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

Weijun Li\textsuperscript{a,c}, Li Yan\textsuperscript{b}, Xu Chen\textsuperscript{c}, Fu Zhang\textsuperscript{a}, Jiulei Jiang\textsuperscript{c}

\textsuperscript{a}School of Computer Science & Engineering, Northeastern University, Shenyang 110169, China
\textsuperscript{b}College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 2111069, China
\textsuperscript{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
databases have increasingly attracted considerable attention. 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 Zadeh’s 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 Zadeh’s 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 $\mu_F : U \rightarrow [0, 1]$, where $\mu_F(u)$, for each $u \in U$, the membership degree $\mu$ belonging to the fuzzy set $F$. For the case where $U$ is a discrete domain, the fuzzy set $F$ is expressed as

\[ F = \mu_F(u_1), \mu_F(u_2), \ldots, \mu_F(u_n) \]

Here $\mu_F$ is used to represent the membership function of the fuzzy set $F$, $\mu_F(u)$ is used to represent the membership degree $\mu$ 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 $\pi_X$ [28]:

\[ \pi_X = \pi_X(u_1)/u_1, \pi_X(u_2)/u_2, \ldots, \pi_X(u_n)/u_n \]

Here, for any $u_1 \in U$, $\pi_X(u_i)$ indicates the possibility that $u_i$ is true. Let $\pi_X$ and $F$ denote a fuzzy possibility distribution and a fuzzy set, respectively, then $\pi_X$ 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 hierarchies and the cardinality constraints on associations).

**Definition 1 (Formal definition of FOOD model)**: A FOOD model is a tuple $\text{FOOD}_{FS} = (\text{FO}_{FS}, \text{FD}_{FS}, \text{FC}_{FS}, \text{FA}_{FS}, \text{FP}_{FS})$ consisting of a set of object identities, attribute domains, classes, attributes, and class declarations [30], [31], where:

1. $\text{FO}_{FS}$ is a set of objects $\text{FO};$ each object has a unique object identity;
2. $\text{FD}_{FS}$ is a set of domains $\text{FD},$ which includes crisp and fuzzy domains.
3. $\text{FC}_{FS}$ is a set of classes $\text{FC};$
4. $\text{FA}_{FS}$ is a set of attributes $\text{FA};$ each attribute $\text{FA}$ is associated with a domain $\text{FD},$ and using the fuzzy keyword FUZZY in front of an attribute denotes that this attribute is fuzzy;
5. $\text{FP}_{FS}$ is a set of class declarations. For each class $\text{FC} \in \text{FC}_{FS}, \text{FP}_{FS}$ contains a declaration:

   Class $\text{FC}$ is a Fuzzy Type $\text{FT},$

   $\text{FT}$ denotes a schema type expression built according to the following syntax:

   $\text{FT} \rightarrow \{ F\text{O}_{1}/\mu_{1}, F\text{O}_{2}/\mu_{2},...,F\text{O}_{n}/\mu_{n} \} \text{ End }$ ; $\text{Union} \text{FC}_{1},...,$ $\text{FC}\text{d}(\text{disjoint, complete}) \text{ End }$ ; $\text{Record} \text{FA}_{1} : \text{FT}_{1},...,$ $\text{FA}_{k} : \text{FT}_{k}, \mu : \text{Real}, f(P_{1},...,P_{m}) : \text{R} \text{ End }$.

   where $\text{FT}_{i}$ is one of the following cases (where $i \in \{1...k\}$): $\text{FT}_{i} \rightarrow \text{FD}_{i}$ ; Set of $\text{FC}_{i}/\eta([m_{1},n_{1}], [m_{2},n_{2}])$.

   $\text{FOOD}_{FS}$ is a set of object declarations to represent values of attributes of objects. For each object $\text{FO} \in \text{FO}_{FS}, \text{FO}$ belong-to $\text{FC}/\mu,$ has-value $[\text{FA}_{1} : \text{FD}_{1}, \text{FA}_{2} : \text{FD}_{2},... \text{ End }]$.

Where:

- The type-is part specifies the structure of a class $\text{FC}$ by a type expression $\text{FT};$
- The is-a part, which is optional, denotes inheritance relationship between fuzzy classes with a membership degree $\beta \in [0, 1];$
- The expression $\{ F\text{O}_{1}/\mu_{1}, F\text{O}_{2}/\mu_{2},...,F\text{O}_{n}/\mu_{n} \}$ denotes that $\text{FC}$ is an extensional class which has a list of object instances $\{ F\text{O}_{1}/\mu_{1}, F\text{O}_{2}/\mu_{2},...,F\text{O}_{n}/\mu_{n} \},$ and each object $\text{FO}_{i}$ has a membership degree of $\mu \in [0,1]$ relative to the class $\text{FC};$
- The $\text{Union...End}$ part denotes a class hierarchy;
- The $\text{Record...End}$ part denotes that a fuzzy class $\text{FC}$ is defined by a set of attributes and their admissible values, this class is called an extensional class; an additional attribute $\mu \in [0,1]$ is used to represent the membership degree of an object belonging to the class $\text{FC};$ $f(P_{1},...,P_{m}) : \text{R}$ represents a method, where $f$ is the name of the method, $P_{1},...,P_{m}$ are types of $m$ parameters, and $\text{R}$ is the type of the result;
- The $\text{Set-of}$ part (i.e., Class $\text{FC}$ type-is Record $\text{FA}_{i}$: Set-of $\text{FC}_{i}/\eta([m_{1},n_{1}], [m_{2},n_{2}]) \text{ End }$ ) denotes an association relationship between classes $\text{FC}$ and $\overline{\text{FC}}_{i}$ by an attribute $\text{FA}_{i}; \text{eta} \in [0, 1]$ the association occurs.
in classes $FC_1$ and $FC_i$ with a membership degree of $\eta_i$; \{[(m_1, n_1),(m_2, n_2)] \} indicates that the association involves at least $m_1$ 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 $F_{FJ} = (FV_{FJ}, \pi_F, \rho_F)$ [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:

- $FD_{FJ} = \bigcup_{i=1}^{m} FD_i$, 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 $\mu_i$;

- $FR_{FJ}$ 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, $FV_i \in FV_{FJ}$, $i \in \{1, \ldots, k\}$;

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

2) A function $\pi_F$ maps a class to a set of its objects.

3) A function $\rho_F$ 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 $FT_{FJ} = FC_{FJ} = \pi_F(FC)$;

- If $FT$ is a record type $Record$ then $FT_{FJ}$ is a set of record values $FR_{J}$ (resp. a set of set-values $FS_{FJ}$);

- If $FT$ is a union type $Union FC_1, ..., FC_q$ (disjoint, complete) then $FT_{FJ}$ is a set of record values $FR_{J}$ and $FC_{FJ} = FC_1 \cup ... \cup FC_q$ and $FC_{FJ} \cap FC_{FJ} = \emptyset$, 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 Zadeh 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; $a$ indicates an individual (named or anonymous); $\mu$ indicates a literal; $FA$ indicates a constraining facet; $\#\$ indicates the cardinality set $S$, and $\ll not\ll \{\geq, >, \leq, <\}$.

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 $O_F = (FOP_O, FDP_O, FDT_O, FCO_O, FPO_O, FLO_O, FLO_{Atm},)$, 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.

<table>
<thead>
<tr>
<th>Fuzzy OWL abstract syntax</th>
<th>Fuzzy DL syntax</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class(FC)</td>
<td>FC</td>
<td>FCI = C0 = C1</td>
</tr>
<tr>
<td>one : Thing</td>
<td>⊤</td>
<td>one : Thing = 1</td>
</tr>
<tr>
<td>one : Nothing</td>
<td>⊥</td>
<td>one : Nothing = 0</td>
</tr>
<tr>
<td>ObjectIntersection(FC1, FC2)</td>
<td>FC1 \cap FC2</td>
<td>(FC1) = \cap (FC1, FC2)</td>
</tr>
<tr>
<td>ObjectUnion(FC1, FC2)</td>
<td>FC1 \cup FC2</td>
<td>(FC1) = \cup (FC1, FC2)</td>
</tr>
<tr>
<td>ObjectComplement(FC1)</td>
<td>~FC1</td>
<td>(FC1) = \neg (FC1)</td>
</tr>
<tr>
<td>ObjectSomeValuesFrom(FOPE, FCE)</td>
<td>FOPE \sqcap FCE</td>
<td>{</td>
</tr>
<tr>
<td>ObjectAllValuesFrom(FOPE, FCE)</td>
<td>\forall</td>
<td>y</td>
</tr>
<tr>
<td>ObjectHasValue(FOPE, a)</td>
<td>FOPE \sqcap { a }</td>
<td>({a}) \in (FOPE)</td>
</tr>
<tr>
<td>ObjectMinCardinality(FOPE)</td>
<td>≤ nFOPE</td>
<td>{</td>
</tr>
<tr>
<td>ObjectMaxCardinality(FOPE)</td>
<td>≥ nFOPE</td>
<td>{</td>
</tr>
<tr>
<td>ObjectExactCardinality(FOPE)</td>
<td>= nFOPE</td>
<td>{</td>
</tr>
<tr>
<td>DisjointObjectProperties</td>
<td>FOPE \sqcap FOPE</td>
<td>(FOPE, FOPE) = \emptyset</td>
</tr>
<tr>
<td>EquivalentObjectProperties</td>
<td>FOPE \equiv FOPE</td>
<td>(FOPE, FOPE) = \emptyset</td>
</tr>
<tr>
<td>DataMinCardinality(FOPE)</td>
<td>≤ nFOPE</td>
<td>{</td>
</tr>
<tr>
<td>DataMaxCardinality(FOPE)</td>
<td>≥ nFOPE</td>
<td>{</td>
</tr>
<tr>
<td>DataExactCardinality(FOPE)</td>
<td>= nFOPE</td>
<td>{</td>
</tr>
<tr>
<td>DataIntersection(FOPE, FOPE)</td>
<td>FOPE \cap FOPE</td>
<td>(FOPE, FOPE) = \emptyset</td>
</tr>
<tr>
<td>DataUnion(FOPE, FOPE)</td>
<td>FOPE \cup FOPE</td>
<td>(FOPE, FOPE) = \emptyset</td>
</tr>
<tr>
<td>DataComplement(FOPE)</td>
<td>~ FOPE</td>
<td>(FOPE) = \neg (FOPE)</td>
</tr>
<tr>
<td>DataSomeValuesFrom(FOPE, FOPE)</td>
<td>FOPE \sqcap FOPE</td>
<td>{</td>
</tr>
<tr>
<td>DataAllValuesFrom(FOPE, FOPE)</td>
<td>\forall</td>
<td>y</td>
</tr>
</tbody>
</table>

### Fuzzy Data Ranges

| DataIntersection(FOPE, FOPE) | FOPE \cap FOPE | (FOPE, FOPE) = \emptyset |
| DataUnion(FOPE, FOPE) | FOPE \cup FOPE  | (FOPE, FOPE) = \emptyset |
| DataComplement(FOPE) | ~ FOPE          | (FOPE) = \neg (FOPE) |
| DataSomeValuesFrom(FOPE, FOPE) | FOPE \sqcap FOPE | \{ | y | (y, y) \in (FOPE) \} for each 1 \leq k \leq n and (y1, y2, ... , yk) \in (FDPE) |
| DataAllValuesFrom(FOPE, FOPE) | \forall | y | (y, y) \in (FOPE) \} for each 1 \leq k \leq n imply (y1, y2, ... , yk) \in (FDPE) |

### Fuzzy Class Axioms

| SubClassOf(FC1, FC2) | FC1 \sqsubseteq FC2 | (FC1) = (FC2) |
| EquivalentClass(FC1, FC2) | FC1 = FC2 | (FC1) = (FC2) |
| DisjointClasses(FC1, FC2) | FC1 \cap FC2 = \emptyset | (FC1, FC2) = \emptyset |

### Fuzzy Object property axioms

| SubObjectPropertyOf(FOPE, FOPE) | FOPE1 \sqsubseteq FOPE2 | (FOPE1) = (FOPE2) |
| EquivalentObjectProperty(FOPE, FOPE) | FOPE1 = FOPE2 | (FOPE1) = (FOPE2) |
| DisjointObjectProperties(FOPE, FOPE) | FOPE1 \cap FOPE2 = \emptyset | (FOPE1, FOPE2) = \emptyset |
| ObjectPropertyDomain(FOPE, FCE) | FOPE \sqcap FCE | (FOPE) \cap (FCE) = \emptyset |
| ObjectPropertyRange(FOPE, FCE) | FOPE \sqsupseteq FCE | (FOPE) \supseteq (FCE) |
| FunctionalObjectProperty(FOPE) | FOPE \sqsubseteq FOPE | (FOPE) \subseteq (FOPE) |

### Fuzzy Data property Axioms

| SubDataPropertyOf(FOPE, FOPE) | FOPE1 \sqsubseteq FOPE2 | (FOPE1) = (FOPE2) |
| EquivalentObjectProperty(FOPE, FOPE) | FOPE1 = FOPE2 | (FOPE1) = (FOPE2) |
| DisjointObjectProperties(FOPE, FOPE) | FOPE1 \cap FOPE2 = \emptyset | (FOPE1, FOPE2) = \emptyset |
| DataPropertyDomain(FOPE, FCE) | FOPE \sqcap FCE | (FOPE) \cap (FCE) = \emptyset |
| DataPropertyRange(FOPE, FCE) | FOPE \sqsupseteq FCE | (FOPE) \supseteq (FCE) |

### Fuzzy Assumptions

| SameIndividual(a1, a2) | a1 = a2 | (a1, a2) |
| DifferentIndividual(a1, a2) | a1 \neq a2 | (a1, a2) |
| ClassAssertion(FC1) | (\forall y) | y \in (FC1) |
| ObjectPropertyAssertion(FOPE, a1) | FOPE \sqcap \{ a \} | (a) \in (FOPE) |
| NegativeObjectPropertyAssertion(FOPE, a1) | FOPE \sqcap \{ a \} | (a) \not\in (FOPE) |
| DataPropertyAssertion(FOPE, a1) | FOPE \sqcap \{ a \} | (a) \in (FOPE) |
| NegativeDataPropertyAssertion(FOPE, a1) | FOPE \sqcap \{ a \} | (a) \not\in (FOPE) |
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) $FL_O$ is a collection of fuzzy individuals (named and anonymous);
7) $FL_O$ 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:

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 $\mu$. 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 FZZY is used to represent fuzzy attribute values of elements. For example, the attribute FZZY-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 $O_F = \{FOD_O, FDP_O, FDT_O, FC_O, FPC, FLO, Flt_O, FO_{Axiom}\}$, 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., $\varphi(FI_O) \subseteq FO \in FOOD_{FS}$.

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

**Rule 3:** Each fuzzy datatype property identifier $FDP_O$ is mapped into a simple fuzzy attribute of FOOD model $FA$, i.e., $\varphi(FDP_O) \subseteq FA \in FOOD_{FS}$, 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 $\varphi(FL_U) \in FA$, $FW_U = invof_\varphi(FL_U) \in FA$, $FW_2 = invof_\varphi(FL_U) \in FA$, where $FW_1$ and $FW_2$ denote inverse properties of $\varphi(FL_U)$ and $\varphi(FL_U)$, respectively. This fuzzy class is mapped into complex fuzzy attribute of FOOD model $\varphi(FC_O) \subseteq FA \in FOOD_{FS}$, this attribute denotes an association relationship between classes.
Rule 5: Each fuzzy datatype $FDT_b$ is mapped into fuzzy domain of FOOD model $FD$, i.e., $\varphi(FDT_b) \subseteq FD \in FDS$.

Rule 6: Each fuzzy object properties identifier $FOP_i$ is mapped into an object of FOOD model $FO$, i.e., $\varphi(FOP_i) \subseteq FO \in FOS$.

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

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 $EnumeratedClass(FOOD_{OI}(\mu_1,...,\mu_n))$ is mapped into FOOD class declaration $Class \varphi(FOOD) \text{ type-is } \{\varphi(FOOD_{OI}))/\mu_1,...,\varphi(FOOD_{OI}))/\mu_n\}$ End, where $\varphi(FOOD)$, $\varphi(FOOD_{OI}) \in FCS$, $\mu_i \in [0, 1]$, $i \in [1, ..., n]$.

Rule 10: A taxonomic or hierarchy relationship between fuzzy ontology classes $SubClassOf(FOOD_{OI}, FOOD_{OJ}, \beta)$ is mapped into FOOD class declaration

$$\varphi(FOOD_{OI}) \text{ is-a } \varphi(FOOD_{OJ})/\beta,$$ where $\varphi(FOOD_{OI}), \varphi(FOOD_{OJ}) \in FCS$, $\beta \in [0, 1], i, j \in [1, ..., n]$.

Rule 11: A relationship axiom of fuzzy ontology $Class (FOOD \ \text{complete UnionOf} (FOOD_{OI}...FOOD_{OJ}))$, Disjoint-Class ($FOOD_{OI}, FOOD_{OJ}$), $i \neq j, i, j \in [1, ..., q]$ is mapped into FOOD class declaration

Class $\varphi(FOOD)$ type-is Union $\varphi(FOOD_{OI}),...,$ $\varphi(FOOD_{OJ})$ (disjoint, complete) End, where $\varphi(FOOD)$, $\varphi(FOOD_{OI}) \in FCS$.

Rule 12: A fuzzy class description $Class (FOOD_{OI} \ \text{partial...restriction} (FOP_i \ \text{allValuesFrom} (FDT_b) \text{cardinality(1))))$ is mapped into fuzzy class declaration of FOOD model

Class $\varphi(FOOD)$ type is $Record \varphi(FOP_i) : (FDT_b),...,$ $\varphi(FOP_i) : (FDT_b),...,$ End, where $\varphi(FOOD) \in FCS$, $\varphi(FOP_i) \in FAS$, $\varphi(FDT_b) \in FDS$.

Rule 13: A fuzzy complex class axiom $Class (FOO\ \text{partial restriction} (FOP_i \ \text{allValuesFrom} (FDT_b) \text{cardinality(1)))) \text{restriction} (FW_1 \ \text{allValuesFrom} (FOOD_{OI} \text{Cardinality(1)}))$, $Class (FOOD_{OI} \text{partial restriction} (FW_1 \ \text{allValuesFrom} (FOOD_{OI} \text{Cardinality(1)})))$, $Class (FOOD_{OI} \text{partial restriction} (FW_1 \ \text{minCardinality} (m_i))))$, $Class (FOOD_{OI} \text{partial restriction} (FW_1 \ \text{maxCardinality} (n_i))))$, $ObjectProperty (FU_i \ \text{domain} \ (FOOD_{OI}) \ \text{range} \ (FOOD_{OI})))$, $ObjectProperty (FW_1 \ \text{domain} \ (FOOD_{OI}) \ \text{range} \ (FU_i \ \text{inverseOf} \ (FW_l)))$, where $FU_i, FW_i \in FOP_i$ and $FW_1 = \text{invoF} (FU_i)$, $FU_i$ denote inverse properties of $FU_i$, respectively, $i \in [1, 2]$ is mapped into fuzzy class declaration of FOOD model

Class $\varphi(FOOD)$ type is $Record \varphi(FOP_i) : \text{Set-of} \varphi(FOOD)/\eta_1([m_i, n_1), (n_2, n_2)]$, where $FT \rightarrow \text{Set-of} \varphi(FOOD)/\eta_1([m_i, n_1), (n_2, n_2)]$, where $\varphi(FOOD), \varphi(FOP_i) \in FCS$, $\varphi(FOOD) \in FAS$, $\eta \in [0, 1]$.

Rule 15: Fuzzy complex class axiom $Class (FOO\ \text{partial restriction} (F \ \text{allValuesFrom}(R) \ \text{maxCardinality(1))))$ 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 $FC_{(PL...PM)} \in FOOD_C$, m fuzzy data range identifiers $P_1,...,P_m \in FOP_i$ and a fuzzy data range identifier $R \in FDT_b$ can be mapped into method of FOOD model

$\varphi(f(P_1,...,P_m)) : R$, the method $\varphi(f)$ of parameter is $P_1,...,P_m$, $\varphi(R) \in FDS$.

Rule 17: Fuzzy ontology datatype property $DatatypeProperty(FOP_i, \text{domain}(FOOD), \text{range}(FDT_b))$ is mapped into fuzzy class declaration of FOOD model

Class $\varphi(FOOD)$ type is $Record \varphi(FOP_i) : (FDT_b) \text{End}$, where $\varphi(FC) \in FCS$, $\varphi(FOP_i) \in FAS$, $\varphi(FDT_b) \in FDS$.

Rule 18: Fuzzy ontology object property $ObjectProperty (FOP_i \ hasop FOP_i) \ (FOOD) \ (FOP_i) \ (FC)$ is mapped into fuzzy class declaration of FOOD model

Object $\varphi(FOP_i)$ belong to $\varphi(FC)$ has-value $\varphi(FOP_i)$ : $\varphi(FC)$ End, here $\varphi(FC)$, $\varphi(FOOD) \in FCS$, $\varphi(FOP_i), \varphi(FOP_i) \in FAS$.

Rule 19: Fuzzy ontology axioms $Class (FC_{(PL...PM)} \text{partial restriction} (R_1 \ \text{someValuesFrom} (owl:Thing) \text{Cardinality(1)))) \ \text{restriction} (R_m \ \text{someValuesFrom} (owl:Thing) \text{Cardinality(1))))$, $Class (FC_{(PL...PM)} \text{partial restriction} (R_1 \ \text{someValuesFrom} (owl:Thing) \text{Cardinality(1))))$.
(r₁ allValuesFrom (P₁)) ... restriction (rₙ allValuesFrom (Pₙ))); Class (FC₀ partial restriction (inverseOf(r₁) allValuesFrom(unionOf(complementOf(FC₁ᵣ₁,...,ₙ₋₁) restriction (rₙ₊₁ allValuesFrom(R₁))))), is mapped into fuzzy class declaration of FOOD model.

Class \( \phi(FC₀) \) type is \( \text{Record} \ \phi(f(P₁,...,Pₙ)) : R \ \text{End} \), here, \( \phi(f(P₁,...,Pₙ)) \) is a method with \( m \) parameters \( P₁,...,Pₙ \), \( \phi(FC₀) \in FC_{FS}, R \in \{r₁,...,rₙ\} \in FA_{FS} \).

**Rule 20:** Fuzzy individual axiom SameIndividual (FI₀₁...FI₀ₙ) or DifferentIndividuals(FI₀₁...FI₀ₙ) is mapped into fuzzy objects of FOOD model

\( n \) objects \( \phi(FI₀) \) are equivalent or different, here \( \phi(FI₀) \in FO_{FS}, i \in \{1,...,n\} \).

**Rule 21:** With a membership fuzzy individual Individual (FI₀ type (FC₀) \( \approx \mu \)) is mapped into fuzzy objects of FOOD model

Objects \( \phi(FI₀) \) belong to \( \phi(FI₀)/\mu \ \text{End} \), where \( \mu \in \{0,1\} \), \( \phi(FI₀) \in FO_{FS}, \phi(FC₀) \in FC_{FS} \).

### 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 \( O₁ \) and its transformed FOOD model \( O₂ \), there exist two mappings \( \delta \) from fuzzy OWL 2 ontologies structure to models \( \phi(O₂) \), and \( \zeta \) from models \( \phi(O₂) \) to fuzzy OWL 2 ontology structure, such that:

- For each fuzzy OWL 2 ontology model \( FI \) conforming to \( O₂ \), \( \delta(FI) \) is a FOOD model \( \phi(O₂) \).
- For each database state \( FI \) of \( \phi(O₂) \), \( \zeta(FI) \) is a fuzzy OWL 2 ontology model conforming to \( O₂ \).

**Proof.** The following first proves the first part of Theorem 1. Let \( FI = (ΔFI, \phi(FI)) \) be a fuzzy interpretation of fuzzy OWL 2 ontology \( O₂ \), and \( o \in ΔFI \) be an ontology instance, an instance model \( \delta(o) \) is conforming to the FOOD \( \phi(O₂) \).

Given a a fuzzy ontology model \( FI \), each symbol \( X \in FOPO \cup FDP0 \cup FC₀ \cup FDT₀ \cup FP₀ \), a fuzzy database state \( \delta(FI) \) of \( \phi(O₂) \) can be defined as follows:

- The domain elements \( Δ^{(FI)} \) of a database state \( \delta(FI) \) of \( \phi(O₂) \) are constituted by the values of the fuzzy OWL 2 ontology semantic interpretation \( FI \).
- The fuzzy database state \( FI \) of \( \phi(O₂) \) in Section A is defined as follows:
  - \( (\phi(X))^{(FI)} = X^{FI} \), where \( X \in \phi(FC₀) \);
  - For each fuzzy class declaration \( \phi(FC₀) \), \( \{< FP₀, F₁ ≥ Δ^{(FI)} × Δ^{(FI)} | FP₀ \in FC₀ \cap d₁ \in FDT₀ \} \), where \( i \in \{1,...,k\} \), we have Class \( \phi(FC₀) \) type-is \( \text{Record} \ \phi(FDP₀₁) : \phi(FDT₀₁),... \), \( \phi(FDP₀ₙ) : \phi(FDT₀ₙ) \ \text{End} \);
  - For each fuzzy class declaration \( \phi(FC₀) \), \( \{< r, FPO_0 ≥ Δ^{(FI)} × Δ^{(FI)} | r \in FC₀ \land FPO_0 \in FC₀ \cap FDT₀ \} \), \( j \in \{1,...,2\} \), we have Class \( \phi(FC₀) \) type-is \( \text{Record} \ \phi(FPO₀) : \text{Set-of} \ \phi(FC₀) \ \text{End} \).

Further, we prove \( \delta(FI) \) is a model of \( \phi(O₂) \), i.e., \( \phi(FI) \) satisfies the definition of \( \phi(O₂) \) in Definition 2.

Note that, the semantics of \( \phi(O₂) \) models can be partitioned into several main cases:

- For a fuzzy OWL 2 ontology interpretation \( FI \). If there are \( \text{FOP₀}^{(FI)} ∈ Δ^{(FI)} × Δ^{(FI)} \) and FOOD fuzzy class \( \phi(FC₀) \) such that \( \phi(FC₀) \) is a \( \phi(FC₀^{(FI)})/β \), then \( \phi(FC₀^{(FI)})(FOP₀) ≤ \phi(FC₀^{(FI)})(FOP₀) \), i.e., \( FC₀ \subseteq FC₀^{(FI)} \). That is, \( \delta(FI) \) satisfies the corresponding fuzzy semantic of FOOD model in Definition 2.

- For a fuzzy OWL 2 ontologies class \( FC₀ \) such that DisjointUnion(\( FC₀ \)) \( FC₀ \subseteq FC₀\rightarrow FC₀..FC₀ \). According to Definition 3, if \( FI \) is a fuzzy interpretation, we have \( FC₀^{(FI)} = FC₀ \cup ... \cup FC₀^{(FI)} \cap FC₀ \rightarrow FC₀ \rightarrow FC₀ \cap FC₀ \rightarrow FC₀ \rightarrow FC₀ \rightarrow FC₀ \). where \( i, j \in \{1,...,q\} \) and \( i \neq j \). By definition of \( \delta(FI) \) above, it follows \( \phi(FC₀^{(FI)}) = \phi(FC₀^{(FI)})(FOP₀) \cup ... \cup \phi(FC₀^{(FI)})(FOP₀) \) and \( \phi(FC₀^{(FI)})(FOP₀) \subseteq \phi(FC₀^{(FI)})(FOP₀) \), such as Class \( \phi(FC₀) \) type-is \( \text{Union} \ \phi(FC₀₁),... \), \( \phi(FC₀) \) (disjoint, complete) \text{End} \).

That is, \( \delta(FI) \) satisfies the corresponding fuzzy semantic of FOOD model in Definition 2.

- For a fuzzy OWL 2 ontology class \( FC₀ \) such that \( FC₀^{(FI)} = Δ^{(FI)} × Δ^{(FI)} \) and fuzzy class of FOOD model such that Class \( \phi(FC₀) \) type-is \( \text{Record} \ \phi(FDP₀₁) : \phi(FDT₀₁),... \), \( \phi(FDP₀ₙ) : \phi(FDT₀ₙ) \ \text{End} \), where \( \phi(FC₀) \in FC \), \( \phi(FDP₀) \in FA, \phi(FDT₀₁) \in FV, i \in \{1,...,k\} \). For an instance \( FI \in \{FDP₀₁ : FDT₀₁,..., FDP₀ₙ : FDT₀ₙ\} \), according to Definition 3, if \( FI \) is a fuzzy interpretation, then \( \text{FOP₀}^{(FI)} = Δ^{(FI)} × Δ^{(FI)} \). By definition of \( \delta(FI) \) above, there is exactly one element \( d₁ \in FDT₀₁ = \phi(FDT₀₁)^{(FI)} \) such that \( (FC₀, d₁) \in \phi(FC₀^{(FI)}) \), i.e.,
\[\varphi(F_{O})^{(\delta)} \leq \bigcap_{i=1}^{k} \{FC_{O}[\forall d_{i}, < FC_{O}, d_{i} \geq (\varphi(F_{A_{i}}))^{(\delta)} \rightarrow d_{i} \in (\varphi(F_{D_{i}}))^{(\delta)} \}, \#d_{i} < FO, d_{i} \geq \varphi(FDT_{O_{i}})^{(\delta)} = 1\}. \] That is, \(\delta(F)\) 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 \(\varphi(FC_{O_{i}})\) type-is Record \(\varphi(FOP_{O_{i}})\) : \(\subseteq \rightarrow \varphi(FC_{O_{i}})/\cap \{m_{1}, m_{2}, \ldots, n_{3}\}\), where \(\varphi(FC_{O_{i}}) \in FC, \varphi(FOP_{O_{i}}) \in FA, i \in \{1, 2\}\). For an instance \(r \in \varphi(FOP_{O_{i}})^{(\delta)}\), it follows \(r \in \{FC_{O_{1}}, \ldots, FC_{O_{k}}\}\). By definition of \(\delta(F)\) above, there is exactly one example \(FI_{O_{i}} \in FC_{O_{i}}\) such that \((r, FI_{O_{i}}) \in \varphi(FC_{O_{i}})^{(\delta)}\), i.e., \(\varphi(FP_{C}^{(\delta)}) \subseteq \bigcap_{i=1}^{k} \{r \in FC_{O_{i}} \mid FI_{O_{i}} \geq \varphi(FC_{O_{i}})^{(\delta)} \rightarrow FI_{O_{i}} \in \varphi(FC_{O_{i}})^{(\delta)} \land \#FI_{O_{i}} \leq \varphi(FI_{O_{i}})^{(\delta)}\} = 1\). Moreover, according to the semantics of cardinality constraints on associations, we have \(\text{card}_{\text{max}}\{FC_{O_{i}}, FP_{C}, FOP_{O_{i}}\} \leq \#\{r \in FPC_{i} \mid FP_{C}[FI_{i}] = FI_{O_{i}}\} = \text{card}_{\text{max}}\{FC_{O_{i}}, FP_{C}, FOP_{O_{i}}\}\), which denotes the minimum and maximum times of an object instance of a fuzzy class participating in an association. Further, by definition of \(\delta(F)\), it follows \(\varphi(FC_{O_{i}})^{(\delta)} \subseteq \{FI_{O_{i}} \mid \text{card}_{\text{min}}\{FC_{O_{i}}, FP_{C}, FOP_{O_{i}}\} \leq \#\{r \in FPC_{i} \mid < r, FI_{O_{i}} \geq \varphi(FP_{C}^{(\delta)})\} \leq \text{card}_{\text{max}}(FC_{O_{i}}, FP_{C}, FOP_{O_{i}})\}.\n
In addition, we know that \(FW_{1} = \text{invof} \varphi(FU_{1})\) and \(FW_{2} = \text{invof} \varphi(FU_{2})\), are the inverse object property identifiers of \(FU_{1}\) and \(FU_{2}\), respectively, and thus we have \(\varphi(FW_{j})^{(\delta)} = \{< FI_{O_{j}}, r \geq \varphi(FI_{O_{j}}) \mid FI_{O_{j}} \in FC_{O_{j}}\} \land r \in \varphi(FP_{C}^{(\delta)})\}.\n
That is, \(\delta(F)\) satisfies the corresponding semantics of this case in Definition 2.

It is shown that the translation \(\text{FOOD}_{FS} = \varphi(O_{F})\) is semantic preservation since that for each fuzzy interpretation \(I_{O}\) of fuzzy OWL 2 ontology, there is a mapping \(\delta : I_{O} \rightarrow I\) so that \(I = \delta(I_{O})\) 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.

![Figure 2](image)

**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


