Construction of Fuzzy Ontologies from Fuzzy UML Models*

Fu Zhang
College of Information Science and Engineering, Northeastern University
Shenyang, 110819, China

Z. M. Ma†
College of Information Science and Engineering, Northeastern University
Shenyang, 110819, China

Received 22 September 2011

Accepted 25 December 2012

Abstract

The success and proliferation of the Semantic Web depends heavily on construction of Web ontologies. However, classical ontology construction approaches are not sufficient for handling imprecise and uncertain information that is commonly found in many application domains. Therefore, great efforts on construction of fuzzy ontologies have been made in recent years. In this paper, we propose a formal approach and develop an automated tool for constructing fuzzy ontologies from fuzzy UML models. *Firstly*, we propose formalization methods of fuzzy UML models and fuzzy ontologies, where fuzzy UML models and fuzzy ontologies can be represented and interpreted by their respective formal definitions and semantic interpretation methods. *Then*, we propose an approach for constructing fuzzy ontologies from fuzzy UML models, i.e., transforming fuzzy UML models (including the structure and instance information of fuzzy UML models) into fuzzy ontologies. *Furthermore*, following the proposed approach, we implement a prototype transformation tool called *FUML2FOnto* that can construct fuzzy ontologies from fuzzy UML models. Constructing fuzzy ontologies from fuzzy UML models will facilitate the development of Web ontologies. *Moreover*, in order to show that the constructed fuzzy ontologies may be useful for reasoning on fuzzy UML models, we investigate how to reason on fuzzy UML models based on the constructed fuzzy ontologies, and it turns out that the reasoning tasks of fuzzy UML models can be checked by means of the reasoning mechanism of fuzzy ontologies.

Keywords: Fuzzy UML model; Fuzzy ontology; Construction; Reasoning.

1. Introduction

The Semantic Web aims at creating ontology-based and machine-processable Web content, and thus the success and proliferation of the Semantic Web depends largely on construction of Web ontologies[1,2,3]. On this basis, many approaches and tools have been developed to construct ontologies from various data resources such as texts, dictionaries, XML documents, and database models (see Ref. 4 for an overview). Among these kinds of data resources, constructing ontologies from database models (e.g., ER model[5], UML model[6], and relational

*This paper is a major extended version of "Zhang Fu, Ma Z M, et al. Fuzzy Semantic Web Ontology Learning from Fuzzy UML Model. Proc. of the CIKM 2009, 1007-1016, 2009.", i.e., the reference [30] in this paper. The current paper provides full formal representations of fuzzy UML models and fuzzy ontologies, full construction mechanism, full proofs of the construction approach, and in-depth investigation on reasoning of fuzzy UML models based on the constructed fuzzy ontologies. The detailed introduction about the new materials of the current paper than the reference [30] can be found at the end of Section 6 in this paper.

†Corresponding author: Z. M. Ma (Zongmin Ma) is with the College of Information Science and Engineering, Northeastern University, Shenyang, 110819, China, PhD, Professor, IEEE Senior Member, (phone/fax: 86-24-83681582; email: mazongmin@ise.neu.edu.cn).

Co-published by Atlantis Press and Taylor & Francis
Copyright: the authors

database model[7]) has attracted considerable attention because of the wide utilization of database models in information modeling and management[8]. In particular, the *Unified Modeling Language* (*UML*)[9,10] has become a de facto industry standard for modeling applications in data and software engineering communities. Many data sources have been modeled in UML, in which abundant domain knowledge has been specified. Thus it is not surprising that constructing ontologies from UML models[6,11,12,13,14], which are helpful to the development of Web ontologies.

However, the classical ontologies and ontology construction approaches as mentioned above were not sufficient for handling imprecise and uncertain information that is commonly found in various application domains such as databases, information systems, and the Semantic Web[15,16,17]. In order to provide the necessary means to handle imprecise and uncertain information in the Semantic Web there are today many proposals for fuzzy extensions to ontologies[18,19,20,21,22,23,24]. Also, how to construct fuzzy ontologies has increasingly attracted attention and several strategies for constructing fuzzy ontologies were proposed in[25,26,27,28,29,30,31]. Being similar to the requirement of handling imprecise and uncertain information in the context of the Semantic Web, the incorporation of imprecise and uncertain information in UML models has been an important topic of data modeling research because such information extensively exists in many real-world applications. Consequently, fuzzy UML models were widely investigated for modeling imprecise and uncertain information[32,33,34,35]. The detailed introduction about fuzzy UML models, fuzzy ontologies, and construction of fuzzy ontologies can be found in Section 6 of this paper.

Therefore, with the requirement of constructing fuzzy ontologies and the appearance of fuzzy UML models, it is possible and meaningful to construct fuzzy ontologies from fuzzy UML models. On one hand, being similar to the construction of ontologies from UML models, constructing fuzzy ontologies from fuzzy UML models will facilitate fuzzy ontology development in the Semantic Web. On the other hand, it is possible to make use of fuzzy ontology techniques to reason on fuzzy UML models on the basis of the constructing work. The current-day fuzzy UML models still suffer from some deficiencies such as inconsistency and complexity, which may result in a degradation of the quality of the design and/or increased development times and costs. Hence, during the design phase, it would be highly desirable to improve the ability of reasoning on fuzzy UML models. Constructing fuzzy ontologies from fuzzy UML models makes it possible that the reasoning tasks of fuzzy UML models (such as whether a fuzzy class is the subclass of another fuzzy class, or whether there is redundancy in a fuzzy UML model) can be detected automatically through the reasoning mechanism of fuzzy ontologies in stead of being checked by hand. However, less research has been done in constructing fuzzy ontologies from fuzzy UML models.

To this end, in this paper we propose a complete formal approach and develop an automated tool for constructing fuzzy ontologies from fuzzy UML models. Additionally, we investigate how to reason on fuzzy UML models based on the constructed fuzzy ontologies. In brief, the paper makes the following main contributions:

- **How to formalize fuzzy UML models and fuzzy ontologies?** In order to establish the precise correspondences between fuzzy UML models and fuzzy ontologies, in **Section 3**, we propose formalization methods of fuzzy UML models and fuzzy ontologies, where fuzzy UML models and fuzzy ontologies can be represented and interpreted by their respective formal definitions and semantic interpretation methods.
- **How to construct fuzzy ontologies from fuzzy UML models?** In **Section 4**, we propose a formal approach and develop an automated tool for constructing fuzzy ontologies from fuzzy UML models, including: ***(i)*** transforming a fuzzy UML model into fuzzy ontology structure at conceptual level; ***(ii)*** transforming a set of object instances (w.r.t. the fuzzy UML model) into fuzzy ontology instance (w.r.t. the fuzzy ontology structure) at instance level. The fuzzy ontology structure and instance form a target fuzzy ontology; ***(iii)*** implementing a prototype transformation tool called *FUML2FOnto* that can automatically construct fuzzy ontologies from fuzzy UML models.
- **How to reason on fuzzy UML models based on the constructed fuzzy ontologies?** To demonstrate the constructed fuzzy ontologies may be useful for reasoning on fuzzy UML models, in **Section 5**, we investigate how to reason on fuzzy UML models based on the constructed fuzzy ontologies, and it turns out that reasoning tasks of fuzzy UML models (e.g., consistency, satisfiability, subsumption, and

Copyright: the authors

redundancy) can be checked by means of the reasoning mechanism of fuzzy ontologies.

The remainder of this paper is organized as follows. Section 2 introduces some preliminaries on fuzzy UML models and fuzzy ontologies. Section 3 gives formal definitions and semantic interpretation methods of fuzzy UML models and fuzzy ontologies. Section 4 proposes an approach and implements a prototype tool for constructing fuzzy ontologies from fuzzy UML models. How to reason on fuzzy UML models with the constructed fuzzy ontologies is investigated in Section 5. Section 6 introduces related work. Section 7 shows conclusions and the future work.

2. Preliminaries on Fuzzy UML Models and Fuzzy Ontologies

In this section, some preliminaries on fuzzy UML models and fuzzy ontologies are recalled briefly. Before that we first introduce some basic knowledge concerning fuzzy sets and possibility distributions.

2.1. *Fuzzy sets and possibility distributions*

In real-world applications, information is often imprecise and uncertain. The imprecision is relevant to the content of an attribute value, and it means that a choice must be made from a given range (interval or set) of values but we do not know exactly which one to choose at present, and the uncertainty is related to the degree of truth of its attribute value, and it means that we can apportion some, but not all, of our belief to a given value or a group of values, for example, a set, say {0.7/18, 0.95/19, 0.98/20, 0.85/21} for the age of Michael, which contains information imprecision (the age may be 18, 19, 20, or 21 and we do not know which one is true) and uncertainty (the degrees of truth of all possible age values are respectively 0.7, 0.95, 0.98, and 0.85) simultaneously. In order to deal with imprecise and uncertain information, the fuzzy set theory[36] addressed by Zadeh has been identified as a successful technique for modeling the imprecise and uncertain information in various application domains such as databases, information systems, and the Semantic Web[15,16,17].

A fuzzy set F with respect to a universe of discourse U is characterized by a membership function $\mu_F : U \rightarrow [0, 1]$, assigning a membership degree $\mu_F(u)$ to each $u \in U$. Membership function $\mu_F(u)$ provides a measure of the degree of the belonging of u to F. For example, $\mu_F(u) = 0.8$ means that u is "likely" to be an element of F by a degree of 0.8. For ease of representation, a fuzzy set F over universe U is organized into a set of ordered pairs:

$F = \{u_1/\mu_F(u_1), u_2/\mu_F(u_2),\ldots, u_n/\mu_F(u_n)\}$

When $\mu_F(u)$ is explained to be a measure of the possibility that a variable X has the value u, where X takes values in U, a fuzzy value is described by a possibility distribution π_X[37]:

$\pi_X = \{u_1/\pi_X(u_1), u_2/\pi_X(u_2),\ldots, u_n/\pi_X(u_n)\}$

Here, $\pi_X(u_i)$, $u_i \in U$, denotes the possibility that $X = u_i$ is true. A fuzzy set is a representation of a concept while possibility distribution relates with the possibility of occurring a value within a distribution. Let π_X and F be the possibility distribution representation and the fuzzy set representation for a fuzzy value, respectively. It is clear that as soon as we assume that X is F then $\pi_X = F$ is true[37]. By means of fuzzy sets and possibility distributions, a fuzzy value on U can be characterized by a fuzzy set or a possibility distribution in U.

For more concepts and operations about fuzzy sets and possibility distributions, please refer to Refs. 36, 37.

2.2. *Fuzzy UML Models*

The fuzzy UML model is a fuzzy extension of UML model based on fuzzy set and possibility distribution theory. Based on Refs. 30, 32, 33, 34, 35, 38, and 39, in the following we introduce some basic notions of fuzzy UML models, including fuzzy class, fuzzy generalization, fuzzy aggregation, fuzzy dependency, and fuzzy association, where both of the structural and dynamic aspects of fuzzy UML models are considered.

2.2.1. *Fuzzy Class*

An object, which is an entity of the real world, is fuzzy because of a lack of information. Formally, objects that have at least one attribute whose value is a fuzzy set are fuzzy objects. The objects with the same structure and behavior are grouped into a class, and classes are organized into hierarchies. Theoretically, a class can be considered from two different viewpoints:

- An extensional class, where the class is defined by a list of its object instances;
- An intensional class, where the class is defined by a set of attributes and their admissible values.

In a fuzzy UML model, a class is fuzzy because of the following several reasons:

- A class is extensionally defined, where some objects with similar properties are fuzzy ones. The

Co-published by Atlantis Press and Taylor & Francis
Copyright: the authors

class defined by these objects may be fuzzy, and these objects belong to the class with degree of [0, 1].

- When a class is intensionally defined, the domains of some attributes may be fuzzy, and thus a fuzzy class is formed.
- The subclass produced by a fuzzy class by means of specialization, and the superclass produced by some classes (in which there is at least one class who is fuzzy) by means of generalization, are fuzzy.

Fig. 1 shows a fuzzy UML class *Old-Employee*, where a fuzzy class is denoted by using a *dashed-outline rectangle* to differentiate a classical class. Here:

- A fuzzy keyword *FUZZY* is appeared in front of an attribute indicating the attribute may take fuzzy values. For example, *FUZZY* Age or *FUZZY* Email. Moreover, for an attribute *a* of type *T* in a class *C*, an optional multiplicity [*i*…*j*] for *a* specifies that *a* associates to each instance of *C* at least *i* and most *j* instances of *T*. When the multiplicity is missing, [1…1] is assumed, i.e., the attribute is single-valued. For example, the attribute "*FUZZY* Email [1…∞]" in Fig. 1 means that each object instance of the class *Old-Employee* has at least one email, and possibly more.
- The method *IsDepartment( ):String* denotes the dynamic aspect of fuzzy UML models. It returns a possibility distribution value {*Department*/u_i} denoting that an *Old-Employee* works in the *Department* with degree $u_i \in [0, 1]$. The type of the parameter is *null*.
- An additional attribute $u \in [0, 1]$ in the class is defined for representing the object instance membership degree to the class.

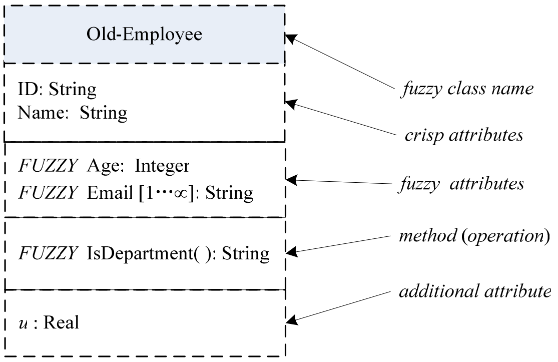


Fig. 1. Representation of a *fuzzy class Old-Employee* in a fuzzy UML model.

2.2.2. *Fuzzy Class*

The concept of *generalization* is one of the basic building blocks of the fuzzy UML model. A *generalization* is a taxonomic relationship between a more general classifier named *superclass* and a more specific classifier named *subclass*. The subclass is produced from the superclass by means of inheriting all attributes and methods of the superclass, overriding some attributes and methods of the superclass, and defining some attributes and methods.

A class produced from a fuzzy class must be fuzzy. If the former is still called subclass and the latter superclass, the subclass/superclass relationship is fuzzy. If a fuzzy class is a subclass of another fuzzy class, for any object, say *o*, let the membership degree that it belongs to the subclass, say *B*, be $u_B(o)$ and the membership degree that it belongs to the superclass, say *A*, be $u_A(o)$. Then $u_B(o) \le u_A(o)$. This characteristic can be used to determine if two classes have a subclass/superclass relationship.

Several generalizations can be grouped together to form a *class hierarchy* as shown in Fig. 2. **Fig. 2** shows a fuzzy generalization relationship, where a *dashed triangular arrowhead* is used to represent a fuzzy generalization relationship. The *disjointness* and *completeness* constraints, which are optional, can be enforced on a class hierarchy. The *disjointness* means that all the specific classes are mutually disjoint, and *completeness* means that the union of the more specific classes completely covers the more general class. That is, the union of object instances of several subclasses completely covers the object instances of the superclass, and the membership degree that any object belongs to the subclass must be less than or equal to the membership degree that it belongs to the superclass.

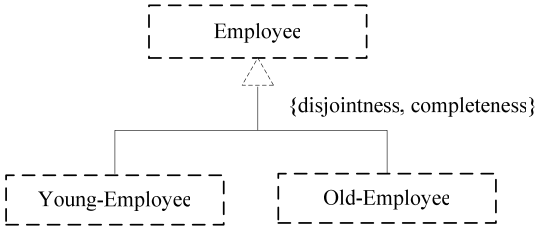


Fig. 2. Representation of a *fuzzy generalization* in a fuzzy UML model.

2.2.3. *Fuzzy Aggregation*

An *aggregation* captures a *whole-part* relationship between a class named aggregate and a group of classes named constituent parts. The constituent parts can exist independently. Aggregate class *New-Computer*, for instance, is aggregated by constituent parts *Monitor*, *New CPU box*, and *Keyboard*. Each object of an aggregate can be projected into a set of objects of constituent parts. Formally, Let C be an aggregate of constituent parts C_1, C_2, …, C_n. For $o \in C$, the projection of o to C_i is denoted by $o{\downarrow}_{Ci}$. Then we have $(o{\downarrow}_{C1}) \in C_1$, $(o{\downarrow}_{C2}) \in C_2$, …, and $(o{\downarrow}_{Cn}) \in C_n$.

A class aggregated from fuzzy constituent parts must be fuzzy. For any object, say o, let the membership degree that it belongs to the aggregate, say C, be $u_C(o)$. Also, let the projection of o to the constituent parts, say C_1, C_2, …, C_n, be $o{\downarrow}_{C1}$, $o{\downarrow}_{C2}$, …, $o{\downarrow}_{Cn}$. Let the membership degrees that these projections belong to C_1, C_2, …, C_n be $u_{C1}(o{\downarrow}_{C1})$, $u_{C2}(o{\downarrow}_{C2})$, …, $u_{Cn}(o{\downarrow}_{Cn})$, respectively. Then it follows $u_C(o) \leq u_{C1}(o{\downarrow}_{C1})$, $u_C(o) \leq u_{C2}(o{\downarrow}_{C2})$, ..., and $u_C(o) \leq u_{Cn}(o{\downarrow}_{Cn})$. This characteristic can be used to determine if several classes have a fuzzy aggregation relationship.

Fig. 3 shows a fuzzy aggregation relationship, where a *dashed open diamond* is used to denote a fuzzy aggregation relationship. Here:

- The class *New CPU box* is a fuzzy class, and thus the class *New-Computer* aggregated by *Monitor*, *New CPU box*, and *Keyboard* is also fuzzy.
- The *multiplicity* $[m_i, n_i]$ specifies that each instance of the aggregate class consists of at least m_i and at most n_i instances of the i-th constituent class. For example, a New-Computer may contain at least one Monitor, and possibly more.

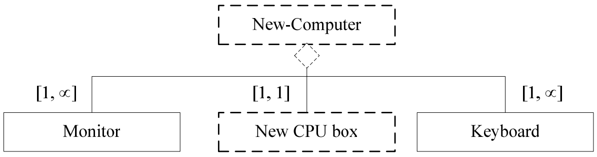


Fig. 3. Representation of a *fuzzy aggregation* in a fuzzy UML model.

2.2.4. *Fuzzy Dependency*

A *dependency*, which is a relationship between a source class and a target class, denotes that the target class exists dependently on the source class. In addition, the dependency between the source class and the target class is only related to the classes themselves and does not require a set of instances for its meaning.

A fuzzy dependency relationship is a dependency relationship with a degree of possibility η as shown in Fig. 4, which can be indicated explicitly by the designers or be implied implicitly by the source class based on the fact that the target class is decided by the source class.

Fig. 4 shows a fuzzy dependency relationship, which is denoted by a *dashed line with an arrowhead*. It is clear that *Employee Dependent* is dependent on fuzzy class *Employee* with membership degree $\eta \in [0, 1]$.

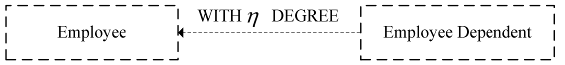


Fig. 4. Representation of a *fuzzy dependency* in a fuzzy UML model.

2.2.5. *Fuzzy Association / Fuzzy Association Class*

An *association* is a relation between the instances of two or more classes. Names of associations are unique in a fuzzy UML model. An association has a related *association class* that describes properties of the association. Three kinds of fuzziness can be identified in an association relationship:

- The association is fuzzy itself, it means that the association relationship fuzzily exists in n classes, namely, this association relationship occurs with a degree of possibility.
- The association is not fuzzy itself, i.e., it is known that the association relationship must occur in n classes. But it is not known for certain if n class instances (i.e., n objects) respectively belonging to the n classes have the given association relationship.
- The association is fuzzy caused by such fact that (1) and (2) occur in the association relationship simultaneously.

A fuzzy association relationship is an association relationship with a degree of possibility. We introduce a symbol β, as shown in Fig. 5, into a fuzzy UML model to denote the degree of possibility of a fuzzy association, and the calculating methods of β with respect to the three kinds of fuzziness above have been introduced in Ref. 33.

Fig. 5 shows a fuzzy association class *Use* between two classes *Old-Employee* and *New-Computer*. A *single line with an arrowhead* is used to denote a fuzzy association, and the association class is connected with the association by a *dashed-outline*. Here:

- *Date* is an attribute of the association class *Use*, which describes the start date that an *Old-Employee* uses a *New-Computer*.
- The additional symbol β denotes the membership degree of the fuzzy association occurring in several classes as mentioned above.
- The participation of a class in a fuzzy association is called a *role* which has a unique name. For example, *Uby* and *Uof* in Fig. 5.
- The cardinality constraint (m, n) on an association S specifies that each instance of the class C_i can participate at least m times and at most n times to S. For example, in Fig. 5, (1, 1) and (1, 2) denote that each *Old-Employee* can use at least 1 and at most 2 *New-Computers* and each *New-Computer* can be used by exactly one *Old-Employee*.

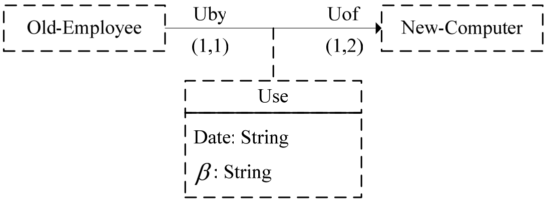


Fig. 5. Representation of a *fuzzy association* (*class*) in a fuzzy UML model.

The main notions of a fuzzy UML model are introduced above. Based on these notions, a complete formal definition and semantic interpretation method of fuzzy UML models will be proposed in Section 3. In the following section, some basic notions of fuzzy ontologies will be recalled in order to represent the fuzzy UML models above.

2.3. *Fuzzy Ontologies*

Ontologies, which enable formal, explicit and common descriptions of application domains, can be defined by ontology definition languages such as RDFS, OIL, DAML+OIL, or OWL[40]. *OWL* (*Web Ontology language*), a W3C recommendation, is to be de-facto standard language for representing ontologies in the Semantic Web. OWL consists of three languages of increasing expressive power: OWL Lite, OWL DL and OWL Full. Although OWL is a quite expressive language it features limitations, mainly with what can be said about fuzzy information. Such information is apparent in many applications and tasks both of the Semantic Web as well as of applications using OWL. Therefore, a fuzzy ontology language called *fuzzy OWL*[18,41,42] was developed.

Being similar to OWL, fuzzy OWL has three increasingly expressive sublanguage fuzzy OWL Lite, fuzzy OWL DL, and fuzzy OWL Full. Fuzzy OWL Lite and fuzzy OWL DL are are almost equivalent to f-$\mathcal{SHIF}$(D) and f-$\mathcal{SHION}$(D) DLs[42]. Further, fuzzy OWL has two types of syntactic form: the RDF/XML syntax and the frame-like style abstract syntax, and they are interchangeable. Moreover, fuzzy OWL DL is the language chosen by the major fuzzy ontology editors because it supports those users who want the maximum expressiveness without losing computational completeness and decidability of reasoning systems. Therefore, when we mention fuzzy OWL in this paper, we usually mean fuzzy OWL DL.

Table 1 gives *the fuzzy OWL abstract syntax*, *the respective fuzzy Description Logic syntax*, and *the semantics*[41,42]. The semantics for fuzzy OWL is based on the interpretation of f-$\mathcal{SHION}$(D)[43]. In detail, the semantics is provided by a fuzzy interpretation $FI = (\Delta^{\mathrm{FI}}, \Delta_D, \bullet^{\mathrm{FI}}, \bullet^{\mathrm{D}})$, where the abstract domain Δ^{FI} is a set of objects, and the datatype domain Δ_D is the domain of interpretation of all datatypes (disjoint from Δ^{FI}) consisting of data values, and $\bullet^{\mathrm{FI}}$ and $\bullet^{\mathrm{D}}$ are two fuzzy interpretation functions, which map: An abstract individual o to an element $o^{\mathrm{FI}} \in \Delta^{\mathrm{FI}}$, For individuals o_1 and o_2, if $o_1 \neq o_2$, $o_1^{\mathrm{FI}} \neq o_2^{\mathrm{FI}}$, A concrete individual v to an element $v^{\mathrm{D}} \in \Delta_D$, A concept name A to a membership degree function $A^{\mathrm{FI}}: \Delta^{\mathrm{FI}} \rightarrow [0, 1]$, An abstract role name R to a membership degree function $R^{\mathrm{FI}}: \Delta^{\mathrm{FI}} \times \Delta^{\mathrm{FI}} \rightarrow [0, 1]$, A concrete datatype D to a membership degree function $D^{\mathrm{D}}: \Delta_D \rightarrow [0, 1]$, A concrete role name T to a membership degree function $T^{\mathrm{FI}} : \Delta^{\mathrm{FI}} \times \Delta_D \rightarrow [0, 1]$. Based on the fuzzy interpretation FI, the complete semantics of fuzzy OWL abstract syntax is depicted in Table 1. In Table 1, C denotes fuzzy class description (i.e., fuzzy DL concept); D denotes fuzzy data range (i.e., fuzzy DL concrete datatype); $E \in \{C, D\}$; R denotes fuzzy ObjectProperty identifier (i.e., fuzzy DL abstract role); T denotes fuzzy DatatypeProperty identifier (i.e., fuzzy DL concrete role); $P \in \{R , T\}$; d and o are abstract individuals; v is a concrete individual; $a \in \{d, v\}$; $\#S$ denotes the cardinality of a set S, and $\bowtie \in \{\geq, >, \leq, <\}$.

A fuzzy ontology can be represented by *fuzzy OWL language*. In brief, a fuzzy OWL ontology is a set of fuzzy OWL axioms in Table 1.

Table 1. Fuzzy OWL abstract syntax, fuzzy Description Logic (DL) syntax, and semantics.

Fuzzy OWL Abstract Syntax	**Fuzzy DL Syntax**	**Semantics**
Fuzzy class descriptions		
Class (A)	A	$A^{\mathrm{FI}} : \Delta^{\mathrm{FI}} \rightarrow [0,1]$
owl:Thing	$\top$	$\top^{\mathrm{FI}}(d) = 1$
owl:Nothing	$\bot$	$\bot^{\mathrm{FI}}(d) = 0$
intersectionOf ($C_1 \ldots C_n$)	$C_1 \sqcap \ldots \sqcap C_n$	$(C_1 \sqcap \ldots \sqcap C_n)^{\mathrm{FI}}(d) = \min\{C_1^{\mathrm{FI}}(d), \ldots, C_n^{\mathrm{FI}}(d)\}$
unionOf ($C_1 \ldots C_n$)	$C_1 \sqcup \ldots \sqcup C_n$	$(C_1 \sqcup \ldots \sqcup C_n)^{\mathrm{FI}}(d) = \max\{C_1^{\mathrm{FI}}(d), \ldots, C_n^{\mathrm{FI}}(d)\}$
complementOf (C)	$\neg C$	$(-C)^{\mathrm{FI}}(d) = 1\text{-}C^{\mathrm{FI}}(d)$
oneOf($o_1 \ldots o_n$)	$\{o_1\} \sqcup \ldots \sqcup \{o_n\}$	$(\{o_1\} \sqcup \ldots \sqcup \{o_n\})^{\mathrm{FI}}(d)$=1 if $d \in \{o_1^{\mathrm{FI}}, \ldots, o_n^{\mathrm{FI}}\}$, 0 otherwise
restriction (P someValuesFrom(E))	$\exists P.E$	$(\exists P.E)^{\mathrm{FI}}(d) = \sup_{a \in \Delta^{\mathrm{FI}}} \{\min\{P^{\mathrm{FI}}(d,a), E^{\mathrm{FI}}(a)\}\}$
restriction (P allValuesFrom(E))	$\forall P.E$	$(\forall P.E)^{\mathrm{FI}}(d) = \inf_{a \in \Delta^{\mathrm{FI}}} \{\max\{1 - P^{\mathrm{FI}}(d,a), E^{\mathrm{FI}}(a)\}\}$
restriction (P hasValue(a))	$\exists P.\{a\}$	$(\exists P.\{a\})^{\mathrm{FI}}(d) = P^{\mathrm{FI}}(d,a)$
restriction (P minCardinality(n))	$\geq nP$	$(\geq nP)^{\mathrm{FI}}(d) = \sup_{a_1,\ldots,a_n \in \Delta^{\mathrm{FI}}} \min(\min_{i=1}^{n} P^{\mathrm{FI}}(d,a_i), \min_{i<j}\{a_i \neq a_j\})$
restriction (P maxCardinality(n))	$\leq nP$	$(\leq nP)^{\mathrm{FI}}(d) = \inf_{a_1,\ldots,a_{n+1} \in \Delta^{\mathrm{FI}}} \max(\max_{i=1}^{n+1}(1 - P^{\mathrm{FI}}(d,a_i)), \max_{i<j}\{a_i = a_j\})$
restriction (P cardinality(n))	$=nP$	$(\geq nP \sqcap \leq nP)^{\mathrm{FI}}(d)$
Fuzzy class axioms		
Class (A partial $C_1 \ldots C_n$)	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathrm{FI}}(d) \leq \min\{ C_1^{\mathrm{FI}}(d), \ldots, C_n^{\mathrm{FI}}(d) \}$
Class (A complete $C_1 \ldots C_n$)	$A \equiv C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathrm{FI}}(d) = \min\{ C_1^{\mathrm{FI}}(d), \ldots, C_n^{\mathrm{FI}}(d) \}$
EnumeratedClass (A $o_1 \ldots o_n$)	$A \equiv o_1 \sqcup \ldots \sqcup o_n$	$A^{\mathrm{FI}}(d) = 1$ if $d \in \{o_1^{\mathrm{FI}}, \ldots, o_n^{\mathrm{FI}}\}$, $A^{\mathrm{FI}}(d) = 0$ otherwise
SubClassOf (C_1 C_2)	$C_1 \sqsubseteq C_2$	$C_1^{\mathrm{FI}}(d) \leq C_2^{\mathrm{FI}}(d)$
EquivalentClasses ($C_1 \ldots C_n$)	$C_1 \equiv \ldots \equiv C_n$	$C_1^{\mathrm{FI}}(d) = \ldots = C_n^{\mathrm{FI}}(d)$
DisjointClasses ($C_1 \ldots C_n$)	$C_i \neq C_j \quad 1 \leq i < j \leq n$	$\min\{C_i^{\mathrm{FI}}(d), C_j^{\mathrm{FI}}(d)\} = 0 \qquad 1 \leq i < j \leq n$
Fuzzy property axioms		
SubPropertyOf (P_1, P_2)	$P_1 \sqsubseteq P_2$	$P_1^{\mathrm{FI}}(d, a) \leq P_2^{\mathrm{FI}}(d, a)$
EquivalentProperties (P_1, ..., P_n)	$P_1 \equiv \ldots \equiv P_n$	$P_1^{\mathrm{FI}}(d, a) = \ldots = P_n^{\mathrm{FI}}(d, a)$
ObjectProperty (R super(R_1) ... super(R_n)	$R \sqsubseteq R_i$	$R^{\mathrm{FI}}(d_1, d_2) \leq R_i^{\mathrm{FI}}(d_1, d_2)$
domain (C_1)…domain (C_m)	$\exists R.\top \sqsubseteq C_i$	$R^{\mathrm{FI}}(d_1, d_2) \leq C_i^{\mathrm{FI}}(d_1)$
range (C_1)…range (C_k)	$\top \sqsubseteq \forall R.C_i$	$R^{\mathrm{FI}}(d_1, d_2) \leq C_i^{\mathrm{FI}}(d_2)$
[Functional])	$\top \sqsubseteq \leq 1\ R$	$\inf_{d_1,d_2 \in \Delta^{\mathrm{FI}}} \max(\max_{i=1}^{2}(1 - R^{\mathrm{FI}}(d,d_i)), d_1 = d_2) \geq 1$
[InverseOf(R_0)]	$R = (R_0)^-$	$R^{\mathrm{FI}}(d_1, d_2) = (R_0^-)^{\mathrm{FI}}(d_1, d_2)$
DatatypeProperty (T super(T_1) ... super(T_n)	$T \sqsubseteq T_i$	$T^{\mathrm{FI}}(d, v) \leq T_i^{\mathrm{FI}}(d, v)$
domain (C_1)…domain (C_m)	$\exists T.\top \sqsubseteq C_i$	$T^{\mathrm{FI}}(d, v) \leq C_i^{\mathrm{FI}}(d)$
range (D_1)…range (D_k)	$\top \sqsubseteq \forall T.D_i$	$1 \leq \inf_{v \in \Delta^{\mathrm{FI}}} \{\max\{1 - T^{\mathrm{FI}}(d,v), D_i^{\mathrm{FI}}(v)\}\}$
[Functional])	$\top \sqsubseteq \leq 1T$	$\inf_{v_1,v_2 \in \Delta^{\mathrm{FI}}} \max(\max_{i=1}^{2}(1 - T^{\mathrm{FI}}(d,v_i)), v_1 = v_2) \geq 1$
Fuzzy individual axioms		
Individual(o type(C_1)[$\ltimes m_1$]...type(C_{n1})[$\ltimes m_{n1}$]	$o : C_i \ltimes m_i$	$C_i^{\mathrm{FI}}(o) \ltimes m_i \qquad m_i \in [0, 1] \qquad 1 \leq i \leq n_1$
value(R_1, o_1)[$\ltimes k_1$]…value(R_{n2}, o_{n2})[$\ltimes k_{n2}$]	(o, o_i): $R_i \ltimes k_i$	$R_i^{\mathrm{FI}}(o, o_i) \ltimes k_i \qquad k_i \in [0, 1] \qquad 1 \leq i \leq n_2$
value(T_1, v_1) [$\ltimes l_1$]…value(T_{n3}, v_{n3}) [$\ltimes l_{n3}$])	(o, v_i): $T_i \ltimes l_i$	$T_i^{\mathrm{FI}}(o, v_i) \ltimes l_i \qquad l_i \in [0, 1] \qquad 1 \leq i \leq n_3$
SameIndividual ($o_1 \ldots o_n$)	$o_1 = \ldots = o_n$	$o_1^{\mathrm{FI}} = \ldots = o_n^{\mathrm{FI}}$
DifferentIndividuals ($o_1 \ldots o_n$)	$o_i \neq o_j \quad 1 \leq i < j \leq n$	$o_i^{\mathrm{FI}} \neq o_j^{\mathrm{FI}} \qquad 1 \leq i < j \leq n$

Copyright: the authors

3. Formalizations of Fuzzy UML Models and Fuzzy Ontologies

In order to well establish correspondences between fuzzy UML models and fuzzy ontologies, it is necessary to give formalization methods of fuzzy UML models and fuzzy ontologies. In this section, we propose formal definitions and semantic interpretations of fuzzy UML models and fuzzy ontologies, respectively.

3.1. *Formalization of Fuzzy UML Models*

In this section we first propose a complete formal definition of fuzzy UML models, where both the structural and dynamic aspects of fuzzy UML models are considered, and the most important features of fuzzy UML models are included. Furthermore, we give the semantic interpretation of fuzzy UML models. Finally, an example of a full fuzzy UML model is provided.

3.1.1. *Formal definition of fuzzy UML models*

The fuzzy UML model is a fuzzy extension of the traditional UML model. Some notions of fuzzy UML models have been introduced in Refs. 30, 32, 33, 34, 35, and 39, which differ in minor aspects in expressiveness and in notation. In this paper it is necessary for us to give a formalization of the fuzzy UML model which abstracts with respect to the most important features and allows us to develop the correspondence to the fuzzy ontology. In general, the most important features of fuzzy UML models are fuzzy object, fuzzy attribute, fuzzy class, fuzzy generalization, fuzzy aggregation, fuzzy dependency, and fuzzy association as mentioned in Section 2.2.

In the following we propose a complete formal definition of fuzzy UML models. Here, based on the definitions of fuzzy UML models[30,35,38], we further consider both the structural and dynamic aspects of fuzzy UML models, and also add several new functions and redefine some components for capturing some important features of fuzzy UML models (see Definition 1 in detail).

Firstly, for two finite sets X and Y we call a function from a subset of X to Y an *X-labeled tuple over Y*. The labeled tuple T that maps $x_i \in X$ to $y_i \in Y$, for $i \in \{1, \ldots, k\}$, is denoted $[x_1 : y_1, \ldots, x_k : y_k]$. We also write $T[x_i]$ to denote y_i.

Definition 1 (fuzzy UML models). A fuzzy UML model is a tuple $\mathcal{F}_{UML} = (\mathcal{L}_\mathcal{F}, \preccurlyeq_\mathcal{F}, att_\mathcal{F}, agg_\mathcal{F}, dep_\mathcal{F}, ass_\mathcal{F}, card_\mathcal{F}, mult_\mathcal{F}, mult'_\mathcal{F})$, where:

- $\mathcal{L}_\mathcal{F} = FO_\mathcal{F} \cup FA_\mathcal{F} \cup FM_\mathcal{F} \cup FT_\mathcal{F} \cup FC_\mathcal{F} \cup FH_\mathcal{F} \cup FG_\mathcal{F} \cup FD_\mathcal{F} \cup FS_\mathcal{F} \cup FR_\mathcal{F}$ is a finite *alphabet* partitioned into a set $FO_\mathcal{F}$ of *fuzzy object* symbols, a set $FA_\mathcal{F}$ of *fuzzy attribute* symbols (i.e., static attributes), a set $FM_\mathcal{F}$ of *fuzzy method* symbols (i.e., dynamic attributes), a set $FT_\mathcal{F}$ of *datatype* symbols, a set $FC_\mathcal{F}$ of *fuzzy class* symbols, a set $FH_\mathcal{F}$ of *fuzzy hierarchy* symbols, a set $FG_\mathcal{F}$ of *fuzzy aggregation* symbols, a set $FD_\mathcal{F}$ of *fuzzy dependency* symbols, a set $FS_\mathcal{F}$ of *fuzzy association* symbols, and a set $FR_\mathcal{F}$ of *role* symbols. Here it should be noted that:
  - A method in $FM_\mathcal{F}$ is the form of $f(P_1, \ldots, P_m) : R$, where f is the name of the method, $P_1,\ldots,P_m$ are types of m parameters, and R is the type of the result.
  - Each attribute $FA \in FA_\mathcal{F}$ is associated with a domain $FT \in FT_\mathcal{F}$, and each fuzzy domain symbol FT has an associated predefined basic domain and the various basic domains are usually assumed to be pairwise disjoint. A fuzzy keyword *FUZZY* is appeared on the front of attributes indicating these attributes are fuzzy attributes.
- $\preccurlyeq_\mathcal{F}(FC) = FC_1 \times FC_2 \times \ldots \times FC_n$ is a relation that models the *hierarchy* (several generalizations can be grouped together to form a class hierarchy) between a superclass FC and several subclasses $FC_1, \ldots, FC_n$. Moreover, there may be an optional constraint (*disjointness*, *completeness*) in the hierarchy relation.
- $att_\mathcal{F} : FC \rightarrow T(FA, FT)$ is a function that maps each *fuzzy class* symbol in $FC_\mathcal{F}$ to a *$FA_\mathcal{F}$-labeled tuple over $FT_\mathcal{F}$*, i.e., $att_\mathcal{F}(FC) \rightarrow [FA_1 : FT_1, \ldots, FA_n : FT_n]$.
- $agg_\mathcal{F}(FG) = FC \times (FC_1 \cup FC_2 \cup \ldots \cup FC_m)$ is a relation that models the *aggregation* between an aggregate class FC and a group of constituent classes FC_i, $i = 1, 2, \ldots, m$.
- $dep_\mathcal{F} \subseteq FC_1 \times FC_2$ is a binary relation over $FC_\mathcal{F}$, which models the *dependency* between a source class FC_1 and a target class FC_2.
- $ass_\mathcal{F} : FS \rightarrow T(FR, FC)$ is a function that maps each *fuzzy association* symbol in $FS_\mathcal{F}$ to a *$FR_\mathcal{F}$-labeled tuple over $FC_\mathcal{F}$*, i.e., $ass_\mathcal{F}(FS) = [FR_1 : FC_1, \ldots, FR_k : FC_k]$. The function $ass_\mathcal{F}$ actually associates a set of *roles* to each fuzzy association, determining implicitly also the arity of the fuzzy association. We assume without loss of generality that:
  - Each role is specific to exactly one fuzzy association, i.e., for two fuzzy associations FS,

$FS' \in FS_{\mathcal{F}}$ with $FS \neq FS'$, if $ass_{\mathcal{F}}(FS) = [FR_1 : FC_1, \ldots, FR_k : FC_k]$ and $ass_{\mathcal{F}}(FS') = [FR_1' : FC_1', \ldots, FR_k' : FC_k']$, where roles FR_i, $FR_i' \in FR_{\mathcal{F}}$, fuzzy classes FC_i, $FC_i' \in FC_{\mathcal{F}}$, and $i \in \{1, \ldots, k\}$, then $\{FR_1, \ldots, FR_k\} \cap \{FR_1', \ldots, FR_k'\} = \varnothing$.

 − For each role $FR \in FR_{\mathcal{F}}$ there is a fuzzy association $FS \in FS_{\mathcal{F}}$ and a fuzzy class $FC \in FC_{\mathcal{F}}$ such that $ass_{\mathcal{F}}(FS) = [\ldots, FR : FC, \ldots]$.

- $card_{\mathcal{F}}$ is a function from $FC_{\mathcal{F}} \times FS_{\mathcal{F}} \times FR_{\mathcal{F}}$ to $\mathbb{N}_0 \times (\mathbb{N}_0 \cup \{\infty\})$ that satisfies the following condition (where $\mathbb{N}_0$ denotes non-negative integers): for a fuzzy association $FS \in FS_{\mathcal{F}}$ such that $ass_{\mathcal{F}}(FS) = [FR_1 : FC_1, \ldots, FR_k : FC_k]$, defining $card_{\mathcal{F}}(FC_i, FS, FR_i) = (card_{\min}(FC_i, FS, FR_i), card_{\max}(FC_i, FS, FR_i))$. The function $card_{\mathcal{F}}$ is used to specify the *cardinality constraints* on the minimum and maximum number of times an object instance of a fuzzy class may participate in a fuzzy association via some role. If not stated otherwise, $card_{\min}(FC_i, FS, FR_i)$ is assumed to be 0 and $card_{\max}(FC_i, FS, FR_i)$ is assumed to be ∞.
- $mult_{\mathcal{F}}$ is a function from $FT_{\mathcal{F}} \times FC_{\mathcal{F}} \times FA_{\mathcal{F}}$ to $\mathbb{N}_1 \times (\mathbb{N}_1 \cup \{\infty\})$ (where $\mathbb{N}_1$ denotes positive integers). The function $mult_{\mathcal{F}}$ is used to specify *multiplicities*, i.e., constraints on the minimum and maximum number of values that an attribute of an object may have. In detail, for an attribute $FA_i \in FA_{\mathcal{F}}$ of type $FT_i \in FT_{\mathcal{F}}$ in a class $FC \in FC_{\mathcal{F}}$ such that $att_{\mathcal{F}}(FC) \to [FA_1 : FT_1, \ldots, FA_n : FT_n]$, defining $mult_{\mathcal{F}}(FT_i, FC, FA_i) = (mult_{\min}(FT_i, FC, FA_i), mult_{\max}(FT_i, FC, FA_i))$. If not stated otherwise, $mult_{\min}(FT_i, FC, FA_i)$ is assumed to be 1 and $mult_{\max}(FT_i, FC, FA_i)$ is also assumed to be 1, i.e., the attribute is mandatory and single-valued.
- $mult'_{\mathcal{F}}$ is a function from $FC_{\mathcal{F}} \times FC_{\mathcal{F}}$ to $\mathbb{N}_0 \times (\mathbb{N}_0 \cup \{\infty\})$, which is used to specify *multiplicities*, i.e., constraints on the minimum and maximum number of times that an object instance of a constituent class may participate in a fuzzy aggregation. In detail, for a fuzzy aggregation $agg_{\mathcal{F}}(FG) = FC \times (FC_1 \cup FC_2 \cup \ldots \cup FC_m)$, where $FC \in FC_{\mathcal{F}}$ is an aggregate class, $FC_i \in FC_{\mathcal{F}}$ is a constituent class, and $i \in \{1, \ldots, m\}$, defining $mult'_{\mathcal{F}}(FC_i, FC) = (mult'_{\min}(FC_i, FC), mult'_{\max}(FC_i, FC))$. If not stated otherwise, $mult'_{\min}(FC_i, FC)$ is assumed to be 0 and $mult'_{\max}(FC_i, FC)$ is assumed to be ∞.

From Definition 1, it is shown that the features of a fuzzy UML model mentioned in Section 2.2 can be represented in the formal definition. In the following section we give the semantics of fuzzy UML models.

3.1.2. *Semantics of fuzzy UML models*

In the following we give the semantic interpretation of fuzzy UML models. The *semantics* of a fuzzy UML model can be given by *fuzzy object states* (see Definition 2), i.e., by specifying which fuzzy object state is consistent with the information structure of the fuzzy UML model.

Definition 2. A fuzzy object state $\mathcal{FB}$ with respect to a fuzzy UML model $\mathcal{F}_{UML}$ is constituted by a nonempty finite set $\Delta^{\mathcal{FB}}$, and a function $\bullet^{\mathcal{FB}}$ that maps:

- Every domain symbol $FT \in FT_{\mathcal{F}}$ to a set $FT^{\mathcal{FB}} \in FT_{\mathcal{F}}^{\mathcal{FB}}$, where $FT_{\mathcal{F}}^{\mathcal{FB}}$ is a set of domains. For non-fuzzy attributes, each domain is associated with the basic types such as Integer, Real, and String; for fuzzy attributes, each fuzzy domain is a fuzzy set or a possibility distribution.
- Every fuzzy class $FC \in FC_{\mathcal{F}}$ to a membership degree function $FC^{\mathcal{FB}}: \Delta^{\mathcal{FB}} \to [0, 1]$. That is, each fuzzy class FC is mapped into a possibility distribution $\{o_1/u_1, \ldots, o_n/u_n\}$, where o_i is an object identifier denoting a real world object belonging to FC, and each o_i is associated with a membership degree u_i denoting the object o_i membership degree to the fuzzy class FC.
- Every fuzzy attribute $FA \in FA_{\mathcal{F}}$ or $FM_{\mathcal{F}}$ to a set $FA^{\mathcal{FB}} \subseteq \Delta^{\mathcal{FB}} \times \bigcup_{FT \in FT_{\mathcal{F}}} FT^{\mathcal{FB}}$.
- Every fuzzy association $FS \in FS_{\mathcal{F}}$ to a set $FS^{\mathcal{FB}}$ of $FR_{\mathcal{F}}$-labeled tuples over $\Delta^{\mathcal{FB}}$, i.e., $FS^{\mathcal{FB}} \subseteq T(FR, \Delta^{\mathcal{FB}})$.

The elements of $FC^{\mathcal{FB}}$, $FA^{\mathcal{FB}}$ and $FS^{\mathcal{FB}}$ are called *instances* of FC, FA, and FS respectively.

A fuzzy object state is considered *acceptable* if it satisfies all integrity constraints of the fuzzy UML model. This is captured by the definition of legal fuzzy object state (see Definition 3).

Definition 3. A fuzzy object state $\mathcal{FB}$ is said to be legal for a fuzzy UML model $\mathcal{F}_{UML}$, if it satisfies the following conditions:

- For each pair of fuzzy classes $FC_1, FC_2 \in FC_{\mathcal{F}}$ such that $\preccurlyeq_{\mathcal{F}}(FC_2) = FC_1$, it holds that $FC_1^{\mathcal{FB}} \subseteq FC_2^{\mathcal{FB}}$.
- For each fuzzy hierarchy $\preccurlyeq_{\mathcal{F}}(FC) = FC_1 \times FC_2 \times \ldots \times FC_n$, it hold that $FC_i^{\mathcal{FB}} \subseteq FC^{\mathcal{FB}}$; if there are disjointness and completeness constraints, then it follows $FC^{\mathcal{FB}} = FC_i^{\mathcal{FB}} \cup \ldots \cup FC_n^{\mathcal{FB}}$ and $FC_i^{\mathcal{FB}} \cap FC_j^{\mathcal{FB}} = \varnothing$, $i \neq j$, $i, j \in \{1, \ldots, n\}$.
- For each fuzzy class $FC \in FC_{\mathcal{F}}$, if $att_{\mathcal{F}}(FC) = [FA_1 : FT_1, \ldots, FA_n : FT_n]$, then for each instance $c \in FC^{\mathcal{FB}}$ and for each $i \in \{1, \ldots, n\}$ the following holds: (i) there is at least one element $a_i \in FA_i^{\mathcal{FB}}$ whose first component is c, and the second

component of a_i is an element $t_i \in FT_i^{\mathcal{FB}}$; (*ii*) $mult_{\min}(t_i, c, a_i) \leq \#\{c \in FC^{\mathcal{FB}} \mid c[a_i] = t_i\} \leq mult_{\max}(t_i, c, a_i)$, where #{ }denotes the base of set { }.

- For each fuzzy aggregation $FG \in FG_{\mathcal{F}}$ such that $agg_{\mathcal{F}}(FG) = FC \times (FC_1 \cup FC_2 \cup \ldots \cup FC_m)$, for each instance $c \in FC^{\mathcal{FB}}$ and $c_i \in FC_i^{\mathcal{FB}}$, where $i \in \{1, \ldots, m\}$, it holds that: $mult'_{\min}(c_i, c) \leq \#\{c \in FC^{\mathcal{FB}} \mid c[FG] = c_i\} \leq mult'_{\max}(c_i, c)$, where #{ }denotes the base of set { }.
- For each pair of fuzzy classes $FC_1, FC_2 \in FC_{\mathcal{F}}$ such that $dep_{\mathcal{F}} \subseteq FC_1 \times FC_2$, it holds that: if $FC_1^{\mathcal{FB}} = \varnothing$, then $FC_2^{\mathcal{FB}} = \varnothing$.
- For each fuzzy association $FS \in FS_{\mathcal{F}}$ such that $ass_{\mathcal{F}}(FS) = [FR_1 : FC_1, \ldots, FR_k : FC_k]$, all instances of FS are of the form $[r_1 : c_1, \ldots, r_k : c_k]$, where $r_i \in FR_i^{\mathcal{FB}}$, $c_i \in FC_i^{\mathcal{FB}}$, $i \in \{1, \ldots, k\}$.
- For each fuzzy association $FS \in FS_{\mathcal{F}}$ such that $ass_{\mathcal{F}}(FS) = [\ldots, FR : FC, \ldots]$, for each instance $c \in FC^{\mathcal{FB}}$, it holds that: $card_{\min}(FC, FS, FR) \leq \#\{s \in FS^{\mathcal{FB}} \mid s[FR] = c\} \leq card_{\max}(FC, FS, FR)$, where #{ }denotes the base of set { }.

3.1.3. *An example of fuzzy UML model*

In the following we provide an example of a full fuzzy UML model, which will be used to construct fuzzy ontology in the later sections.

Fig. 6 shows a graphic fuzzy UML model $\mathcal{F}_{\text{UML1}}$ (the fuzzy UML model is similar with the model in Ref. 30 and some new constraints such as *methods* and *multiplicities* are added). Furthermore, **Fig. 7** gives formal representation of the fuzzy UML model $\mathcal{F}_{\text{UML1}}$ in Fig. 6 according to the formal definition of fuzzy UML models (i.e., Definition 1 proposed in Section 3.1.1). The detailed instruction is as follows:

- A *fuzzy class* is denoted by using a *dashed-outline rectangle* to differentiate a classical class, e.g., *Old-Employee* as shown in Fig. 6. A fuzzy class may contain four parts, i.e., crisp attributes, fuzzy attributes, methods, or an additional attribute *u* as shown in Fig. 6, where:
  - A fuzzy keyword *FUZZY* is appeared in front of an attribute indicating the attribute may take fuzzy values. For example, *FUZZY* Age or *FUZZY* Email. Moreover, the *multiplicity* [1…∞] of the attribute "*FUZZY* Email [1…∞]" means that each object instance of the fuzzy class *Old-Employee* has at least one email, and possibly more.
  - An additional attribute $u \in [0, 1]$ in a fuzzy class is defined for representing an object membership degree to the fuzzy class.
  - The method *IsDepartment( ):String* returns a possibility distribution value {*Department/u*$_i$}, which denotes that an *Old-Employee* works in the *Department* with degree $u_i \in [0, 1]$. The type of the parameter is *null*.
- A *fuzzy generalization* is denoted by using a *dashed triangular arrowhead*, e.g., the class *Employee* is a generalization of classes *Young-Employee* and *Old-Employee*, and two classes *Young-Employee* and *Old-Employee* are *disjoint* and *the union of them* completely covers the class *Employee*.
- A *fuzzy dependency* is denoted by using a *dashed line with an arrowhead*, e.g., there is a fuzzy dependency relationship between the target class *Employee Dependent* and the source class *Employee*.
- A *fuzzy association* is denoted by using a *single line with an arrowhead* and the association class is connected with the association by a *dashed-outline*, e.g., *Use* is a fuzzy association class between two classes *Old-Employee* and *New-Computer*. Here:
  - *Date* is an attribute of the association class *Use*, which describes the start date that an *Old-Employee* uses a *New-Computer*.
  - The additional attribute β denotes the degree of possibility that an association relationship occurs in *n* classes.
  - The participation of a class in a fuzzy association is called a *role* which has a unique name. For example, *Uby* and *Uof*. The cardinality constraints (1, 1) and (1, 2) denote that each *Old-Employee* can use at least 1 and at most 2 *New-Computers* and each *New-Computer* can be used by exactly one *Old-Employee*.
- A *fuzzy aggregation* is denoted by using a *dashed open diamond*, e.g., the class *New-Computer* is aggregated by *Monitor*, *New CPU box*, and *Keyboard*. The *multiplicity* [m_i, n_i] specifies that each instance of the aggregate class consists of at least m_i and at most n_i instances of the *i*-th constituent class. For example, a *New-Computer* may contain at least one *Monitor*, and possibly more.

Copyright: the authors

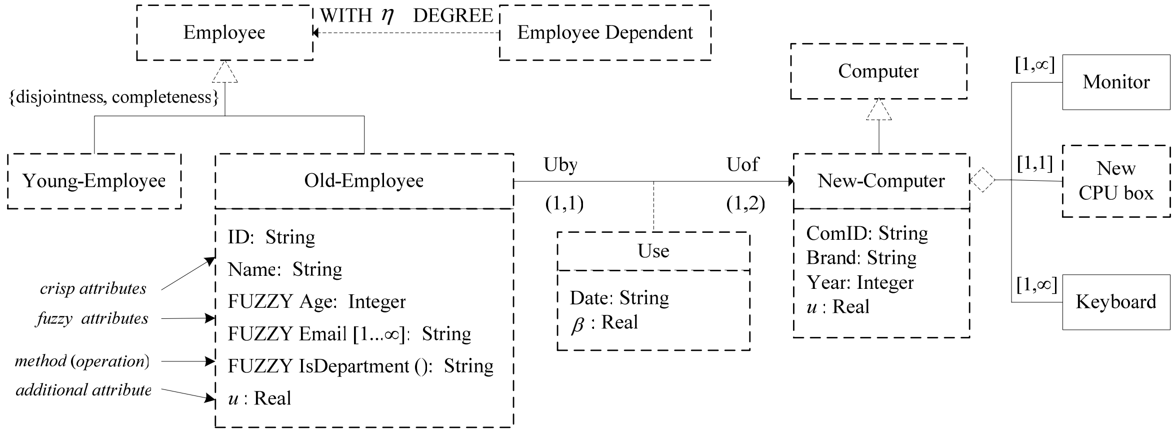


Fig. 6. A graphic fuzzy UML model $\mathcal{F}_{\text{UML1}}$.

According to Definition 1, the formal syntax of the fuzzy UML model $\mathcal{F}_{\text{UML1}}$ in Fig. 6 is as follows:

$\mathcal{F}_{\text{UML1}} = (\mathcal{L}_{\mathcal{F}}, \preccurlyeq_{\mathcal{F}}, att_{\mathcal{F}}, agg_{\mathcal{F}}, dep_{\mathcal{F}}, ass_{\mathcal{F}}, card_{\mathcal{F}}, mult_{\mathcal{F}}, mult'_{\mathcal{F}})$, where

$\mathcal{L}_{\mathcal{F}}$ is a set of symbols partitioned into:

$FC_{\mathcal{F}}$ = {Employee, Young-Employee, Old-Employee, Computer, New-Computer, …, Keyboard}

$FA_{\mathcal{F}}$ = {ID, Name, FUZZY Age, FUZZY Email, Date, u, β, ComID, Brand, Year}

$FT_{\mathcal{F}}$ = {String, Integer, Real}

$FM_{\mathcal{F}}$ = {FUZZY IsDepartment()}

…

$FR_{\mathcal{F}}$ = {Uby, Uof}

The set of functions/relations over the above symbols consists of:

$att_{\mathcal{F}}$(Old-Employee) = [ID: String, Name: String, FUZZY Age: Integer, FUZZY Email: String,
FUZZY IsDepartment(): String, u: Real]

$att_{\mathcal{F}}$(New-Computer) = [ComID: String, Brand: String, Year: Integer, u : Real]

$att_{\mathcal{F}}$(Use) = [Date: String, β : Real]

$\preccurlyeq_{\mathcal{F}}$(Employee) = Young-Employee × Old-Employee (*disjointness*, *completeness*)

$\preccurlyeq_{\mathcal{F}}$(Computer) = New-Computer

$agg_{\mathcal{F}}(FG)$ = New-Computer × (Monitor ∪ New CPU box ∪ Keyboard)

$dep_{\mathcal{F}} \subseteq$ Employee × Employee-Department

$ass_{\mathcal{F}}$(Use) = [Uby : Old-Employee, Uof : New-Computer]

$mult_{\mathcal{F}}$(String, Old-Employee, FUZZY Email) = (1, ∞)

$mult'_{\mathcal{F}}$(Monitor, New-Computer) = (1, ∞)

$mult'_{\mathcal{F}}$(New CPU box, New-Computer) = (1, 1)

$mult'_{\mathcal{F}}$(Keyboard, New-Computer) = (1, ∞)

$card_{\mathcal{F}}$(Old-Employee, Use, Uby) = (1, 1)

$card_{\mathcal{F}}$(New-Computer, Use, Uof) = (1, 2).

Fig. 7. The formal syntax of the fuzzy UML model $\mathcal{F}_{\text{UML1}}$ in Fig. 6.

Fig. 8 gives a fuzzy UML instantiation (i.e., object diagram) with respect to the fuzzy UML model $\mathcal{F}_{\text{UML1}}$ in Fig. 6, which includes some objects and the detailed information of several objects only. Here:

- *Employee* is a generalization of classes *Young-Employee* and *Old-Employee* with *disjointness* and *completeness* constraints as shown in Fig. 6. Therefore, the union of object instances of two subclasses *Young-Employee* and *Old-Employee* completely covers the object instances of the superclass *Employee*, and the membership degree that any object belongs to the subclass must be less than or equal to the membership degree that it belongs to the superclass as shown in Fig. 8.
- The attributes such as *Age* and *Email* are fuzzy, e.g., the age of *Tom* is represented by a possibility distribution: {51/0.7}.
- u = 0.6 denotes that the object o_1' belongs to the class *Old-Employee* with degree 0.6.
- β = 0.7 denotes that the object o_1' uses the object o_3 with degree 0.7. Here, the attribute *Date* of the association class *Use* is omitted.

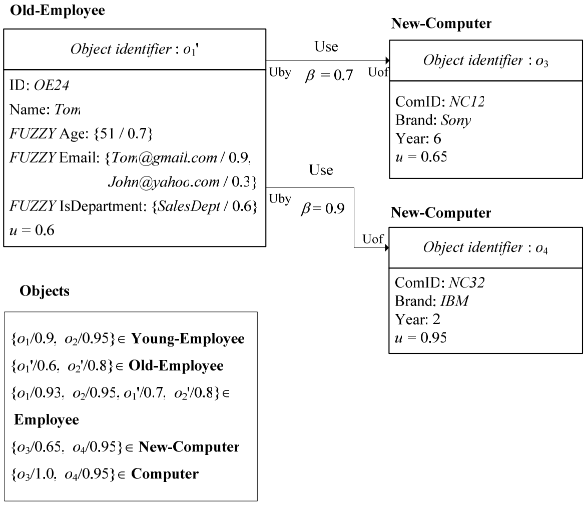

Old-Employee
Object identifier : o_1'
ID: OE24
Name: Tom
FUZZY Age: {51 / 0.7}
FUZZY Email: {Tom@gmail.com / 0.9, John@yahoo.com / 0.3}
FUZZY IsDepartment: {SalesDept / 0.6}
u = 0.6
Use
Uby
β = 0.7
Uof
Use
Uby
β = 0.9
Uof
New-Computer
Object identifier : o_3
ComID: NC12
Brand: Sony
Year: 6
u = 0.65
New-Computer
Object identifier : o_4
ComID: NC32
Brand: IBM
Year: 2
u = 0.95
Objects
{o_1/0.9, o_2/0.95} ∈ Young-Employee
{o_1'/0.6, o_2'/0.8} ∈ Old-Employee
{o_1/0.93, o_2/0.95, o_1'/0.7, o_2'/0.8} ∈ Employee
{o_3/0.65, o_4/0.95} ∈ New-Computer
{o_3/1.0, o_4/0.95} ∈ Computer

Fig. 8. A fragment of a fuzzy UML instantiation (i.e., object diagram) w.r.t. the fuzzy UML model $\mathcal{F}_{\text{UML1}}$ in Fig. 6.

As mentioned in Section 3.1.2, the semantics of a fuzzy UML model can be given by legal fuzzy object states. Based on the definition of semantics of fuzzy UML models in Section 3.1.2 and the fuzzy UML instantiation in Fig. 8, now let us explain some conditions of the legal fuzzy object state for a fuzzy UML model using several elements of the fuzzy UML model $\mathcal{F}_{\text{UML1}}$ in Fig. 7. With regard to the condition (1) in Definition 3 of Section 3.1.2, for pair of fuzzy classes *New-Computer* and *Computer* such that $\preccurlyeq_{\mathcal{F}}$(*Computer*) = *New-Computer* as shown in Fig. 7 (i.e., *New-Computer* is a subclass of *Computer*), from Fig .8 we can know that *New-Computer*$^{\mathcal{FB}}$ = {o_3/0.65, o_4/0.95} and *Computer*$^{\mathcal{FB}}$ = {o_3/1.0, o_4/0.95}, thus it is shown that $\mathcal{FB}$ satisfies *New-Computer*$^{\mathcal{FB}}$ ⊆ *Computer*$^{\mathcal{FB}}$, that is, $\mathcal{FB}$ is a legal fuzzy object state of the fuzzy UML model $\mathcal{F}_{\text{UML1}}$ in Fig. 7, and this is consistent with the condition (1) in Definition 3. With regard to the condition (6) in Definition 3, for fuzzy association *Use* such that $ass_{\mathcal{F}}$(*Use*) = [*Uby* : *Old-Employee*, *Uof* : *New-Computer*] as shown in Fig. 7, from Fig .8 we know that $ass_{\mathcal{F}}$(*Use*) = [*Uby* : o_1', *Uof* : o_3] and $ass_{\mathcal{F}}$(*Use*) = [*Uby* : o_1', *Uof* : o_4], that is, the instances of *Use* are of the form [*Uby* : o_i, *Uof* : o_j], where o_i ∈ *Old-Employee*$^{\mathcal{FB}}$, o_j ∈ *New-Computer*$^{\mathcal{FB}}$, and this is consistent with the condition (6) in Definition 3. The other conditions of the legal fuzzy object state in Definition 3 can be illustrated analogously following the procedures above.

3.2. *Formalization of Fuzzy Ontologies*

In the following we give a formal definition of fuzzy ontologies for representing the constructed fuzzy ontology knowledge from fuzzy UML models, and the formal definition considers as many fuzzy ontology constructors as possible that usually mentioned in the literature.

A fuzzy ontology, which is simply an ontology that uses fuzzy logic to provide a natural representation of imprecise and uncertain knowledge, may be usually represented by fuzzy OWL language (a fuzzy extension of the standard Web Ontology Language OWL) as mentioned in Section 2.3. A fuzzy ontology formulated in fuzzy OWL language is called fuzzy OWL ontology. Currently, there have been several definitions of fuzzy ontology by considering different features in Refs. 18, 21, 23, 30, 31, 44, 45, etc (a list of definitions is summarized in Ref. 46 in detail). In summary, there are three categories of definitions, a group of the existing definitions (e.g., Refs. 18, 23) formalize the notion of fuzzy ontologies by means of an enumeration of the elements of fuzzy ontologies (e.g., a fuzzy ontology is defined as a quintuple (C, R, T, A, X)[23], where C is a set of fuzzy concepts, R is a set of fuzzy relations between fuzzy concepts, T is a concept hierarchy, A is a set of non-taxonomic fuzzy associative relationships, and X is a set of fuzzy axioms), and these definitions do not allow other fuzzy elements than the explicitly mentioned such as individuals and fuzzy role hierarchy; *Moreover*, several definitions of fuzzy ontologies are proposed based on the language fuzzy OWL (e.g., Refs. 30, 31, and 45), but there are still some features which cannot be represented in these definitions such as individuals or fuzzy individual axioms; *In addition*, some definitions of fuzzy ontologies are proposed for particular application domains (e.g., Refs. 21, 44), and these definitions are even more restrictive and are considered as the minimal extensions which are sufficient to cover some particular applications such as news summarization in Ref. 44. As mentioned in the literature, we also understand that a universal fuzzy ontology definition which can be suitable for each application domain is difficult. All of the definitions above are not well-suited and not sufficient for representing both the structure and instance information of fuzzy UML models. Therefore, based on the existing definitions above, the following Definition 4 gives a formal definition of fuzzy OWL ontologies, which considers both the structure and instance information of fuzzy ontologies. Here, we want to stress out that we do not expect our fuzzy ontology definition to become a standard now, but it considers most of general and important features of fuzzy ontologies as usually mentioned in the literature.

Definition 4 (fuzzy OWL ontologies). A fuzzy OWL ontology $\mathcal{FO}$, which consists of the fuzzy ontology structure $\mathcal{FO}_S$ and the fuzzy ontology instance $\mathcal{FO}_I$ defined over $\mathcal{FO}_S$, can be defined as a tuple $\mathcal{FO} = (\mathcal{FO}_S, \mathcal{FO}_I)$:

(1) $\mathcal{FO}_S = FID_0 \cup FAxiom_0$, where $FID_0 = FCID_0 \cup FDRID_0 \cup FOPID_0 \cup FDPID_0$ is a set of *fuzzy class descriptions*, and $FAxiom_0$ is a set of *fuzzy class and property axioms* defined over FID_0 (see Table 1 in Section 2.3):

- $FCID_0$ is a set of fuzzy class identifiers. Each fuzzy class may be a user-defined fuzzy class or one of two predefined fuzzy classes owl: Thing and owl: Nothing;
- $FDRID_0$ is a set of fuzzy datatype identifiers. Each fuzzy datatype may be a predefined fuzzy XML Schema datatype[47];
- $FOPID_0$ is a set of fuzzy object property identifiers. Each fuzzy object property links individuals to individuals, and each property may have its characters (e.g., Symmetric and InverseOf) and its restrictions (e.g., someValuesFrom and Cardinality);
- $FDPID_0$ is a set of fuzzy datatype property identifiers. Each fuzzy datatype property links individuals to data values, and the domain of the property may be a fuzzy datatype in $FDRID_0$. Each fuzzy datatype property also may have its characters and restrictions;
- $FAxiom_0$, which is a set of *fuzzy class and property axioms* defined over FID_0, is used to represent the fuzzy ontology structure information.

(2) $\mathcal{FO}_I = FIID_0 \cup FAxiom_0$, where $FIID_0$ is a set of *individual* identifiers, and $FAxiom_0$ is a set of *fuzzy individual axioms* (see Table 1 in Section 2.3):

- $FIID_0$ is a set of individual identifiers;
- $FAxiom_0$ is a set of *fuzzy individual axioms* to represent the fuzzy ontology instance information.

In summary, a fuzzy OWL ontology $\mathcal{FO}$ includes two parts: the structure (a set of fuzzy class and property axioms defined over fuzzy class descriptions in Table 1) and the instance associated with the structure (a set of fuzzy individual axioms in Table 1). If a fuzzy interpretation $FI = (\Delta^{FI}, \Delta_D, \bullet^{FI}, \bullet^{D})$ mentioned in Section 2.3 satisfies all axioms in a fuzzy ontology $\mathcal{FO}$, then we say that FI is a model of $\mathcal{FO}$.

Based on the formalization of fuzzy UML models and fuzzy OWL ontologies above, in the following sections we study how to establish correspondences between the two knowledge representation models, i.e., how to construct fuzzy OWL ontologies from fuzzy UML models (see Section 4), and how to apply the constructed fuzzy OWL ontologies to reason on fuzzy UML models in a intelligent and automatic way (see Section 5).

4. Construction of Fuzzy OWL Ontologies from Fuzzy UML Models

This section proposes a formal approach and develops an automated tool for constructing fuzzy OWL ontologies from fuzzy UML models, including: **(*i*)** transforming a fuzzy UML model into fuzzy OWL ontology structure (see **Section 4.1**). **(*ii*)** transforming a fuzzy UML instantiation (w.r.t. the fuzzy UML model) into fuzzy OWL ontology instance (w.r.t. the fuzzy OWL ontology structure) (see **Section 4.2**). The fuzzy ontology structure and instance form a target fuzzy OWL ontology; The correctness of the approach is proved and a transformation example is provided; (***iii***) implementing a prototype transformation tool called *FUML2FOnto*, which can construct fuzzy OWL ontologies from fuzzy UML models (see **Section 4.3**).

4.1. *Transforming Fuzzy UML Model into Fuzzy OWL Ontology Structure*

In the following, Definition 5 *first* proposes a formal approach for transforming a fuzzy UML model into the fuzzy OWL ontology structure. *Then*, Theorem 1 proves the correctness of the approach. *Finally*, we provide a transformation example. All of these will help to well understand how to construct fuzzy OWL ontologies from fuzzy UML models.

Giving a fuzzy UML model $\mathcal{F}_{UML} = (\mathcal{L}_{\mathcal{F}}, \preccurlyeq_{\mathcal{F}}, att_{\mathcal{F}}, agg_{\mathcal{F}}, dep_{\mathcal{F}}, ass_{\mathcal{F}}, card_{\mathcal{F}}, mult_{\mathcal{F}}, mult'_{\mathcal{F}})$, starting with the construction of *fuzzy OWL class descriptions* from the alphabet set $\mathcal{L}_{\mathcal{F}}$, Definition 5 induces *a set of fuzzy class and property axioms* from the rest of the fuzzy UML model $\mathcal{F}_{UML}$.

Definition 5 (structure transformation). Given a fuzzy UML model $\mathcal{F}_{UML} = (\mathcal{L}_{\mathcal{F}}, \preccurlyeq_{\mathcal{F}}, att_{\mathcal{F}}, agg_{\mathcal{F}}, dep_{\mathcal{F}}, ass_{\mathcal{F}}, card_{\mathcal{F}}, mult_{\mathcal{F}}, mult'_{\mathcal{F}})$. The fuzzy OWL ontology structure $\mathcal{FO}_S = \varphi(\mathcal{F}_{UML})$ can be derived by transformation function φ as shown in Table 2.

Table 2. Transforming rules from a fuzzy UML model to fuzzy OWL ontology structure.

Fuzzy UML model $\mathcal{F}_{UML}$	**Fuzzy OWL ontology structure $\mathcal{FO}_S = \varphi(\mathcal{F}_{UML}) = (FID_0, FAxiom_0)$**
Alphabet set $\mathcal{L}_{\mathcal{F}}$	Fuzzy OWL class description set FID_0
Each fuzzy class $FC \in FC_{\mathcal{F}}$	A fuzzy class identifier $\varphi(FC) \in FCID_0$
Each fuzzy attribute symbol $FA \in FA_{\mathcal{F}}$	A fuzzy datatype property identifier $\varphi(FA) \in FDPID_0$
Each method without parameters $f()$: $R \in FM_{\mathcal{F}}$	A fuzzy datatype property identifier $\varphi(f) \in FDPID_0$ A fuzzy data range identifier $\varphi(R) \in FDRID_0$
Each method with parameters $f(P_1,\ldots,P_m)$: $R \in FM_{\mathcal{F}}$	A fuzzy class identifier $\varphi(FC_{f(P1,\ \ldots,\ Pm)}) \in FCID_0$ m fuzzy data range identifiers $\varphi(P_1)\ldots\varphi(P_m) \in FDRID_0$ A fuzzy data range identifier $\varphi(R) \in FDRID_0$ A fuzzy object property identifier $\varphi(r_1) \in FOPID_0$ (m+1) fuzzy datatype property identifiers $\varphi(r_2)\ldots\varphi(r_{m+2}) \in FDPID_0$ *Note that*: $\varphi(r_1)$ represents the object of invocation, the next m $\varphi(r_2)\ldots\varphi(r_{m+1})$ represent the parameters, and the last one $\varphi(r_{m+2})$ represents the return result.
Each fuzzy datatype symbol $FT \in FT_{\mathcal{F}}$	A fuzzy data range identifier $\varphi(FT) \in FDRID_0$
Each fuzzy association (class) symbol $FS \in FS_{\mathcal{F}}$	A fuzzy class identifier $\varphi(FS) \in FCID_0$
Each role symbol $FR \in FR_{\mathcal{F}}$	A pair of inverse fuzzy object property identifiers: $\varphi(FR) \in FOPID_0$ and v = invof_$\varphi(FR) \in FOPID_0$
Each fuzzy aggregation $FG \in FG_{\mathcal{F}}$ such that $agg_{\mathcal{F}}(FG) = FC \times (FC_1 \cup \ldots \cup FC_m)$	m pairs of fuzzy object property identifiers: φ(is_part_of_g_1), φ(is_whole_of_g_1), …, φ(is_part_of_g_m), φ(is_whole_of_g_m)
Each extensionally defined class $FC \in FC_{\mathcal{F}}$ such that $FC = \{o_1, \ldots, o_n\}$, where $o_i \in FO_{\mathcal{F}}$	n fuzzy individual identifiers $\varphi(o_1), \ldots, \varphi(o_n) \in FIID_0$
Constraints	Fuzzy OWL axiom set $FAxiom_0$
Each enumeration class $FC = \{o_1, \ldots, o_n\}$, where $FC \in FC_{\mathcal{F}}$, $o_i \in FO_{\mathcal{F}}$	Creating a fuzzy class axiom: EnumeratedClass ($\varphi(FC)$ $\varphi(o_1)\ldots\varphi(o_n)$) .
Each fuzzy generalization $\preccurlyeq_{\mathcal{F}}(FC_2) = FC_1$, where $FC_1, FC_2 \in FC_{\mathcal{F}}$	Creating a fuzzy class axiom: Class ($\varphi(FC_1)$ partial $\varphi(FC_2)$) or SubClassOf ($\varphi(FC_1)$ $\varphi(FC_2)$).
Each fuzzy class hierarchy $\preccurlyeq_{\mathcal{F}}(FC) = FC_1 \times FC_2 \times \ldots \times FC_n$, i.e., a fuzzy class FC generalizing n fuzzy classes $FC_1, \ldots, FC_n$ with the optional *disjointness* and *completeness* constraints	Creating fuzzy class axioms: Class ($\varphi(FC_i)$ partial $\varphi(FC)$) or SubClassOf ($\varphi(FC_i)$ $\varphi(FC)$) ; If there are the *disjointness* and *completeness* constraints, then adding fuzzy class axioms: EquivalentClasses ($\varphi(FC)$ unionOf ($\varphi(FC_1), \ldots, \varphi(FC_n)$)) DisjointClasses ($\varphi(FC_i), \varphi(FC_j)$) $i,j \in 1, 2, \ldots, n,\ i \neq j$.

Copyright: the authors

Each fuzzy class $FC \in FC_{\mathcal{F}}$ such that $att_{\mathcal{F}}(FC) \to [FA_1\ (p_1...q_1): FT_1, \ldots, FA_n\ (p_n...q_n): FT_n]$. Here, $FA_i \in FA_{\mathcal{F}}$, $FT_i \in FT_{\mathcal{F}}$, $(p_i...q_i)$ is the multiplicity $mult_{\mathcal{F}}(FT_i, FC, FA_i) = (\ mult_{min}(FT_i, FC, FA_i), mult_{max}(FT_i, FC, FA_i)\ )$ as mentioned in Definition 1, and $i \in \{1\ldots n\}$	Creating a fuzzy class axiom: Class (φ(FC) partial restriction (φ(FA_1) allValuesFrom (φ(FT_1)) minCardinality (p_1) maxCardinality (q_1)) … restriction (φ(FA_n) allValuesFrom (φ(FT_n)) minCardinality (p_n) maxCardinality (q_n))) ; Creating fuzzy property axioms: DatatypeProperty (φ(FA_i) domain (φ(FC)) range (φ(FT_i))) .
Each fuzzy aggregation $FG \in FG_{\mathcal{F}}$ such that $agg_{\mathcal{F}}(FG) = FC \times (FC_1\ [w_1...z_1] \cup \ldots \cup FC_m\ [w_m...z_m])$. Here, $FC, FC_i \in FC_{\mathcal{F}}$, $(w_i...z_i)$ is the multiplicity $mult'_{\mathcal{F}}(FC_i, FC) = (\ mult'_{min}(FC_i, FC), mult'_{max}(FC_i, FC)\ )$ as mentioned in Definition 1, and $i \in \{1\ldots m\}$	Creating a fuzzy axioms: Class (owl:Thing partial restriction (inverseOf φ(is_part_of_g_1) allValuesFrom (φ(FC)) Cardinality (1)) restriction (φ(is_whole_of_g_1) allValuesFrom (φ(FC_1)) minCardinality (w_1) maxCardinality (z_1)) … restriction (inverseOf φ(is_part_of_g_m) allValuesFrom (φ(FC)) Cardinality (1)) restriction (φ(is_whole_of_g_m) allValuesFrom (φ(FC_m)) minCardinality (w_m) maxCardinality (z_m))) ; ObjectProperty (φ(is_part_of_g_i) domain (φ(FC_i)) range (φ(FC))) ; ObjectProperty (φ(is_whole_of_g_i) domain (φ(FC)) range (φ(FC_i))) .
Each fuzzy dependency such that $dep_{\mathcal{F}} \subseteq FC_1 \times FC_2$	As mentioned in Section 2.2, the dependency between FC_1 and FC_2 is only related to the classes themselves and does not require a set of instances for its meaning, and the current fuzzy OWL ontologies are not enough to support this type of property, which can be indicated explicitly by the designers.
Each pair of symbols $x, y \in FC_{\mathcal{F}} \cup FS_{\mathcal{F}}$ with $x \neq y$ and $x \in FS_{\mathcal{F}}$	Creating a fuzzy class axiom: DisjointClasses (φ(x), φ(y)) .
Each fuzzy class $FC \in FC_{\mathcal{F}}$ such that $att_{\mathcal{F}}(FC) \to [\ldots, f() : R, \ldots]$. Here, $f()$: $R \in FM_{\mathcal{F}}$ is a method without parameters.	Creating a fuzzy class axiom: Class (φ(FC) partial restriction (φ(f) allValuesFrom (φ(R)) maxCardinality(1))).
Each fuzzy class $FC \in FC_{\mathcal{F}}$ such that $att_{\mathcal{F}}(FC) \to [\ldots, f(P_1, \ldots, P_m) : R, \ldots]$. Here, $f(P_1, \ldots, P_m) : R \in FM_{\mathcal{F}}$ is a method with m parameters $P_1, \ldots, P_m$.	Creating fuzzy class axioms: Class (φ($FC_{f(P1, \ldots, Pm)}$) partial restriction (φ(r_1) someValuesFrom (owl:Thing) Cardinality(1)) … restriction (φ(r_{m+1}) someValuesFrom (owl:Thing) Cardinality(1))) ; Class (φ($FC_{f(P1, \ldots, Pm)}$) partial restriction (φ(r_2) allValuesFrom (φ(P_1))) … restriction (φ(r_{m+1}) allValuesFrom (φ(P_m)))) ; Class (φ(FC) partial restriction (inverseOf(φ(r_1)) allValuesFrom (unionOf (complementOf (φ($C_{f(P1, \ldots, Pm)}$)) restriction (φ(r_{m+2}) allValuesFrom (φ(R))))))). *Note that*: The first axiom states that each instance of φ($FC_{f(P1, \ldots, Pm)}$), representing a tuple, correctly is connected to exactly one object for each of $r_1,\ldots r_{m+1}$; The other two axioms impose the correct typing of parameters and of the return value.
Each fuzzy association $FS \in FS_{\mathcal{F}}$ such that $ass_{\mathcal{F}}(FS) = [FR_1 : FC_1, \ldots, FR_k : FC_k]$. Here, $FR_i \in FR_{\mathcal{F}}$, $FC_i \in FC_{\mathcal{F}}$, each role FR_i is associated with a *cardinality constraint* $card_{\mathcal{F}}(FC_i, FS, FR_i) = (\ card_{min}(FC_i, FS, FR_i), card_{max}(FC_i, FS, FR_i)\ )$, and $i \in \{1\ldots\mathrm{k}\}$	Creating fuzzy class axioms: Class (φ(FS) partial restriction (φ(FR_1) allValuesFrom (φ(FC_1)) cardinality (1)) … restriction (φ(FR_k) allValuesFrom (φ(FC_k)) cardinality (1))) ; Class (φ(FC_i) partial restriction (v_i allValuesFrom (φ(FS)))), where v_i = invof_φ(FR_i) ∈ $FOPID_0$ denotes an inverse property of the fuzzy object property φ(FR_i) as mentioned above ; Do case $card_{\mathcal{F}}(FC_i, FS, FR_i)$ of: (i) $m = card_{min}(FC_i, FS, FR_i) \neq 0$, creating a fuzzy class axiom: Class (φ(FC_i) partial restriction (v_i minCardinality (m))) ; (ii) $n = card_{max}(FC_i, FS, FR_i) \neq \infty$, creating a fuzzy class axiom: Class (φ(FC_i) partial restriction (v_i maxCardinality (n))) ; (iii) $q = card_{\mathcal{F}}(FC_i, FS, FR_i)$, creating a fuzzy class axiom: Class (φ(FC_i) partial restriction (v_i Cardinality (q))) ; Creating fuzzy property axioms: ObjectProperty (v_i domain (φ(FC_i)) range (φ(FS)) inverseOf φ(FR_i)) ; ObjectProperty (φ(FR_i) domain (φ(FS)) range (φ(FC_i))) .

Below we prove the correctness of the transformation in Definition 5, which can be sanctioned by establishing mappings between the *fuzzy object states* (w.r.t. the fuzzy UML model $\mathcal{F}_{UML}$) and the *models* of the transformed fuzzy OWL ontology $\varphi(\mathcal{F}_{UML})$ (see Theorem 1).

Theorem 1. *For each fuzzy UML model $\mathcal{F}_{UML}$, $\mathcal{FB}$ is a legal fuzzy object state corresponding to $\mathcal{F}_{UML}$, and $\varphi(\mathcal{F}_{UML})$ is the fuzzy OWL ontology derived from the $\mathcal{F}_{UML}$ by Definition 5. Then there exist two mappings $\alpha_{\mathcal{F}}$, from the fuzzy object state $\mathcal{FB}$ to the model of $\varphi(\mathcal{F}_{UML})$, and $\beta_{\mathcal{F}}$, from the model of $\varphi(\mathcal{F}_{UML})$ to the fuzzy object state $\mathcal{FB}$, such that:*

- *For each legal fuzzy object state $\mathcal{FB}$ for $\mathcal{F}_{UML}$, there is $\alpha_{\mathcal{F}}(\mathcal{FB})$ which is a model of $\varphi(\mathcal{F}_{UML})$.*
- *For each model $\mathcal{FI}$ of $\varphi(\mathcal{F}_{UML})$, there is $\beta_{\mathcal{F}}(\mathcal{FI})$ which is a legal fuzzy object state for $\mathcal{F}_{UML}$.*

Proof. The following gives the proof of the first part of Theorem 1. Here, as mentioned in Section 3.1, for each legal fuzzy object state $\mathcal{FB}$, a finite set of fuzzy values $\Delta^{\mathcal{FB}}$ is considered. Moreover, for ease of understanding the following proof omits the membership degrees which occur at the fuzzy UML model as mentioned in Section 3.1. Firstly, given a legal fuzzy object state $\mathcal{FB}$ for $\mathcal{F}_{UML}$, the fuzzy interpretation $\alpha_{\mathcal{F}}(\mathcal{FB})$ of $\varphi(\mathcal{F}_{UML})$ can be defined as follows:

- The domain elements $\Delta^{\alpha_{\mathcal{F}}(\mathcal{FB})}$ of the fuzzy interpretation $\alpha_{\mathcal{F}}(\mathcal{FB})$ of $\varphi(\mathcal{F}_{UML})$ are constituted by the values $\Delta^{\mathcal{FB}}$ of the fuzzy object state $\mathcal{FB}$.
- The fuzzy OWL class description set FID_0 of $\varphi(\mathcal{F}_{UML})$ in Definition 5 are defined as follows:
  - For each symbol $X \in FT_{\mathcal{F}} \cup FA_{\mathcal{F}} \cup FC_{\mathcal{F}} \cup FS_{\mathcal{F}}$, $(\varphi(X))^{\alpha_{\mathcal{F}}(\mathcal{FB})} = X^{\mathcal{FB}}$;
  - For each fuzzy association $FS \in FS_{\mathcal{F}}$ with $ass_{\mathcal{F}}(FS) = [FR_1 : FC_1, \ldots, FR_k : FC_k]$, $(\varphi(FR_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})} = \{<s, c_i> \in \Delta^{\alpha_{\mathcal{F}}(\mathcal{FB})} \times \Delta^{\alpha_{\mathcal{F}}(\mathcal{FB})} \mid s \in FS^{\mathcal{FB}} \wedge c_i \in FC_i^{\mathcal{FB}} \wedge s[FR_i] = c_i\}$, where $i = 1, \ldots, k$.

Based on the definition of $\alpha_{\mathcal{F}}(\mathcal{FB})$ above, in the following we prove that $\alpha_{\mathcal{F}}(\mathcal{FB})$ is a model of $\varphi(\mathcal{F}_{UML})$, i.e., prove that $\alpha_{\mathcal{F}}(\mathcal{FB})$ satisfies each fuzzy OWL axiom of $FAxiom_0$ of $\varphi(\mathcal{F}_{UML})$ in Definition 5. Notice that, the fuzzy OWL axioms of $FAxiom_0$ in Definition 5 may be mainly partitioned into the following several cases:

Case 1: If $\preccurlyeq_{\mathcal{F}}(FC_2) = FC_1$. Firstly, according to Definition 3, if $\mathcal{FB}$ is a legal fuzzy object state, then $FC_1^{\mathcal{FB}} \subseteq FC_2^{\mathcal{FB}}$. Then, by definition of $\alpha_{\mathcal{F}}(\mathcal{FB})$ above, we have $(\varphi(FC_1))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \subseteq (\varphi(FC_2))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$. That is, $\alpha_{\mathcal{F}}(\mathcal{FB})$ satisfies the axiom Class ($\varphi(FC_1)$ partial $\varphi(FC_2)$) in Definition 5.

Case 2: If there is a fuzzy class hierarchy $\preccurlyeq_{\mathcal{F}}(FC) = FC_1 \times FC_2 \times \ldots \times FC_n$. Firstly, similarly for the case 1 above, we have $(\varphi(FC_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \subseteq (\varphi(FC))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$, where $i \in \{1, \ldots, n\}$. Then, if there are disjointness and completeness constraints, according to Definition 3, if $\mathcal{FB}$ is a legal fuzzy object state, then $FC^{\mathcal{FB}} = FC_1^{\mathcal{FB}} \cup \ldots \cup FC_n^{\mathcal{FB}}$ and $FC_i^{\mathcal{FB}} \cap FC_j^{\mathcal{FB}} = \varnothing$, and by definition of $\alpha_{\mathcal{F}}(\mathcal{FB})$ again, we have $(\varphi(FC))^{\alpha_{\mathcal{F}}(\mathcal{FB})} = (\varphi(FC_1))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \cup \ldots \cup (\varphi(FC_n))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$ and $(\varphi(FC_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \cap (\varphi(FC_j))^{\alpha_{\mathcal{F}}(\mathcal{FB})} = \varnothing$. That is, $\alpha_{\mathcal{F}}(\mathcal{FB})$ satisfies the corresponding axioms of the fuzzy class hierarchy $\preccurlyeq_{\mathcal{F}}(FC)$ in Definition 5.

Case 3: If there is a fuzzy class $FC \in FC_{\mathcal{F}}$ with $att_{\mathcal{F}}(FC) = [FA_1 : FT_1, \ldots, FA_n : FT_n]$. Firstly, for an instance $c \in (\varphi(FC))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$, by definition of $\alpha_{\mathcal{F}}(\mathcal{FB})$ above, we have $c \in FC^{\mathcal{FB}}$. Then, according to Definition 3, there is at least one element $a_i \in FA_i^{\mathcal{FB}}$ whose first component is c, and the second component of a_i is an element $t_i \in FT_i^{\mathcal{FB}}$, and we have the formula $mult_{\min}(t_i, c, a_i) \leq \#\{c \in FC^{\mathcal{FB}} \mid c[a_i] = t_i\} \leq mult_{\max}(t_i, c, a_i)$, where #{ }denotes the base of set { }. Furthermore, by definition of $\alpha_{\mathcal{F}}(\mathcal{FB})$ above, we have $a_i = <c, t_i> \in FA_i^{\mathcal{FB}} = (\varphi(FA_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$ and $(\varphi(FC))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \subseteq \{c \mid mult_{\min}(t_i, c, a_i) \leq \#\{c \in (\varphi(FC))^{\alpha_{\mathcal{F}}(\mathcal{FB})} | <c, t_i> \in (\varphi(FA_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})}\} \leq mult_{\max}(t_i, c, a_i)\}$, where $c \in (\varphi(FC))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$ and $t_i \in FT_i^{\mathcal{FB}} = (\varphi(FT_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$. Moreover, according to Definition 2, it is shown that each fuzzy attribute $FA \in FA_{\mathcal{F}}$ is mapped into a set, i.e., $FA^{\mathcal{FB}} \subseteq \Delta^{\mathcal{FB}} \times \bigcup_{FT \in FT_{\mathcal{F}}} FT^{\mathcal{FB}}$, and based on the above statements, there is at least one $<c, t_i> \in a_i$, i.e., $FA_i^{\mathcal{FB}} \subseteq FC^{\mathcal{FB}} \times \bigcup_{FT_i \in FT_{\mathcal{F}}} FT_i^{\mathcal{FB}}$, i.e., $(\varphi(FA_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \subseteq (\varphi(FC))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \times (\varphi(FT_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$. That is, $\alpha_{\mathcal{F}}(\mathcal{FB})$ satisfies the corresponding axioms of the fuzzy class $att_{\mathcal{F}}(FC)$ in Definition 5.

Case 4: If $x, y \in FC_{\mathcal{F}} \cup FS_{\mathcal{F}}$ with $x \neq y$ and $x \in FS_{\mathcal{F}}$. Firstly, according to Definition 1, the various basic domains are pairwise disjoint and disjoint from $\Delta^{\mathcal{FB}}$ and from the set of labeled tuples. Then, we know that labeled tuples corresponding to different associations cannot be equal since they are defined over different sets of roles. That is, $\alpha_{\mathcal{F}}(\mathcal{FB})$ satisfies the corresponding axioms of this case in Definition 5.

Case 5: IF there is a fuzzy association $FS \in FS_{\mathcal{F}}$ such that $ass_{\mathcal{F}}(FS) = [FR_1 : FC_1, \ldots, FR_k : FC_k]$, and each role FR_i is associated with a cardinality constraint $card_{\mathcal{F}}(FC_i, FS, FR_i) = (card_{\min}(FC_i, FS, FR_i),$

$card_{\max}(FC_i, FS, FR_i)$). Firstly, for a fuzzy association instance $s \in (\varphi(FS))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$, by definition of $\alpha_{\mathcal{F}}(\mathcal{FB})$ above, we have $s \in FS^{\mathcal{FB}}$. Then, according to Definition 3, we know that all instances of FS are of the form $[r_1 : c_1, \ldots, r_k : c_k]$, and thus s is also the following FR-labeled tuples over $\Delta^{\mathcal{FB}}$ of the form $[r_1 : c_1, \ldots, r_k : c_k]$, where $r_i \in FR_i^{\mathcal{FB}}$, $c_i \in FC_i^{\mathcal{FB}}$, $i \in \{1, \ldots, k\}$. Furthermore, by definition of $\alpha_{\mathcal{F}}(\mathcal{FB})$ above once again, we have $c_i \in (\varphi(FC_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$. Moreover, by definition of $\alpha_{\mathcal{F}}(\mathcal{FB})$ above, it follows $(\varphi(FR_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})} = \{<s, c_i> \in \Delta^{\alpha_{\mathcal{F}}(\mathcal{FB})} \times \Delta^{\alpha_{\mathcal{F}}(\mathcal{FB})} \mid s \in FS^{\mathcal{FB}} \wedge c_i \in FC_i^{\mathcal{FB}} \wedge s[FR_i] = c_i\}$, i.e., $(\varphi(FR_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \subseteq FS^{\mathcal{FB}} \times FC_i^{\mathcal{FB}}$, so $(\varphi(FR_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \subseteq (\varphi(FS))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \times (\varphi(FC_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$. Since v_i = invof_φ(FR_i) denotes an inverse property of the fuzzy object property φ(FR_i), i.e., $v_i^{\alpha_{\mathcal{F}}(\mathcal{FB})} = ((\varphi(FR_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})})^-$, we have $v_i^{\alpha_{\mathcal{F}}(\mathcal{FB})} = ((\varphi(FR_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})})^- \subseteq (\varphi(FC_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \times (\varphi(FS))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$. In addition, according to Definition 3, we have the following formula: $card_{\min}(FC, FS, FR) \le \#\{s \in FS^{\mathcal{FB}} \mid s[FR] = c\} \le card_{\max}(FC, FS, FR)$, where #{ }denotes the base of set { }, and by definition of $\alpha_{\mathcal{F}}(\mathcal{FB})$ above, it follows $(\varphi(FC_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \subseteq \{c_i \mid mult_{\min}(t_i, c, a_i) \le \#\{s \in (\varphi(FS))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \mid <s, c_i> \in (\varphi(FR_i))^{\alpha_{\mathcal{F}}(\mathcal{FB})}\} \le mult_{\max}(t_i, c, a_i)\}$. That is, $\alpha_{\mathcal{F}}(\mathcal{FB})$ satisfies the corresponding axioms of the fuzzy association $ass_{\mathcal{F}}(FS)$ in Definition 5.

The other cases in the first part of Theorem 1 such as methods and aggregations can be proved analogously, where the methods are included in the case 3 above, and the aggregations are a particular kind of binary associations and can be proved similarly with the case 4 above. Moreover, the two parts of Theorem 1 are a mutually inverse process, and the proof of the second part of Theorem 1, which can be treated analogously according to the first part above, is omitted here. □

In the following, we further illustrate the transformation in Definition 5 with Example 1.

Example 1. Given the fuzzy UML model $\mathcal{F}_{\text{UML1}}$ in Fig. 6, by Definition 5, we can obtain the corresponding fuzzy OWL ontology structure $\mathcal{FO}_{\text{S}} = \varphi(\mathcal{F}_{\text{UML1}})$ in Fig. 9.

The fuzzy OWL ontology structure φ($\mathcal{F}_{\text{UML1}}$) = ($FID_0$, $FAxiom_0$) derived from the fuzzy UML model $\mathcal{F}_{\text{UML1}}$ in Fig. 6 by Definition 5 is as follows (for brevity, FID_0, part of axioms, and the symbol φ( ) are omitted):

*FAxiom*₀ = { Class (Young-Employee partial Employee) ;
Class (Old-Employee partial Employee) ;
EquivalentClasses(φ(Employee), unionOf (φ(Young-Employee), φ(Old-Employee))) ;
DisjointClasses (Young-Employee, Old-Employee) ;
Class (New-Computer partial Computer) ;
Class (Old-Employee partial restriction (ID allValuesFrom (xsd:String) Cardinality (1)) restriction (Name allValuesFrom (xsd:String) Cardinality (1)) restriction (FUZZY-Age allValuesFrom (xsd:Integer) Cardinality (1)) restriction (FUZZY-Email allValuesFrom (xsd:String) minCardinality (1)) restriction (IsDepartment allValuesFrom (xsd:String) maxCardinality (1)) restriction (*u* allValuesFrom (xsd:Real) Cardinality (1))) ;
Class (Use partial restriction (Date allValuesFrom (xsd:String) Cardinality (1)) restriction (*β* allValuesFrom (xsd:Real) Cardinality (1)) restriction (Uby allValuesFrom (Old-Employee) Cardinality (1)) restriction (Uof allValuesFrom (New-Computer) Cardinality (1))) ;
Class (owl:Thing partial restriction (inverseOf φ(is_part_of_g_1) allValuesFrom (New-Computer) Cardinality (1)) restriction (φ(is_whole_of_g_1) allValuesFrom (Monitor) minCardinality (1)) … restriction (inverseOf φ(is_part_of_g_m) allValuesFrom (New-Computer) Cardinality (1)) restriction (φ(is_whole_of_g_m) allValuesFrom (Keyboard) minCardinality (1))) ;
Class (Old-Employee partial restriction (invof_Uby allValuesFrom (Use))) ;
Class (New-Computer partial restriction (invof_Uof allValuesFrom (Use))) ;
Class (Old-Employee partial restriction (invof_Uby Cardinality (1))) ;
Class (New-Computer partial restriction (invof_Uof minCardinality (1) maxCardinality (2))) ;
ObjectProperty (is_part_of_g_1 domain (Monitor) range (New-Computer)) ;
ObjectProperty (is_whole_of_g_1 domain (New-Computer) range (Monitor)) ;
DatatypeProperty (Name domain (Old-Employee) range (xsd:String) [Functional]) ; … }

Fig. 9. The fuzzy OWL ontology structure φ($\mathcal{F}_{\text{UML1}}$) derived from the fuzzy UML model $\mathcal{F}_{\text{UML1}}$ in Fig. 6.

4.2. *Transforming Fuzzy UML Instantiations into Fuzzy Ontology Instances*

Based on the structure transformation in Section 4.1, in this section we further propose an approach for transforming the fuzzy UML instantiations into the fuzzy OWL ontology instances at instance level. Firstly, let us briefly sketch the representation forms of instances in fuzzy OWL ontologies and fuzzy UML models.

In a fuzzy OWL ontology, the instances are represented by the fuzzy individual axioms (see Table 1):

- Individual (o type(C_1) [$\bowtie m_1$] … value(R_1, o_1) [$\bowtie k_1$] … value(U_1, v_1) [$\bowtie l_1$] …);
- SameIndividual (o_1 … o_n);
- DifferentIndividuals (o_1 … o_n), where o_i denotes an abstract individual, v_i a concrete individual, C fuzzy class description, R and U a fuzzy object and datatype property respectively, $m_i \in [0, 1]$, $k_i \in [0, 1]$, $l_i \in [0, 1]$, and $\bowtie \in \{\geq, >, \leq, <\}$.

In a fuzzy UML model, as mentioned in Sections 2.2 and 3.1, it is shown that the fuzziness may occur at several different levels:

- The first level is the *attribute level*. In a class of a fuzzy UML model, each attribute is associated with a type (called the domain of an attribute), and an attribute may be one of the following two cases:
  - A *non-fuzzy* attribute, its domain may be a simple/basic type (such as integer, real and string) or a complex type (such as set type and object type).
  - A *fuzzy* attribute, it domain is a fuzzy-type-based type. Only fuzzy attributes have fuzzy types and fuzzy attributes are explicitly indicated in a class definition (e.g., a fuzzy keyword *FUZZY* is appeared on the front of attributes indicating they are fuzzy attributes). A fuzzy domain of a fuzzy attribute may be a set of possibility distributions or a set of fuzzy linguistic terms (each fuzzy linguistic term is associated with a membership function over a basic type). For example, the domain of an attribute *Age* may be a crisp set of integers, a set of possibility distributions over the basic type integers, or a set of fuzzy linguistic terms over the basic type integers such as {*young*, *old*}.
- The second level is the *object/class level*, i.e., an object belongs to a class with a membership degree of [0, 1]. An additional attribute $u \in [0, 1]$ is introduced into a class to represent the membership degree of an object to the class.
- The third level is the *class/class level*, such as the fuzziness in a subclass/superclass relationship or the fuzziness in two classes with an association relationship as mentioned in Sections 2.2 and 3.1, and these kinds of fuzziness may be represented by the fuzziness in the first two levels above. For example, in a subclass/superclass relationship, the membership degree that any object belongs to the subclass is less than or equal to the membership degree that it belongs to the superclass. Therefore, fuzzy subclass/superclass relationships in a fuzzy UML model can be assessed by utilizing the inclusion degree of objects to the class in the second level above.

Based on the discussion about fuzzy UML models above and Sections 2.2 and 3.1, a fuzzy UML instantiation with respect to a fuzzy UML model, which describes the real world by means of objects, values, and their mutual relationships, can be considered as a finite set of assertions. The assertion formalisms of a fuzzy UML instantiation (i.e., a set of object instances) with respect to a fuzzy UML model include:

- The assertion of the form FO: FC: u, which denotes that a fuzzy object FO is an instance of a fuzzy class FC with membership degree of $u \in [0, 1]$.
- The assertion of the form FO : [FA_1:FV_1:n_1,…,FA_k:FV_k:n_k], which denotes the attribute value associated with FO, where $FV_i \in \Delta^{\mathcal{FB}}$, $FA_i \in FA_{\mathcal{F}}$ denotes the attribute of FO, $n_i \in [0, 1]$ denotes the membership degree, and $i \in \{1…k\}$ (see Definition 2). Here, since the value of an attribute FA_i may be a possibility distribution, for simplicity, FV_i : n_i only denotes one element of the possibility distribution.
- Since the fuzziness in the third level above in a fuzzy UML model can be assessed by the other two levels as mentioned above, the assertion formalisms of the fuzziness in the third level may be repressed by the above assertion formalisms.

On this basis, Definition 6 gives a transformation approach from fuzzy UML instantiations with respect to fuzzy UML models to fuzzy OWL ontology instances.

Definition 6 (Instance transformation). Given a *fuzzy UML instantiation* (i.e., a set of object instances) w.r.t. a fuzzy UML model, the corresponding *fuzzy OWL ontology instances* (w.r.t. the fuzzy ontology structure in Definition 5) can be derived as the rules in Table 3:

Copyright: the authors

Table 3. Transforming rules from the fuzzy UML instantiations to the fuzzy OWL ontology instances.

Fuzzy UML instantiations	Fuzzy OWL ontology instances $\mathcal{FO}_\text{I}$	Comments
Each fuzzy object symbol $FO \in FO_{¸}$	A fuzzy individual identifier $\varphi(FO) \in FIID_0$	
Each fuzzy class symbol $FC \in FC_{¸}$, each fuzzy attribute symbol $FA \in FA_{¸}$, each fuzzy role symbol $FR \in FR_{¸}$ and so on in Definition 5	A fuzzy class identifier $\varphi(FC) \in FCID_0$, a fuzzy datatype property identifier $\varphi(FA) \in FDPID_0$, a pair of inverse fuzzy object property identifiers $\varphi(FR) \in FOPID_0$ and v = invof_$\varphi(FR) \in FOPID_0$, and the other fuzzy class/datatype property/object property identifiers; For brevity, in the following we use $\varphi(FC)$ to denote all fuzzy class identifiers in Definition 5, U_i denotes all fuzzy datatype property identifiers, and R_i denotes all fuzzy object property identifiers.	See Definition 5 in detail
Each fuzzy assertion FO: FC: n	A fuzzy individual axioms: Individual ($\varphi(FO)$ type($\varphi(FC)$) [$\bowtie'$ n]), $\bowtie' \in \{\geq, \leq\}$	See Fig. 10 in detail
Each fuzzy assertion: FO:[FA_1:FV_1:n_1,…,FA_k:FV_k:n_k]	A fuzzy individual axioms: Individual ($\varphi(FO)$ value(R_i, $\varphi(FV_\text{i})$) [$\bowtie'$ n_i]… value(U_i, $\varphi(FV_\text{i})$) [$\bowtie'$ n_i]…) where $\varphi(FV_\text{i})$, which denotes the corresponding value of FV_i, is an element of the fuzzy interpretation domain of the fuzzy ontology, $i \in \{1…k\}$, $\bowtie' \in \{\geq, \leq\}$	See Fig. 10 in detail

In the following, we illustrate the transformation procedure in Definition 6 with Example 2.

Example 2. Given the fuzzy UML instantiation in Fig. 8 (w.r.t. the fuzzy UML model in Fig. 6 and Fig. 7), the transformed fuzzy ontology instance (w.r.t. the fuzzy ontology structure in Fig. 9) is shown in Fig. 10. The fuzzy ontology structure in Fig. 9 and the instance in Fig. 10 form a target fuzzy ontology.

The fuzzy OWL ontology instance FO_I = (FID_0, $FAxiom_0$) derived from the fuzzy UML instantiation in Fig. 8 is as follows: $FIID_0$ = {o_1, o_2, o_3, o_4, o_1', o_2', o''} ; //o'' is an additional individual

$FAxiom_0$ = { DifferentIndividuals (o_1, o_2, o_3, o_4, o_1', o_2', o'') ;

Individual (o_1 type(Young-Employee) [$\bowtie'$ 0.9]) ;

Individual (o_2 type(Young-Employee) [$\bowtie'$ 0.95]) ;

Individual (o_1' type(Old-Employee) [$\bowtie'$ 0.6]) ;

Individual (o_2' type(Old-Employee) [$\bowtie'$ 0.8]) ;

Individual (o_1 type(Employee) [$\bowtie'$ 0.93]) ;

Individual (o_2 type(Employee) [$\bowtie'$ 0.95]) ;

Individual (o_1' type(Employee) [$\bowtie'$ 0.7]) ;

Individual (o_2' type(Employee) [$\bowtie'$ 0.8]) ;

Individual (o_3 type(New-Computer) [$\bowtie'$ 0.65]) ;

Individual (o_4 type(New-Computer) [$\bowtie'$ 0.95]) ;

Individual (o_3 type(Computer)) ; Individual (o_4 type(Computer) [$\bowtie'$ 0.95]) ;

Individual (o'' type(Use)) ;

Individual (o'' value(Uby, o_1') value(Uof, o_3) [$\bowtie'$ 0.7] value(Uof, o_4) [$\bowtie'$ 0.9]) ;

Individual (o_3 value(ComID, *NC12*) value(Brand, *Sony*) value(Year, *6*) value(u, *0.65*) value (invof_Uof, o'') [$\bowtie'$ 0.7]) ;

Individual (o_4 value(ComID, *NC32*) value(Brand, *IBM*) value(Year, *2*) value(u, *0.95*) value (invof_Uof, o'') [$\bowtie'$ 0.9]) ;

Individual (o_1' value(ID, *OE24*) value(Name, Tom) value(FUZZY-Age, 51) [$\bowtie'$ 0.7] … value(FUZZY-Email, John@yahoo.com) [$\bowtie'$ 0.3] value(IsDepartment, SalesDept) [$\bowtie'$ 0.6] value(u, *0.6*) value(invof_Uby, o'')) ; }

Fig. 10. The fuzzy OWL ontology instance derived from the fuzzy UML instantiation in Fig. 8.

Based on the proposed approaches in the previous sections, a fuzzy UML model including its structure and instance information can be transformed into a fuzzy OWL ontology. In the following section we further develop a prototype tool to construct fuzzy OWL ontologies from fuzzy UML models.

4.3. *Prototype Construction Tool*

In this section, following the proposed approaches in Sections 4.1 and 4.2, we developed a prototype tool called *FUML2FOnto*, which can read in an XML-coded fuzzy UML model created with a CASE tool PowerDesigner and automatically construct fuzzy OWL ontologies from fuzzy UML models. In the following we briefly introduce the design and implementation of the prototype tool *FUML2FOnto*.

The implementation of *FUML2FOnto* is based on Java 2 JDK 1.6 platform, and the Graphical User Interface is exploited by using the java.awt and javax.swing packages. In our implementation, the XMI file of the fuzzy UML model is parsed by the DOM API for Java to construct an in-memory document representation (tree structure). Java methods are used to extract fuzzy UML model data from the tree structure; the extracted data are stored as in-memory data in Java classes and simultaneously displayed on the tool screen. Then the in-memory data are used by Java methods to perform the transformation from the fuzzy UML model to the fuzzy OWL ontology. The resulting ontology is saved as text files and displayed simultaneously on the tool screen. The overall architecture of *FUML2FOnto* is shown in **Fig. 11**.

From Fig. 11, *FUML2FOnto* includes four main modules, i.e., *input module*, *parsing module*, *transformation module*, and *output module*:

- The input module inputs an XML-coded fuzzy UML model created with a CASE tool PowerDesigner. It should be noted that some features of a fuzzy UML model (such as the disjointness and completeness constraints in a fuzzy hierarchy and the fuzzy dependency as introduced in Sections 2.2 and 3.1) cannot be supported by the CASE tool, and they need to be added by the designers by hand;
- The parsing module uses the regular expression to parse the fuzzy UML model file, and stores the parsed results as Java ArrayList classes, where fuzzy UML classes, attributes, and so on are identified;
- The transformation module transforms the parsed results of the fuzzy UML model into the fuzzy OWL ontology structure and instance according to the proposed approaches in Sections 4.1 and 4.2. where the instance transformation is based on the structure transformation;
- The output module produces the resulting fuzzy OWL ontology which is saved as a text file and displayed on the tool screen as will be shown in Fig. 12.

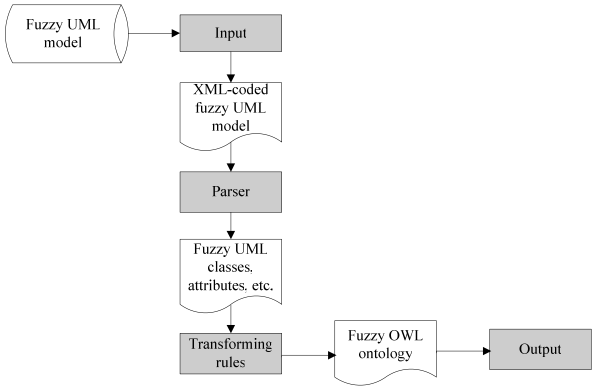
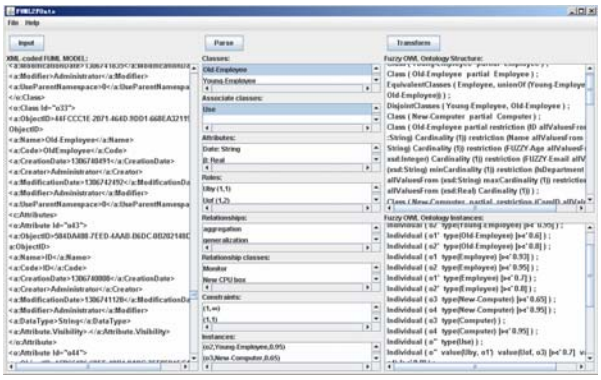


Fig. 11. The overall architecture of *FUML2FOnto*.

In the following we provide an example of *FUML2FOnto* to show that the proposed approach is feasible and the implemented tool is efficient. As shown in Section 3.1, a fuzzy UML model $\mathcal{F}_{\text{UML1}}$ modeling the situation at a company (including the structure information in Fig. 6 and the instance information in Fig. 8) is provided, and the fuzzy UML model $\mathcal{F}_{\text{UML1}}$ is transformed into a fuzzy OWL ontology by *FUML2FOnto*. **Fig. 12** shows the screen snapshot of *FUML2FOnto*, which displays the transformations from the fuzzy UML model (Fig. 6) and the corresponding fuzzy UML instantiation (Fig. 8) to the fuzzy OWL ontology structure (Fig. 9) and the fuzzy OWL ontology instance (Fig. 10). In Fig. 12, the XML-coded fuzzy UML model file, the parsed results, and the constructed fuzzy OWL ontology are displayed in the left, middle and right areas, respectively.

Copyright: the authors



```
<a:Modifier>Administrator</a:Modifier>
<a:UseParentNamespace>0</a:UseParentNamespa
</o:Class>
<o:Class Id="o33">
<a:ObjectID>44FCCC1E-2071-46ED-9DD1-66BEA3211
ObjectID>
<a:Name>Old-Employee</a:Name>
<a:Code>OldEmployee</a:Code>
<a:CreationDate>1306740491</a:CreationDate>
<a:Creator>Administrator</a:Creator>
<a:ModificationDate>1306742492</a:ModificationDa
<a:Modifier>Administrator</a:Modifier>
<a:UseParentNamespace>0</a:UseParentNamespa
<c:Attributes>
<o:Attribute Id="o43">
<a:ObjectID>584DA488-7EED-4AAB-B6DC-0D202148C
a:ObjectID>
<a:Name>ID</a:Name>
<a:Code>ID</a:Code>
<a:CreationDate>1306740008</a:CreationDate>
<a:Creator>Administrator</a:Creator>
<a:ModificationDate>1306741128</a:ModificationDa
<a:Modifier>Administrator</a:Modifier>
<a:DataType>String</a:DataType>
<a:Attribute.Visibility>-</a:Attribute.Visibility>
</o:Attribute>
<o:Attribute Id="o44">
```

```
Classes: Old-Employee, Young-Employee
Associate classes: Use
Attributes: Date: String, ID: Real
Roles: Uby (1,1), Uof (1,2)
Relationships: aggregation, generalization
Relationship classes: Monitor, New CPU box
Constraints: (1,∞), (1,1)
Instances: (o2,Young-Employee,0.95), (o3,New-Computer,0.65)
```

```
Class ( Old-Employee  partial  Employee ) ;
EquivalentClasses ( Employee, unionOf (Young-Employe
Old-Employee)) ) ;
DisjointClasses ( Young-Employee, Old-Employee ) ;
Class ( New-Computer  partial  Computer ) ;
Class ( Old-Employee partial restriction (ID allValuesFro
:String) Cardinality (1)) restriction (Name allValuesFrom
String) Cardinality (1)) restriction (FUZZY-Age allValuesF
xsd:Integer) Cardinality (1)) restriction (FUZZY-Email allV
(xsd:String) minCardinality (1)) restriction (IsDepartment
allValuesFrom (xsd:String) maxCardinality (1)) restrictio
allValuesFrom (xsd:Real) Cardinality (1)) ) ;
```

```
Individual ( o1'  type(Old-Employee) [⋈' 0.6] ) ;
Individual ( o2'  type(Old-Employee) [⋈' 0.8] ) ;
Individual ( o1  type(Employee) [⋈' 0.93] ) ;
Individual ( o2  type(Employee) [⋈' 0.95] ) ;
Individual ( o1'  type(Employee) [⋈' 0.7] ) ;
Individual ( o2'  type(Employee) [⋈' 0.8] ) ;
Individual ( o3  type(New-Computer) [⋈' 0.65] ) ;
Individual ( o4  type(New-Computer) [⋈' 0.95] ) ;
Individual ( o3  type(Computer) ) ;
Individual ( o4  type(Computer) [⋈' 0.95] ) ;
Individual ( o"  type(Use) ) ;
Individual ( o"  value(Uby, o1)  value(Uof, o3) [⋈' 0.7]  v
```

Fig. 12. Screen snapshot of *FUML2FOnto*.

Based on the tool *FUML2FOnto*, a fuzzy OWL ontology can be constructed from a fuzzy UML model. Now let us analyze the time complexity of the construction process by measuring the amount of work done by the tool. Here we consider the *transformation* operations and ignore the *preprocessing* operations (i.e., fuzzy UML model parsing and element extraction as mentioned in the overall architecture of *FUML2FOnto* above). That is to say, we exclude the amount of work done by an XML parser (e.g., the DOM API for Java in our implementation) that parses the fuzzy UML model (i.e., an XMI-coded file) and prepares the element data in computer memory for the usage in the transformation procedure. Moreover, since the transformation from fuzzy UML symbols to fuzzy OWL ontology identifiers can be simultaneously made as sub-operations in creating the fuzzy OWL ontology axioms as shown in Sections 4.1 and 4.2, we can ignore the amount of work done in the symbol-to-identifier transformation and consider only the creation of axioms to be the main operation of the tool *FUML2FOnto*. On this basis, we now could measure the amount of work done by the tool *FUML2FOnto* by roughly counting the total number of operations for axiom creation in the construction procedure proposed in our approach (see Definitions 5 and 6 in Sections 4.1 and 4.2).

Given a fuzzy UML model $\mathcal{F}_{\text{UML}} = (\mathcal{L}_{\mathcal{F}}, \preccurlyeq_{\mathcal{F}}, att_{\mathcal{F}}, agg_{\mathcal{F}}, dep_{\mathcal{F}}, ass_{\mathcal{F}}, card_{\mathcal{F}}, mult_{\mathcal{F}}, mult'_{\mathcal{F}})$, the amount of work done by the tool *FUML2FOnto* mainly depends on the structure of the fuzzy UML model $\mathcal{F}_{\text{UML}}$. We use the total number of main elements in $\mathcal{F}_{\text{UML}}$ to measure the scale of the fuzzy UML model. Suppose the scale of the fuzzy UML model $\mathcal{F}_{\text{UML}}$ is $\mathcal{N} = \mathcal{N}_{\text{FC}} + \mathcal{N}_{\text{FA}} + \mathcal{N}_{\text{FS}} + \mathcal{N}_{\text{FR}} + \mathcal{N}_{\text{FH}} + \mathcal{N}_{\text{FG}} + \mathcal{N}_{\text{I}}$, where $\mathcal{N}_{\text{FC}}$, $\mathcal{N}_{\text{FA}}$, $\mathcal{N}_{\text{FS}}$, $\mathcal{N}_{\text{FR}}$, $\mathcal{N}_{\text{FH}}$, $\mathcal{N}_{\text{FG}}$ and $\mathcal{N}_{\text{I}}$ denotes the cardinality of the sets of fuzzy classes, fuzzy attributes, fuzzy associations, fuzzy roles, fuzzy hierarchies, fuzzy aggregations and object instances, respectively (note that the fuzzy dependency in a fuzzy UML model is omitted here since it cannot be represented by fuzzy OWL ontology and needs to be indicated explicitly by the designers as mentioned in Section 4.1). Then, from the construction procedure proposed in our approach (see Definitions 5 and 6 in Sections 4.1 and 4.2), it is shown that the executing times of creating the corresponding axioms of the case $att_{\mathcal{F}}(FC)$ is $\mathcal{N}_{\text{FC}} + 2\mathcal{N}_{\text{FA}}$ at most (where $\mathcal{N}_{\text{FC}}$ includes fuzzy classes and fuzzy association classes, and $\mathcal{N}_{\text{FA}}$ includes attributes and methods), the case of fuzzy association $ass_{\mathcal{F}}(FS)$ is $\mathcal{N}_{\text{FS}} + 5\mathcal{N}_{\text{FR}}$, the case of fuzzy class hierarchy $\preccurlyeq_{\mathcal{F}}(FC)$ is $2\mathcal{N}_{\text{FH}}$, the case of fuzzy aggregation $agg_{\mathcal{F}}(FG)$ is $2\mathcal{N}_{\text{FG}}$, the case of DislointClasses(φ(x), φ(y)) in Table 2 of Section 4.1 is $\mathcal{N}_{\text{FS}} \times (\mathcal{N}_{\text{FS}}\text{-}1)/2 + \mathcal{N}_{\text{FS}} \times \mathcal{N}_{\text{FC}} = \mathcal{N}_{\text{FS}} \times (\mathcal{N}_{\text{FC}}+\mathcal{N}_{\text{FS}}/2-1/2)$, and the executing times of creating the corresponding axioms of object instances is $\mathcal{N}_{\text{I}} + 1$. Therefore, the total running times $T = \mathcal{N}_{\text{FC}} + 2\mathcal{N}_{\text{FA}} + \mathcal{N}_{\text{FS}} + 5\mathcal{N}_{\text{FR}} + 2\mathcal{N}_{\text{FH}} + 2\mathcal{N}_{\text{FG}} + \mathcal{N}_{\text{FS}}\times(\mathcal{N}_{\text{FC}}+\mathcal{N}_{\text{FS}}/2-1/2) + \mathcal{N}_{\text{I}} + 1 < \mathcal{N}^2 + 5\mathcal{N}$, that is, the worst case time complexity of the transformation from a fuzzy UML model to a fuzzy OWL ontology is O($\mathcal{N}^2$).

So far, on the basis of the proposed approach and the implemented tool, fuzzy OWL ontologies can be constructed from fuzzy UML models. The correctness of our proposed construction approach has been formally proved in the previous section 4.1. In this section, we developed the tool *FUML2FOnto* to demonstrate that the approach actually works. In summary, from the proposed approach and tool above, it is shown that two main steps need to be carried out when constructing a fuzzy OWL ontology from a fuzzy UML model, i.e., transforming the fuzzy UML model into the fuzzy OWL ontology at structure level and transforming the object instance w.r.t. the fuzzy UML model into the fuzzy OWL ontology at instance level. Therefore, the correctness of the resulting fuzzy OWL ontology is decided by the respective correctness of two steps above. For the first step, the transformation rules at structure level (see Definition 5) can transform all the elements of the fuzzy UML model (including fuzzy classes $FC_{\mathcal{F}}$, fuzzy attributes $FA_{\mathcal{F}}$, and so on) into fuzzy OWL ontology identifiers, and induce a set of fuzzy class and property axioms from the constraints of the fuzzy UML model (including fuzzy hierarchies, fuzzy aggregations, fuzzy associations, and so on). The correctness of the transformation above has been sanctioned by the theorem 1. For the second step, the transformation at instance level from the object instance w.r.t. the fuzzy UML model into the fuzzy OWL ontology instance is established based on the transformation at structure level in the first step. Based on the constructed fuzzy OWL ontology identifiers and axioms in the first step, the transformation rules at instance level (see Definition 6) can further map the objects to their corresponding fuzzy classes and map the attributes of the objects to their corresponding values. The correctness of the transformation at instance level can be ensured by the correctness of the transformation at structure level in the first step. Moreover, we also provide an example (including examples 1 and 2 in Sections 4.1 and 4.2) to show that the approach is validated and verified, and the example is tested by the tool *FUML2FOnto*, where the example 1 can construct the fuzzy OWL ontology structure $\mathcal{FO}_\mathrm{S}$ (i.e., identifiers and fuzzy class/property axioms in Fig. 9) from the fuzzy UML model $\mathcal{F}_{\mathrm{UML1}}$ in Fig. 6 and Fig. 7, and then the example 2 can further construct the fuzzy OWL ontology instance $\mathcal{FO}_\mathrm{I}$ (i.e., fuzzy individual axioms in Fig. 10) from the object instance in Fig. 8 w.r.t. the fuzzy UML model $\mathcal{F}_{\mathrm{UML1}}$. Finally, two parts $\mathcal{FO}_\mathrm{S}$ and $\mathcal{FO}_\mathrm{I}$ form a target fuzzy OWL ontology.

After constructing a fuzzy OWL ontology from a fuzzy UML model on the basis of the previous work, this means that the information in a fuzzy UML model may be represented using a knowledge representation technique. As a result, reasoning of fuzzy UML models may be handled using the common fuzzy ontology technique in an intelligent and automatic way. For example, using conventional techniques in reasoning on fuzzy UML models would mean to manually check the reasoning tasks of fuzzy UML models (e.g., whether a fuzzy class in a fuzzy UML model is satisfiable or whether there is redundancy in a fuzzy UML model). Using knowledge representation techniques such as fuzzy ontology, the tasks of checking the reasoning problems of fuzzy UML models may be done automatically, which may help to improve the ability of reasoning on fuzzy UML models.

Therefore, to demonstrate the constructed fuzzy OWL ontologies may be useful for reasoning on fuzzy UML models, in the following section we will focus on investigating how to reason on fuzzy UML models based on the constructed fuzzy OWL ontologies.

5. Reasoning on Fuzzy UML Models with the Constructed Fuzzy OWL Ontologies

As mentioned in Section 1 and the end of Section 4.3, the approach of constructing fuzzy OWL ontologies from fuzzy UML models may facilitate the fuzzy ontology development. Also, the constructed fuzzy OWL ontologies may be useful for reasoning on fuzzy UML models. Therefore, after representing a fuzzy UML model in a fuzzy OWL ontology in the previous sections of this paper, maybe it is convenient and worth it investigating how to reason on fuzzy UML models based on the constructed fuzzy OWL ontologies.

In this section, to demonstrate the constructed fuzzy OWL ontologies may be useful for reasoning on fuzzy UML models, based on the initial idea in Ref. 30, we investigate how to reason on fuzzy UML models based on the constructed fuzzy OWL ontologies in depth, including:

- we first introduce the motivation of reasoning on fuzzy UML models with the constructed fuzzy OWL ontologies in detail;
- we give formal definitions of the familiar reasoning tasks of fuzzy UML models, where several reasoning problems are added and some reasoning problems are redefined and studied in depth;
- we reduce reasoning on fuzzy UML models to reasoning on the constructed fuzzy OWL ontologies, and give the proofs of correctness of the reduction methods. By reducing, the reasoning tasks of fuzzy UML models may be done by checking the reasoning problems of fuzzy OWL ontologies. Further, the reasoning problems of fuzzy OWL ontologies can be automatically checked by means of the existing fuzzy ontology reasoner DeLorean[48].

5.1. *The Motivation of Reasoning on Fuzzy UML Models with the Constructed Fuzzy OWL Ontologies*

In the following we give a brief example of reasoning on fuzzy UML models, which can more directly illustrate that: (*i*) some reasoning tasks may usually occur in the fuzzy UML modeling activities; (*ii*) it is possible and meaningful to reason on fuzzy UML models with the constructed fuzzy OWL ontologies.

Copyright: the authors

Example 3. Fig. 13 shows a simple fuzzy UML model $\mathcal{F}_{\text{UML2}}$ and its corresponding object instances (part of classes and object instances). Here, as mentioned in Sections 2.2 and 3.1: (*i*) △ denotes the generalization relationship between fuzzy classes; (*ii*) *disjointness and completeness* constraints denote that two fuzzy classes *Young-Employee* and *Old-Employee* are *disjoint* and *the union of them* completely covers the superclass *Employee*. There are unsatisfiability and redundancy in the fuzzy UML model $\mathcal{F}_{\text{UML2}}$ since the following reasons:

- Since *Old-Employee* and *Young-Employee* are disjoint, i.e., there is no instance which belongs to the two classes, we have $\{o_1/0,\ o_2/0\} \in$ *Old-Employee*.
- Since *Young-Employee* is a subclass of *Old-Employee*, i.e., for any object, the membership degree that it belongs to the subclass *Young-Employee* is *less than or equal to* the membership degree that it belongs to the superclass *Old-Employee*, we have $\{o_1/{\geq}0.9,\ o_2/{\geq}0.95,\ o_1'/0.6,\ o_2'/0.8\} \in$ *Old-Employee*;
- It is shown from (*a*)-(*b*) that the two instance sets of *Old-Employee* are conflictive. Therefore, we know that the instance set of *Young-Employee* is an *empty set*, because the *empty set* is the only set that can be *at the same time disjoint from* and *contained in* the class *Old-Employee*. That is, *Young-Employee* is unsatisfiable since it is an empty class.
- Since *Young-Employee* is an empty class, and *Employee* is the union of *Young-Employee* and *Old-Employee*, we know that *Employee* is equivalent to *Old-Employee*, i.e., there is redundancy in the fuzzy UML model.
- The above cases may result in that other undesirable problems occur in a more complete fuzzy UML model.

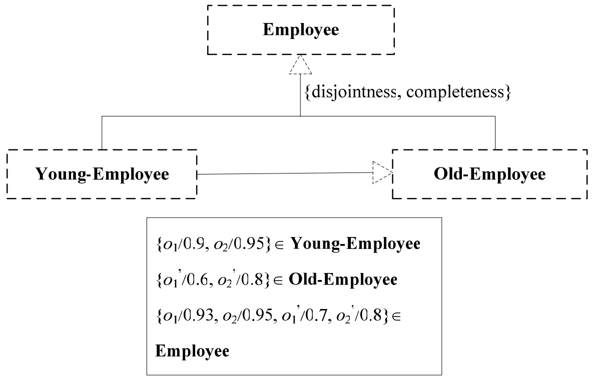

Employee
{disjointness, completeness}
Young-Employee
Old-Employee
{o1/0.9, o2/0.95} ∈ Young-Employee
{o1'/0.6, o2'/0.8} ∈ Old-Employee
{o1/0.93, o2/0.95, o1'/0.7, o2'/0.8} ∈ Employee

Fig. 13. A simple fuzzy UML model $\mathcal{F}_{\text{UML2}}$.

All the reasoning problems mentioned above may occur in the fuzzy UML modeling activities, and the burden of checking these problems is left to the designers, which is a complex and time-consuming task and may have some disadvantages such as a low reasoning efficiency and reliability. Therefore, it would be highly desirable to improve the ability of reasoning on fuzzy UML models. Based on the previous sections of this paper, a fuzzy UML model can be transformed into a fuzzy OWL ontology, and in the following we further make an attempt to resolve the problem of reasoning on fuzzy UML models by means of the reasoning mechanism of the constructed fuzzy OWL ontologies.

5.2. *The Reasoning Problems of Fuzzy UML Models*

Generally speaking, the familiar reasoning problems of fuzzy UML models include *consistency*, *satisfiability*, *subsumption*, and *redundancy*. Based on Refs. 30, 38, in the following we give formal definitions of the reasoning problems above in more detail, where some reasoning problems (such as consistency, satisfiability and redundancy) are redefined and studied in depth.

Definition 7 (consistency of fuzzy UML models). As shown in Section 3.1, a fuzzy UML model may contain its structure information and its instance information (see Fig. 6 and Fig. 8). A fuzzy UML model is *consistent*, if the set of object instances satisfies all the constraints of the structure information.

In particular, when no object instance exists, the consistency problem of the fuzzy UML model above is reduced to the satisfiability problem of the fuzzy UML model in Definition 8.

Definition 8 (satisfiability of fuzzy UML models). When there are not object instances in a fuzzy UML model, the fuzzy UML model is *satisfiable*, if it admits at least one fuzzy UML instantiation (i.e., a set of object instances).

If a fuzzy UML model is not satisfiable, the classes altogether are contradictory, i.e., it does not allow that any class can be populated without violating any of the requirements imposed by the structure information of the fuzzy UML model. This may be due to a design error or over-constraining.

Definition 9 (satisfiability of fuzzy classes). A fuzzy class $FC \in FC_{\mathcal{F}}$ in a fuzzy UML model $\mathcal{F}_{\text{UML}}$ is *satisfiable*, if there is at least one fuzzy object state/instance $\mathcal{FB}$ of $\mathcal{F}_{\text{UML}}$ mentioned in Definition 2 such that $FC^{\mathcal{FB}} \neq \varnothing$.

A fuzzy class FC in a fuzzy UML model $\mathcal{F}_{\text{UML}}$ is satisfiability, i.e., $\mathcal{F}_{\text{UML}}$ admits a fuzzy object state in which FC has a non-empty set of objects. An unsatisfiable fuzzy class weakens the understandability of a fuzzy UML model, since the fuzzy class stands for an empty class, and thus, at the very least, it is inappropriately named. In this case, designers should modify or delete the fuzzy class to increase understandability.

Definition 10 (subsumption of fuzzy classes). Given two fuzzy classes FC_1 and FC_2 in a fuzzy UML model $\mathcal{F}_{\text{UML}}$, if for each fuzzy object state $\mathcal{FB}$ of $\mathcal{F}_{\text{UML}}$, $FC_1^{\mathcal{FB}}$ is a subset of $FC_2^{\mathcal{FB}}$ (see Definition 3), then FC_1 is a subclass of FC_2.

Definition 10 denotes that a fuzzy class FC_1 is a subclass of another fuzzy class FC_2 if, for any object, the membership degree of it to FC_1 is *less than or equal to* the membership degree of it to FC_2. The class subsumption is the basis for a classification of all the classes in a fuzzy UML model.

Definition 11 (redundancy of fuzzy UML models). A fuzzy UML model is *redundant*, if there is some class standing for an empty class, i.e., it is inappropriately named; or there are two equivalent fuzzy classes.

In Definition 11, *firstly*, there is some class standing for an empty class, i.e., there is some unsatisfiable fuzzy class (see Definition 9). *Secondly*, we need to introduce the notion of *equivalent fuzzy classes*. As we have known, in the classical UML model, two classes are equivalent if they denote the same set of object instances. In a fuzzy UML model, however, an object may belong to a fuzzy class with membership degree of [0, 1], and thus two fuzzy classes may have same objects with same/different membership degrees. In this case, the notion of equivalent classes should be extended under fuzzy environment as follows.

In a fuzzy UML model, we said that a fuzzy class FC_1 is equivalent to another fuzzy class FC_2, if at least one of the following conditions is satisfied: (*i*) two fuzzy classes have same objects with same membership degrees; (*ii*) two fuzzy classes do not have subclass/superclass relationship but they have same objects with different membership degrees. Note that, for case (i), when two fuzzy classes have same objects with same membership degrees, the two fuzzy classes are strictly equivalent. In this case, one of the fuzzy classes can be removed and replaced by another. For case (ii), let FC_1 and FC_2 be two fuzzy classes, and assume that there is an object FO belonging to the two fuzzy classes with different membership degrees $u_{FC1}(FO) \in (0, 1]$ and $u_{FC2}(FO) \in (0, 1]$. At this moment, which one in FC_1 and FC_2 is the class of object FO depends on the following cases: if $u_{FC1}(FO) > u_{FC2}(FO)$, then FC_1 is considered as the class of object FO and we say FC_1 fuzzily includes FC_2, else FC_2 is considered as the class of object FO and we say FC_2 fuzzily includes FC_1. Correspondingly, the fuzzy class which is included by another fuzzy class is removed, and the fuzzy class which includes the removed fuzzy class is retained.

The reasoning problems mentioned in Definitions 7-11 may occur in the fuzzy UML modeling activities, and the following section will study how to reason on these problems of fuzzy UML models by means of the reasoning mechanism of the constructed fuzzy OWL ontologies.

5.3. *Reasoning on Fuzzy UML Models with the Constructed Fuzzy OWL Ontologies*

After transforming a fuzzy UML model into a fuzzy OWL ontology based on the previous sections, the following theorems allow us to reduce reasoning on the fuzzy UML model to reasoning on the fuzzy OWL ontology, so that the reasoning problems of the fuzzy UML model mentioned in Section 5.2 may be checked through the reasoning mechanism of the constructed fuzzy OWL ontology.

Theorem 2 (consistency of fuzzy UML models). Given a fuzzy UML model $\mathcal{F}_{\text{UML}}$, and $\varphi(\mathcal{F}_{\text{UML}})$ is the constructed fuzzy OWL ontology according to the proposed approach in Section 4. The fuzzy UML model $\mathcal{F}_{\text{UML}}$ is *consistent* iff $\varphi(\mathcal{F}_{\text{UML}})$ is *consistent*, i.e., the fuzzy OWL ontology instance is consistent w.r.t. the fuzzy OWL ontology structure.

Based on Definitions 7 and 8, when there are not object instances in a fuzzy UML model (i.e., the constructed fuzzy OWL ontology contains only the fuzzy OWL ontology structure), the consistency problem of the fuzzy UML model (Theorem 2) will be reduced to the satisfiability problem of the fuzzy UML model (see Theorem 3).

Theorem 3 (satisfiability of fuzzy UML models). Given a fuzzy UML model $\mathcal{F}_{\text{UML}}$ (only contains the structure information), and $\varphi(\mathcal{F}_{\text{UML}})$ is the constructed

Copyright: the authors

fuzzy OWL ontology according to the proposed approach in Section 4. $\mathcal{F}_{\text{UML}}$ is *satisfiable* iff $\varphi(\mathcal{F}_{\text{UML}})$ is satisfiable, i.e., there is a fuzzy interpretation $\mathcal{FI}$ which is a model of $\varphi(\mathcal{F}_{\text{UML}})$.

Theorems 2 and 3 are the straightforward consequences of Definitions 7 and 8, and thus their proofs are omitted here.

Theorem 4 (satisfiability of fuzzy classes). Given a fuzzy UML model $\mathcal{F}_{\text{UML}}$, $\varphi(\mathcal{F}_{\text{UML}})$ is the constructed fuzzy OWL ontology according to the proposed approach in Section 4, and FC is a fuzzy class in $\mathcal{F}_{\text{UML}}$. FC is *satisfiable* in $\mathcal{F}_{\text{UML}}$ iff $\varphi(\mathcal{F}_{\text{UML}}) \nvDash \varphi(FC) \sqsubseteq \bot$.

Proof. "⇒": If FC is satisfiable, then there is a legal fuzzy object state $\mathcal{FB}$ with $FC^{\mathcal{FB}} \neq \emptyset$. By part 1 of Theorem 1, $\alpha_{\mathcal{F}}(\mathcal{FB})$ is a model of $\varphi(\mathcal{F}_{\text{UML}})$ with $FC^{\mathcal{FB}} = (\varphi(FC))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$, so $(\varphi(FC))^{\alpha_{\mathcal{F}}(\mathcal{FB})} \neq \emptyset$. That is, $\varphi(\mathcal{F}_{\text{UML}}) \nvDash \varphi(FC) \sqsubseteq \bot$.

"⇐": If $\varphi(\mathcal{F}_{\text{UML}}) \nvDash \varphi(FC) \sqsubseteq \bot$, then $\varphi(FC)$ is consistent in $\varphi(\mathcal{F}_{\text{UML}})$, i.e., there is a fuzzy interpretations $\mathcal{FI}$ of $\varphi(\mathcal{F}_{\text{UML}})$ with $(\varphi(FC))^{\mathcal{FI}} \neq \emptyset$. By part 2 of Theorem 1, $\beta_{\mathcal{F}}(\mathcal{FI})$ is a legal fuzzy object state for $\mathcal{F}_{\text{UML}}$ with $(\varphi(FC))^{\mathcal{FI}} = FC^{\beta_{\mathcal{F}}(\mathcal{FI})}$, so $FC^{\beta_{\mathcal{F}}(\mathcal{FI})} \neq \emptyset$. That is, FC is satisfiable. □

Theorem 5 (subsumption of fuzzy classes). Given a fuzzy UML model $\mathcal{F}_{\text{UML}}$, $\varphi(\mathcal{F}_{\text{UML}})$ is the constructed fuzzy OWL ontology according to the proposed approach in Section 4, and FC_1 and FC_2 are two fuzzy classes in $\mathcal{F}_{\text{UML}}$. FC_1 is a subclass of FC_2 in $\mathcal{F}_{\text{UML}}$ iff $\varphi(\mathcal{F}_{\text{UML}}) \vDash \varphi(FC_1) \sqsubseteq \varphi(FC_2)$.

Proof. "⇒": If $\varphi(\mathcal{F}_{\text{UML}}) \nvDash \varphi(FC_1) \sqsubseteq \varphi(FC_2)$, i.e., $\varphi(FC_1) \sqcap \neg\varphi(FC_2)$ is consistent in $\varphi(\mathcal{F}_{\text{UML}})$, then there is a fuzzy interpretations $\mathcal{FI}$ of $\varphi(\mathcal{F}_{\text{UML}})$ with $(\varphi(FC_1) \sqcap \neg\varphi(FC_2))^{\mathcal{FI}} \neq \emptyset$, i.e., $\exists d.\ d \in (\varphi(FC_1))^{\mathcal{FI}}$ and $d \notin (\varphi(FC_2))^{\mathcal{FI}}$. By part 2 of Theorem 1, $\beta_{\mathcal{F}}(\mathcal{FI})$ is a legal fuzzy object state for $\mathcal{F}_{\text{UML}}$ with $(\varphi(FC_1))^{\mathcal{FI}} = FC_1^{\beta_{\mathcal{F}}(\mathcal{FI})}$ and $(\varphi(FC_2))^{\mathcal{FI}} = FC_2^{\beta_{\mathcal{F}}(\mathcal{FI})}$, i.e., $\exists d.\ d \in FC_1^{\beta_{\mathcal{F}}(\mathcal{FI})}$ and $d \notin FC_2^{\beta_{\mathcal{F}}(\mathcal{FI})}$. That is, FC_1 is not a subclass of FC_2, and thus there is a contradiction, i.e., $\varphi(\mathcal{F}_{\text{UML}}) \vDash \varphi(FC_1) \sqsubseteq \varphi(FC_2)$.

"⇐": If FC_1 is not a subclass of FC_2, then there is a legal fuzzy object state $\mathcal{FB}$ for $\mathcal{F}_{\text{UML}}$ such that $c \in FC_1^{\mathcal{FB}}$ and $c \notin FC_2^{\mathcal{FB}}$. By part 1 of Theorem 1, $\alpha_{\mathcal{F}}(\mathcal{FB})$ is a model of $\varphi(\mathcal{F}_{\text{UML}})$ with $FC_1^{\mathcal{FB}} = (\varphi(FC_1))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$ and $FC_2^{\mathcal{FB}} = (\varphi(FC_2))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$, i.e., $c \in (\varphi(FC_1))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$ and $c \notin (\varphi(FC_2))^{\alpha_{\mathcal{F}}(\mathcal{FB})}$. That is, $\varphi(\mathcal{F}_{\text{UML}}) \nvDash \varphi(FC_1) \sqsubseteq \varphi(FC_2)$, and thus there is a contradiction, i.e., FC_1 is a subclass of FC_2. □

Theorem 6 (redundancy of fuzzy UML models). Given a fuzzy UML model $\mathcal{F}_{\text{UML}}$, $\varphi(\mathcal{F}_{\text{UML}})$ is the constructed fuzzy OWL ontology according to the proposed approach in Section 4, and FC, FC_1 and FC_2 are three fuzzy classes in $\mathcal{F}_{\text{UML}}$. $\mathcal{F}_{\text{UML}}$ is redundant if and only if at least one of the following conditions is satisfied: (*i*) $\varphi(\mathcal{F}_{\text{UML}}) \vDash \varphi(FC) \sqsubseteq \bot$; (*ii*) $\varphi(\mathcal{F}_{\text{UML}}) \vDash \varphi(FC_1) \equiv \varphi(FC_2)$; (*iii*) $\varphi(\mathcal{F}_{\text{UML}}) \nvDash \varphi(FC_1) \sqsubseteq \varphi(FC_2)$, $\varphi(\mathcal{F}_{\text{UML}}) \nvDash \varphi(FC_2) \sqsubseteq \varphi(FC_1)$, and $\varphi(FC_1)$ and $\varphi(FC_2)$ contain same individuals.

Note that, as mentioned in Definition 11, a fuzzy UML model is *redundant*, if there is some class standing for an empty class or there are two equivalent fuzzy classes. *Firstly*, for the first condition, a class is an empty class, i.e., the class is unsatisfiable, which can be reasoned by checking whether $\varphi(\mathcal{F}_{\text{UML}}) \vDash \varphi(FC) \sqsubseteq \bot$ according to Theorem 4. *Secondly*, for the second condition, which can be partitioned into the following two cases: (*a*) when two fuzzy classes have same individuals with same membership degrees, the equivalence problem of fuzzy classes can be reasoned by checking whether $\varphi(\mathcal{F}_{\text{UML}}) \vDash \varphi(FC_1) \equiv \varphi(FC_2)$, i.e., $\varphi(\mathcal{F}_{\text{UML}}) \vDash \varphi(FC_1) \sqsubseteq \varphi(FC_2)$ and $\varphi(\mathcal{F}_{\text{UML}}) \vDash \varphi(FC_2) \sqsubseteq \varphi(FC_1)$ in Theorem 5; (*b*) when two fuzzy classes do not have subclass/superclass relationship but they have same objects with different membership degrees, it cannot be handled directly by checking whether $\varphi(\mathcal{F}_{\text{UML}}) \vDash \varphi(FC_1) \equiv \varphi(FC_2)$. In this case, the constraint that two fuzzy classes do not have subclass/superclass relationship can be reasoned by checking whether $\varphi(\mathcal{F}_{\text{UML}}) \nvDash \varphi(FC_1) \sqsubseteq \varphi(FC_2)$ and $\varphi(\mathcal{F}_{\text{UML}}) \nvDash \varphi(FC_2) \sqsubseteq \varphi(FC_1)$ in Theorem 5. Furthermore, whether two fuzzy classes contain same objects can be done by detecting whether $\varphi(FC_1)$ and $\varphi(FC_2)$ have the same individuals, which can be checked by the retrieval reasoning service of fuzzy ontologies[48]. Theorem 6 is the straightforward consequence of Definition 11, and the proof is hereby omitted here.

In summary, **Fig. 14** describes the process of reasoning on the fuzzy UML model $\mathcal{F}_{\text{UML1}}$ in Section 3.1.3 based on the approaches proposed in the previous sections of this paper:

- Firstly, according to the construction approach of fuzzy OWL ontologies proposed in Section 4, the fuzzy UML model $\mathcal{F}_{\text{UML1}}$ and its instantiation (see Section 3.1.3) can be transformed into the fuzzy OWL ontology $\varphi(\mathcal{F}_{\text{UML1}})$ as have been shown in Fig. 9 and Fig. 10.

- Then, based on the reasoning reduction approaches in this section, the reasoning tasks of the fuzzy UML model $\mathcal{F}_{UML1}$ (e.g., consistency, satisfiability, subsumption, and redundancy as mentioned in Section 5.2) may be checked by detecting the reasoning problems of the constructed fuzzy OWL ontology $\varphi(\mathcal{F}_{UML1})$. Finally, reasoning on the fuzzy OWL ontology $\varphi(\mathcal{F}_{UML1})$ can be automatically handled by means of the existing fuzzy ontology reasoner DeLorean[48].

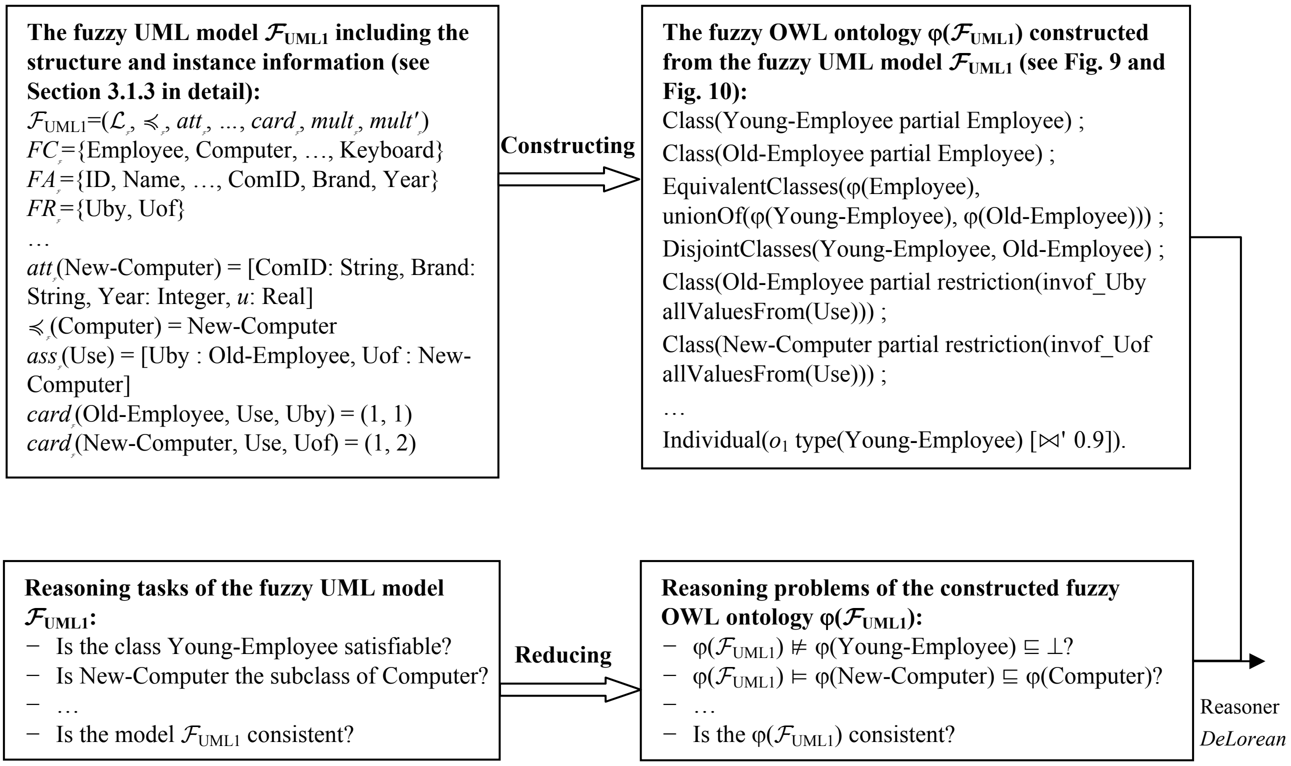

The fuzzy UML model $\mathcal{F}_{UML1}$ including the structure and instance information (see Section 3.1.3 in detail):
$\mathcal{F}_{UML1}=(\mathcal{L}_\mathcal{F}, \preccurlyeq_\mathcal{F}, att_\mathcal{F}, ..., card_\mathcal{F}, mult_\mathcal{F}, mult')$
$FC_\mathcal{F}$={Employee, Computer, …, Keyboard}
$FA_\mathcal{F}$={ID, Name, …, ComID, Brand, Year}
$FR_\mathcal{F}$={Uby, Uof}
…
$att_\mathcal{F}$(New-Computer) = [ComID: String, Brand: String, Year: Integer, u: Real]
$\preccurlyeq_\mathcal{F}$(Computer) = New-Computer
$ass_\mathcal{F}$(Use) = [Uby : Old-Employee, Uof : New-Computer]
$card_\mathcal{F}$(Old-Employee, Use, Uby) = (1, 1)
$card_\mathcal{F}$(New-Computer, Use, Uof) = (1, 2)
Constructing
The fuzzy OWL ontology $\varphi(\mathcal{F}_{UML1})$ constructed from the fuzzy UML model $\mathcal{F}_{UML1}$ (see Fig. 9 and Fig. 10):
Class(Young-Employee partial Employee) ;
Class(Old-Employee partial Employee) ;
EquivalentClasses(φ(Employee), unionOf(φ(Young-Employee), φ(Old-Employee))) ;
DisjointClasses(Young-Employee, Old-Employee) ;
Class(Old-Employee partial restriction(invof_Uby allValuesFrom(Use))) ;
Class(New-Computer partial restriction(invof_Uof allValuesFrom(Use))) ;
…
Individual(o_1 type(Young-Employee) [⋈' 0.9]).
Reasoning tasks of the fuzzy UML model $\mathcal{F}_{UML1}$:
− Is the class Young-Employee satisfiable?
− Is New-Computer the subclass of Computer?
− …
− Is the model $\mathcal{F}_{UML1}$ consistent?
Reducing
Reasoning problems of the constructed fuzzy OWL ontology $\varphi(\mathcal{F}_{UML1})$:
− $\varphi(\mathcal{F}_{UML1}) \nvDash \varphi$(Young-Employee) $\sqsubseteq \bot$?
− $\varphi(\mathcal{F}_{UML1}) \vDash \varphi$(New-Computer) $\sqsubseteq \varphi$(Computer)?
− …
− Is the $\varphi(\mathcal{F}_{UML1})$ consistent?
Reasoner
DeLorean

Fig. 14. Reasoning on the fuzzy UML model $\mathcal{F}_{UML1}$ in Section 3.1.3 with the constructed fuzzy OWL ontology $\varphi(\mathcal{F}_{UML1})$.

So far, based on the previous sections, it is possible to establish the correspondences between fuzzy UML models and fuzzy ontologies:

- Based on **Section 3**, fuzzy UML models and fuzzy ontologies can be represented and interpreted by their respective formal definitions and semantic interpretation methods;
- On the basis of the formal definitions above and **Section 4**, fuzzy ontologies can be constructed from fuzzy UML models, i.e., fuzzy UML models (including the structure and instance information) can be transformed into the corresponding fuzzy ontologies. Constructing fuzzy ontologies from fuzzy UML models may facilitate the development of the Semantic Web fuzzy ontologies.
- After constructing fuzzy ontologies from fuzzy UML models, the constructed fuzzy ontologies may be useful for reasoning on fuzzy UML models as shown in **Section 5**. By reducing the reasoning tasks of a fuzzy UML model to the reasoning problems of the constructed fuzzy ontology, the reasoning tasks of the fuzzy UML model may be checked by detecting automatically the reasoning problems of the constructed fuzzy ontology by means of the existing fuzzy ontology reasoner DeLorean[48].

6. Related Work

How to construct classical/fuzzy ontologies has become a key technology to enable the Semantic Web, and kinds of methods and tools were developed to construct crisp/fuzzy ontologies. The following categories of approaches are related to our work according to their focuses, including *classical ontology construction*, *fuzzy*

UML models and *fuzzy ontologies*, and *fuzzy ontology construction*:

(i) How to construct classical ontologies:

Regarding how to construct classical ontologies, **Table 4** shows some approaches for constructing ontologies from various data sources, including UML model. It should be noted that Table 4 does not cover all publications in the research area. For a comprehensive review of ontology construction, please refer to Refs. 4, 49, and 50. Moreover, less research on how to reason on UML models with the Semantic Web knowledge representation techniques, especially ontologies, has been done. There are several works[51,52] establishing correspondences between UML models and description logics. However, the above research works were not sufficient for handling imprecise and uncertain information that is commonly found in real-world applications.

Table 4. Survey of classical ontology construction approaches.

	Different types of inputs

Ontology construction	Text, Dictionary, Knowledge base (see Refs. 4, 49, and 50 for overviews);
	ER model[5,53];
	Relational database model[7,54,55];
	Object-oriented database model[56];
	XML data model[57,58];

	UML model[6,11,12,13,14].

(ii) How to represent imprecise and uncertain information in UML models and ontologies:

Regarding how to modeling imprecise and uncertain information in UML models, several works have been done in extending UML models based on the fuzzy set theory[32,33,34,35,39,59]. How to apply fuzzy UML for uncertain systems modeling is investigated[32,59]. Without including formal representations, different levels of fuzziness were introduced into the UML model in Refs. 33, 34 and the mappings of the fuzzy UML model into the fuzzy XML model and the fuzzy relational schema were hereby developed. An approach for mapping activity diagram in fuzzy UML to fuzzy Petri net was presented[35]. How to integrate the fuzziness into the fuzzy UML model was investigated[39]. A comprehensive literature review of fuzzy data models can be found in Ref. 60.

Also, *in order to represent and reason on imprecise and uncertain information in ontologies*, some approaches have been developed to characterize or define fuzzy ontologies. Calegari[18,19] integrated fuzzy logic in ontologies, and developed a plug-in for the KAON Project in order to introduce fuzziness in ontologies. Lee[44] presented a four-layered fuzzy ontology and applied it to news summarization. Sanchez[23] introduced a fuzzy ontology structure from the aspects of lexicon and knowledge base. Lam[21] proposed a fuzzy ontology map by extending the crisp ontology with the fuzzy theory and graph theory. The comprehensive reviews on fuzzy ontologies and some relevant applications can be found in Ref. 46.

(iii) How to construct fuzzy ontologies:

Regarding the requirement of constructing fuzzy ontologies, some efforts have been made to construct fuzzy ontologies from various data sources. Here, we *first* briefly summarize some work about the construction of fuzzy ontologies from data sources such as fuzzy narrower term relations and fuzzy context. *Then*, we introduce several approaches for constructing fuzzy ontologies from fuzzy database models. *Finally*, we introduce the most similar work in Ref. 30 and provide detailed comparisons between it and our work.

In order to construct fuzzy ontologies from some data sources, Widyantoro[29] described a PASS (Personalized Abstract Search Services) system and an automatic technique to build a fuzzy ontology of term associations for query refinement in the system, where the fuzzy ontology construction is grouped into two stages. The first stage is to create a full ontology from fuzzy narrower term relations, and the full fuzzy ontology is then pruned by eliminating unnecessary relations in the second stage. Quan[28] proposed a framework known as FOGA (Fuzzy Ontology Generation frAmework) that can automatically generate a fuzzy ontology from uncertainty data based on Formal Concept Analysis (FCA) theory. De Maio[25] presented an approach for automatic generation of a fuzzy ontology, where the approach gave the mapping steps for translating the fuzzy lattice generated by Formal Concept Analysis (FCA) theory into an OWL based ontology. Moreover, Gu[61] proposed a reasoning-enabled general fuzzy ontology based on the observation that three general fuzzy relations (i.e., fuzzy instance relation, fuzzy concept relation, and fuzzy concept base relation) exist widely among the real world, and also introduced a reference process for building these general fuzzy relations in fuzzy ontology applications.

Abulaish[62] presented a fuzzy ontology generation framework in which concept descriptors and inter-concept relations are represented as fuzzy relations. This work has been integrated with a text-mining system such that, starting with a seed ontology a domain ontology can be extended with new knowledge extracted from text documents. In addition, Inyaem[26] proposed a methodology for constructing terrorism fuzzy ontology for event extraction work using Web Ontology Language (OWL). Ceravolo[20] presented a way of building specification of a shared conceptualization proceeding in a bottom-up fashion, i.e., defining concepts as clusters of concrete objects, where the bottom-up ontologies are based on discovery of implicit part-whole and is-a relations. Also, the authors discussed how metadata based on bottom-up ontologies can be associated with a flexible degree of trust by collecting user feedback. Chen[63] proposed an automatic fuzzy ontology generation approach from fuzzy context, where fuzzy formal concept analysis and fuzzy concept hierarchy structure are adopted to automatically generate primitive fuzzy ontology, and a complete fuzzy ontology model can be generated through the extension of primitive ontology.

Besides, there are several approaches for constructing fuzzy ontologies from fuzzy database models. A fuzzy ontology construction approach from fuzzy relational models was proposed in Ref. 27, where the authors first implicitly transformed the fuzzy relational schema into the fuzzy ER model by means of reverse engineering, and then transformed the fuzzy ER model and data instances into the fuzzy ontology structure and fuzzy RDF data model, respectively. Also, an ontology system was proposed in Ref. 64 to represent the fuzzy information of a fuzzy relational database. Owing to the formality of this representation, fuzzy metaknowledge base access is more accessible to users or applications which use the ontology as an interface for access. Moreover, a fuzzy ontology construction approach from fuzzy ER models was developed in Ref 45, where the authors transformed the fuzzy ER model and the corresponding data instances into the fuzzy ontology. How to construct fuzzy ontologies from fuzzy object-oriented database models was investigated in Ref. 31. In addition, a fuzzy description logic approach for representing and reasoning on fuzzy UML models was presented in Ref. 38. All of the work above gives us good hints for developing our approach in this paper, but the goals and approaches of the work above are different comparing to our research, and our aim in this paper is to develop a formal and automatic approach for constructing fuzzy ontologies from fuzzy UML models as introduced in the previous sections.

In addition, how to construct fuzzy ontologies from fuzzy UML models was initially investigated in Ref. 30. The current paper differs from Ref. 30 in three major aspects: ***(i)*** *Regarding the formalization problems of fuzzy UML models and fuzzy ontologies*, this paper gives the *semantic interpretation method* of fuzzy UML models (see Section 3.1.2 in this paper), which was missed in Ref. 30; And this paper proposes a *more complete formal definition of fuzzy UML models*, where this paper considers both the structural and dynamic aspects of fuzzy UML models and adds some new functions into the formal definition for capturing some important features of fuzzy UML models (see Definition 1 in this paper); Moreover, this paper gives a *formal definition of fuzzy ontologies* including the fuzzy ontology structure and the fuzzy ontology instance, and the fuzzy ontology instance (i.e., a set of fuzzy individual axioms) is missed in Ref. 30; ***(ii)*** *Regarding the construction of fuzzy ontologies from fuzzy UML models* (see Section 4 in this paper), *first*, this paper proposes a complete fuzzy ontology construction approach which considers both the structural and dynamic aspects of fuzzy UML models, where the dynamic aspect of fuzzy UML models was not discussed in Ref. 30, and the proof of correctness of the construction approach is given in this paper but missed in Ref. 30; *Second*, the construction approaches from fuzzy UML models to fuzzy ontologies at instance level are different in this paper and Ref. 30, the work in Ref. 30 transformed the structure information of a fuzzy UML model into a fuzzy ontology structure but transformed the object instance information of the fuzzy UML model into a fuzzy RDF data model, where the fuzzy ontology structure (i.e., a set of fuzzy class and property axioms) and the fuzzy ontology instance (i.e., the fuzzy RDF data model) were not written using the same fuzzy ontology representation syntax, but this paper transforms the structure and instance information of a fuzzy UML model into the fuzzy ontology structure and instance using the same syntax, and both of the fuzzy ontology structure and instance are represented as the forms of fuzzy axioms, which may be more useful for the reasoning of fuzzy UML models and the access to and evaluation of fuzzy ontologies; *Third*, the prototype construction tool was developed in this paper and was missed in Ref. 30; ***(iii)*** *Regarding the problem of reasoning on fuzzy UML models with the constructed fuzzy ontologies*, the work in Ref. 30 only briefly discussed and gave several theorems, and the detailed reasoning procedures of fuzzy UML models were missed in Ref. 30. Based on the initial idea in Ref. 30, this paper well introduces the motivation of reasoning on fuzzy UML models with the constructed fuzzy ontologies, gives the complete formal definitions of reasoning problems of fuzzy UML models (where several reasoning problems are added and some reasoning problems are redefined and studied in depth), and proposes the reasoning methods and gives the proofs of correctness of the reasoning methods (see Section 5 in this paper).

Based on the observations above, so far, to our best knowledge, there is no complete report on constructing fuzzy ontologies from fuzzy UML models and reasoning on fuzzy UML models with the constructed fuzzy ontologies.

7. Conclusions

We proposed a formal approach and developed an automated tool for constructing fuzzy ontologies from fuzzy UML models, and further studied how the constructed fuzzy ontologies may be useful for reasoning on fuzzy UML models. *Firstly*, a complete formal definition of fuzzy UML models was proposed. *Then*, we proposed an approach for constructing fuzzy ontologies from fuzzy UML models, i.e., transforming fuzzy UML models (including the structure and instance information of fuzzy UML models) into fuzzy ontologies. *Furthermore*, following the proposed approach, we implemented a prototype transformation tool *FUML2FOnto*, which can automatically construct fuzzy ontologies from fuzzy UML models. *Finally*, how to reason on fuzzy UML models based on the constructed fuzzy ontologies was investigated, and it turned out that the reasoning tasks of fuzzy UML models can be checked by the reasoning mechanism of the fuzzy ontologies.

Summarizing, given a fuzzy UML model represented in the form of Definition 1, it can be transformed into a fuzzy ontology based on the proposed approach in the paper. Further, the constructed fuzzy ontology may be used to reason on the fuzzy UML model by means of the reasoning mechanism of the fuzzy ontology. It should be noted that, although several works investigated fuzzy UML models by considering different features of fuzzy UML models, there is not a standard definition of fuzzy UML models. We do not expect our definition of fuzzy UML models to become a standard now, but it considers most of general and important features of fuzzy UML models as usually mentioned in the literature. The formalization in fuzzy ontologies of fuzzy UML models may facilitate the development of the Semantic Web fuzzy ontologies and also be considered as the basic steps towards developing intelligent systems that provide computer-aided support to reason on fuzzy UML models.

As far as future work, we intend to test and evaluate the construction approach and tool with more and complex examples. Also, we aim at developing other approaches of constructing classical/fuzzy ontologies, such as constructing fuzzy ontologies from fuzzy object-relational database models[65].

Acknowledgements

The authors wish to thank the anonymous referees for their valuable comments and suggestions, which improved the technical content and the presentation of the paper. The work is supported by *National Natural Science Foundation of China* (61073139, 60873010, and 61202260) and by *Program for New Century Excellent Talents in University* (NCET-05-0288).

References

1. T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, *Scientific American* **284**(5) (2001) 34–43.
2. T. Berners-Lee, W. Hall, J. Hendler, N. Shadbolt and D. Wietzner, Creating a science of the Web, *Science* **313**(11) (2006) 769–771.
3. E. K. Jacob, Ontologies and the semantic web, *Bulletin of the American Society for Information Science and Technology* **29**(4) (2003) 19–22.
4. A. Maedche and S. Steffen, Ontology Learning for the Semantic Web, *IEEE Intelligent Systems* **16**(2) (2001) 72–79.
5. M. Fahad, ER2OWL: Generating OWL Ontology from ER Diagram, in *Proc. 5th Int. Conf. Intelligent Information Processing* (Beijing, China, 2008), pp. 28–37.
6. K. Baclawski, M. Kokar and P. Kogut, et al, Extending UML to Support Ontology Engineering for the Semantic Web. In *Proc. 4th Int. Conf. on UML* (Toronto, Canada, 2001), pp. 342–360.

7. Y. An, A. Borgida and J. Mylopoulos, Refining Semantic Mappings from Relational Tables to Ontologies, *In Proc. of 2nd International workshop on Semantic Web and Databases* (Toronto, Canada, 2004), pp. 84–90.
8. T. A. Halpin, Metaschemas for ER, ORM and UML data models: a comparison, *Journal of Database Management* **13**(2) (2002) 20–30.
9. G. Booch, J. Rumbaugh and I. Jacobson, *The Unified Modeling Language User Guide* (Addison-Welsley Longman, Inc, 1998).
10. *Object Management Group (OMG), Unified Modeling Language (UML), Version 1.5*, Technical report, OMG, 2003, www.omg.org.
11. S. Cranefield, UML and the Semantic Web. In *Proc. International Semantic Web Working Symposium* (Palo Alto, California, 2001), pp. 113–130.
12. D. Djurić, D. Gašević and V. Devedžić, Ontology Modeling and MDA, *Journal on Object Technology* **4**(1) (2005) 109–128.
13. K. Falkovych, M. Sabou and H. Stuckenschmidt, UML for the Semantic Web: Transformation-Based Approaches, in *Knowledge Transformation for the Semantic Web*, eds. B. Omelayenko and M. Klein (IOS Press, 2003), pp. 92–106.
14. H. S. Na, O. H. Choi and J. E. Lim, A Method for Building Domain Ontologies based on the Transformation of UML Models, in *Proc. 4th Int. Conf. Software Engineering Research, Management and Applications* (Seattle, Washington, USA, 2006), pp. 332–338.
15. J. Galindo (eds.), *Handbook of Research on Fuzzy Information Processing in Databases* (Information Science Reference, Hershey, 2008).
16. G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications* (Prentice-Hall, Englewood, NJ, 1995).
17. T. Lukasiewicz and U. Straccia, Managing uncertainty and vagueness in description logics for the Semantic Web, *Web Semantics: Science. Services and Agents on the World Wide Web* **6**(4) (2008) 291–308.
18. S. Calegari and D. Ciucci, Fuzzy ontology, fuzzy description logics and fuzzy-owl, in *Proc. of WILF 2007* (Camogli, Italy, 2007), pp. 118–126.
19. S. Calegari and D. Ciucci, Integrating fuzzy logic in ontologies, in *Proc. 9th International Conference on Enterprise Information Systems* (Setúbal, Portugal, 2006), pp. 66–73.
20. P. Ceravolo, A. Corallo and E. Damiani, et al, Bottom-up Extraction and Maintenance of Ontology-based Metadata, in E. Sanchez: *Fuzzy Logic and the Semantic Web* (Elsevier, Amsterdam, 2006), pp. 265–282.
21. T. H. W. Lam, Fuzzy ontology map-a fuzzy extension of the hard-constraint ontology, in *Proc. of 5th the IEEE/WIC/ACM International Conference on Web Intelligence* (Hong Kong, China, 2006), pp. 506–509.
22. D. Parry, Fuzzy ontologies for information retrieval on the WWW, in E. Sanchez: *Fuzzy Logic and the Semantic Web* (Elsevier, Amsterdam, 2006), pp. 21–48.
23. E. Sanchez and T. Yamanoi, Fuzzy ontologies for the semantic web, in *Proc. 7th International Conference on Query Answering Systems* (Milan, Italy, 2006), pp. 691–699.
24. C. Thomas and A. Sheth, On the Expressiveness of the Languages for the Semantic Web-Making a Case for 'A Little More', in E. Sanchez: *Fuzzy Logic and the Semantic Web* (Elsevier, Amsterdam, 2006), pp. 3–20.
25. C. De Maio, G. Fenza, V. Loia and S. Senatore, Towards an Automatic Fuzzy Ontology Generation, In *Proc. of the 2009 IEEE International Conference on Fuzzy Systems* (Jeju Island, Korea, 2009), pp. 1044–1049.
26. U, Inyaem, P. Meesad, C. Haruechaiyasak and D. Tran, Construction of Fuzzy Ontology-Based Terrorism Event Extraction, in *Proc. of the third International Conference on Knowledge Discovery and Data Mining* (Phuket, Thailand, 2010), pp. 391–394.
27. Z. M. Ma, Y. Lv and L. Yan, A Fuzzy Ontology Generation Framework from Fuzzy Relational Databases, *Int. J. Semantic Web Information Systems* **4**(3) (2008) 1–15.
28. T. T. Quan, S. C. Hui, A. C. M. Fong and T. H. Cao, Automatic fuzzy ontology generation for Semantic Web, *IEEE Transaction on Knowledge and Data Engineering* **18**(6) (2006) 842–856.
29. D. H. Widyantoro and J. Yen, A Fuzzy Ontology-based Abstract Search Engine and Its User Studies, in *Proc. of the 10th IEEE International Conference on Fuzzy Systems* (Melbourne, Australia, 2001), pp. 1291–1294.
30. F. Zhang, Z. M. Ma, J. Cheng, X. Meng, Fuzzy Semantic Web Ontology Learning from Fuzzy UML Model, in *Proc. of the 18th ACM Conference on Information and Knowledge Management* (Hong Kong, China, 2009), pp. 1007–1016.
31. F. Zhang, Z. M. Ma, G. Fan and X. Wang, Automatic Fuzzy Semantic Web Ontology Learning from Fuzzy Object-Oriented Database Model, in *Proc. International Conference on Database and Expert Systems Applications* (Bilbao, Spain, 2010), pp. 16–30.
32. A. Haroonabadi and M. Teshnehlab, Behavior Modeling in Uncertain Information Systems by Fuzzy-UML, *International Journal of Soft Computing* **4**(1) (2009) 32–38.
33. Z. M. Ma and L. Yan, Fuzzy XML data modeling with the UML and relational data models, *Data & Knowledge Engineering* **63**(3) (2007) 970–994.
34. Z. M. Ma, F. Zhang and L. Yan, Fuzzy information modeling in UML class diagram and relational database models, *Applied Soft Computing* **11**(6) (2011) 4236–4245.
35. H. Motameni, A. Movaghar, I. Daneshfar, H. Nemat Zadeh and J. Bakhshi, Mapping to Convert Activity Diagram in Fuzzy UML to Fuzzy Petri Net, *World Applied Sciences Journal* **3**(3) (2008) 514–521.

36. L A. Zadeh, Fuzzy sets, *Information and Control* **8**(3) (1965) 338–353.
37. L A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems* **1**(1) (1978) 3–28.
38. Z. M. Ma, F. Zhang, L. Yan and J. Cheng, Representing and reasoning on fuzzy UML models: A description logic approach, *Expert Systems with Applications* **38**(3) (2011) 2536–2549.
39. M. A. Sicilia, E. Garcia, J. A. Gutierrez, Integrating fuzziness in object oriented modeling language: towards a fuzzy-UML, in *Proc. of International Conference on Fuzzy Sets Theory and its Applications* (Liptovský Ján, The Slovak Republic, 2002), pp. 66–67.
40. *OWL: Ontology Web Language*, http://www.w3.org/2004/OWL/
41. G. Stoilos, G. Stamou, V. Tzouvaras, J. Z. Pan and I. Horrocks, Fuzzy OWL: Uncertainty and the Semantic Web, in *Proc. International Workshop of OWL: Experiences and Directions* (Galway, Ireland, 2005), pp. 80–89.
42. G. Stoilos, G. Stamou and J. Z. Pan, Fuzzy extensions of OWL: Logical properties and reduction to Fuzzy Description Logics, *Int. J. of Approximate Reasoning* **51** (2010) 656–679.
43. U. Straccia, Towards a fuzzy description logic for the semantic Web, in *Proc. of the 2nd European Semantic Web Conf.* (Heraklion, Crete, 2005), pp. 167–181.
44. C. S. Lee, Z. W. Jian and L. K. Huang, A fuzzy ontology and its application to news summarization, *IEEE Transactions on Systems, Man and Cybernetics Part B* **35**(5) (2005) 859–880.
45. Z. M. Ma, F. Zhang, L. Yan, Y. Lv, Formal semantics-preserving translation from fuzzy ER model to fuzzy OWL DL ontology, *Web Intelligence and Agent Systems: An International Journal* **8**(4) (2010) 397–412.
46. F. Bobillo, Managing Vagueness in Ontologies, *PhD Dissertation*, University of Granada, Spain, 2008.
47. B. Oliboni and G. Pozzani, An XML Schema for Managing Fuzzy Documents, *Technical report*, Department of Computer Science, University of Verona, Italy, May 2008.
48. F. Bobillo, M. Delgado and J. Gomez-Romero, DeLorean: A Reasoner for Fuzzy OWL 2, *Expert Systems with Applications* **39**(1) (2012) 258–272.
49. O. Corcho, M. Fernández-López and A. Gómez-Pérez, Methodologies, tools and languages for building ontologies. Where is their meeting point?, *Data & Knowledge Engineering* **46** (2003), 41–64.
50. L. Zhou, Ontology learning: state of the art and open issues, *Information Technology and Management* **8**(3) (2007) 241–252.
51. D. Berardi, D. Calvanese and G. De Giacomo, Reasoning on UML class diagrams, *Artificial Intelligence* **168**(1/2) (2005) 70–118.
52. A. Calı`, D. Calvanese, G. De Giacomo and M. Lenzerini, Reasoning on UML Class Diagrams in Description Logics, in *Proc. of IJCAR Workshop on Precise Modelling and Deduction for Object-Oriented Software Development* (Siena, Italy, 2001), pp. 15–28.
53. S. R. Upadhyaya and P. S. Kumar, ERONTO: A Tool for Extracting Ontologies from Extended E/R Diagrams, in *Proc. 20th Annual ACM Symposium on Applied Computing* (Santa Fe, New Mexico, 2005), pp. 666–670.
54. I. Astrova, Reverse engineering of relational database to ontologies, in *Proc. of the ESWC 2004* (Heraklion, Greece, 2004), pp. 327–341.
55. L. Lubyte and S. Tessaris, Automatic extraction of ontologies wrapping relational data sources, in *Proc. of 20th International Conference on Database and Expert Systems Applications* (Linz, Austria, 2009), pp. 128–142.
56. F. Zhang, Z. M. Ma, X. Wang and Y. Wang, Formal approach and automated tool for constructing ontology from object-oriented database model, in *Proc. of the 19th ACM Conference on Information and Knowledge Management* (Toronto, Ontario, Canada, 2010), pp. 1329–1332.
57. Y. An, A. Borgida and J. Mylopoulos, Constructing Complex Semantic Mappings between XML Data and Ontologies, in *Proc. of 4th International Semantic Web Conference* (Galway, Ireland, 2005), pp. 6–20.
58. T. Rodrigues, P. Rosa and J. Cardoso, Mapping XML to existing OWL Ontologies, in *Proc. Int. Conf. on WWW/Internet 2006* (Murcia, Spain, 2006), pp. 72–77.
59. A. Haroonabadi and M. Teshnehlab, Applying Fuzzy-UML for Uncertain Systems Modeling, in *First Joint Congress on Fuzzy and Intelligent Systems* (Mashhad, Iran, 2007), pp. 32–38.
60. Z. M. Ma and L. Yan, A Literature Overview of Fuzzy Conceptual Data Modeling, *Journal of Information Science and Engineering* **26**(2) (2010) 427–441.
61. H. Gu, H. Lv, J. Gao and J. Shi, Towards a General Fuzzy Ontology and Its Construction, in *Proc. of ISKE a part of series: Advances in intelligent system research* (Chengdu, China, 2007), pp. 409–414 .
62. M. Abulaish and L. Dey, A fuzzy ontology generation framework for handling uncertainties and non-uniformity in domain knowledge description, in *Proc. of the International Conference on Computing: Theory and Applications* (Kolkata, India, 2007), pp. 287–293.
63. W. Chen, Q. Yang, L. Zhu and B. Wen, Research on Automatic Fuzzy Ontology Generation from Fuzzy Context, in *Proc. of the 2009 Second International Conference on Intelligent Computation Technology and Automation* (Zhangjiajie, China, 2009), pp. 764–767.
64. I. J. Blanco, M. A. Vila and C. Martinez-Cruz, The use of ontologies for representing database schemas of fuzzy information, *J. Intelligent Syst* **23**(4) (2008) 419–445.
65. J. C. Cubero, N. Marín, J. M. Medina, O. Pons and M. A. Vila, Fuzzy object management in an object-relational framework, in *Proc. 10th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems* (Perugia, Italy, 2004), pp. 1767–1774.

Copyright: the authors