Filomat 32:5 (2018), 1687–1696
https://doi.org/10.2298/FIL1805687L

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Reverse Engineering of Fuzzy OWL 2 Ontologies to Object-oriented Database Models

Weijun Li[a,c], Li Yan[b], Xu Chen[c], Fu Zhang[a], Jiulei Jiang[c]

[a]*School of Computer Science & Engineering, Northeastern University, Shenyang 110169, China*
[b]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 2111069, China*
[c]*School of Computer Science and Engineering, North Minzu University , Yinchuan 750021, China*

Abstract. With the increase in demand of complex information modeling, object-oriented database models are put on the agenda. But information imperfection is inherent in the real-world applications. To deal with these complex imprecise and uncertain information, fuzzy object-oriented database (*FOOD*) and fuzzy OWL 2 ontology modeling are recently received more attention. But construction fuzzy ontology is a time-consuming and laborious task from scratch, reusing existing fuzzy ontology is an effective method of ontology construction. For the sake of reusing fuzzy OWL 2 ontologies, this paper proposes a reverse engineering approach for transforming fuzzy OWL 2 Ontologies into *FOOD* models. And reverse engineering can shorten development cycles of ontology and various database models. On this basis, we propose formal definition of *FOOD* models and fuzzy OWL 2 ontologies. Furthermore, we give transformation rules and explain how to transform fuzzy OWL 2 ontologies into fuzzy *FOOD* models with an example in detail. The correctness of this transformation approach is proved. The advantage of reengineering fuzzy ontologies into *FOOD* models is the reusability of domain knowledge on the Web.

1. Introduction

Nowadays, ontologies have been widely used in many fields, e.g. computer science, e-commerce, intelligent retrieval, data mining, and so on. But, constructing ontology has become the focus of recent research. In view of this need, knowledge bases, XML and databases become the data sources for constructing ontologies (see [10] for surveys). It is considered that construction of ontologies is a laborious and time-consuming task [13]. Reusing previous ontologies are considered as an effective approach of constructing ontologies. Ontology reusing is defined as the process which constructs a new ontology by making full use of used ontologies, ontological components and ontological knowledge.

A crucial issue in ontology reusing is to identify their components and interrelationships of the existing ontologies, in which is a reverse engineering process of the ontology [9], [21]. The reverse engineering [1], [2], [7], [15], which is referred to reengineering [12] also, is used to denote a development process of researching an existing system and reconstructing it into a new form. The reverse engineering is to

2010 *Mathematics Subject Classification*. 03B52.
Keywords. Object-oriented Database (FOOD) Model; Fuzzy OWL 2 Ontology; Reverse Engineering; Transforming, Mapping.
Received: 30 September 2017; Accepted: 17 November 2017
Communicated by Hari M. Srivastava
Research supported by the National Natural Science Foundation of China (61370075,61672139 and 61650206) and Natural Science Foundation of Ningxia (NZ17103).
Email addresses: liweijun@nun.edu.cn (Weijun Li), yanli@nuaa.edu.cn (Li Yan), chenxu@nun.edu.cn (Xu Chen)

discover the underlying features of a system, to identify or recover one or more of the system requirements, specifications, design and implementation that can aid in understanding and modifying the systems. Meanwhile, for the purpose of modeling and manipulation of complex object and relation, object-oriented databases have increasingly attracted considerable attention.

Due to the information and data in the real world is uncertainty and imperfect, *FOOD* models are put forward. In order to process imperfect and complex objects, Ma et al. [16] extend an object-oriented database model based on the semantic measure of fuzzy data and the possibility distribution. Yan et al. [26] propose using fuzzy measure of probabilistic theory in modeling object-oriented databases. Shukla et al. [23] present an overview of the different approaches to fuzzy techniques integration in object-oriented databases. Ma et al. [17] provide an overview of the current main approaches of fuzzy extension of object-oriented databases.

Meanwhile, to express and reason on fuzzy knowledge, fuzzy ontology definitions [5], [24], [25], [31] have been proposed by incorporating fuzzy description logic and fuzzy set theory [27], [28]. W3C Web Ontology Working Group recommends the Web Ontology language (OWL) 2 [18], [19], [20] to be the standard for ontologies description in the Semantic Web [14]. Bobillo et al.[5] propose a method of using OWL 2 annotation properties to represent fuzzy ontologies. Our work mainly investigates the fuzzy OWL 2 ontology reverse engineering. Fuzzy OWL 2 ontology is an extension of the classical OWL 2 ontology based on Zadehs fuzzy set theory [27], [28]. The logical foundation of fuzzy OWL 2 is the fuzzy DL called *f-SROIQ*(D) [6].

With the extensive application of ontology, numerous fuzzy ontologies were established. How to reuse the existing fuzzy ontologies has been considered as an effective way of knowledge reusing. Benslimane et al. [2] propose a reverse engineering approach of extracting domain ontology schema to construct conceptual data model so that ontologies can be reused at a conceptual level. In particular, fuzzy object-oriented database (*FOOD*) models have been considered as important tools of describing and storing fuzzy information in real-world applications [17], [23]. Moreover, several work has established the relationships between fuzzy ontologies and *FOOD* models. Zhang et al. [30] propose description logic approach of indicating and inference fuzzy object-oriented database models. Then, they [31] propose a method of constructing fuzzy ontologies using *FOOD* models, and establish reasoning mechanism on *FOOD* models.

However, above-mentioned existing works cannot achieve to transform fuzzy OWL 2 ontologies into object-oriented database models utilizing reverse engineering. Based on Zadehs fuzzy set theory, we develop *FOOD* models that addresses all types of fuzzy data and complex objects. Then, we develop a methodology of transforming fuzzy OWL 2 ontologies into FOOD models. After that, we prove that the transforming method is correct, and provide a transformed instance to explain the proposed approach.

The remainder of this paper is organized as follows. The *FOOD* models and fuzzy OWL 2 ontologies are introduced in *Section 2*. In *Section 3*, an approach for reengineering fuzzy OWL 2 ontologies into *FOOD* models is proposed. *Section 4* concludes the paper.

2. Preliminaries

Zadeh [27] originally introduced the concept of fuzzy sets. Let U a universe of discourse and F be a fuzzy set in U. The definition of F requires a membership function $\mu_F : U \to [0,1]$, where $\mu_F(u)$, for each $u \in U$, the membership degree μ belonging to the fuzzy set F. For the case where U is a discrete domain, the fuzzy set F is expressed as

$F = u_1/\mu_F(u_1), u_2/\mu_F(u_2), ..., u_n/\mu_F(u_n)$

Here μ_F is used to represent the membership function of the fuzzy set F,$\mu_F(u)$ is used to represent the membership degree u belonging to the fuzzy set F. In fact $\mu_F(u)$ can be interpreted as a possibility measure that a variable X has value u, where X takes values in U, a fuzzy value is described by a possibility distribution π_X [28]:

$\pi_X = \pi_X(u_1)/u_1, \pi_X(u_2)/u_2, ..., \pi_X(u_n)/u_n$

Here, for any $u_1 \in U$, $\pi_X(u_i)$ indicates the possibility that u_i is true. Let π_X and F denote a fuzzy possibility distribution and a fuzzy set, respectively, then π_X and F can be equal, scilicet $\pi_X = F$ [22].

2.1. FOOD *model*

The *FOOD* model [3], [4], [16], [17], [29], [30], [31] is a fuzzy extension of the traditional object-oriented database model. The following introduces the basic notions of *FOOD* model, including object, class, attribute, method and hierarchy of classes.

• *Fuzzy object:* An object is a real world entity or an abstract concept. An object has features that can be the attribute of the object itself, or can be the relationship between the object and another object or multiple objects. If an object at least has one fuzzy attribute the object is fuzzy.

• *Fuzzy class:* Objects with the same attributes make up classes. Comparing with the traditional classical class, the boundary of a fuzzy class is not accurate. The class boundary imprecision is because of the inaccurate attribute value.

• *Fuzzy inheritance hierarchies:* In a *FOOD* model, the class generated by a fuzzy class must be fuzzy. In fuzzy inheritance relationship, one is called superclass, the other is called a subclass. Further, several inheritances of subclasses can be combined to form a class hierarchy.

• *Fuzzy attribute:* The range of attribute values is called domain of the attribute. If the domain of an attribute is fuzzy, the attribute is fuzzy.

• *Method:* Method is a series of operations on the object state.

We review the existing definitions of (fuzzy) object-oriented database models [29], [30], [31], and give a formal definition of *FOOD* models. In this definition there is the structural and dynamic aspects of *FOOD* models, which includes the major notions of fuzzy objects, fuzzy classes, fuzzy inheritance hierarchies, and other constraints (e.g., the disjoint and complete constraints on class hierarchies and the cardinality constraints on associations).

Definition 1 (Formal definition of *FOOD* model): A *FOOD* model is a tuple $FOOD_{FS} = (FO_{FS}, FD_{FS}, FC_{FS}, FA_{FS}, FP_{FS})$ consisting of a set of object identities, attribute domains, classes, attributes, and class declarations [30],[31], where:

1) FO_{FS} is a set of objects FO; each object has a unique object identity;

2) FD_{FS} is a set of domains FD, which includes crisp and fuzzy domains.

3) FC_{FS} is a set of classes FC;

4) FA_{FS} is a set of attributes FA; each attribute FA is associated with a domain FD, and using the fuzzy keyword FUZZY in front of an attribute denotes that this attribute is fuzzy;

5) FP_{FS} is a set of class declarations. For each class $FC \in FC_{FS}, FP_{FS}$ contains a declaration:

Class FC <u>is-a</u> FC_{sup}/β <u>type-is</u> FT,

FT denotes a schema type expression built according to the following syntax:

$FT \rightarrow \{FO_1/\mu_1, FO_2/\mu_2, ..., FO_n/\mu_n\}$ <u>End</u> ; <u>*Union*</u> $FC_1, ..., FC_k(disjoint, complete)$ <u>End</u> ; <u>*Record*</u> $FA_1 : FT_1, ..., FA_k : FT_k, \mu : Real, f(P_1, ..., P_m) : R$ <u>End</u> .

where FT_i is one of the following cases $(wherei \in \{1...k\}) : FT_i \rightarrow FD_i$; $\underline{Set-of}\ FC_i/\eta_i[(m_1, n_1), (m_2, n_2)]$.

$FOOD_{FO}$ is a set of object declarations to represent values of attributes of objects. For each object $FO \in FO_{FS}$, FO <u>belong-to</u> FC/μ , <u>has-value</u> $[FA_1 : FD_1, FA_2 : FD_2, ...]$ <u>End</u>.

Where:

• The <u>type-is</u> part specifies the structure of a class FC by a type expression FT;

• The <u>is-a</u> part, which is optional, denotes inheritance relationship between fuzzy classes with a membership degree $\beta \in [0, 1]$;

• The expression $\{FO_1/\mu_1, FO_2/\mu_2, ..., FO_n/\mu_n\}$ denotes that FC is an *extensional class* which has a list of object instances $\{FO_1/\mu_1, FO_2/\mu_2, ..., FO_n/\mu_n\}$, and each object FO_i has a membership degree of $\mu \in [0, 1]$ relative to the class FC;

• The <u>*Union*</u>...<u>*End*</u> part denotes a class hierarchy;

• The <u>*Record*</u>...<u>*End*</u> part denotes that a fuzzy class FC is defined by a set of attributes and their admissible values, this class is called an intensional class; an additional attribute $\mu \in [0, 1]$ is used to represent the membership degree of an object belonging to the class FC; $f(P_1, ..., P_m) : R$ represents a method, where f is the name of the method, $P_1, ..., P_m$ are types of m parameters, and R is the type of the result;

• The <u>Set-of</u> part (i.e., Class FC <u>type-is</u> Record FA_i: <u>Set-of</u> $FC_i/\eta_i[(m_1, n_1), (m_2, n_2)]$ <u>End</u>) denotes an association relationship between classes FC and FC_i by an attribute FA_i; $eta_i \in [0, 1]$ the association occurs in classes FC and FC_i with a membership degree of η_i; /, $[(m_1, n_1), (m_2, n_2)]$ indicates that the association involves at least m_i and at most n_i objects of a class;

• The belong-to part denotes that an object FO belongs to a fuzzy class FC with a membership degree of $\mu \in [0, 1]$;

• The has-value part denotes the attribute values of an object FO, and the attributes belong to the fuzzy class FC.

Then, we take advantage of fuzzy database states (e.g., sets of object instance) to describe semantics of FOOD models [30]. The fuzzy database state (e.g., object information) ties in with the schema structure of FOOD model (e.g., schema information).

Definition 2 (Semantics of *FOOD* models): The semantics of a *FOOD* model can be given by a fuzzy database state FJ, which is defined by a fuzzy interpretation $FJ_{FD} = (FV_{FJ}, \pi^{FJ}, \rho^{FJ}, \bullet^{FJ})$[30], [31]:

1) A set $FV_{FJ} = FD^{FJ} \cup FO^{FJ} \cup FR^{FJ} \cup FS^{FJ}$ of fuzzy values is inductively defined as follows:

• $FD^{FJ} = \bigcup_{i=1}^{n} FD_i^{FJ}$, where FD_i is a crisp or fuzzy domain as mentioned in the previous sections;

• $FO^{FJ} = \{FO_1/\mu_1, ..., FO_n/\mu_n\}$, where FO_i is an object associated with a membership degree μ_i;

• FR^{FJ} is a set of record values. A record value is denoted by $[FA_1 : FV_1, ..., FA_k : FV_k]$, where FA_i is an attribute, $FV_i \in FV_{FJ}$, $i \in \{1, ..., k\}$;

• FS^{FJ} is a set of set-values. A set-value is denoted by $\{FV_1, ..., FV_k\}$, where $FV_i \in FV_{FJ}$, $i \in \{1, ..., k\}$.

2) A function π^{FJ} maps a class to a set of its objects.

3) A function ρ^{FJ} maps an object to values of its attributes.

4) A function $\bullet^{FJ}$ maps each type expression FT into a set FT^{FJ} such that:

• If FT is a class FC, then $FT^{FJ} = FC^{FJ} = \pi^{FJ}(FC)$;

• If FT is a record type Record ... End (resp. a set type Set-of), then FTFJ is a set of record values FRFJ (resp. a set of set-values FS^{FJ});

• If FT is a union type Union $FC_1, ..., FC_q$ (disjoint, complete) End, then it follows $FT^{FJ} = FC_1^{FJ} \cup ... \cup FC_q^{FJ}$ and $FC_i^{FJ} \cap FC_j^{FJ} = \emptyset$, where $i, j \in \{1, ..., q\}$, and $i \neq j$.

If a fuzzy database state satisfies all of the constraints of a *FOOD* model, the fuzzy database state is considered acceptable. The fuzziness may occur at three different levels in a *FOOD* model [8],[11],[16], i.e., the attribute level, the object/class level, and the subclass/superclass level.

2.2. Fuzzy OWL 2 Ontology

To define fuzzy OWL 2 ontology, it is necessary to introduce fuzzy OWL language [29], which is based on the Zadehs fuzzy set theory [27]. The semantics for fuzzy OWL 2 are equivalent in the expressive description logic *f-SROIQ(D)* [6]. After summarizing the fuzzy OWL in [29], [30], we provide *Table* 1 to show the fuzzy OWL 2 abstract syntax, the corresponding description logics syntax, and the semantics.

In *Table* 1, *FC* indicates a fuzzy class; *FCE* indicates a fuzzy class expression; *FDT* indicates a fuzzy datatype; *FDR* indicates a fuzzy data range; *FDP* indicates a fuzzy data property; *FDPE* indicates a fuzzy data property expression; *FOP* indicates a fuzzy ObjectProperty; *FOPE* indicates a fuzzy ObjectProperty expression; α indicates an individual (named or anonymous); *lt* indicates a literal; *FA* indicates a constraining facet; $\sharp S$ indicates the cardinality set S, and $\bowtie \in \{\geq, >, \leq, <\}$.

An ontology described by fuzzy OWL 2 language (e.g. [29], [31]) is called fuzzy OWL 2 ontology. To represent fuzzy OWL 2 ontologies, we present a formal definition of fuzzy OWL 2 ontologies in the following.

Definition 3 (Fuzzy OWL 2 ontology): A fuzzy OWL ontology is formally represented as 8-tuple $O_F = (FOP_O, FDP_O, FDT_O, FC_O, FP_C, FI_O, Flt_O, FO_{Axiom})$, consisting of the following elements [31]:

1) FOP_O is a set of object properties identifiers, containing at least the object properties *owl:topObjectProperty* and *owl:bottomObjectProperty*. Each object properties links individuals to individuals, and each property may have its characters and its restrictions;

2) FDP_O is a set of datatype properties as defined in the OWL 2, the data properties link individuals to data values, containing at least the data properties *owl:topDataProperty* and *owl:bottomDataProperty* ;

3) FDT_O is a set of all datatype, containing the datatype *rdfs:Literal* and possibly other datatypes ;

Table 1: Fuzzy Owl Abstract Syntax, Description Logic (DL) Syntax and Interpretation.

Fuzzy OWL abstract syntax	Fuzzy DL syntax	Interpretation
Fuzzy Class description		
$Class(FC)$	FC	$FC^{FI} \subseteq \triangle^{FI}$
$owl : Thing$	$\top$	$(owl : Thing)^{FC} = \triangle^{FI}$
$owl : Nothing$	$\bot$	$(owl : Nothing)^{FC} = \emptyset$
$ObjectIntersectionOf(FCE_1 ... FCE_n)$	$FCE_1 \sqcap ... \sqcap FCE_n$	$(FCE_1)^{FC} \cap ... \cap (FCE_n 0^{FC}$
$ObjectUnionOf(FCE_1 ... FCE_n)$	$FCE_1 \sqcup ... \sqcup FCE_n$	$(FCE_1)^{FC} \cup ... \cup (FCE_n)^{FC}$
$ObjectComplementOf(FCE)$	$\neg\; FCE$	$\triangle^{FI} \setminus (FCE)^{FC}$
$ObjectOneOf(a_1 ... a_n)$	$\{a_1\} \sqcup ... \sqcup \{a_n\}$	$\{(a_1)^{FI}, ..., (a_n)^{FI}\}$
$ObjectSomeValuesFrom(FOPE\ FCE)$	$\exists FOPE \cdot FCE$	$\{x \mid \exists y : (x, y) \in (FOPE)^{FOP}$ and $y \in (FCE)^{FC}\}$
$ObjectAllValuesFrom(FOPE\ FCE)$	$\forall \neq FOPE \cdot FCE$	$\{x \mid \forall y : (x, y) \in (FOPE)^{FOP}$ implies $y \in (FCE)^{FC}\}$
$ObjectHasValue(FOPE\ a)$	$\exists FOPE \cdot \{a\}$	$\{x \mid (x, (a)^I) \in (FOPE)^{FOP}\}$
$ObjectHasSelf(FOPE)$	$FOPE \equiv (FOPE)^-$	$\{x \mid (x, x) \in (FOPE)^{FOP}\}$
$ObjectMinCardinality(n\ FOPE)$	$\geq nFOPE$	$\{x \mid \sharp\{y \mid (x, y) \in (FOPE)^{FOP}\} \geq n\}$
$ObjectMaxCardinality(n\ FOPE)$	$\leq nFOPE$	$\{x \mid \sharp\{y \mid (x, y) \in (FOPE)^{FOP}\} \leq n\}$
$ObjectExactCardinality(n\ FOPE)$	$\equiv nFOPE$	$\{x \mid \sharp\{y \mid (x, y) \in (FOPE)^{FOP}$ and $y \in (FOPE)^{FOP}\} = n\}$
$DataSomeValuesFrom(FDPE_1 ... FDPE_n\ FDR)$	$\exists FDR \cdot \{FDPE_1 ... FDPE_n\}$	$\{x \mid \exists y_1, ..., y_n : (x, y_k) \in (FDPE_k)^{FDP}$ for each $1 \leq k \leq n$ and $(y_1, ..., y_n) \in (FDR)^{FDT}\}$
$DataAllValuesFrom(FDPE_1 ... FDPE_n\ FDR)$	$\forall FDR \cdot \{FDPE_1 ... FDPE_n\}$	$\{x \mid \forall y_1, ..., y_n : (x, y_k) \in (FDPE_k)^{FDP}$ for each $1 \leq k \leq n$ imply $(y_1, ..., y_n) \in (FDR)^{FDT}\}$
$DataHasValue(FDPE\ lt)$	$\exists FDPE \cdot \{lt\}$	$\{x \mid (x, (lt)^{LT}) \in (FDPE)^{FDP}\}$
$DataMinCardinality(n\ FDPE)$	$\geq nFDPE$	$\{x \mid \sharp\{y \mid (x, y) \in (FDPE)^{FDP}\} \geq n\}$
$DataMaxCardinality(n\ FDPE)$	$\leq nFDPE$	$\{x \mid \sharp\{y \mid (x, y) \in (FDPE)^{FDP}\} \leq n\}$
$DataExactCardinality(n\ FDPE)$	$\equiv nFDPE$	$\{x \mid \sharp\{y \mid (x, y) \in (FDPE)^{FDP}\} = n\}$
$DataMinCardinality(n\ FDPE\ FDR)$	$\geq nFDR \cdot FDPE$	$\{x \mid \sharp\{y \mid (x, y) \in (FDPE)^{FDP}$ and $y \in (FDR)^{FDT}\} \geq n\}$
$DataMaxCardinality(n\ FDPE\ FDR)$	$\leq nFDR \cdot FDPE$	$\{x \mid \sharp\{y \mid (x, y) \in (FDPE)^{FDP}$ and $y \in (FDR)^{FDT}\} \leq n\}$
$DataExactCardinality(n\ FDPE\ FDR)$	$\equiv nFDR \cdot FDPE$	$\{x \mid \sharp\{y \mid (x, y) \in (FDPE)^{FDP}$ and $y \in (FDR)^{FDT}\} = n\}$
Fuzzy Data Ranges		
$DataIntersectionOf(FDR_1 ... FDR_n)$	$FDR_1 \sqcap ... \sqcap FDR_n$	$(FDR_1)^{FDT} \cap ... \cap FDR_n)^{FDT}$
$DataUnionOf(FDR_1 ... FDR_n)$	$FDR_1 \sqcup ... \sqcup FDR_n$	$(FDR_1)^{FDT} \cup ... \cup FDR_n)^{FDT}$
$DataComplementOf(FDR)$	$\neg\; FDR$	$(\triangle_n) \setminus (FDR)^{FDT}$ where n is the arity of FDR
$DataOneOf(lt_1 ... lt_n)$	$\{lt_1\} \sqcup ... \sqcup \{lt_n\}$	$\{(lt_1)^{LT}, ... ,(lt_n)^{LT}\}$
$DatatypeRestriction(FDT\ F_1\ lt_1 ... F_n\ lt_n)$		$(FDR)^{FDT} \cap (F_1, lt_1)^{FA} \cap ... \cap (F_n, lt_n)^{FA}$
Fuzzy Class axioms		
$Class(FC\ partial\ FCE_1 ... FCE_n)$	$FC \sqsubseteq FCE_1 \sqcap ... \sqcap FCE_n$	$(FC)^{FC} \subseteq (FCE_1)^{FC} \cap ... \cap (FCE_n)^{FC}$
$SubClassOf(FCE_1\ FCE_2)$	$FCE_1 \sqsubseteq FCE_2$	$(FCE_1)^{FC} \subseteq (FCE_2)^{FC}$
$EquivalentClasses(FCE_1 ... FCE_n)$	$FCE_1 \equiv ... \equiv FCE_n$	$(FCE_j)^{FC} = (FCE_k)^{FC}$ for each $1 \leq j < k \leq n$
$DisjointClasses(FCE_1 ... FCE_n)$	$FCE_j \neq FCE_k 1 \leq j < k \leq n$	$(FCE_j)^{FC} \cap (FCE_k)^{FC} = \emptyset$ for each $1 \leq j < k \leq n$
$DisjointUnion(FC\ FCE_1 ... FCE_n)$	$FC \equiv (FCE_1 \sqcup ... \sqcup FCE_n), FCE_j \neq FCE_k, 1 \leq j < k \leq n$	$(FC)^{FC} = (FCE_1)^{FC} \cup ... \cup (FCE_n)^{FC}$ and $(FCE_j)^{FC} \cap (FCE_k)^{FC} = \emptyset$ for each $1 \leq j < k \leq n$
Fuzzy Object property axioms		
$SubObjectPropertyOf(FOPE_1\ FOPE_2)$	$FOPE_1 \sqsubseteq FOPE_2$	$(FOPE_1)^{FOP} \subseteq (FOPE_2)^{FOP}$
$EquivalentObjectProperties(FOPE_1 ... FOPE_n)$	$FOPE_1 \equiv ... \equiv FOPE_n$	$(FOPE_j)^{FOP} = (FOPE_k)^{FOP}$ for each $1 \leq j < k \leq n$
$DisjointObjectProperties(FOPE_1 ... FOPE_n)$	$FOPE_j \neq FOPE_k, 1 \leq j < k \leq n$	$(FOPE_j)^{FOP} \cap (FOPE_k)^{FOP} = \emptyset$ for each $1 \leq j < k \leq n$
$ObjectPropertyDomain(FOPE\ FCE)$	$\exists FOPE \cdot FCE$	$\forall x, y : (x, y) \in (FOPE)^{FOP}$ implies $x \in (FCE)^{FC}$
$ObjectPropertyRange(FOPE\ FCE)$	$\top \sqsubseteq \forall FOPE \cdot FCE$	$\forall x, y : (x, y) \in (FOPE)^{FOP}$ implies $x \in (FCE)^{FC}$
$InverseObjectProperties(FOPE_1\ FOPE_2)$	$FOPE_1 \equiv (FOPE_2)^-$	$(FOPE_1)^{FOP} = \{(x, y) \mid (y, x) \in (FOPE_2)^{FOP}\}$
$FunctionalObjectProperty(FOPE)$	$\top \sqsubseteq \leq 1FOPE$	$\forall x, y_1, y_2 : (x, y_1) \in (FOPE)^{FOP}$ and $(x, y_2) \in (FOPE)^{FOP}$ imply $y_1 = y_2$
$InverseFunctionalObjectProperty(FOPE)$	$\top \sqsubseteq \leq 1(FOPE)^-$	$\forall x_1, x_2, y : (x_1, y) \in (FOPE)^{FOP}$ and $(x_2, y) \in (FOPE)^{FOP}$ imply $x_1 = x_2$
$ReflexiveObjectProperty(FOPE)$	$FOPE \equiv (FOPE)^-$	$\forall x : x \in \triangle^{FI}$ implies $(x, x) \in (FOPE)^{FOP}$
$IrreflexiveObjectProperty(FOPE)$	$FOPE \neq (FOPE)^-$	$\forall x : x \in \triangle^{FI}$ implies $(x, x) \notin (FOPE)^{FOP}$
$SymmetricObjectProperty(FOPE)$	$FOPE \equiv (FOPE)^{FT}$	$\forall x, y : (x, y) \in (FOPE)^{FOP}$ implies $(y, x) \in (FOPE)^{FOP}$
$AsymmetricObjectProperty(FOPE)$	$FOPE \neq (FOPE)^{FT}$	$\forall x, y : (x, y) \in (FOPE)^{FOP}$ implies $(y, x) \notin (FOPE)^{FOP}$
$TransitiveObjectProperty(FOPE)$	$(FOPE)^2 \sqsubseteq FOPE$	$\forall x, y, z : (x, y) \in (FOPE)^{FOP}$ and $(y, z) \in (FOPE)^{FOP}$ implies $(x, z) \in (FOPE)^{FOP}$
Fuzzy Data property Axioms		
$SubDataPropertyOf(FDPE_1\ FDPE_2)$	$FDPE_1 \sqsubseteq FDPE_2$	$(FDPE_1)^{FDP} \subseteq (FDPE_2)^{FDP}$
$EquivalentDataProperties(FDPE_1 ... FDPE_n)$	$FDPE_1 \equiv ... \equiv FDPE_n$	$(FDPE_i)^{FDP} = (FDPE_j)^{FDP}$ for each $1 \leq j < k \leq n$
$DisjointDataProperties(FDPE_1 ... FDPE_n)$	$FDPE_j \neq FDPE_k, 1 \leq j < k \leq n$	$(FDPE_i)^{FDP} \cap (FDPE_j)^{FDP} = \emptyset$, for each, $1 \leq j < k \leq n$
$DataPropertyDomain(FDPE\ FCE)$	$\exists FDPE.\top \sqsubseteq FCE$	$\forall x, y : (x, y) \in (FDPE)^{FDP}$ implies $x \in (FCE)^{FDP}$
$DataPropertyRange(FDPE\ FDR)$	$\top \sqsubseteq \forall FDPE.FCE$	$\forall x, y : (x, y) \in (FDPE)^{FDP}$ implies $y \in (FDR)^{FDP}$
$FunctionalDataProperty(FDPE)$	$\top \sqsubseteq \leq 1FDPE$	$\forall x, y_1, y_2 : (x, y_1) \in (FDPE)^{FDP}$ and $(x, y_2) \in (FDPE)^{FDP}$ implies $y_1 = y_2$
Fuzzy Assertions Axioms		
$SameIndividual(a_1 ... a_n)$	$\{a_j\} \equiv ... \equiv \{a_k\}$	$(a_j)^{FI} = (a_k)^{FI}\ 1 \leq j < k \leq n$
$DifferentIndividuals(a_1 ... a_n)$	$\{a_j\} \neq \{a_k\} 1 \leq j < k \leq n$	$(a_j)^{FI} \neq (a_k)^{FI}\ 1 \leq j < k \leq n$
$ClassAssertion(FCE\ a)$	$\exists FCE \cdot \{a\}$	$(a)^{FI} \in (FCE)^{FC}$
$ObjectPropertyAssertion(FOPE\ a_1\ a_2)$	$\exists FOPE \cdot \{a_1, a_2\}$	$((a_1)^{FI}, (a_2)^{FI}) \in (FOPE)^{FOP}$
$NegativeObjectPropertyAssertion(FOPE\ a_1\ a_2)$	$\nexists FOPE \cdot \{a_1, a_2\}$	$((a_1)^{FI}, (a_2)^{FI}) \notin (FOPE)^{FOP}$
$DataPropertyAssertion(FOPE\ a\ lt)$	$\exists FOPE \cdot \{a, lt\}$	$(a^{FI}, (lt)^{FI}) \in (FOPE)^{FOP}$
$NegativeDataPropertyAssertion(FOPE\ a\ lt)$	$\nexists FOPE \cdot \{a, lt\}$	$(a^{FI}, (lt)^{FI}) \notin (FOPE)^{FOP}$

4) FC_O is a set of fuzzy class defined in the OWL. Each class can be an *AbstractClass* or a *ConcreteClass*;
5) FP_C is a collection of property sets about fuzzy class;
6) FI_O is a collection of fuzzy individuals (named and anonymous);
7) Flt_O is a literal containing each datatype FDT_O and each lexical form of Flt_O;
8) FO_{Axiom} is a set of finite fuzzy OWL 2 axioms.

Based on the above definition, we illustrate a fuzzy OWL 2 ontology of *E-commerce* in *Figure 1*. There are several kinds of fuzziness in the *E-commerce* fuzzy ontology.

A fuzzy OWL 2 ontology structure of customers of E-commerce:
FO_{Axiom} ={ *Class*(Customer complete *UnionOf*(Corporate-Customer, Personal-Customer)); *DisjointClasses*(Corporate-Customer, Personal-Customer) ; *Class*(Order partial restriction(dataReceived allValuesFrom(xsd:String) cardinality(1)) restriction(isPrepaid allValuesFrom(xsd:Boolean) cardinality(1)) restriction(μ allValuesFrom(xsd:Real))); *Class*(Young-Customer partial Customer restriction(CustNo allValuesFrom(xsd:String) cardinality(1)) restriction(CustName allValuesFrom(xsd:String) cardinality(1)) restriction(Fuzzy_age allValuesFrom(xsd:String) cardinality(1)));
SubClassOf(Young-Customer, Customer); *Class*(Corporate-Customer partial Customer restriction(ContactName allValuesFrom(xsd:String) cardinality(1)) restriction(FUZZY_creditRating allValuesFrom(xsd:String) cardinality(1)) restriction(FUZZY-discount allValuesFrom(xsd:Real) cardinality(1)) restriction(allValuesFrom(xsd:single) cardinality(1))); *EnumeratedClass*(Corporate-Customer (cc1, 1.0), (cc2, 0.7), (cc3, 0.8)); *SubClassOf*(Young-Customer Corporate-customer 0.8);
Class(Serving partial restriction (Serveof allValuesFrom (Employee) cardinality(1)) restriction (Serveby allValuesFrom (Corporate-Customer) cardinality(1))); *Class*(Employee partial restriction(invof_Serveof allValuesFrom(Serving))); *Class*(Corporate-Customer partial restriction(invof_Serveby allValuesFrom(Serving))); *Class*(Employee partial restriction (invof_Serveof minCardinality (1) maxCardinality (3))); *Class*(Corporate-Customer partial restriction (invof_Serveby minCardinality (3) maxCardinality ()));
ObjectProperty(Serveof domain(Serving) range(Employee)); *ObjectProperty*(Serveby domain(Serving) range (Corporate-Customer)); *ObjectProperty*(invof_Serveof domain (Employee) range (Serving) inverseOf Serveof); *ObjectProperty*(invof_Serveby domain (Corporate-Customer) range (Taking) inverseOf Serveby);
Individual(yc1 type(Young-Customer) [$\bowtie$ 0.9] value(CustNo, 2013013) value(CustName, Lucy) value(*FUZZY_age*, young) value(μ, 0.9) value(invof_Takof, o') value(invof_Supby, o'') [$\bowtie'$ 0.9]); *Individual*(o1,o2,o3 type(Student));}

Figure 1. A fuzzy OWL 2 ontology structure.

In *Figure 1*, there is fuzzy ontology *E-commerce*. If an element is fuzzy that there are membership degrees after the element. The possibility of an element belonging to its parent element denotes the membership degree associated with the element. The fuzziness in a class is represented by an attribute $\mu \in [0, 1]$. For example, in the *E-commerce* fuzzy ontology, the element *Corporate-Customer* may be fuzzy since we cannot precisely describe the element. In this case, we can found that there is an attribute $\mu \in [0, 1]$ in the axiom of the element *Corporate-Customer*. A fuzzy keyword *FUZZY* is used to represent fuzzy attribute values of elements. For example, the attribute *FUZZY-creditRating* of the element *Corporate-Customer* may be fuzzy. Moreover, there may be other fuzzy elements and attributes in the *E-commerce* fuzzy ontology in the real-word applications.

3. Transforming Fuzzy Owl 2 Ontologies to *FOOD* Models

In this section, based on *FOOD* models and fuzzy OWL 2 ontologies, we propose a formal approach to transform fuzzy OWL 2 ontologies into *FOOD* models. The correctness of the approach is proved in *Theorem 1*. Moreover, an example helps to understand how to transform fuzzy OWL 2 ontologies to *FOOD* models.

3.1. Transforming fuzzy OWL 2 ontology into FOOD model

Giving a fuzzy OWL 2 Ontology model $O_F = (FOP_O, FDP_O, FDT_O, FC_O, FP_C, FI_O, Flt_O, FO_{Axiom})$, we propose some rules to construction $FOOD_{FS}$, starting with the construction of FOOD objects, classes and attributes from the fuzzy OWL 2 ontologies O_F.

Rule 1: Each fuzzy individual ontology identifier FI_O is mapped into an object of *FOOD* model *FO*, i.e., $\varphi(FI_O) \subseteq FO \in FO_{FS}$.

Rule 2: Each fuzzy ontology class identifier FC_O is mapped into a class of *FOOD* model *FC*, i.e., $\varphi(FC_O) \subseteq FC \in FC_{FS}$.

Rule 3: Each fuzzy datatype property identifier FDP_O is mapped into a simple fuzzy attribute of *FOOD* model *FA*, i.e., $\varphi(FDP_O) \subseteq FA \in FA_{FS}$, where the domain of an attribute is a crisp or fuzzy domain.

Rule 4: Each fuzzy class identifier FC_O contains four fuzzy object property identifiers $\varphi(FU_1) \in FA$, $FW_1 = \text{invof}\_\varphi(FU_1) \in FA$, $FW_2 = \text{invof}\_\varphi(FU_2) \in FA$, where FW_1 and FW_2 denote inverse properties of $\varphi(FU_1)$ and $\varphi(FU_2)$, respectively. This fuzzy class is mapped into complex fuzzy attribute of *FOOD* model $\varphi(FC_O) \subseteq FA \in FA_{FS}$, this attribute denotes an association relationship between classes.

Rule 5: Each fuzzy datatype FDT_O is mapped into fuzzy domain of *FOOD* model *FD*, i.e., $\varphi(FDT_O) \subseteq FD \in FD_{FS}$.

Rule 6: Each fuzzy object properties identifier FOP_O is mapped into an object of *FOOD* model *FO*, i.e., $\varphi(FOP_O) \subseteq FO \in FO_{FS}$.

Rule 7: Each fuzzy class properties set FP_C is mapped into an attribute of *FOOD* model *FA*, i.e., $\varphi(FP_C) \subseteq FA \in FA_{FS}$.

Rule 8: Each cardinality (m_i and n_i) of the fuzzy object property maps the object instance of a *FOOD* model class to participate at least m_i times and most n_i times to the association.

Rule 9: Each enumerated class of the fuzzy ontology $EnumeratedClass(FC_O(FC_{O1}, \mu_1)...(FC_{On}, \mu_n))$ is mapped into *FOOD* class declaration *Class* $\varphi(FC_O)$ $\underline{\text{type-is}}$ $\{\varphi(FC_{O1})/\mu_1, ..., \varphi(FC_{On})/\mu_n\}$ *End* , where $\varphi(FC_O)$, $\varphi(FC_{Oi}) \in FC_{FS}$, $\mu_i \in [0, 1]$, $i \in \{1, ..., n\}$.

Rule 10: A taxonomic or hierarchy relationship between fuzzy ontology classes *SubClassOf* $(FC_{Oi}, FC_{Oj}, \beta)$ is mapped into *FOOD* class declaration

Class $\varphi(FC_{Oi})$ $\underline{\text{is-a}}$ $\varphi(FC_{Oj})/\beta$, where $\varphi(FC_{Oi}), \varphi(FC_{Oj}) \in FC_{FS}$, $\beta \in [0, 1]$, $i, j \in \{1, ..., n\}$.

Rule 11: A relationship axiom of fuzzy ontology *Class* (FC_O *complete UnionOf* $(FC_{O1}...FC_{Oq})$), *Disjoint-Classes* (FC_{Oi}, FC_{Oj}) ,$i \neq j, i, j \in \{1, ..., q\}$ is mapped into *FOOD* class declaration

Class $\varphi(FC_O)$ $\underline{\text{type-is}}$ *Union* $\varphi(FC_{O1}), ..., \varphi(FC_{Oq})$ (*disjoint*, *complete*) *End*, where $\varphi(FC_O), \varphi(FC_{Oi}) \in FC_{FS}$.

Rule 12: A fuzzy class description *Class* (FC_O *partial ... restriction*(FDP_{Oi} *allValuesFrom* (FDT_{Oi}) *cardinality* (1))...) is mapped into fuzzy class declaration of *FOOD* model

Class $\varphi(FC_O)$ $\underline{\text{type is}}$

$\underline{\text{Record}}$..., $\varphi(FDP_{Oi}) : \varphi(FDT_{Oi})$, ... $\underline{\text{End}}$, where $\varphi(FC_O) \in FC_{FS}$, $\varphi(FDP_{Oi}) \in FA_{FS}$, $\varphi(FDT_{Oi}) \in FD_{FS}$.

Rule 13: A fuzzy complex class description *Class* (FC_O *partial restriction* (FOP_{O1} *allValuesFrom* (FDT_{O1}) *cardinality* (1))... *restriction* (FOP_{Ok} *allValuesFrom* (FDT_{Ok}) *cardinality* (1))); *DatatypeProperty* (FOP_{Oi} *domain* (FC_O) *range* (FDT_{Oi}) [*Functional*]) is mapped into fuzzy class declaration of *FOOD* model

Class $\varphi(FC_O)$ $\underline{\text{type is}}$ $\underline{\text{Record}}$ $\varphi(FOP_{O1}) : \varphi(FDT_{O1}), ..., \varphi(FOP_{Ok}) : \varphi(FDT_{Ok}), ...$ $\underline{\text{End}}$, where $\varphi(FC_O) \in FC_{FS}, \varphi(FOP_{Oi}) \in FA_{FS}, \varphi(FDT_{Oi}) \in FD_{FS}$.

Rule 14: Fuzzy complex class axiom *Class* (FC_O *partial restriction* (FW_1 allValuesFrom(FC_{O1}) *Cardinality*(1)) *restriction* (FW_2 *allValuesFrom* (FC_{O2}) *Cardinality*(1))); *Class*(FC_{Oi} *partial restriction* (FW_i *allValuesFrom* (FC_O))); *Class*(FC_{Oi} *partial restriction* (FW_i *minCardinality* (m_i))); *Class*(FC_{Oi} *partial restriction* (FW_i *maxCardinality* (n_i))); *ObjectProperty* (FU_i *domain* (FC_O) *range* (FC_{Oi})); *ObjectProperty* (FW_i *domain* (FC_{Oi}) *range* (FC_O) *inverseOf* (FU_i)), where $FU_i, FW_i \in FOP_O$ *and* $FW_1 = \text{invof\_}(FU_i)$, FW_i *denote inverse properties of* FU_i , *respectively*, $i \in \{1, 2\}$ is mapped into fuzzy class declaration of FOOD model

Class $\varphi(FC_{O1})$ $\underline{\text{type-is}}$ $\underline{\text{Record}}$ $\varphi(FC_O)$: $\underline{\text{Set-of}}$ $\varphi(FC_{O2})/\eta[(m_1, n_1), (m_2, n_2)]$,where $FT \rightarrow$ $\underline{\text{Set-of}}$ $\varphi(FC_{O2})/\eta[(m_1, n_1), (m_2, n_2)]$, where $\varphi(FC_{O1})$, $\varphi(FC_{O2}) \in FC_{FS}$, $\varphi(FC_O) \in FA_{FS}$, $\eta \in [0, 1]$.

Rule 15: Fuzzy complex class axiom *Class* (FC_O *partial restriction* (*f allValuesFrom(R) maxCardinality(1)*)) is mapped into method of *FOOD* model

$\varphi(f) : R$, the method $\varphi(f)$ of parameter is null.

Rule 16: Fuzzy complex class axiom *a fuzzy class identifier* $FC_{f(P1,...,Pm)} \in FC_O$; *m fuzzy data range identifiers* $P_1, ..., P_m \in FDP_O$; *a fuzzy data range identifier* $R \in FDT_O$ can be mapped into method of *FOOD* model

$\varphi(f(P_1, ..., P_m)) : R$, the method $\varphi(f)$ of parameter is $P_1, ..., P_m$, $\varphi(R) \in FD_{FS}$.

Rule 17: Fuzzy ontology datatype property *DatatypeProperty*(FDP_{Oi}, *domain*(FC_O) *range*(FDT_{Oi})) is mapped into fuzzy class declaration of *FOOD* model

Class $\varphi(FC_O)$ $\underline{\text{type is}}$

$\underline{\text{Record}}$ $\varphi(FDP_{Oi}) : \varphi(FDT_{Oi})$ $\underline{\text{End}}$, where $\varphi(FC_O) \in FC_{FS}, \varphi(FDP_{Oi}) \in FA_{FS}, \varphi(FDT_{Oi}) \in FD_{FS}$.

Rule 18: Fuzzy ontology object property *ObjectProperty* (FOP_O *hasop* $FOP_{Oi}(FC_{Oi})$ *range* (FC_O)) is mapped into fuzzy class declaration of *FOOD* model

Object $\varphi(FOP_O)$ $\underline{\text{belong to}}$ $\varphi(FC_O)$ $\underline{\text{has-value}}$ $\varphi(FOP_{Oi}) : \varphi(FC_{Oi})$ $\underline{\text{End}}$, here $\varphi(FC_O)$, $\varphi(FC_{Oi}) \in FC_{FS}$, $\varphi(FOP_O), \varphi(FOP_{Oi}) \in FA_{FS}$.

Rule 19: Fuzzy ontology axioms *Class* ($FC_{Of(P1,...,Pm)}$ *partial restriction* (r_1 *someValuesFrom* (*owl:Thing*) *Cardinality*(1)) ... *restriction* (r_m *someValuesFrom* (*owl:Thing*) *Cardinality*(1))); *Class* ($FC_{Of(P1,...,Pm)}$ *partial restriction*

(r_1 *allValuesFrom* (P_1)) ... *restriction* (r_m *allValuesFrom* (P_m))); *Class* (FC_O *partial restriction* (*inverseOf*(r_1) *allValuesFrom*(*unionOf*(*complementOf* ($FC_{f(P1,...,Pm)}$) *restriction* (r_{m+1} *allValuesFrom* (*R*)))))), is mapped into fuzzy class declaration of *FOOD* model

Class $\varphi(FC_O)$ type is Record $\varphi(f(P_1,...,P_m)) : R$ End, here, $\varphi(f(P_1,...,P_m))$ is a method with m parameters $P_1,...,P_m$, $\varphi(FC_O) \in FC_{FS}$, $R \in \{r_1,...,r_m\} \in FA_{FS}$.

Rule 20: Fuzzy individual axiom *SameIndividual* ($FI_{O1}...FI_{On}$) *or DifferentIndividuals*($FI_{O1}...FI_{On}$) is mapped into fuzzy objects of *FOOD* model

n objects $\varphi(FI_{Oi})$ *are equivalent or different*, here $\varphi(FI_{Oi}) \in FO_{FS}$, $i \in \{1,...,n\}$.

Rule 21: With a membership fuzzy individual *Individual* (FI_O type (FC_O) $\bowtie \mu$) is mapped into fuzzy objects of *FOOD* model

Objects $\varphi(FI_O)$ belong-to $\varphi(FI_O)/\mu$ *End* ,where $\bowtie \in \{\geq, \leq\}$, $\mu \in [0,1]$, $\varphi(FI_O) \in FO_{FS}$, $\varphi(FC_O) \in FC_{FS}$.

3.2. The correctness of the transformation approach

The *Sections A* specify some mapping rules that can transform fuzzy OWL 2 ontology to *FOOD* model. How to prove the correctness of the transforming rules is an important and challenge task. This part, we prove correctness of the approach which can be established mapping instance of fuzzy OWL 2 ontology and *FOOD* model.

Theorem 1. For every fuzzy OWL 2 ontology O_F and its transformed *FOOD* model $\varphi(O_F)$, there exist two mappings δ from fuzzy OWL 2 ontologies structure to models $\varphi(O_F)$, and ζ from models $\varphi(O_F)$ to fuzzy OWL 2 ontology structure, such that:

• For each fuzzy OWL 2 ontology model *FI* conforming to O_F, $\delta(FI)$ is a *FOOD* model $\varphi(O_F)$.

• For each database state *FJ* of $\varphi(O_F)$, $\zeta(FJ)$ is a fuzzy OWL 2 ontology model conforming to O_F.

Proof. The following first proves the first part of *Theorem 1*. Let $FI = (\triangle^{FI}, \bullet^{FI}))$ be a fuzzy interpretation of fuzzy OWL 2 ontology O_F, and $o \in \triangle^{FI}$ be an ontology instance, an instance model $\delta(o)$ is conforming to the *FOOD* $\varphi(O_F)$.

Given a given a fuzzy ontology model *FI*, each symbol $X \in FOP_O \cup FDP_O \cup FC_O \cup FDT_O \cup FP_C$, a fuzzy database state $\delta(FI)$ of $\varphi(O_F)$ can be defined as follows:

The domain elements $\triangle^{\delta(FI)}$ of a database state $\delta(FI)$ of $\varphi(O_F)$ are constituted by the values of the fuzzy OWL 2 ontology semantic interpretation *FI*.

The fuzzy database state *FJ* of $\varphi(O_F)$ in *Section A* is defined as follows:

• $(\varphi(X))^{\delta(FI)} = X^{FI}$, where $X \in \varphi(FC_O)$;

• For each fuzzy class declaration $(\varphi(FDP_{Oi}))^{\delta(FI)} = \{< FP_C, d_i > \in \triangle^{\delta(FI)} \times \triangle^{\delta(FI)} \mid FP_C \in FC_O^{FI} \wedge d_i \in FDT_{Oi}^{FI}\}$, where $i \in \{1,...,k\}$, we have *Class* $\varphi(FC_O)$ type-is Record $\varphi(FDP_{O1}) : \varphi(FDT_{O1}),...,\varphi(FDP_{Ok}) : \varphi(FDT_{Ok})$ End;

• For each fuzzy class declaration $(\varphi(FU_j))^{\delta(FI)} = \{< r, FOP_{Oj} > \in \triangle^{\delta(FI)} \times \triangle^{\delta(FI)} \mid r \in FP_C^{FI} \wedge FOP_{Oj} \in FC_{Oj}^{FI}\}$, $j \in \{1,2\}$, we have *Class* $\varphi(FC_{O1})$ type-is Record $\varphi(FP_C)$: *Set-of* $\varphi(FC_{O2})$ End.

Further, we prove $\delta(FI)$ is a model of $\varphi(O_F)$, i.e., prove $\delta(FI)$ satisfies the definition of $\varphi(O_F)$ in *Definition 2*. Note that, the semantics of $\varphi(O_F)$ models can be partitioned into several main cases:

• for a fuzzy OWL 2 ontologies interpretation *FI*. If there are $(FOP_O)^{FOP} \in \triangle^{FI} \times \triangle^{FI}$ and *FOOD* fuzzy class $\varphi(FC_O)$ such that *Class* $\varphi(FC_O)$ is-a $\varphi(FC_{Osup})/\beta$, then $\varphi(FC_O)^{\delta(FI)}(FOP_O) \subseteq \varphi(FC_{Osup})^{\delta(FI)}(FOP_O)$, i.e., $FC_O^{FI} \subseteq FC_{Osup}^{FI}$. That is, $\delta(FI)$ satisfies the corresponding fuzzy semantic of *FOOD* model in *Definition 2*.

• for a fuzzy OWL 2 ontologies class FC_O such that $DisjointUnion(FC_O FC_{O1} FC_{O2}...FC_{On})$. According to *Definition 3*, if *FI* is a fuzzy interpretation, we have $FC_O^{FI} = FC_{O1}^{FI} \cup ... \cup FC_{Oq}^{FI}$ and $FC_{Oi}^{FI} \cap FC_{Oj}^{FI} = \emptyset$, where $i, j \in \{1,...,q\}$ and $i \neq j$. By definition of $\delta(FI)$ above, it follows $\varphi(FC_O)^{\delta(FI)} = \varphi(FC_{O1})^{\delta(FI)} \cup ... \cup \varphi(FC_{Oq})^{\delta(FI)}$ and $\varphi(FC_{Oi})^{\delta(FI)} \cap \varphi(FC_{Oj})^{\delta(FI)} = \emptyset$, such as *Class* $\varphi(FC_O)$ type-is Union $\varphi(FC_{O1}),...,\varphi(FC_{Oq})$ (*disjoint, complete*) End. That is, $\delta(FI)$ satisfies the corresponding fuzzy semantic of *FOOD* model in *Definition 2*.

• for a fuzzy OWL 2 ontology class FC_O such that $(FC_O)^{FC} = \triangle^{FI} \times \triangle^{FI}$ and fuzzy class of *FOOD* model such that *Class* $\varphi(FC_O)$ type-is Record $\varphi(FDP_{O1}) : \varphi(FDT_{O1}),...,\varphi(FDP_{Ok}) : \varphi(FDT_{Ok})$ End, where $\varphi(FC_O) \in FC$, $\varphi(FDP_{Oi}) \in FA$, $\varphi(FDT_{Oi}) \in FV$, $i \in \{1,...,k\}$. For an instance $FI_O \in [FDP_{O1} : FDT_{O1},...,FDP_{Ok} : FDT_{Ok}]$, according to *Definition 3*, if *FI* is a fuzzy interpretation, then $(FDP_O)^{FDP} = \triangle^{FI} \times \triangle^{FD}$. By definition of $\delta(FI)$ above, there is exactly one element $d_i \in FDT_{Oi}^{FI} = \varphi(FDT_{Oi})^{\delta(FI)}$ such that $(FC_O, d_i) \in \varphi(FDP_{Oi})^{\delta(FI)}$, i.e.,

$\varphi(FI_O)^{\delta(FI)} \subseteq \bigcap_{i=1}^{k}\{FC_O|\forall d_i. < FC_O, d_i >\in (\varphi(FA_i))^{\zeta(FJ)} \to d_i \in (\varphi(FD_i))^{\zeta(FJ)} \wedge \sharp\{d_i| < FO, d_i >\in \varphi(FDT_{Oi})^{\delta(FI)}\} = 1\}$. That is, $\delta(FI)$ satisfies the semantics of *Definition 2*.

• for fuzzy classes of OWL 2 ontology FC_{O1}, FC_{O2} and fuzzy class of *FOOD* model such that *Class* $\varphi(FC_{O1})$ type-is Record $\varphi(FOP_O)$: $\underline{Set-of}\,\varphi(FC_{O2})/\eta[(m_1, n_1), (m_2, n_2)]$, where $\varphi(FC_{Oi}) \in FC$, $\varphi(FOP_O) \in FA$, $i \in \{1, 2\}$. For an instance $r \in \varphi(FOP_O)^{\delta(FI)}$, it follows $r \in \{FC_{O1}, ..., FC_{Ok}\}$. By definition of $\delta(FI)$ above, there is exactly one example $FI_{Oj} \in FC_{Oj}^{FI} = \varphi(FC_{Oj})^{\delta(FI)}$ such that $(r, FI_{Oj}) \in \varphi(FC_{Oj})^{\delta(FI)}$, i.e., $\varphi(FP_C)^{\delta(FI)} \subseteq \bigcap_{j=1}^{2}\{r\,|\,\forall FI_{Oj} < r, FI_{Oj} >\in \varphi(FU_j)^{\delta(FI)} \to FI_{Oj} \in \varphi(FC_{Oj})^{\delta(FI)} \wedge \sharp\{FO_j\,| < r, FI_{Oj} >\in \varphi(FU_j)^{\delta(FI)}\} = 1\}$. Moreover, according to the semantics of cardinality constraints on associations, we have $card_{min}(FC_{Oj}, FP_C, FOP_{Oj}) \leq \sharp\ \{r \in FP_C^{FI}\,|FP_C[FU_j] = FI_{Oj}\} \leq card_{max}(FC_{Oj}, FP_C, FOP_{Oj})$, which denotes the minimum and maximum times of an object instance of a fuzzy class participating in an association. Further, by definition of $\delta(FI)$ above, it follows $\varphi(FC_{Oj})^{\delta(FI)} \subseteq \{FI_{Oj}\,|\,card_{min}(FC_{Oj}, FP_C, FOP_{Oj}) \leq \sharp\{r \in FP_C^{FI}\,|\, < r, FI_{Oj} >\in FP_C^{FI}\} \leq card_{max}(FC_{Oj}, FP_C, FOP_{Oj})\}$. In addition, we know that FW_1=invof_$\varphi(FU_1)$ and FW_2=invof_$\varphi(FU_2)$, are the inverse object property identifiers of FU_1 and FU_2, respectively, and thus we have $\varphi(FW_j)^{\delta(FI)} = \{< FI_{Oj}, r >\in \triangle^{FI} \times \triangle^{FI}\,|\,FI_{Oj} \in FCO_j^{FI} \wedge r \in FP_C^{FI}\}$, $j \in \{1, 2\}$, i.e., $\varphi(FW_j)^{\delta(FI)} = \varphi(FU_j)^{\delta(FI)-} \subseteq \varphi(FC_{Oj})^{\delta(FI)} \times \varphi(FP_C)^{\delta(FI)}$. That is, $\delta(FI)$ satisfies the corresponding semantics of this case in *Definition* 2.

It is shown that the translation $FOOD_{FS} = \varphi(O_F)$ is semantic preservation since that for each fuzzy interpretation *FI* of fuzzy OWL 2 ontology, there is a mapping $\delta : FI \to FJ$ so that $FJ = \delta(FI)$ is a model of $\varphi(O_F)$. Thus the first part of *Theorem* 1 is proved. The second part of *Theorem* 1 is an inverse process of the first part of *Theorem* 1. The proof of the second part of *Theorem* 1 is analogous to the first part.

3.3. A transforming example from fuzzy OWL 2 ontology to a FOOD model

In this section, we provide a fuzzy OWL 2 ontology instance in *Figure* 1, and the corresponding *FOOD* model derived from the instance applying these rules in *part A* is shown in *Figure* 2.

Class Order type-is Record OrderID: String dateReceived: String isPrepaid:Boolean End
Class *Young-Customer* is-a *Corporate-Customer*/0.8 type-is Record CardNo: String CustName: String FUZZY-age: String Making: Set-of*Order* [(0,∞),(1,1)] μ : Real End
Class *Corporate-Customer* is-a *Customer* type-is Record ContactName: String FUZZY-CreditRating: String FUZZY-Discount: Real μ : Real End
Class *Employee* type-is Record FUZZY Serving: Set-of Corporate-Customer /η [(1, 3), (3,∞)] End
Class *Customer* type-is Union Corporate-Customer, Personal-Customer (disjoint, complete) End
Object y1 belong-to Young_Customer/0.9 has-value CustNo: 2013013, CustName: Lucy, FUZZY-age: young, Making: {*o1, o2, o3*}, μ : 0.9 End
Object e1 belong-to Employee has-value FUZZYServing: {y1/0.96, y2/0.6} End
Object cc1 belong-to Corporate-Customer;
Object o1, o2, o3 belong-to Order
Object s2 belong-to Staff/0.9...

Figure 2. A FOOD model derived from fuzzy OWL 2 ontology in Figure. 1

4. Conclusions

In this paper, we mainly investigate fuzzy OWL 2 ontology and *FOOD* model. Firstly, their formal definitions are proposed. Then, we present a methodology of transforming fuzzy OWL 2 ontology into *FOOD* model on structure and instance levels. The correctness of the approach is proved, and a transformation example is described to explain the transforming approach.

In the future, we intend to test and evaluate the reusing fuzzy OWL 2 ontologies approach with more complex example based on *FOOD* models. Moreover, extending existing database system has reasoning capabilities for *FOOD* models.

References

[1] M.I.O.Ali,F.Laroche,A.Bernard,S.Remy, Toward a methodological knowledge based approach for partial automation of reverse engineering, In CIRP Design conference. 21(2014)270-275.
[2] S.M.Benslimane, M.Malki, D.Bouchiha, Deriving Conceptual Schema from Domain Ontology: A Web Application Reverse Engineering Approach. Int. Arab J. Inf. Technol.7(2)(2010)167-176.

[3] D.V.Ban, H.C.Ha, V.D.Quang, Normalizing object classes in fuzzy object-oriented database schema, Journal of computer science and cybernetics. 28(2)(2011)131-141.
[4] F.Berzal, N.Marn,O.Pons, M.A.Vila, Managing fuzziness on conventional object-oriented platforms, International Journal Intelligent Systems. 22(7)(2007)781-803.
[5] F. Bobillo, U.Straccia, Fuzzy ontology representation using OWL 2, International Journal of Approximate Reasoning. 52(7)(2011)1073-1094.
[6] F.Bobillo, Managing vagueness in ontologies. Ph.D. thesis, University of Granada, Spain, 2008.
[7] T.Cipresso, M.Stamp,Software reverse engineering, Handbook of Information and Communication Security. Springer Science & Business Media, Berlin, Germany.(2010)659-696.
[8] D.Dubois, H.Prade, J.P.Rossazza, Vagueness, typicality, and uncertainty in class hierarchies, International Journal of Intelligent Systems.6(1991) 167C183.
[9] A.Gmez-Prez, M.D.Rojas-Amaya, Ontological reengineering for reuse, Knowledge Acquisition, Modeling and Management. Springer Berlin Heidelberg.(1999) 139-156.
[10] M.Hazman, S.R.El-Beltagy, A.Rafea,A Survey of Ontology Learning Approaches, International Journal of Computer Applications. 22(8)(2011) 36-43.
[11] M.Koyuncu, A.Yazici, IFOOD: an intelligent fuzzy object-oriented database architecture, IEEE Transactions on Knowledge and Data Engineering. 15(5)(2003) 1137C1154.
[12] W.J.Li, X.Chen, Z.M.Ma, Reengineering fuzzy nested relational databases into fuzzy XML model, Proceedings of the 2014 IEEE International Conference on Fuzzy Systems.(2014) 1612-1617.
[13] D.Lonsdale, D.W.Embley, Y.Ding, et al., Reusing ontologies and language components for ontology generation, Data & Knowledge Engineering, 69(4)(2010) 318-330.
[14] T.Lukasiewicz, U.Straccia, Managing uncertainty and vagueness in description logics for the semantic web, Journal of Web Semantics. 6(4)(2008)291-308.
[15] C.W.Lu, W.C.Chu, C.H.Chang,et al., Reverse engineering, J. Handbook of Software Engineering and Knowledge Engineering. 2(2002)5.
[16] Z.M.Ma, W.J.Zhang, W.Y.Ma, Extending object-oriented databases for fuzzy information modeling, Information Systems. 29(5)(2004)421-435.
[17] Z.M.Ma, L.Yan, A Literature Overview of Fuzzy Database Models, Journal of Information Science and Engineering. 24(2008)189-202.
[18] OWL 2 Web Ontology Language Direct Semantics (Second Edition), http://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/
[19] OWL 2 Web Ontology Language Document Overview (Second Edition), https://www.w3.org/TR/owl2-overview/
[20] OWL 2 Web Ontology Language New Features and Rationale (Second Edition), http://www.w3.org/TR/2012/REC-owl2-new-features-20121211/
[21] D.N.Peralta, H.S.Pinto, N.J.Mamede, Reusing a time ontology, ICEIS. (2003) 121.
[22] K.V.S.V.N.Raju, A.K.Majumdar, Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems, ACM Transactions on Database Systems. 13(2)(1988)129-166.
[23] P.K.Shukla, M.Darbari, V.K.Singh, et al., A survey of fuzzy techniques in object-oriented databases, International Journal of Scientific and Engineering Research.2(11)(2011)1-11.
[24] G.Stoilos, G.Stamou, J.Z.Pan, Fuzzy extensions of OWL: Logical properties and reduction to fuzzy description logics, International Journal of Approximate Reasoning. 51(6)(2010)656-679.
[25] C.A.Yaguinuma, M.T.Santos, H.A.Camargo, et al., A meta-ontology for modeling fuzzy ontologies and its use in classification tasks based on fuzzy rules, International Journal of Computer Information Systems and Industrial Management Applications. 6(2014)89-101.
[26] L.Yan, Z.M.Ma, A Probabilistic Object-oriented Database Model with Fuzzy Measures, In Advances in Probabilistic Databases for Uncertain Information Management. Springer Berlin Heidelberg.(2013)23-38.
[27] L.A.Zadeh, Fuzzy Sets, Information and Control. 8(3)(1965)338-353.
[28] L.A.Zadeh, Fuzzy Sets as a Basis for a Theory of Possibility, Fuzzy Sets and Systems. 1(1)(1978)3-28.
[29] F.Zhang, Z.M.Ma, G.F.Fan, X.Wang, Automatic fuzzy semantic web ontology learning from fuzzy object-oriented database model, Database and Expert Systems Applications. Springer Berlin Heidelberg.(2010)16-30.
[30] F.Zhang, Z.M.Ma, L.Yan, Y.Wang, A description logic approach for representing and reasoning on fuzzy object-oriented database models, Fuzzy Sets and Systems.186(1)(2012)1-25.
[31] F.Zhang, Z.M.Ma, X.Chen, Formalizing fuzzy object-oriented database models using fuzzy ontologies, Journal of Intelligent & Fuzzy Systems. 29(4)(2015)1407-1420.