

Towards a Semantic Portal for Oncology using a Description Logic with Fuzzy Concrete Domains

Mathieu d'Aquin, Jean Lieber, Amedeo Napoli

► **To cite this version:**

Mathieu d'Aquin, Jean Lieber, Amedeo Napoli. Towards a Semantic Portal for Oncology using a Description Logic with Fuzzy Concrete Domains. Elie Sanchez. Fuzzy Logic and the Semantic Web, 1 (1), Elsevier, 2006, Capturing intelligence, 0-444-51948-3. inria-00001254

HAL Id: inria-00001254

<https://hal.inria.fr/inria-00001254>

Submitted on 13 Apr 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards a Semantic Portal for Oncology using a Description Logic with Fuzzy Concrete Domains

Mathieu d'Aquin, Jean Lieber, Amedeo Napoli

*Orpailleur research group, LORIA, (CNRS, INRIA, Nancy Universities),
BP 239, 54506 Vandœuvre-lès-Nancy, France,
lieber@loria.fr, daquin@loria.fr, napoli@loria.fr*

Abstract

This paper presents three systems that are fully implemented and a proposal for a fourth one. KASIMIR is a knowledge based-system using an ad hoc formalism similar to a simple description logic with concrete domains which is used for representing decision protocols in oncology. FUZZY-KASIMIR is an extension of KASIMIR with fuzzy concrete domains taking into account discontinuities in the decision that are due to numerical thresholds. Another extension of KASIMIR has led to embed it into a semantic portal for oncology, which has been motivated by the need to share knowledge for geographically distributed physicians and has led to change the ad hoc formalism to the standard OWL DL. A combination of these two extensions of KASIMIR is currently under implementation and will lead to a semantic portal for oncology with fuzzy datatypes.

Key words: fuzzy description logics, semantic portal for oncology, OWL DL, fuzzy datatypes

1 Introduction

The KASIMIR project studies knowledge management and decision support in oncology [4]. It has led to the implementation of the KASIMIR system in which various medical decision protocols have been represented. The most complex protocol represented so far in the KASIMIR system is the one for breast cancer treatment. Besides protocol representation, tools for edition, visualization and maintenance of the knowledge contained in protocols have been implemented in the KASIMIR system [3]. KASIMIR uses an ad hoc representation formalism close to object-based representation formalisms (OBRFs [17]) and to description logics (DLs [1]).

Two further versions of KASIMIR have been realized in parallel. The first one is FUZZY-KASIMIR: a fuzzy extension of the formalism and the inferences of KASIMIR, that can be considered as a fuzzy DL. The second one relies on semantic

Web principles and technologies. KASIMIR is embedded into a semantic portal, dedicated to the diffusion of knowledge and services for oncology. Particularly, protocols are now represented and shared thanks to the OWL DL formalism, and decision support relies on standard DL reasoning within this language.

Section 2 shows how medical decision protocols are expressed in KASIMIR and why this classical representation may be insufficient because of a threshold problem in concrete domains. Section 3 presents the FUZZY-KASIMIR system, that has been implemented in order to solve this problem. The current semantic portal for oncology has been developed using the OWL DL formalism which does not integrate fuzzy notions; section 4 describes the main features of this portal and explains how it is planned to be extended in order to have a semantic portal for fuzzy decision protocols in oncology. Section 5 discusses our approach and compares it to related work, in particular to fuzzy DLs and similar formalisms; it discusses also how fuzziness can be introduced in OWL DL, taking into account the semantic Web origin of this formalism.

2 Decision Protocol Representation in KASIMIR

KASIMIR uses an ad hoc formalism for representing decision protocols, that is inspired by OBRFs and DLs, but for the sake of simplicity, the syntax of the $\mathcal{ALC}(\mathcal{D})$ formalism [1] is used in this section.

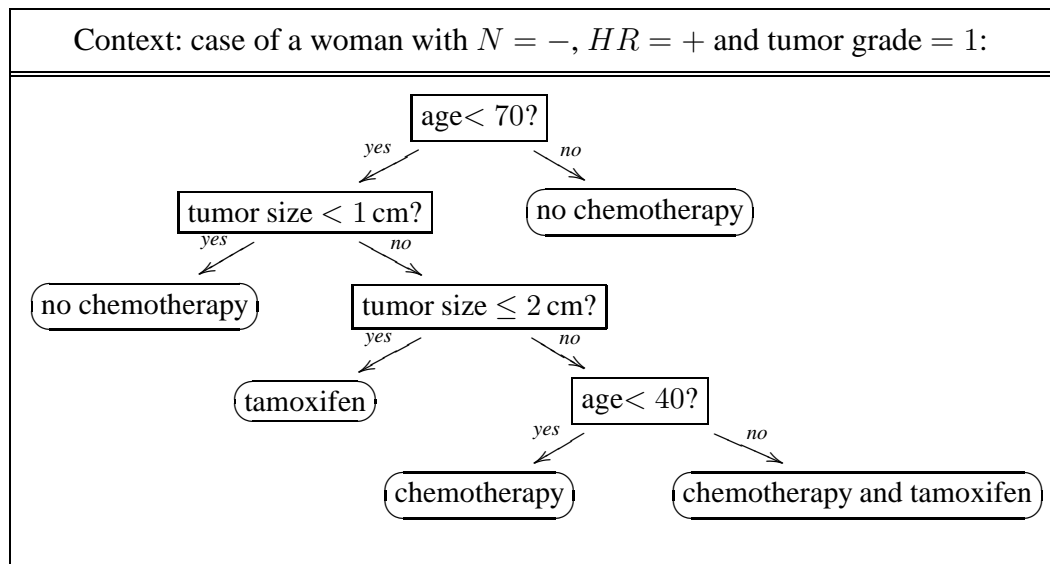


Fig. 1. A protocol represented by a decision tree (extracted and adapted from the decision protocol of breast cancer treatment described in www.oncolor.org/).

Let us consider the protocol of figure 1. It has been represented by the knowledge base of figure 2. The C_i 's are *concepts* representing sets of individuals. For instance, C_0 represents the women of figure 1's context and C_1 represents, among

[A0] $C_0 \equiv \exists \text{sex.female} \sqcap \exists \text{N.false} \sqcap \exists \text{RH.true} \sqcap \exists \text{tumor}.\exists \text{grade.} =_1$	
[A1] $C_1 \equiv C_0 \sqcap \exists \text{age.} <_{70}$	[A9] $C_2 \sqsubseteq \exists \text{reco.no-chemo}$
[A2] $C_2 \equiv C_0 \sqcap \exists \text{age.} \geq_{70}$	[A10] $C_3 \sqsubseteq \exists \text{reco.no-chemo}$
[A3] $C_3 \equiv C_1 \sqcap \exists \text{tumor}.\exists \text{size.} <_{1.0}$	[A11] $C_5 \sqsubseteq \exists \text{reco.tamoxifen}$
[A4] $C_4 \equiv C_1 \sqcap \exists \text{tumor}.\exists \text{size.} \geq_{1.0}$	[A12] $C_7 \sqsubseteq \exists \text{reco.chemotherapy}$
[A5] $C_5 \equiv C_4 \sqcap \exists \text{tumor}.\exists \text{size.} \leq_{2.0}$	[A13] $C_8 \sqsubseteq \exists \text{reco.chemo+tam}$
[A6] $C_6 \equiv C_4 \sqcap \exists \text{tumor}.\exists \text{size.} >_{2.0}$	
[A7] $C_7 \equiv C_6 \sqcap \exists \text{age.} <_{40}$	
[A8] $C_8 \equiv C_6 \sqcap \exists \text{age.} \geq_{40}$	

Fig. 2. A knowledge base representing the protocol of figure 1.

these women, the ones whose age is lower than 70. The axioms [A0] to [A8] define the concepts of patients C_0 to C_8 by necessary and sufficient conditions. The axioms [A9] to [A13] relate the concepts of patients to concepts of treatments: an axiom $\text{Pat} \sqsubseteq \exists \text{reco.Ttt}$ means that for each patient of the set of individuals represented by Pat , there is a recommended treatment in the set of treatments represented by Ttt . For example, [A11] means that for the women represented by the class C_5 , a treatment by tamoxifen (a hormonotherapy drug) is recommended by the protocol. Let KB be a knowledge base: KB is a set of axioms.

A concept C represents a set and a role r represents a binary relation. An interpretation \mathcal{I} is a pair $(\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta_{\mathcal{I}}$ is a non-empty set and $\cdot^{\mathcal{I}}$ maps a concept C (resp., a role r) to a subset $C^{\mathcal{I}}$ of $\Delta_{\mathcal{I}}$ (resp., to a subset $r^{\mathcal{I}}$ of $\Delta_{\mathcal{I}} \times \Delta_{\mathcal{I}}$). A concept is either atomic (a concept name) or defined by an expression. Given two concepts C and D and a role r , $C \sqcap D$ (conjunction) and $\exists r.C$ (role restriction) are two new concepts such that, for every interpretation \mathcal{I} , $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$ and $(\exists r.C)^{\mathcal{I}}$ is the set of the $x \in \Delta_{\mathcal{I}}$ for which exists $y \in C^{\mathcal{I}}$ with $(x, y) \in r^{\mathcal{I}}$. The third KASIMIR constructor is concrete feature restriction and is based on concrete domains.

The definitions below are inspired from [15]. A *concrete domain* D is a pair (Δ_D, Φ_D) where Δ_D is a set and Φ_D is a set of predicate names on Δ_D . In the following, only unary predicates are considered, i.e., $\varphi \in \Phi_D$ is interpreted as a subset of Δ_D denoted by φ^D . Two concrete domains are represented in KASIMIR: $\mathbb{R} = (\Delta_{\mathbb{R}}, \Phi_{\mathbb{R}})$ and $\mathbb{Z} = (\Delta_{\mathbb{Z}}, \Phi_{\mathbb{Z}})$.

$\Delta_{\mathbb{R}} = \mathbb{R}$ is the set of real numbers and $\Phi_{\mathbb{R}}$ is the set of the predicate names $\top_{\mathbb{R}}$, $\perp_{\mathbb{R}}$ and P_x where $P \in \{<, \leq, \geq, >\}$ and $x \in \Delta_{\mathbb{R}}$. $\top_{\mathbb{R}}$ is a name for $\Delta_{\mathbb{R}}$ and $\perp_{\mathbb{R}}$, a name for \emptyset : $\top_{\mathbb{R}}^{\mathbb{R}} = \Delta_{\mathbb{R}}$, $\perp_{\mathbb{R}}^{\mathbb{R}} = \emptyset$. P_x represents the set $P_x^{\mathbb{R}}$ of $t \in \Delta_{\mathbb{R}}$ such that $t P x$. For example, $>_1^{\mathbb{R}} =]1.; +\infty[$. More generally, $P_x^{\mathbb{R}}$ is an interval with one infinite bound.

$\Delta_{\mathbb{Z}} = \mathbb{Z}$ is the set of integers and $\Phi_{\mathbb{Z}}$ is the set of the predicate names $\top_{\mathbb{Z}}$, $\perp_{\mathbb{Z}}$ and P_x where $P \in \{<, \leq, \geq, >\}$ and $x \in \Delta_{\mathbb{R}}$. The interpretation of these predicates are $\top_{\mathbb{Z}}^{\mathbb{Z}} = \Delta_{\mathbb{Z}}$, $\perp_{\mathbb{Z}}^{\mathbb{Z}} = \emptyset$ and $P_x^{\mathbb{Z}} = \{n \in \Delta_{\mathbb{Z}} \mid n P x\}$ (e.g., $\geq_3^{\mathbb{Z}} = \{3, 4, 5, \dots\}$).

Both \mathbf{R} and \mathbf{Z} are *admissible*, that is, for $\mathbf{D} \in \{\mathbf{R}, \mathbf{Z}\}$: (1) $\Phi_{\mathbf{D}}$ is closed under negation¹ and contains a name $\top_{\mathbf{D}}$ for $\Delta_{\mathbf{D}}$; (2) the satisfiability of any conjunction of predicates from $\Phi_{\mathbf{D}}$ is decidable.

Let $\mathbf{D} = (\Delta_{\mathbf{D}}, \Phi_{\mathbf{D}})$ be a concrete domain, $\varphi \in \Phi_{\mathbf{D}}$ and g be a \mathbf{D} -concrete feature, i.e., a name interpreted by a mapping $g^{\mathcal{I}}$ from $\Delta_{\mathcal{I}}$ to $\Delta_{\mathbf{D}}$. The concept defined by $\exists g.\varphi$ represents the set of individuals $x \in \Delta_{\mathcal{I}}$ such that $g^{\mathcal{I}}(x) \in \varphi^{\mathbf{D}}$. For example, if *age* is a \mathbf{Z} -concrete feature, *size* is a \mathbf{R} -concrete feature and *tumour* is a role, then $\exists \text{age}.\gt_{70}$ represents the set of individuals older than seventy and $\exists \text{tumour}.\exists \text{size}.\leq_2$ represents the set of individuals whose tumor size s is such that $s \leq 2$ cm.

Concepts of the form $\exists g. =_x$ are also used in the examples of this paper (for example, $\exists \text{grade}.\ =_1$ in the figure 2). In fact, it is not built on other predicate names, but it is the abbreviation of $\exists g.\geq_x \sqcap \exists g.\leq_x$.

Let $\mathcal{I} = (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$, an interpretation. \mathcal{I} satisfies an axiom $\mathbf{C} \equiv \mathbf{D}$ if $\mathbf{C}^{\mathcal{I}} = \mathbf{D}^{\mathcal{I}}$. \mathcal{I} satisfies an axiom $\mathbf{C} \sqsubseteq \mathbf{D}$ if $\mathbf{C}^{\mathcal{I}} \subseteq \mathbf{D}^{\mathcal{I}}$. A model of KB is an interpretation satisfying all its axioms.

The main inferences implemented in KASIMIR are the subsumption test and the classification. Given two concepts \mathbf{C} and \mathbf{D} , \mathbf{C} is subsumed by \mathbf{D} —denoted by $\mathbf{C} \sqsubseteq \mathbf{D}$ —if, for every model \mathcal{I} of the knowledge base, $\mathbf{C}^{\mathcal{I}} \subseteq \mathbf{D}^{\mathcal{I}}$. Given a knowledge base KB and a concept \mathbf{C} , classification highlights the *subsumers* of \mathbf{C} , i.e., the concepts \mathbf{D} of KB such that $\mathbf{C} \sqsubseteq \mathbf{D}$.

For example, let us consider the knowledge base of figure 2 and a 39 year old woman with $N = -$, $HR = +$, a tumor grade of 1 and a tumor size of 3 cm. This medical case can be modeled by a concept MC (there is no mechanism for managing instances in KASIMIR):

$$\text{MC} = \mathbf{C}_0 \sqcap \exists \text{age}.\ =_{39} \sqcap \exists \text{tumour}.\exists \text{size}.\ =_3.$$

The classification points out the concepts more general than MC in the knowledge base. Among them, there is the concept $\exists \text{reco}.\text{chemotherapy}$ (cf. axioms [A0, 1, 4, 6, 7 and 12]). Thus, a chemotherapy is proposed for this patient.

Now, it can be remarked that the age 39 is close to the decision threshold 40 of the protocol. This raises a problem if the choice of this threshold is not assumed to be very accurate or if other factors, not expressed in the protocol, can be taken into account (e.g., another health problem). Indeed, if the woman was just one year older, this would lead to a different proposition of treatment (*chemo+tam*, cf. axioms [A0, 1, 4, 6, 8 and 13]). Therefore, it seems reasonable that the KASIMIR system proposes *both* treatments *chemotherapy* and *chemo+tam* (the latter is a treat-

¹ I.e., for each $\varphi \in \Phi_{\mathbf{D}}$, exists $\psi \in \Phi_{\mathbf{D}}$ such that $\psi^{\mathbf{D}} = \Delta_{\mathbf{D}} \setminus \varphi^{\mathbf{D}}$.

ment with a cure of chemotherapy and a cure of tamoxifen). The KASIMIR user, a physician, can use these propositions to build his/her own proposition that can be a choice between them or a combination of them (it may be the same chemotherapy and another hormonotherapy, with a lower dose of tamoxifen). To implement this mechanism, (classical) thresholds have been replaced by fuzzy thresholds; the extension of KASIMIR for managing them is presented in the next section.

3 FUZZY-KASIMIR: Formalism and Inferences

This section presents how fuzziness is introduced into the formalism of KASIMIR. First, some notions about fuzzy sets are reminded (for more information about fuzzy sets, see [7]).

Given a set X , called the *universe*, a fuzzy subset of X is a function $A : x \in X \mapsto A(x) \in [0; 1]$ (for the sake of simplicity, a fuzzy set and its membership function are not distinguished). Let A and B be two fuzzy subsets of X . A is a subset of B –denoted by $A \subseteq B$ – if, for every $x \in X$, $A(x) \leq B(x)$. $A \cap B$ is the fuzzy subset of X such that $(A \cap B)(x) = \min(A(x), B(x))$, for $x \in X$. Fuzzy logics are extensions of classical logic such that the truth values are elements of $[0; 1]$ (0 stands for “false” and 1, for “true”) and such that the logical connectors are extended on $[0; 1]$. We use in the following the Zadeh fuzzy logic for which, in particular, \wedge is fuzzified in \min , \vee in \max and \neg in $n : x \mapsto 1 - x$. The implication \Rightarrow can be fuzzified in different ways by a function $F_{\Rightarrow} : (x, y) \in [0; 1]^2 \mapsto F_{\Rightarrow}(x, y)$. In the following, the Łukasiewicz implication defined by $F_{\Rightarrow}(x, y) = \min(1, 1 - x + y)$ is used (we could also have chosen the Gödel implication as in [5]). Based on F_{\Rightarrow} , the gradual inclusion between two fuzzy subsets A and B of X can be defined by $F_{\subseteq}(A, B) = \inf_{x \in X} F_{\Rightarrow}(A(x), B(x))$. Note that $F_{\subseteq}(A, B) = 1$ iff $A \subseteq B$. Given a fuzzy subset A of X and $\alpha \in]0; 1]$, A_{α} , the α -cut of A , is a classical set such that $x \in A_{\alpha}$ iff $A(x) \geq \alpha$. If $\alpha = 1$, A_{α} is called the *core* of A . The *support* of A is $\{x \in X \mid A(x) > 0\}$.

The formalism of FUZZY-KASIMIR extends the one of KASIMIR by the addition of new predicate names in the concrete domains $\mathbf{D} \in \{\mathbf{R}, \mathbf{Z}\}$. Such a new predicate name φ is interpreted as a fuzzy unary predicate $\varphi^{\mathbf{D}}$, i.e., as a fuzzy subset of $\Delta_{\mathbf{D}}$.

The new predicate names of the concrete domain \mathbf{R} are $\leq_{a \pm b}$ and $\geq_{a \pm b}$ where $a, b \in \Delta_{\mathbf{R}}$ and $b > 0$. They are interpreted by:

$$\text{for } x \in \Delta_{\mathbf{R}}, \quad \leq_{a \pm b}^{\mathbf{R}}(x) = \begin{cases} 1 & \text{if } x \leq a - b \\ \frac{a + b - x}{2b} & \text{if } x \in [a - b; a + b] \\ 0 & \text{if } x \geq a + b \end{cases}$$

$$\geq_{a \pm b}^{\mathbf{R}}(x) = n(\leq_{a \pm b}^{\mathbf{R}}(x)) = 1 - \leq_{a \pm b}^{\mathbf{R}}(x)$$

The new predicate names of \mathbf{Z} are $\leq_{a\pm b}$ and $\geq_{a\pm b}$ with $a, b \in \Delta_{\mathbf{Z}}$ and $b > 0$. The interpretations of these predicate names are the same as for the concrete domain \mathbf{R} , except that the universe is $\Delta_{\mathbf{Z}}$.

\mathbf{R} and \mathbf{Z} remain admissible with the addition of these new predicate names, if the conjunction and negation used in the definition of admissibility are \min and n .

The semantics of a FUZZY-KASIMIR concept \mathbf{C} is, for an interpretation \mathcal{I} , a fuzzy subset $\mathbf{C}^{\mathcal{I}}$ of $\Delta_{\mathcal{I}}$. Given two concepts \mathbf{C} and \mathbf{D} , a role r , a concrete domain $\mathbf{D} = (\Delta_{\mathbf{D}}, \Phi_{\mathbf{D}})$, a \mathbf{D} -concrete feature g and $\varphi \in \Phi_{\mathbf{D}}$, $\cdot^{\mathcal{I}}$ maps the concept $\mathbf{C} \sqcap \mathbf{D}$ into the fuzzy set $\mathbf{C}^{\mathcal{I}} \cap \mathbf{D}^{\mathcal{I}}$, the concept $\exists r.\mathbf{C}$ into the fuzzy set

$$(\exists r.\mathbf{C})^{\mathcal{I}} : x \in \Delta_{\mathcal{I}} \mapsto \sup_{x \in \Delta_{\mathcal{I}}} \min(r^{\mathcal{I}}(x, y), \mathbf{C}^{\mathcal{I}}(y)) \in [0; 1]$$

and the concept $\exists g.\varphi$ into the fuzzy set

$$(\exists g.\mathbf{C})^{\mathcal{I}} : x \in \Delta_{\mathcal{I}} \mapsto \varphi^{\mathcal{D}}(g^{\mathcal{I}}(x)) \in [0; 1]$$

The subsumption between concepts of FUZZY-KASIMIR can be defined as in KASIMIR: $\mathbf{C} \sqsubseteq \mathbf{D}$ if for every model of KB, $\mathbf{C}^{\mathcal{I}} \subseteq \mathbf{D}^{\mathcal{I}}$, the only difference is that here, \subseteq stands for inclusion between fuzzy sets. Therefore, the hierarchical classification can be still applied. Another inference implemented in FUZZY-KASIMIR is the *degree of subsumption* between two concepts \mathbf{C} and \mathbf{D} :

$$F_{\sqsubseteq}(\mathbf{C}, \mathbf{D}) = \inf\{F_{\subseteq}(\mathbf{C}^{\mathcal{I}}, \mathbf{D}^{\mathcal{I}}) \mid \mathcal{I}: \text{model of KB}\}$$

It must be noticed that $F_{\sqsubseteq}(\mathbf{C}, \mathbf{D}) = 1$ iff $\mathbf{C} \sqsubseteq \mathbf{D}$. The fuzzy hierarchical classification highlights the concepts \mathbf{D} of KB such that $F_{\sqsubseteq}(\mathbf{C}, \mathbf{D}) > 0$. An algorithm for fuzzy hierarchical classification is presented in [14].

For instance, let KB be the knowledge base obtained by substituting in the knowledge base of figure 2, the axioms [A6], [A7] and [A8], respectively by

- [A6'] $\mathbf{C}_6 \equiv \mathbf{C}_4 \sqcap \exists \text{tumor}.\exists \text{size}.\geq_{2.\pm 1}$.
- [A7'] $\mathbf{C}_7 \equiv \mathbf{C}_6 \sqcap \exists \text{age}.\leq_{40\pm 5}$
- [A8'] $\mathbf{C}_8 \equiv \mathbf{C}_6 \sqcap \exists \text{age}.\geq_{40\pm 5}$

Let us now consider the example of section 2 with the new knowledge base. First, it can be noticed that MC is not subsumed with a degree 1 by any concept related to a treatment by an axiom. By contrast, it is subsumed with degrees of $]0; 1[$ by $\exists \text{reco}.\text{chemotherapy}$ and $\exists \text{reco}.\text{chemo+tam}$, with the respective degrees 0.6 and 0.4 (cf. axioms [A0, 1, 4, 6, 7' and 12] and axioms [A0, 1, 4, 6, 8' and 13]).

Therefore, the system recommends the two treatments chemotherapy and chemo+tam, with a preference for the former. Thus, the physician using FUZZY-

KASIMIR decides either one of these treatments, or a combination of them (e.g., with a compromise in the doses of the chemotherapy and hormonotherapy drugs).

4 Towards a Semantic Portal for Oncology based on Fuzzy Datatypes

One purpose of the KASIMIR project is the diffusion of standard knowledge in oncology, in order to improve health-care practice. KASIMIR has to provide a direct, updated and intelligent access to the current knowledge, for geographically distributed users. This leads to standardization and distribution requirements for both knowledge and software components within KASIMIR. For this reason, the KASIMIR system is currently embedded into a semantic portal for oncology. This semantic portal is a Web server used to supply and share knowledge and intelligent services for oncology, thanks to semantic Web principles and technologies. In particular, medical decision protocols are represented within the OWL DL formalism and supplied services are based on standard DL reasoning with this formalism. Note that this leads to an improved expressivity with respect to the ad hoc KASIMIR formalism, and that it allows KASIMIR to take advantage of the other knowledge sources and tools available on the semantic Web.

In the following, we focus on the representation of medical decision protocols within the OWL DL formalism, and particularly on the currently debated subject of concrete domain representation (section 4.1). As explained in the previous section, representation of decision protocols leads us to introduce fuzziness in the representation formalism. Then, on the basis of lessons learnt from the development of FUZZY-KASIMIR, section 4.2 discusses the introduction of fuzzy datatypes into OWL DL.

4.1 Protocol Representation in the Current Semantic Portal

The semantic portal has been developed within an architecture based on Web services dedicated to OWL reasoning. The KASIMIR user interface, used to associate a treatment to a patient description, has been replaced by a more general instance editor called EDHIBOU that uses these Web services. The principles of the translation of KASIMIR knowledge bases in OWL DL is presented hereafter.

The KASIMIR ad hoc syntax is closer to object-based representation formalisms than to DLs. A first work was translated in the DL syntax of $\mathcal{ALC}(D)$. This part of this translation is not detailed here: KASIMIR has already been described in the DL syntax in the previous sections.

The representation of concrete domains in OWL is based on *XML Schema datatypes*.

A datatype corresponds to a (possibly infinite) set of values that are addressed through literals. A literal is a string that is interpreted as a value of a datatype. For example, "40"^^xsd:integer is a literal representing the integer 40, xsd:integer being the XML Schema datatype for integers. There are two disjoint types of properties in OWL: object properties, that relate instances to other instances, and datatype properties, that relate instances to literals. Object properties correspond to roles and datatype properties to concrete features in classical DLs. Now, some concept constructors can use datatype properties and datatypes to build complex concepts. For example, the concept $\exists \text{age}.\text{integer}$, where age is a datatype property, corresponds to $\exists \text{age}.\top_z$ in the previous KASIMIR representation. In the XML syntax of OWL, $\exists \text{age}.\text{integer}$ can be written:

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#age"/>
  <owl:someValuesFrom rdf:resource=
    "http://www.w3.org/2001/XMLSchema#integer"/>
</owl:Restriction>
```

XML Schema provides mechanisms to build user defined datatypes like *derived simple types*. In particular, restrictions on existing datatypes can be used to express basic unary predicates like the ones that are used for protocol representation in KASIMIR. For example, $\exists \text{age}.<_{40}$ in KASIMIR is expressed in OWL by $\exists \text{age}.\text{integerUnder40}$, with integerUnder40 the derived type of the integers $n < 40$, defined by:

```
<xsd:simpleType name="integerUnder40">
  <xsd:restriction base="integer">
    <xsd:maxExclusive value="40"/>
  </xsd:restriction>
</xsd:simpleType>
```

Then, subsumption between concepts is performed thanks to the *subtype test* between XML Schema datatypes. For example, if integerBetween20and30 is a subtype of integerUnder40, then $\exists \text{age}.\text{integerBetween20and30} \sqsubseteq \exists \text{age}.\text{integerUnder40}$.

The problem is that derived types are not supported in the current OWL DL recommendation and that protocol representation for the KASIMIR portal uses it in a non-standard way. Actually, only *built-in* datatypes (xsd:integer, xsd:float etc.) are taken into account, but current studies like OWL-E [18] aims at extending these recommendations for expressive datatype expressions. Thus, there is a reasonable hope that a future recommendation extending OWL DL will include user-defined or derived datatypes.

An important preliminary remark must be done about OWL reasoning with datatypes. Datatype properties and object properties are disjoint in OWL, thus a reasoner on OWL can be composed of two distinct parts:

- The *concept reasoner* deals with concepts and, whenever it meets a datatype, calls the datatype reasoner.
- The *datatype reasoner* deals with datatypes and provides services (such as the subtype test) independently from the concepts of the DL.

This principle can be used to implement in a straightforward way two inferences on OWL with fuzzy datatypes: the subsumption (\sqsubseteq) and the α -subsumption (\sqsubseteq_α). The implementation of the degree of subsumption (F_\sqsubseteq) is more complex and will be discussed.

Remark: Many current DL reasoners use the method of semantic tableau to compute the subsumption test. In fact, the test computed by this method is the satisfiability of a concept and it calls the test of consistency of a datatype implemented in the datatype reasoner [19]. However, subsumption test between concepts (resp., subtype test between datatypes) can be computed thanks to concept satisfiability (resp., datatype consistency). Thus, in the following, only the subsumption test between concepts and the subtype test between datatypes are considered.

4.2.1 Fuzzy datatypes

A (classical) datatype represents a set of values. For example, the datatype `integerUnder40` represents the set $\{39, 38, 37, \dots\}$, i.e. $<_{40}^{\mathbb{Z}}$ with the notation of section 2. We introduce the notion of *fuzzy datatype*, which represents a *fuzzy set* of values. For example, let `integerUnder40±5` be the fuzzy datatype representing the fuzzy set $\leq_{40\pm 5}^{\mathbb{Z}}$, with the notation of section 3. More generally, it is assumed that a syntax is defined to introduce some fuzzy datatypes, in particular, the fuzzy datatypes `integerUnder $a\pm b$` and `integerAbove $a\pm b$` ($a, b \in \Delta_{\mathbb{Z}}$, $b > 0$), representing respectively the fuzzy sets $\leq_{a\pm b}^{\mathbb{Z}}$ and $\geq_{a\pm b}^{\mathbb{Z}}$. For example, we can imagine that a definition of the fuzzy datatype `integerUnder40±5` could be something like:

```
<xsd:simpleType name="integerUnder40pm5">
  <xsd:restriction base="integer">
    <xsd:maxInclusive value="40" plusOrMinus="5"/>
  </xsd:retriction>
</xsd:simpleType>
```

Another possibility is to follow the idea presented in [5] and to define this fuzzy datatype by two classical datatypes: `integerUnderEq35` representing the core of $\leq_{40\pm5}^Z$ and `integerUnder45` representing its support. Then the definition of `integerUnder40±5` could be:

```
<fxsd:fuzzyType name="integerUnder40pm5">
  <fxsd:core value="integerUnderEq35"/>
  <fxsd:support value="integerUnder45"/>
</fxsd:fuzzyType>
```

where `fxsd` would refer to a new namespace for fuzzy XML Schema datatype definitions.

Let `fd1` and `fd2` be two fuzzy datatypes and A_1 and A_2 , the corresponding fuzzy sets. Let $F_{\preccurlyeq}(\text{fd1}, \text{fd2}) = F_{\subseteq}(A_1, A_2)$, the so-called *subtype degree* of `fd1` in `fd2`. For example:

$$F_{\preccurlyeq}(\text{integerUnder40}\pm 5, \text{integerUnder42}\pm 8) = 0.935$$

For $\alpha \in]0; 1]$, let \preccurlyeq_{α} be the crisp relation between fuzzy datatypes defined by `fd1` \preccurlyeq_{α} `fd2` if $F_{\preccurlyeq}(\text{fd1}, \text{fd2}) \geq \alpha$. We call \preccurlyeq_{α} the α -*subtype relation* (it is the α -cut of F_{\preccurlyeq}). Finally, let \preccurlyeq be the so-called *subtype relation* between fuzzy datatypes, i.e., `fd1` \preccurlyeq `fd2` if $F_{\preccurlyeq}(\text{fd1}, \text{fd2}) = 1$ (in other words, $\preccurlyeq = \preccurlyeq_1$). \preccurlyeq verifies `fd1` \preccurlyeq `fd2` iff $A_1 \subseteq A_2$. It is assumed that, for the fuzzy datatypes that are introduced, F_{\preccurlyeq} (and thus \preccurlyeq_{α} and \preccurlyeq) is computable. If the only fuzzy datatypes introduced are of the forms `integerUndera`±`b` and `integerAbovea`±`b`, this property of computability holds.²

The crisp and fuzzy relations \preccurlyeq , \preccurlyeq_{α} and F_{\preccurlyeq} between fuzzy datatypes are used in the following to compute the relations \subseteq , \subseteq_{α} and F_{\subseteq} between concepts.

4.2.2 Subsumption between concepts with fuzzy datatypes (\subseteq)

Let `C` and `D` be two concepts described in OWL DL, possibly using fuzzy datatypes. As mentioned above, the concept reasoner used to compute the test `C` \subseteq `D` uses the services of the datatype reasoner, in particular the subtype test. Thus, to compute the subsumption test, it is not necessary to modify the concept reasoner, but only the datatype reasoner, so that it can take into account fuzzy datatypes and the computation of \preccurlyeq .

² Indeed, in order to compute $F_{\preccurlyeq}(\text{fd1}, \text{fd2}) = \inf_x F_{\Rightarrow}(A_1(x), A_2(x))$, it is sufficient to compute this for $x \in \{a_1 - b_1, a_1 + b_1, a_2 - b_2, a_2 + b_2\}$, for `fdi` = `integerUnderai`±`bi` or `fdi` = `integerAboveai`±`bi`, $i \in \{1, 2\}$.

Let us consider these two examples:

$$\begin{aligned} \exists \text{age.integerUnder}40 \pm 5 &\sqsubseteq \exists \text{age.integerUnder}50 \pm 10 \\ \exists \text{age.integerUnder}40 \pm 5 &\not\sqsubseteq \exists \text{age.integerUnder}42 \pm 8 \end{aligned}$$

To compute these tests, the concept reasoner calls the datatype reasoner with the queries:

$$\begin{aligned} \text{integerUnder}40 \pm 5 &\preceq \text{integerUnder}50 \pm 10? && (\text{answer: yes}) \\ \text{integerUnder}40 \pm 5 &\preceq \text{integerUnder}42 \pm 8? && (\text{answer: no}) \end{aligned}$$

4.2.3 α -subsumption between concepts with fuzzy datatypes (\sqsubseteq_α)

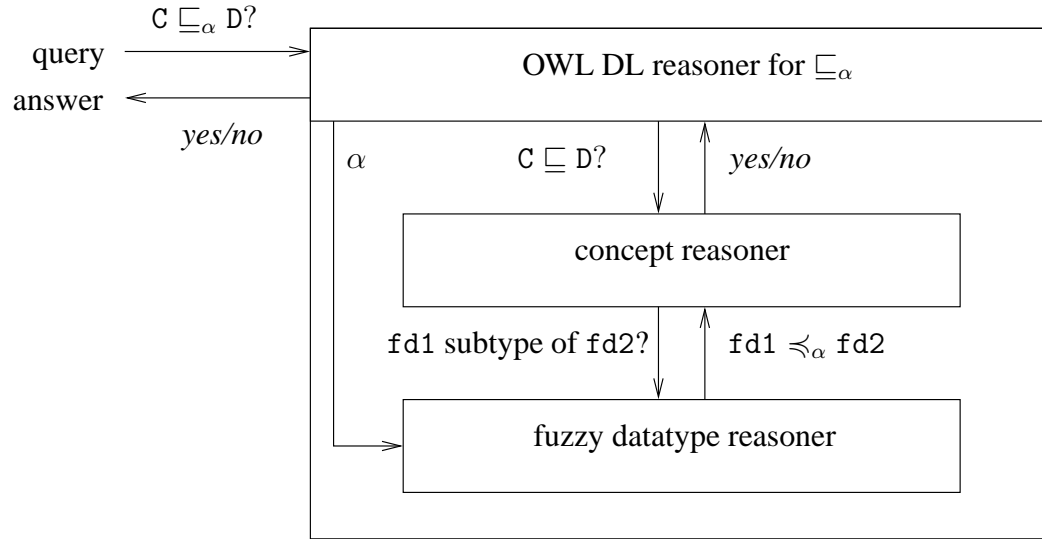


Fig. 3. Architecture of an OWL DL reasoner supporting the \sqsubseteq_α inference.

Let $\alpha \in [0; 1]$ and \sqsubseteq_α be the binary relation between concepts defined by $C \sqsubseteq_\alpha D$ if $F_{\sqsubseteq}(C, D) \geq \alpha$. As it was the case for \sqsubseteq , the test $C_1 \sqsubseteq_\alpha C_2$ can be computed without modifying the concept reasoner. The principle of the implementation of \sqsubseteq_α , as it is planned, is illustrated by the figure 3. Let us assume that a query $C \sqsubseteq_\alpha D$ is sent to the service for the α -subsumption. This service gives the parameter α to the fuzzy datatype reasoner, so that, each time a subtype test is requested for a pair $(fd1, fd2)$ of fuzzy datatypes, it computes $fd1 \preceq_\alpha fd2$. Therefore, the computation of the test $C \sqsubseteq_\alpha D$ amounts to the computation by the concept reasoner of $C \sqsubseteq D$, which calls the fuzzy datatype reasoner parameterized by the value α .

This service should give the following results on these examples:

$$\begin{aligned} \exists \text{age.integerUnder}40 \pm 5 &\sqsubseteq_\alpha \exists \text{age.integerUnder}50 \pm 10 && \text{for any } \alpha \in [0; 1] \\ \exists \text{age.integerUnder}40 \pm 5 &\sqsubseteq_\alpha \exists \text{age.integerUnder}42 \pm 8 && \text{iff } \alpha \leq 0.935 \end{aligned}$$

Remark: For this computation it is assumed that the concept reasoner does not store the results of previous subsumption tests for future reuse, or that this storage can be reset. Indeed, if the result of any subsumption test is stored, then, after a test $C \sqsubseteq_{\alpha} D$ has been performed with a positive answer, any test $C \sqsubseteq_{\beta}$, with the same concepts C and D , will reuse this answer!

4.2.4 Degree of subsumption between concepts with fuzzy datatypes (F_{\sqsubseteq})

What makes possible the implementation of \sqsubseteq and \sqsubseteq_{α} without modifying the concept reasoner is that the result of these tests are binary. Now, a service implementing F_{\sqsubseteq} would return a fuzzy truth value, thus using the output of the classical concept reasoner as the output of this service is not possible. However, it is possible to reuse –without changing it– the concept reasoner in order to implement F_{\sqsubseteq} . The principle is as follows. Let us assume that the α -subsumption has been implemented according to the architecture of figure 3. The computation of $F_{\sqsubseteq}(C, D)$ is based on the following equation:

$$F_{\sqsubseteq}(C, D) = \max\{\alpha \in [0; 1] \mid C \sqsubseteq_{\alpha} D\} \quad (1)$$

Thus, to compute $F_{\sqsubseteq}(C, D)$, it is sufficient to find this maximum value of α . This search can be done by dichotomy, given ε the precision of the required result (i.e., $|F_{\sqsubseteq}(C, D) - \alpha| \leq \varepsilon$, where α is the value that is computed by this search):

1. Let $a = 0$, $b = 1$ and $\alpha = \frac{1}{2}$.
2. While $b - a > \varepsilon$ do
 - 2.i If $C \sqsubseteq_{\alpha} D$ (cf. section 4.2.3) then let $a = \alpha$ else let $b = \alpha$.
 - 2.ii Let $\alpha = \frac{a+b}{2}$.
3. Return α

This computation requires at most $\lceil -\log_2 \varepsilon \rceil$ queries $C \sqsubseteq_{\alpha} D$ (where $\lceil x \rceil$ is the lowest integer a such that $x \leq a$).

This algorithm can be modified to provide an exact result if the set of the fuzzy truth values explored is finite. This occurs if a finite scale of fuzzy truth values is used, as suggested in [5].

An ongoing study aims at finding an *exact* value of $F_{\sqsubseteq}(C, D)$, even with the scale of truth values $[0; 1]$. The principle would be, given the concepts C and D , to make the fuzzy datatype reasoner generate a finite set of values FTV . Then, the dichotomy can be performed in FTV , which would require at most $1 + \lceil \log_2 |FTV| \rceil$ tests $C \sqsubseteq_{\alpha} D$. How this set can be efficiently generated is still an open question.

Another way to implement F_{\sqsubseteq} could be to modify the concept reasoner so that it can manipulate fuzzy truth values, but designing and implementing a fuzzy concept reasoner which is as optimized as current (classical) DL reasoners is a much more complex task, from both theoretical and applied viewpoints.³

5 Discussion and Related Work

The contribution of this paper is twofold. First, a description of a simple representation language similar to a fuzzy DL –the FUZZY-KASIMIR formalism– has been described. Section 5.1 situates this language among the family of fuzzy DLs and related formalisms. The second contribution is our proposal of a semantic portal using the formalism OWL DL with fuzzy datatypes. OWL DL is not only a formalism equivalent to an expressive DL, it is also a language that has been designed for the semantic Web, which involves additional requirements. Section 5.2 shows how our proposal fulfills these requirements.

5.1 Fuzzy DLs: how Fuzziness is Introduced into DLs

	[26]	[25]	[16]	[22]	[23]	[9]	[24]	[20]	[11; 10]	[13]	[6]	F-K
(FPCD)	×		×				×				×	×
(M)		×	×			×	×					
(FQ)								×				
(FAX)				×			×		×	×		
(FA)				×	×	×	×		×	×		

Table 1

Fuzzy DLs and related work, according to the way fuzziness is introduced in the formalism (F-K stands for FUZZY-KASIMIR) .

The fuzzy DLs we have met in the literature are built following the same general principle: starting from a (classical) DL, such as \mathcal{ALC} , additional features are added to it in order to express fuzziness. In any case, the semantics consists in interpreting a concept by a fuzzy set and a role by a fuzzy binary relation. By contrast, there are different ways to introduce fuzziness into the DL formalism. To our knowledge, there are five of these ways:

- (FPCD)** Fuzzy predicates in concrete domains,
- (M)** Modifiers of concepts,
- (FQ)** Fuzzy quantifiers,
- (FAX)** Fuzzy axioms and

³ For this purpose, Umberto Straccia has suggested to use techniques for mixed integer programming [24].

(FA) Fuzzy assertions.

The table 1 presents some fuzzy DLs and related formalisms, under the viewpoint of these five fuzziness introduction types. The related formalisms are either DLs taking into account some uncertainty (possibilistic logic [11; 10] or probabilistic reasoning [13]), or object-based representation formalisms with fuzzy features (like [6] and FUZZY-KASIMIR).

(FPCD) is the approach used in FUZZY-KASIMIR. Sub-categories of (FPCD) can be defined by precisising what are the concrete domains involved and what are the families of fuzzy predicate names that are supported.

(M) is based on the constructors mC , where m is a modifier (also called a manipulator or a hedge) and C is a concept. A modifier is a function $m : [0; 1] \rightarrow [0; 1]$. For example, the modifier *very* is often defined by $very(t) = t^2$. Given an interpretation \mathcal{I} , mC is interpreted by $(mC)^{\mathcal{I}}(x) = m(C^{\mathcal{I}}(x))$ for $x \in \Delta_{\mathcal{I}}$. Different families of modifiers give birth to different sub-categories of (M).

(FQ) consists in extending the classical quantifiers of DLs (\exists and \forall) with fuzzy quantifiers. The language of the chosen fuzzy quantifiers characterizes the fuzzy extension of the DL.

(FAX), contrasting with the previous ways of introducing fuzziness, does not consist in defining new concept constructors, but it consists in associating to axioms some pieces of informations about them, such as constraints on their fuzzy truth values.

(FA) is based on the same idea as (FAX), except that it concerns assertions (i.e., specifications on particular instances) and not concepts.

Finally, as table 1 shows, combinations of the ways of introducing fuzziness into DLs can be done.

5.2 Introduction of Fuzziness into OWL DL

OWL DL can be seen as an expressive DL designed to be the standard language for representing ontologies in the semantic Web. It is a “semantic Web DL” which involves some particular features that must be taken into account in the building of a fuzzy extension for this formalism:

- OWL is standard. It has the status of a W3C recommendation.
- There exist open, stable and efficient systems that are actually used for reasoning with OWL.
- OWL is divided into sub-languages (Lite, DL and Full) that differ in their level

of complexity, and in the expected usage of the included constructors in practical applications.

The syntax and the semantics of OWL are defined in a recommendation of the W3C (World Wide Web Consortium) [2]. This means that it is a standard language and so, that an OWL ontology must be considered in the same way, i.e. leads to the same behavior, in different OWL implementations. Being a standard also constraints the language to be stable in time: the extension of a standard technology has to be compatible with the extended technology (principle of backward compatibility). Considering a fuzzy extension of OWL, any ontology described in the classical OWL formalism must also be an ontology in the fuzzy OWL formalism and so, must lead to the same inferences in the two languages. This constraint is fulfilled by the fuzzy extension proposed in this paper. Indeed, on the basis of XML Schema datatypes, datatype definitions are extended for introducing fuzziness in our framework. Since the OWL recommendation is not directly concerned with datatype definitions, the actual syntax of the OWL formalism remains the same. Moreover, the semantics introduced to take into account fuzzy datatypes in OWL DL is fully compatible with ontologies made to be interpreted in a classical way, using the classical semantics. For example, if C and D are two concepts of an ontology that does not use any fuzzy datatype, then C is subsumed by D in the classical sense, iff C is subsumed by D in the fuzzy sense.

Several systems have been developed for reasoning on $\mathcal{SHOIN}(\mathcal{D})$, the OWL DL underlying description logic (e.g., FACT, PELLET and RACER [12; 21; 8]). Since these systems are open, stable and efficient, the development of a reasoner for OWL DL with fuzzy datatypes can benefit from reusing them. As said in section 4.2, datatype reasoning and concept reasoning are two distinct parts of an OWL DL reasoner. In this way, the reasoner for OWL DL with fuzzy datatypes can be built on top of an existing concept reasoner without modifying it and so, can have benefit of its efficiency.

Furthermore, this layered architecture comes in complement of the existing layers of OWL (Lite, DL and Full), and fuzzy datatypes can be used in any sub-language of OWL. Moreover, in the same way that the constructors included in OWL Lite provide “a minimal useful subset of language features, that are relatively straightforward for tool developers to support” [2], using fuzzy datatypes seems to be a simple and useful way of introducing fuzziness in OWL. Indeed, in our application for oncology, this choice is motivated by requirements concerning the representation of medical decision protocols. In many other applications, the need to define fuzzy knowledge elements also comes from thresholds on concrete data (most often numeric data). For this reason, the proposed extension of OWL with fuzzy datatypes and the future associated reasoner, should provide a simple answer to the need for fuzzy representation in many semantic Web applications.

6 Conclusion

This paper presents three systems that are fully implemented (including user interfaces), and a proposal for a fourth one. KASIMIR is a knowledge-based system using an ad hoc formalism similar to a simple description logic with concrete domains which is used for representing decision protocols in oncology. FUZZY-KASIMIR is an extension of KASIMIR with fuzzy concrete domains taking into account discontinuities in the decision that are due to numerical thresholds. Another extension of KASIMIR has led to embed it into a semantic portal for oncology, which has been motivated by the need to share knowledge for geographically distributed physicians and has led to change the ad hoc formalism to the standard OWL DL. A combination of these two extensions of KASIMIR (FUZZY-KASIMIR and the semantic portal for oncology) is currently under implementation and will lead to a semantic portal for oncology with fuzzy datatypes. The implementation of this new portal follows the principles of the semantic Web (in particular, backward compatibility) and is based on the decomposition of an OWL DL reasoner into a concept reasoner and a datatype reasoner. The inferences that are studied are the subsumption, the α -subsumption and the degree of subsumption between OWL DL classes with fuzzy datatypes.

References

- [1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, Cambridge, UK, 2003.
- [2] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language Reference, www.w3.org/TR/owl-ref. last consultation: February, 2005.
- [3] M. d'Aquin, C. Bouthier, S. Brachais, J. Lieber, and A. Napoli. Knowledge Edition and Maintenance Tools for a Semantic Portal in Oncology. *International Journal of Human Computer Studies*, 62(5), to be published in 2005.
- [4] M. d'Aquin, S. Brachais, J. Lieber, and A. Napoli. Decision Support and Knowledge Management in Oncology using Hierarchical Classification. In K. Kaiser, S. Miksch, and S. W. Tu, editors, *Proceedings of the Symposium on Computerized Guidelines and Protocols (CGP-2004)*, volume 101 of *Studies in Health Technology and Informatics*, pages 16–30. IOS Press, 2004.
- [5] D. Dubois, J. Mengin, and H. Prade. Possibilistic uncertainty and fuzzy features in description logic, A preliminary discussion. In E. Sanchez, editor, *Proceedings of the Workshop Fuzzy Logic and the Semantic Web*, pages 5–7, Marseille, France, 2005.
- [6] D. Dubois, H. Prade, and J.-P. Rossazza. Vagueness, Typicality and Uncertainty in Class Hierarchies. *International Journal of Intelligent Systems*, 6:167–183, 1991.

- [7] Didier Dubois and Henri Prade, editors. *Fundamentals of Fuzzy Sets*, volume 7 of *The handbooks of fuzzy sets*. Kluwer, 2000.
- [8] V. Haarslev and R. Möller. Description of the Racer System and its Applications. In C. Goble, D. L. McGuinness, R. Möller, and P. F. Patel-Schneider, editors, *Proceedings of the 2001 International Description Logics Workshop*, pages 46–55, 2001.
- [9] S. Hölldobler, T. D. Khang, and H.-P. Störr. A Fuzzy Description Logic with Hedges as Concept Modifiers. In Nguyen Hoang Phuong, Hung T. Nguyen, Nguyen Cat Ho, and Pratit Santiprabhob, editors, *Proceedings of the Joint Third International Conference on Intelligent Technologies and Third Vietnam-Japan Symposium on Fuzzy Systems and Applications (InTech/VJFuzzy'2002)*, Hanoi, Vietnam, pages 25–34, 2002.
- [10] B. Hollunder. An Alternative Proof Method for Possibilistic Logic and its Application to Terminological Logics. Research report RR-93-01, DFKI, 1993.
- [11] B. Hollunder. An Alternative Proof Method for Possibilistic Logic and its Application to Terminological Logics. In Ramon Lopez de Mantaras and David Poole, editors, *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 327–335, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers.
- [12] I. Horrocks. Using an expressive description logic: FaCT or fiction? In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR'98)*, pages 636–647. Morgan Kaufmann Publishers, June 1998.
- [13] M. Jaeger. Probabilistic Reasoning in Terminological Logics. In Jon Doyle, Erik Sandewall, and Pietro Torasso, editors, *KR'94: Principles of Knowledge Representation and Reasoning*, pages 305–316. Morgan Kaufmann, 1994.
- [14] J. Lieber. Strong, Fuzzy and Smooth Hierarchical Classification for Case-Based Problem Solving. In F. van Harmelen, editor, *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI-02)*, Lyon, France, pages 81–85. IOS Press, Amsterdam, 2002.
- [15] C. Lutz. Description Logics with Concrete Domains – A Survey. In *Advances in Modal Logics Volume 4*. King's College Publications, 2003.
- [16] R. Molitor and C.B. Tresp. Extending Description Logics to Vague Knowledge in Medicine. In P. Szczepaniak, P.J.G. Lisboa, and S. Tsumoto, editors, *Fuzzy Systems in Medicine*, volume 41 of *Studies in Fuzziness and Soft Computing*, pages 617–635. Springer Verlag, 2000.
- [17] A. Napoli, C. Laurenço, and R. Ducournau. An object-based representation system for organic synthesis planning. *International Journal of Human-Computer Studies*, 41(1/2):5–32, 1994.
- [18] J. Z. Pan. Reasoning Support for OWL-E (Extended Abstract). In *Proc. of Doctoral Programme in the 2004 International Joint Conference of Automated Reasoning (IJCAR2004)*, July 2004.

- [19] J. Z. Pan and I. Horrocks. Web Ontology Reasoning with Datatype Groups. In Dieter Fensel, Katia Sycara, and John Mylopoulos, editors, *Proceedings of the 2nd International Semantic Web Conference (ISWC2003)*, 2003.
- [20] D. Sanchez and A. Tettamanzi. Reasoning and Quantification in Fuzzy Description Logics. In E. Sanchez, editor, *Proceedings of the Workshop Fuzzy Logic and the Semantic Web*, pages 15–17, Marseille, France, 2005.
- [21] E. Sirin and B. Parsia. Pellet: An OWL DL Reasoner, Porter. In *Third International Semantic Web Conference (ISWC2004)*, November 2004.
- [22] U. Straccia. A Fuzzy Description Logic. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98) and of the 10th Conference on Innovative Applications of Artificial Intelligence (IAAI-98)*, pages 594–599. AAAI Press, 1998.
- [23] U. Straccia. Reasoning within Fuzzy Description Logics. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.
- [24] U. Straccia. Fuzzy Description Logics with Concrete Domains. Technical Report 2005-TR-03, Istituto di Scienza e Tecnologie dell’Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy, 2005.
- [25] C.B. Tresp and R. Molitor. A Description Logic for Vague Knowledge. In *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI’98)*, pages 361–365. J. Wiley and Sons, 1998.
- [26] J. Yen. Generalizing Term Subsumption Languages to Fuzzy Logic. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI’91)*, Sidney, pages 472–477, august 1991.