical tool that may support the engineers and the domain experts in performing all the tasks of the methodology.
- By developing vagueness-specific knowledge elicitation techniques that may effectively enable the acquisition of vague ontological knowledge.

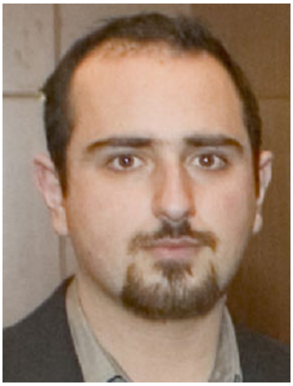
References

1. Agarwal S, Lamparter S (2005) A semantic matchmaking portal for electronic markets. In: Proceedings of the seventh IEEE international conference on E-commerce technology, 19–22 July 2005
2. Alexopoulos P, Wallace M, Kafentzis K, Thomopoulos (2009) A fuzzy knowledge-based decision support system for tender call evaluation. In: Proceedings of the 5th IFIP conference on artificial intelligence applications and innovations (AIAI 2009)
3. Alexopoulos P, Wallace M, Kafentzis K, Askounis D (2010) Utilizing imprecise knowledge in ontology-based CBR systems through fuzzy algebra. Int J Fuzzy Syst 12(1):1–14
4. Baeza-Yates R, Tiberi A (2007) Extracting semantic relations from query logs. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '07). ACM, New York, pp 76–85
5. Bechhofer S, van Harmelen F, Hendler J, Horrocks I, McGuinness D, Patel Schneider P, Stein LA (2004) OWL web ontology language reference. W3C recommendation, 10 Feb 2004
6. Bobillo F, Delgado M, Gomez-Romero J (2008) DeLorean: a reasoner for fuzzy OWL 1.1. In: Proceedings of the 4th international workshop on uncertainty reasoning for the semantic web (URSW 2008). CEUR workshop proceedings 423. Karlsruhe, Oct 2008
7. Bobillo F, Straccia U (2008) fuzzyDL: an expressive fuzzy description logic reasoner. In: Proceedings of the 2008 international conference on fuzzy systems
8. Bouamrane M-M, Rector A, Hurrell M (2010) Using OWL ontologies for adaptive patient information modeling and preoperative clinical decision support. Knowledge and information systems. Springer, London, p 114
9. Calegari S, Sanchez E (2007) A fuzzy ontology-approach to improve semantic information retrieval. In: Bobillo F, da Costa PCG, D'Amato C, Fanizzi N, Fung F, Lukasiewicz T, Martin T, Nickles M, Peng Y, Pool M, Smrz P, Vojtas P (eds) Proceedings of the third ISWC workshop on uncertainty reasoning for the semantic web—URSW'07, vol 327
10. Cimiano P (2006) Ontology learning and population from text: algorithms, evaluation and applications. Springer-Verlag, New York, Inc., Secaucus
11. Chandrasekaran B, Josephson JR, Benjamins VR (1999) What are ontologies, and why do we need them. IEEE Intell Syst 14(1):20–26
12. Chen W, Yang Q, Zhu L, Wen B (2009) Research on automatic fuzzy ontology generation from fuzzy context. In: Proceedings of the 2009 second international conference on intelligent computation technology and automation—volume 02 (ICICTA '09), vol 2. IEEE Computer Society, Washington, pp 764–767
13. Chen CL, Tseng F, Liang T (2010) An integration of fuzzy association rules and WordNet for document clustering. Knowledge and information systems. Springer, London, p 122

14. Escovar ELG, Yaguinima CA, Biajic M (2006) Using fuzzy ontologies to extend semantically similar data mining. In: XXI Simposio Brasileiro de Banco de Dados (SBBD), Florianopolis, p 16–30
15. Fernandez Lopez M, Gomez Perez A, Juristo N (1997) Methontology: from ontological art towards ontological engineering. In: Spring symposium on ontological engineering of AAAI, pp 33–40
16. Fodeh S, Punch B, Tan, P-N (2011) On ontology-driven document clustering using core semantic features. Knowledge and information systems. Springer, London, p 127
17. Gomez-Perez A, Corcho O, Fernandez-Lopez M (2004) Ontological engineering. Springer-Verlag London Limited
18. Gruninger M, Fox MS (1995) Methodology for the design and evaluation of ontologies. In: IJCAI95 on workshop basic ontological issues in knowledge sharing
19. Jarrar M, Meersman R (2008) Ontology engineering—the DOGMA approach. In: Advances in web semantics, vol I, LNCS 4891. Springer, Berlin
20. Jurisica I, Mylopoulos J, Yu E (2004) Ontologies for knowledge management: an information systems perspective. Knowledge and information systems, vol 6, no. 4. Springer, London, pp 380–401
21. Hyde D (2008) Vagueness, logic and ontology. Ashgate new critical thinking in philosophy
22. Klir G, Yuan B (1995) Fuzzy sets and fuzzy logic, theory and applications. Prentice Hall, Upper Saddle River
23. Kotis K, Vouros G (2006) Human-centered ontology engineering: the HCOME methodology. Knowledge and information systems (KAIS), vol 10. Springer, London, pp 109–131
24. Lau RY, Song D, Li Y, Cheung TCH, Hao JX (2009) Toward a fuzzy domain ontology extraction method for adaptive e-learning. IEEE Trans Knowl Data Eng 21(6):800–813
25. Lee CS, Jian ZW, Huang LK (2005) A fuzzy ontology and its application to news summarization. IEEE Trans Syst Man Cybern B 35(5):859–880
26. Liu K, Fang B, Zhang W (2010) Ontology emergence from folksonomies. In: Proceedings of the 19th ACM international conference on information and knowledge management (CIKM '10). ACM, New York
27. Mailis T, Stoilos G, Stamou G (2010) Expressive reasoning with horn rules and fuzzy description logics. Knowledge and information systems, vol 25, no. 1. Springer, London, pp 105–136
28. McGee V, McLaughlin B (1994) Distinctions without a difference. South J Philos 33(suppl):203–251
29. Parry D (2004) A fuzzy ontology for medical document retrieval. In: Hogan J, Montague P, Purvis M, Steketee C (eds) Proceedings of the second workshop on Australasian information security, data mining and web intelligence, and software internationalisation—vol 32. Dunedin. ACM international conference proceeding series, vol 54. Australian Computer Society, Darlinghurst, pp 121–126
30. Pinto S, Martins J (2004) Ontologies: how can they be built? Knowl Inf Syst 6:441–464
31. Ramezani M, Witschel HF, Braun S, Zacharias V (2010) Using machine learning to support continuous ontology development. In: Cimiano P, Sofia Pinto H (eds) Proceedings of the 17th international conference on Knowledge engineering and management by the masses (EKAW'10). Springer, Berlin, pp 381–390
32. Reichartz F, Korte H, Paass G (2010) Semantic relation extraction with kernels over typed dependency trees. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '10). ACM, New York, pp 773–782
33. Sanchez D, Isern D, Millan M (2010) Content annotation for the semantic web: an automatic web-based approach. Knowledge and information systems. Springer, London, p 126
34. Shapiro S (2006) Vagueness in context. Oxford University Press, USA
35. Simou N, Kollias S (2007) FiRE: a fuzzy reasoning engine for impecise knowledge. K-space PhD students workshop. Berlin, 14 Sept 2007
36. Stoilos G, Straccia U, Stamou G, Pan JZ (2006) General concept inclusions in fuzzy description logics. In: 17th European conference on artificial intelligence (ECAI 06). Riva del Garda
37. Stoilos G, Stamou G, Pan JZ, Simou N, Tzouvaras V (2008) Reasoning with the fuzzy description logic f-SHIN: theory, practice and applications. In: Costa PCG, d'Amato C et al (eds) Uncertainty reasoning for the semantic web I
38. Szumlanski S, Gomez F (2010) Automatically acquiring a semantic network of related concepts. In: Proceedings of the 19th ACM international conference on information and knowledge management (CIKM-10). Toronto, pp 19–28
39. Tho QT, Hui SC, Fong ACM, Cao TH (2006) Automatic fuzzy ontology generation for semantic web. IEEE Trans Knowl Data Eng 18(6):842–856
40. Thomas C, Sheth A (2006) On the expressiveness of the languages for the semantic web—making a case for 'A little more'. In: Sanchez E (ed) Fuzzy logic and the semantic web. Elsevier, Amsterdam
41. Uschold M, King M (1995) Towards a methodology for building ontologies. In: IJCAI95 workshop on basic ontological issues in knowledge sharing, pp 6.1–6.10

42. Vrandecic D, Pinto HS, Sure Y, Tempich C (2005) The DILIGENT knowledge processes. J Knowl Manag 9(5):85–96
43. Wallace M, Mylonas P, Akrivas G, Avrithis Y, Kollias S (2006) Automatic thematic categorization of multimedia documents using ontological information and fuzzy algebra. In: Ma Z (ed) Studies in fuzziness and soft computing, soft computing in ontologies and semantic web, vol 204. Springer, Berlin
44. Zadeh LA (2003) From search engines to question-answering systems the need for new tools. In: The 12th IEEE international conference on fuzzy systems 2003, vol 2, pp 1107–1109
45. Zhai J, Liang Y, Jiang J, Yu Y (2008) Fuzzy ontology models based on fuzzy linguistic variable for knowledge management and information retrieval. In: Intelligent information processing IV, pp 58–67
46. Zhang F, Ma ZM, Fan G, Wang X (2010) Automatic fuzzy semantic web ontology learning from fuzzy object-oriented database model. In: Bringas PG, Hameurlain A, Quirchmayr G (eds) Proceedings of the 21st international conference on database and expert systems applications: part I (DEXA'10). Springer, Berlin, pp 16–30
47. Zhang F, Ma ZM, Cheng J, Meng X (2009) Fuzzy semantic web ontology learning from fuzzy UML model. In: Proceeding of the 18th ACM conference on Information and knowledge management (CIKM '09). ACM, New York
48. Zhou L (2007) Ontology learning: state of the art and open issues. Inf Technol Manag 8(3):241–252

Author Biographies



Panos Alexopoulos received his degree in Electrical and Computer Engineering from the National Technical University of Athens in Greece in 2004 and the M.Sc. in Advanced Computing from Imperial College London in 2005. Currently he is pursuing a Ph.D. degree in Knowledge Engineering & Management in National Technical University of Athens while working as a knowledge engineer and researcher for IMC Technologies (Athens, Greece). His research interests include ontology engineering, semantic technologies, fuzzy systems and knowledge management.

Manolis Wallace acquired his degree in Electrical and Computer Engineering from the National Technical University of Athens in Greece in 2001 and Ph.D. in Intelligent Knowledge-based Systems in Uncertain Environments from the same university in 2005. Dr. Wallace, formerly assistant professor and chair of the department of Computer Science at the University of Indianapolis, Athens Campus, currently holds a visiting lecturer position at the University of Peloponnese, while at the same time serving as a project coordinator for the Foundation of the Hellenic World. He is a recipient of a 2002–2005 HERACLETUS national fund for basic research. His research interests evolve around semantics, data mining, fuzzy systems, uncertainty and practical applications of the above. Dr. Wallace is an associate editor of the International Journal of Fuzzy Systems and the co-founder and co-chair of the Semantic Media Adaptation (SMAP) Initiative.

Konstantinos Kafentzis acquired his degree in Electrical and Computer Engineering from the National Technical University of Athens in Greece in 2001. Currently, he is Managing Director of IMC Technologies S.A. (Athens, Greece) while in the past he has worked as a researcher of ICCS-NTUA and as an R&D Director in IMC Technologies, participating and leading several research projects in the areas of knowledge management, eParticipation and semantic technologies.

Dimitris Askounis is Associate Professor in the School of Electrical and Computer Engineering of the National Technical University of Athens (NTUA), with a Ph.D. degree in Information Technology (IT). He has over 20 years of experience in the areas of Decision Support Systems, Management Information Systems, Intelligent Information Systems, e-business and e-government technologies, as well as in the application of state of the art IT tools in operation management and energy systems. He has been involved in numerous projects and activities in CEEC, NIS, and MEDA countries concerning monitoring and evaluation of large projects, management training, development of computerized information systems, etc. He has more than 100 publications and several announcements and presentations in national and international conferences and workshops.