Description Logic based Knowledge Merging for Concrete and Fuzzy Domain Ontology

Sri Krishna Kumar* and J. A. Harding*†

*Wolfson School of Mechanical Engineering, Loughborough University, UK

†Communicating Author: J.A.Harding@lboro.ac.uk

Abstract:

Enterprises, especially virtual enterprises (VEs) are nowadays getting more knowledge intensive and adopting efficient Knowledge management (KM) systems to boost their competitiveness. The major challenge for KM for VEs is to acquire, extract and integrate new knowledge with the existing source. Ontologies have been proved to be one of the best tools for representing knowledge with class, role and other characteristics. It is imperative to accommodate the new knowledge in the current ontologies with logical consistencies as it is tedious and costly to construct new ontologies every time after acquiring new knowledge. This paper introduces a mechanism and a process to integrate new knowledge in to the current system (ontology). Separate methods have been adopted for fuzzy and concrete domain ontologies. The process starts by finding the semantic and structural similarities between the concepts using Wordnet and Description logic (DL). DL-based reasoning is used next to determine the position and relationships between the incoming and existing knowledge. The experimental results provided show the efficacy of the proposed Method.

Keywords: *Description Logic, Fuzzy Logic, Fuzzy-DL, Knowledge Merging, Knowledge Management, Ontology*

INTRODUCTION:

In the current era of globalization industries are facing stiff competition through shorter product life-cycles, volatile markets and swift technological advancement. Under these circumstances, enterprises are collaborating to form large supply chain networks or virtual enterprises (VE). Such networks or VEs allow enterprises to provide advanced and multifaceted products or services to customers whilst focusing on their core-competencies and collaborating for other complementary aspects to remain competitive in the market. However, managing and operating a successful VE can be more complex than managing the individual member enterprises and strong communication, cooperation & collaboration in-line with interoperability is required between the member enterprises.

A successful VE needs to develop a mechanism for seamless transfer of data, information and knowledge among member enterprises[1]. Information and communication technology (ICT) can help to achieve collaboration in VEs at a technical level[2] whilst ontologies have been proved to be important tools at the semantic level. Generally each individual enterprise builds its own ontology, based on the domain of their operation, to represent the enterprise's knowledge. It is imperative that enterprise ontology should holds two features: 1. Interoperability (current aspect): to be able to collaborate with other enterprises and 2. Maintenance (continuous or on-going aspect): to accommodate new knowledge. Interoperability means that information and knowledge transferred using ontologies need to be understood accurately, i.e. with correct intention and extension[3]. However, as ontologies may be developed independently to suit personnel requirements, it is impossible to avoid heterogeneity in the terminology used for concepts and their relation and mappings between ontologies are required to interrelate different concepts and to achieve interoperability. Many mapping techniques have been adopted and proposed, in the literature, to achieve uniformity and to tackle interoperability of the enterprise ontologies (current aspect)[4-5].

Nowadays knowledge has become one of the most precious resources for any enterprise. However, this knowledge is more valuable if it can be made inferable and deducible. The future success of enterprises is coupled with their knowledge assets so enterprises need to accumulate knowledge (or create knowledge) from information e.g.

by updating their knowledge in the form of ontology. According to Mo and Zhou[6], knowledge is power and its proper management is necessary to preserve valuable content, learn new things, solve problems, consolidate core competency and discover and implement new technologies. Enterprises should be able to maintain their ontologies to accommodate new knowledge to stay competitive and successfully collaborate in VEs not only in the current time but also in the future. For this reason, maintenance of ontology is termed as a continuous or on-going aspect of virtual enterprises.

Ontologies definitely play important roles in knowledge management[7], but the knowledge discovery possess is equally important to identify and accommodate new knowledge within existing ontologies. The discovered knowledge will not be useful unless it is mapped semantically and structurally with the existing ontologies. To merge knowledge correctly, both the syntax and semantics must be considered, in order to:

1. Deduce similar or new concepts
2. Deduce the possibility of merging concepts, i.e. by restructuring an ontology.
3. Achieve logically consistent mappings.

This paper tackles all three of these problems and develops a mechanism for ontology mapping in the same domain i.e. by enhancing the enterprises' knowledge by accommodating new knowledge into an existing ontology. Moreover, this paper tackles the above problem by using the Description Logic (DL) paradigm as enterprises are increasingly using OWL (web ontology language) to store, use and transfer data and knowledge through the web and OWL is based on DL which is a fragment of first order logic (FOL). The proposed approach can be widely applicable in E-Commerce[8], product design[9], product development[10] and medical domain[11], where new information is being gathered with time. Ontology based knowledge merging approach, proposed here, will help in making their knowledge bases coherent. Furthermore, nowadays enterprises are moving from traditional product lifecycle management (PLM) to knowledge based PLM, in which ontologies play a crucial role[12]. This approach can help enterprises in updating their knowledge bases for improved and efficient knowledge based PLM.

The next section reviews in detail the current progress in the area. Section 3 gives the preliminaries about ontology construction methods (DL, fuzzy logic, fuzzy-DL) and

Wordnet used to identify the meaning and relation of the words used for concept creation. Section 4 describes a method for finding similarity between concepts. The process of merging and reconfiguration is dealt with in section 5. The proposed techniques have been implemented and they are demonstrated through an example which is presented in section 6 and section 7 concludes the paper.

LITERATURE SURVEY:

Exhaustive surveys have been carried out on KM[13-14] and its tools[15]. KM in enterprises is mostly tackled at the subjective level and this can be divided in three different stages: 1. Knowledge creation 2. Ontology development for new knowledge and 3. Merging new knowledge in the existing sources.

Knowledge plays a significant role in the organizational performance[16]. Due to the widespread application of different information systems, a large amount of different knowledge is accumulated during collaboration between enterprises. One of the most important factors in knowledge management is knowledge discovery. Proliferation of data has created a completely new and different area of knowledge management[17] requiring the extraction of knowledge from abundant data and the organization and merging of this knowledge with existing knowledge. Existing knowledge supports organizations in creating new knowledge and updating the overall knowledge base[18]. Knowledge discovery includes discovering implicit knowledge from the data, often using Data Mining techniques to extract knowledge from data sources. Exhaustive literature surveys illustrate that Knowledge Management frameworks, Knowledge-based systems (KBS), information and communication technology (ICT), artificial intelligence and expert systems, database technology etc. have all been adopted by enterprises to exploit knowledge in order to solve their current problems and enhance their expertise. A detailed review has been done by Liao[15]. Pollalis and Dimitriou[4] first proposed the different initiatives needed for knowledge creation and then developed the requirements at each stage of the KM-lifecycle.

Ontology based frameworks have been proven to be the ideal tools for knowledge representation as they provide uniform frameworks to identify similarities and differences between different entities in the specific domain[9]. Many researchers have proposed different methodologies for ontology creation from new knowledge. Huang and Diao[19] proposed a methodology for creating a Concept Map based ontology construction method for knowledge integration. This accumulates knowledge in the business processes and rules and constraints are implemented using SWRL (semantic web rule language). However to implement this in the VE scenario, enterprises need to reconstruct their ontology every time they move to a new collaboration. Ling et al.[20] proposed an ontology-based method to build an integrated knowledge base from heterogeneous sources operating in a single domain. Rajsiri et al.[21] developed a knowledge based ontology model for the collaborative business process model. A distributed enterprise system framework for KM is developed by Ho et al.[22] (2004). Pirro et al.[23] developed a framework for creating, managing and sharing knowledge within an organization with a distributed functional system. Mo and Zhou[6] developed tools and methods for managing the intangible knowledge of VE. Ling et al.[20] proposed an ontology based method for knowledge integration in a collaborative environment. They used heterogeneous ontologies to build domain ontology, i.e. by merging them and through inconsistency elimination. Chen et al.[24] used Wordnet and fuzzy formal concept analysis for merging domain ontologies. Raunich and Rahm[25] proposed the ATOM (Automatic Target-driven Ontology Merging) for integration of multiple ontologies. The process was based on the equivalent relation between source and target taxonomy and merging them preserving the target taxonomy. PROMPT[26] uses the class-name similarities and relies on user for specific merge operation whereas, OntoMerge[27] uses the bridge ontology concept for ontology merging.

It has been widely reported that classical ontologies are not appropriate to deal with imprecise and vague knowledge inherent to several real world domains[28]. It is necessary to merge knowledge in an enterprise, not only for concrete domains, but also for fuzzy domains. Recently approaches have been reported for extending and reasoning with ontologies in fuzzy domains[28-30].

It is clear from the literature survey that the 3rd stage of the KM in enterprises, i.e. merging new knowledge in the existing ones has been given little or no attention. This paper, introduces a method to map discovered knowledge with existing knowledge using an ontology and, if needed, reconfiguring the ontology.

THEORY OF ONTOLOGY:

This section describes the description logic (DL), fuzzy logic, fuzzy-DL and Wordnet used in this paper. DL is a decidable fragment of first order logic which acts as a backbone for ontology development. Fuzzy logic and consequently fuzzy-DL deal with the vague knowledge. Wordnet is helpful in finding semantic similarity between words. A detailed description of each of these approaches is given in the following section.

Description Logic (DL):

Description logic (DL) provides a logical construction for knowledge bases (KB) and is comprised of Concepts, Roles and Individuals as basic building blocks. DL has been proved to be most promising for processing, sharing and interpreting knowledge especially using the web. Ontologies play a key role in constructing KBs in a hierarchical manner of concepts and roles in a particular domain.

The formation of a KB in DL starts by defining the atomic concepts and atomic roles. Atomic concepts and roles generally represent the domain specific, self-explainable entities that are not defined using other concepts and roles (for more detail see Baader [31]). Other general concepts and roles are defined using atomic concepts and general concepts, atomic roles and general roles and constructors (like union, intersection, quantifiers etc.). For example the concepts (C) are formed from atomic concepts using top concept (T), bottom concept ($\perp$), negation ($\neg A$), union (C_1 ò C_2), intersection (C_1 ó C_2), existential quantifier ($\exists R.C$), universal quantifier ($\forall R.C$), cardinality restriction ($\geq_n R.C,\ \leq_n R.C$) etc. Similarly Roles (R) are constructed from atomic roles (P), negation ($\neg R$), transitive (R^+), inverse roles (R^-) etc. Concepts and roles, in DL are seen as unary and binary relations such as: C(x) and R(y, z), where x satisfies the concept C and y and z are in relation R.

However, simple DL is less appropriate in cases of imprecise definition of concepts and relations (roles) and therefore fuzzy-DL, which is a mixture of fuzzy logic and DL has been invented.

Integrating KBs does not simply mean joining an existing KB with a new one. Rather it requires a unified representation of entities (Concepts, Roles and Individuals) in the merged KB. Moreover, an integrated KB must contain all the valuable knowledge and must be free from inconsistencies.

Fuzzy Set and Fuzzy logic:

Fuzzy set theory and fuzzy logic[32] are widely adopted for capturing vague knowledge. Unlike the crisp set, where an element is either a member of a set or not, i.e. the binary (0 and 1) relation, a fuzzy set (X) and its members ($x_1, x_2, \ldots$) are related with a membership function, $\mu : X \rightarrow [0,1]$. In other words, an element is a member of the set with a degree between 0 to 1. In a broader sense the crisp set can also be considered as a fuzzy set which takes only the boundary values 0 and 1.

Like DL, fuzzy logic also supports operations like complement, union, intersection, transitivity etc. with the help of strong mathematical principles. A Fuzzy complement (c) is a unary function defined by $c : [0,1] \rightarrow [0,1]$ with interpretation $I(\neg x) = \Theta I(x)$. A Fuzzy complement satisfies the boundary conditions, i.e. $c[0]=1$, $c[1]=0$ and is monotonically increasing, i.e. $x \le y \Rightarrow c(x) \ge c(y)$. There are many complement functions defined in the literature, among them are Lukasiewicz negation: $c(x) = 1 - x$ and Godel complement $c(x) = 1$ if $x > 0$ else $c(x) = 0$.

Fuzzy intersection, termed as t-norm, is defined by the function $t : [0,1] \times [0,1] \rightarrow [0,1]$ with interpretation $I(x \wedge y) = I(x) \otimes I(y)$. Fuzzy intersection satisfies the boundary conditions: $t(x,1) = x$ and $t(x,0) = 0$, monotonicity: $y \ge z \Rightarrow t(x,y) \ge t(x,z)$ and other set theoretic properties like, commutativity: $t(x,y) = t(y,x)$, associativity: $t(x, t(y,z)) = t(t(x,y), z)$. Most widely used t-norm functions are Lukasiewicz t-norm: $t(x,y) = \max(0, x + y - 1)$ and product t-norm: $t(x,y) = x.y$.

Fuzzy union, termed as t-conorm, is a function defined by $u : [0,1] \times [0,1] \rightarrow [0,1]$ with interpretation $I(x \vee y) = I(x) \oplus I(y)$. Similar to fuzzy complement and fuzzy intersection, fuzzy union also satisfies the boundary conditions: $u(x,0) = x$, $u(x,1) = 1$ and monotonicity. It also follows the commutative and associative rules as in case of fuzzy intersection. Most commonly used t-conorm functions are Lucksiewicz: $u(x,y) = \min(1, x + y)$, Godel: $u(x,y) = \max(x,y)$.

One of the most important operations in fuzzy logic relating to classical logic is fuzzy implication. Fuzzy implication is defined by the function: $I:[0,1]\times[0,1]\rightarrow[0,1]$ with interpretation $I(x\rightarrow y)=I(x)\Rightarrow I(y)$. In classical logic implication $A\rightarrow B$ is equivalent to $\neg A\vee B$ or $\max\{t\in\{0,1\}\big|A\wedge t\leq B\}$. In classical logic both are equivalent but their extension in fuzzy logic leads to S-implication and R-implication respectively (ref.). Fuzzy implication functions are Luckasiewicz: $I(x,y)=\min(1,x+y-1)$, Godel: $I(x,y)=y$ if x>y , else $I(x,y)=1$.

Although, there are many functions related to fuzzy logic, Luckasiewicz, Godel, product etc., Bobillo and Straccia[29] showed the benefit of using Luckasiewicz function and this paper uses these functions for fuzzy interpretation.

Fuzzy Description Logic:

Fuzzy DL is an extended version of DL where concepts (unary relation) and roles (binary relations) are extended to fuzzy set and fuzzy binary relations. DL-axioms are also extended into fuzzy set using degree of truth. Similar to DL, fuzzy-DL consists of fuzzy-Tbox and fuzzy- Abox. Fuzzy-Tbox consists of the concepts (C, D) and role names (P, R) along with the general inclusion axioms i.e. $\left\langle (C\subseteq D)\geq\alpha\right\rangle$ which means that concept C is sub-concept of concept D with truth value α where, $\alpha\in[0,1]$. Similarly for roles $\left\langle (P\subseteq R)\geq\beta\right\rangle$ with $\beta\in[0,1]$. The fuzzy A-box consists of the fuzzy assertion of individuals with concept and roles with fuzzy membership value in the form $\lhd\beta$, where $\lhd\in\{\geq,>,\leq,<\}$ and $\beta\in[0,1]$. Like DL, interpretation of fuzzy-DL I_f is a pair $(\Delta^{I_f},\cdot^{I_f})$, where Δ^{I_f} is the fuzzy domain of interpretation and $\cdot^{I_f}$ is interpretation function with the following characteristics:

For individual a, $a\in\Delta^{I_f}$, For concept C, $C^{I}:\Delta^{I_f}\rightarrow[0,1]$, For role R, $R^{I_f}:\Delta^{I_f}\times\Delta^{I_f}\rightarrow[0,1]$. Comparison between DL and fuzzy-DL interpretation with basic concepts, roles and constructors is been shown in the table 1. (For more detailed explanations read Stoilos *et. al.,*[28]).

Concepts / Roles	DL Interpretation	Fuzzy-DL Interpretation
Atomic Concept : A	$A^I \subseteq \Delta^I$	$A^I : \Delta^I \rightarrow [0,1]$
Top concept: T	Δ^I	$T^I(a) = 1$
Bottom concept: $\perp$	ϕ	$\perp^I(a) = 0$
Concept conjunction: C ⊓ D	$C^I \bigcap D^I$	$(C \text{ ó } D)^I(x) = C^I(x) \otimes D^I(x)$
Concept disjunction: C ⊔ D	$C^I \bigcup D^I$	$(C \text{ ò } D)^I(x) = C^I(x) \oplus D^I(x)$
Concept negation: $\neg C$	$\Delta^I \setminus C^I$	$!\ C^I(x)$
Atomic role: R	$R^I \subseteq \Delta^I \times \Delta^I$	$R^I : \Delta^I \times \Delta^I \rightarrow [0,1]$
Inverse Role: R^-	$\{(y,x) \in \Delta^I \times \Delta^I \vert (x,y) \in R^I$	$R^{-I}(y^I, x^I) \equiv R^I(x^I, y^I)$ $: \Delta^I \times \Delta^I \rightarrow [0,1]$
Concept assertion: $a:C$	$a^I \in C^I$	$C^I(a^I) \rightarrow [0,1]$
Concept Subsumption: C ⊑ D	$C^I \sqsubseteq D^I$	$\inf_{x \in \Delta^I} \{C^I(x) \Rightarrow D^I(x)\}$
Role Assertion: (a,b):R	$(a^I, b^I) \in R^I$	$R^I(a^I, b^I) \rightarrow [0,1]$

Table 1: Comparison between DL and Fuzzy-DL

Wordnet:

Wordnet (wordnet API)[33], created by Princeton university, is a dictionary of semantically similar English words, arranged structurally. Words are characterized based on the parts of speech- noun, verb, adjective etc. and linked together and categorized as synonyms, hyponyms etc.

ONTOLOGY SIMILARITY:

An ontology is the explicit specification of shared conceptualization[34] (Gruber, 1993). In simple words, an ontology is a domain specific knowledge representation specified in terms of concepts and their relations. An ontology can be represented as $O := \{C, R, A\}$ where C is the set of concepts, R is the set of roles and A is the set of axioms. Similarly a fuzzy ontology can be represented as $O_F := \{C, R^F, A^F\}$ where R^F and A^F additionally associate fuzzy membership values between [0, 1].

In this paper, an ontology *O1* is defined as the existing knowledge and *O2* as the new knowledge. Let C_i^1 and C_j^2 be the *i*th and *j*th concepts of two ontologies *O1* and *O2* respectively such that $C_i^1 \in O_1$ and $C_j^2 \in O_2$. All other notations used in this section are as follows:

$SynC_i^1$: Synonym set of C_i^1 , $SynC_j^2$: Synonym set of C_j^2

$HyperC_i^1$: Hypernym set of C_i^1 , $HyperC_j^2$: Hypernym set of C_j^2

$HypoC_i^1$: Hyponym set of C_i^1 , $HypoC_j^2$: Hyponym set of C_j^2

SC_i^1 : Set of super concepts of C_i^1, sC_i^1 : Set of sub concepts of C_i^1,

SC_j^2 : Set of super concepts of C_j^2, sC_j^2 : Set of sub concepts of C_j^2

The methodology adopted in this paper for knowledge merging, i.e. ontology mapping and ontology reconfiguration, is based on two steps. In the first step, a similarity matrix or index is calculated. In the second step merging and reconfiguration are carried out based on logical arguments to get a consistent final ontology.

For calculating the similarity matrix, two parameters have been taken into account: semantic similarity and structural similarity. Semantic similarity determines how closely two concept names are linguistically associated, whereas structural similarity determines the hierarchical relationship (equivalent, super and sub) between new concepts and concepts of existing ontology. The next section illustrates the process of calculating semantic similarity, structural similarity and hence the similarity matrix.

Semantic Similarity:

The Semantic similarity between concepts is defined by the function $\psi : \{C_i^1, C_j^2\} \rightarrow [0,1]$, where $C_i^1 \in O_1 \,\&\, C_j^2 \in O_2$. In language two words can be related to each other in various ways e.g. same root, antonyms etc. However, the synonyms, hypernyms and hyponyms of two words imitate the equivalent, super and sub relationship of ontological concepts, and therefore only synonym, hypernym and hyponym relations have been taken into account for calculating the semantic similarity. As concept names in general do not contain any fuzziness, rather instances and relations are fuzzy, this paper does not take fuzziness in the semantic similarity into account. The procedure for semantic similarity calculation is as follows:

Synonym: $\psi_1(C_i^1, C_j^2) = 1$, if $\exists t_1, t_2 \mid t_1 = t_2 \; \& \; t_1 \in SynC_i^1 \& \; t_2 \in SynC_j^2$

$\psi_1(C_i^1, C_j^2) = 0$, otherwise

Hypernym: $\psi_2(C_i^1, C_j^2) = \beta_2, \{\beta_2 \in [0,1]\}$, if $\exists t_1, t_2 \mid t_1 = t_2 \; \& \; t_1 \in HyperC_i^1 \; \& \; t_2 \in HyperC_j^2$

$\psi_2(C_i^1, C_j^2) = 0$, otherwise

Hyponym: $\psi_3(C_i^1, C_j^2) = \beta_3, \{\beta_3 \in [0,1]\}$, if $\exists t_1, t_2 \mid t_1 = t_2 \; \& \; t_1 \in HypoC_i^1 \; \& \; t_2 \in HypoC_j^2$

$\psi_3(C_i^1, C_j^2) = 0$, otherwise

Here, β_2 and β_3 are weights given to Hypernym and Hyponym relation. The final semantic similarity index will be the maximum of all, i.e. $\psi(C_i^1, C_j^2) = \max\{\psi_1(C_i^1, C_j^2), \psi_2(C_i^1, C_j^2), \psi_3(C_i^1, C_j^2)\}$

Structural Similarity:

The structural similarity between the concepts of two ontologies is the measurement of their association in terms of equivalence, super and sub relationships. The structural similarity is measured at three levels (equivalence, super and sub relation). As relationships between the concepts can be fuzzy, this paper considers both the concrete domain (instance 1) and the fuzzy domain (instance 2) for structural similarity calculation. The procedure is explained next.

Equivalence relation Similarity (ER):

Instance 1(Concrete domain): In the concrete domain, an equivalence relation between two concepts is closely associated with the equivalence between their super and sub concepts respectively. Mathematically, the equivalence relation between concept C_i^1 and C_j^2 can be given as:

$$ER(C_i^1, C_j^2) = 0.5\left\{\frac{Sim(SC_i^1, SC_j^2)}{\left|SC_i^1 \cup SC_j^2\right|}\right\}^2 + 0.5\left\{\frac{Sim(sC_i^1, sC_j^2)}{\left|sC_i^1 \cup sC_j^2\right|}\right\}^2$$

Here, function $Sim(.,.)$ determines the number of similar elements in the two sets and $|A|$ is the cardinality of the set A. The first part of the equation calculates the similarity in terms of super concepts and the second part calculates the similarity in terms of sub concepts. Squaring the function gives more weightage to the structurally equivalent concepts as the ratio will never exceed the value 1. Equal weightage has been given to both the parts as two concepts are equivalent if their super and sub concepts are equivalent respectively.

Case 2(Fuzzy domain): In the fuzzy domain the equivalence concept relation can be given as:

$$ER(C_i^1, C_j^2) = 0.5\frac{\sqrt{\sum_{SC_i^2}\sum_{SC_j^2}\{\mu Sim(SC_i^1, SC_j^2)\}^2}}{\{SC_i^1 \cup SC_j^2\}} + 0.5\frac{\sqrt{\sum_{sC_i^2}\sum_{sC_j^2}\{\mu Sim(sC_i^1, sC_j^2)\}^2}}{\{sC_i^1 \cup sC_j^2\}}$$

Where, $\mu Sim(.,.)$ determines the fuzzy value or truth value of the equivalence relation of the two input concepts. Unlike concrete domain square root of the function has been taken in the fuzzy case as the fuzzy value will not exceed the value 1 which will give more weightage to the structurally similar concepts.

<u>Super Relation Similarity (SR):</u>

A concept is said to be in a super concept relationship with another concept when its sub concepts match with the super concepts of the other. In this paper super relation similarity between the two concepts has been identified for both the cases as follows:

Instance 1 (Concrete domain): In the concrete domain, super relation similarity between two concepts is the similarity of their super and sub concepts. Mathematically this can be stated as:

$$SupR(C_i^1, C_j^2) = \left\{ \frac{Sim(\{sC_i^1 \cup C_i^1\}, \{SC_j^2 \cup C_j^2\})}{\left| sC_i^1 \cup SC_j^2 \cup C_i^1 \cup C_j^2 \right|} \right\}^2$$

The super relation function (*SupR*) includes both concepts (C_i^1 and C_j^2) in the denominator to get a closer evaluation.

Instance 2 (Fuzzy Domain): For the fuzzy case super relation similarity is

$$SupR(C_i^1, C_j^2) = \frac{\sqrt{\sum\sum\{\mu Sim(\{sC_i^1 \cup C_i^1\}, \{SC_j^2 \cup C_j^2\})\}^2}}{\{sC_i^1 \cup SC_j^2 \cup C_i^1 \cup C_j^2\}}$$

and is a measurement of the equivalent fuzzy value or truth value for the super relation between the two input concepts.

<u>**Sub Relation Similarity (sR):**</u>

In contrast with super relation similarity, a concept is in a sub relationship similarity with another concept if its super concepts match with the sub concepts of the other. For both cases this can be calculated as follows:

Instance 1(Concrete domain): In line with the argument given in the super relation, a concrete domain sub relation can be given as:

$$subR(C_i^1, C_j^2) = \left\{ \frac{Sim(\{SC_i^1 \cup C_i^1\}, \{sC_j^2 \cup C_j^2\})}{\left| SC_i^1 \cup sC_j^2 \cup C_i^1 \cup C_j^2 \right|} \right\}^2$$

Instance 2 (Fuzzy Domain): For fuzzy domain it will be:

$$SubR(C_i^1, C_j^2) = \frac{\sqrt{\sum\sum\{\mu Sim(\{SC_i^1 \cup C_i^1\}, \{sC_j^2 \cup C_j^2\})\}^2}}{\{SC_i^1 \cup sC_j^2 \cup C_i^1 \cup C_j^2\}}$$

Now the overall mapping relation based on semantic and structural similarity can be given as:

1. **Equivalence relation:**

   $\mathcal{E}\Re(C_i^1, C_j^2) = k\psi(C_i^1, C_j^2) + (1-k)ER(C_i^1, C_j^2)$

2. **Super relation:**

   $$\mathcal{S}\Re(C_i^1,C_j^2) = k\psi(C_i^1,C_j^2) + (1-k)Sup(C_i^1,C_j^2)$$

3. **Sub relation:**

   $$\mathbf{s}\Re(C_i^1,C_j^2) = k\psi(C_i^1,C_j^2) + (1-k)Sub(C_i^1,C_j^2)$$

The constant $k \in [0,1]$ is the weight given to the semantic relation. The relational matrix obtained here is used for ontology merging and reconfiguration (explained in the next section). The relational matrix serves two purposes as it not only relates the closeness of two concepts from different ontologies but also explores the kind of relation i.e. equivalence, super and sub. The next section describes the process of ontology merging and reconfiguration.

ONTOLOGY MERGING AND RECONFIGURATION:

The relational matrix obtained in the previous section is used for ontology merging and, if necessary, for ontology reconfiguration. This approach first determines the greatest similarity in terms of equivalence, super and sub relations between the concepts of the ontologies. The next step involves establishing logical consistency, i.e. the formation of a logically consistent merged and reconfigured ontology. This process is different for both the concrete and fuzzy domains. This section begins by explaining the process for the concrete domain and later deals with the fuzzy domain.

Concrete Domain: In the consistency checking part, two concepts of two different ontologies are compared. This process first finds the maximum of {$\mathcal{E}\Re(C_i^1,C_j^2)$, $\mathcal{S}\Re(C_i^1,C_j^2)$, $\mathbf{s}\Re(C_i^1,C_j^2)$ } (see figure 20). In case two or more concepts have the maximum value arbitrary selection is carried out. Separate reasoning is carried out for each equivalence, super and sub relation (as explained next). In case, no consistent relation is derived for the maximum value, then the process selects the next best value and so on until a solution is reached or no relation is found. In case no relation is found it is added as a new concept. The detailed explanation is given next.

Equivalence relation: The easiest case is when the equivalence relation matrix is one (case 1) and the new concept (C_j^2) can be established in the existing ontology as shown in fig (1).

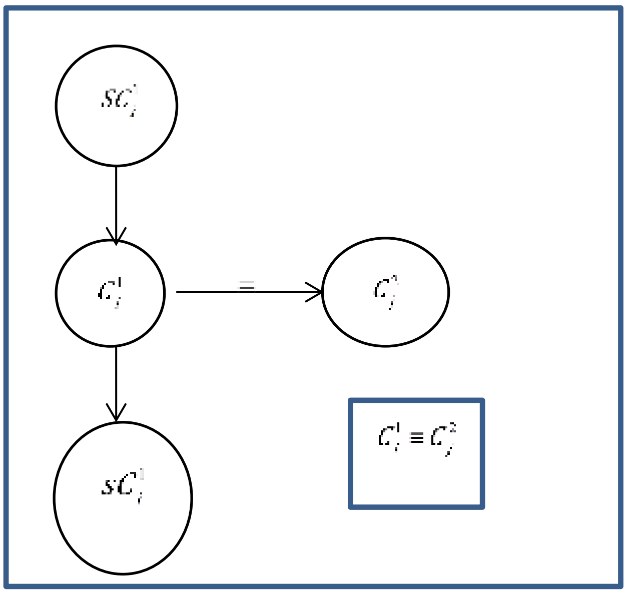


Figure 1: Case 1

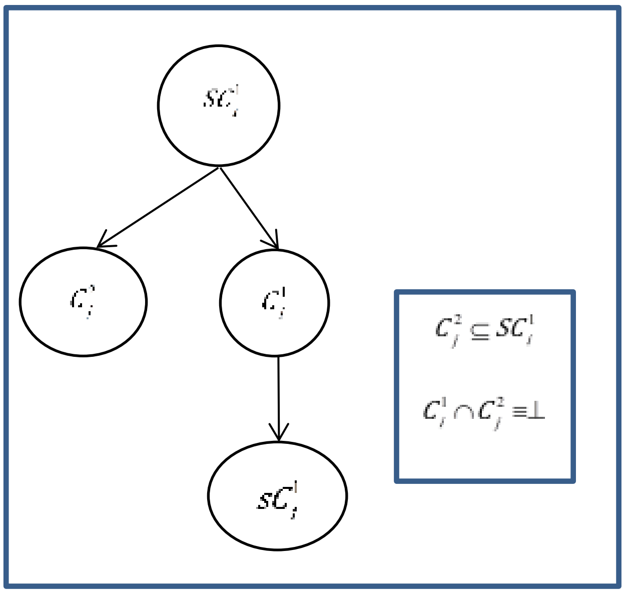
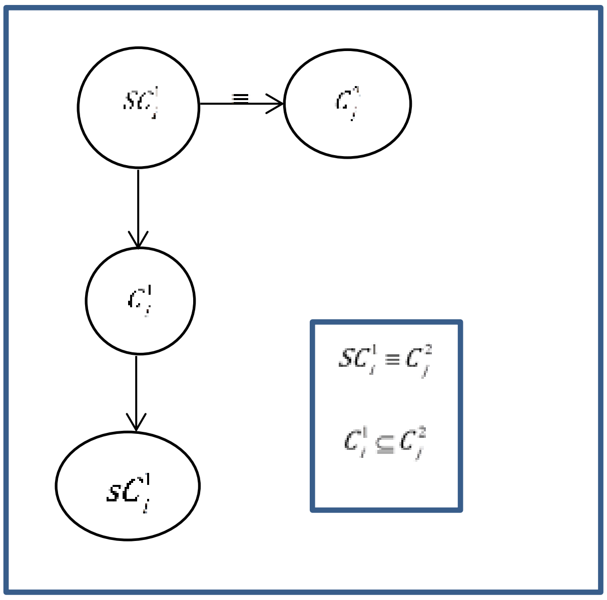


Figure 2: Case 2

Case 2 (fig. 2) is where the equivalence relation matrix is less than one and a possible position for the new concept (C_j^2) is as a sibling of concept (C_i^1). This case arises when C_j^2 is the sub- concept of SC_i^1 but C_i^1 and C_j^2 do not have any common sub-concepts. In this case the equivalence relational matrix will have a greater value than the sub and super relational matrix.

Sub relation: This is where the existing concept is in a sub-concept relation according to relational matrix i.e. $C_i^1 \subseteq C_j^2$. In this case, three positions are possible where the new concept can be merged in the ontology, as shown in the figures 3 to 5.

The first condition (case 3) arises when the new concept (C_j^2) is equivalent to the super concept (SC_i^1) of the compared concept (C_i^1). The second condition (case 4) arises when the new concept (C_j^2) is a super-concept of the compared concept (C_i^1) and is also a sub-concept of the super concept (SC_i^1). This situation arises when a concept in an ontology is further subdivided or refined. The third condition (case 5) arises when the new concept (C_j^2) is super concept of (SC_i^1), as shown in the figure (5).



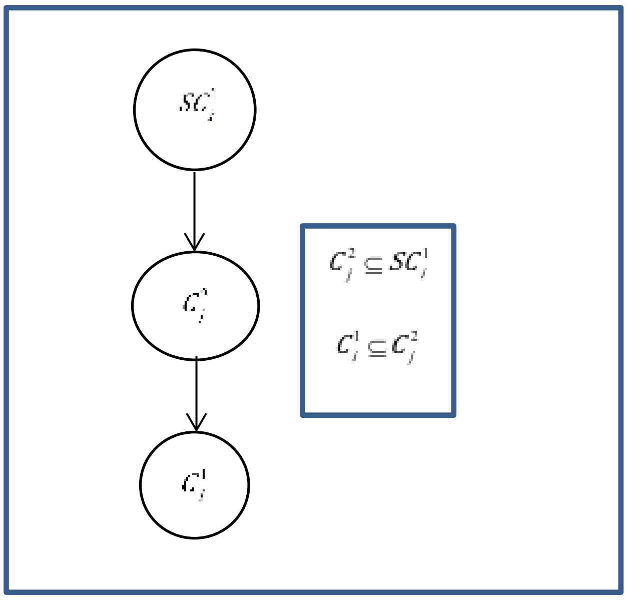


Figure 3: Case 3

Figure 4: Case 4

In this case, the position of C_j^2 is above SC_i^1, but to get the exact place C_j^2 must be compared with SC_i^1 and then the conditions (3) and (4) should be checked again.

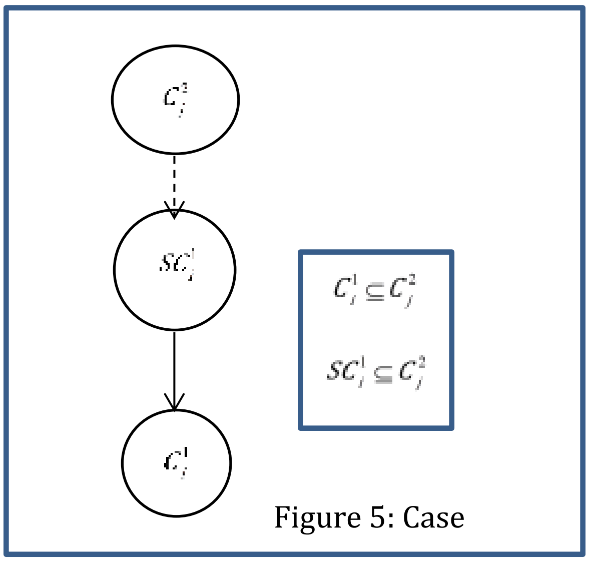


Figure 5: Case

Super Relation: This is where the existing concept is in a super-concept relation according to the relational matrix i.e. $C_j^2 \subseteq C_i^1$. In this case, possible positions where the new concept can be merged in the ontology are shown in figures 6 to 9.

The first condition (case 6) arises when C_j^2 is a sub concept of C_i^1 and a super concept of sC_i^1. The second condition (case 7) is when C_j^2 is a sub concept of C_i^1 and is equivalent to sC_i^1. The third condition (case 8) arises when C_j^2 is a sub concept of C_i^1 and is disjoint with sC_i^1. This scenario describes the condition when a concept is redefined with the addition of new concepts (or new characteristics).

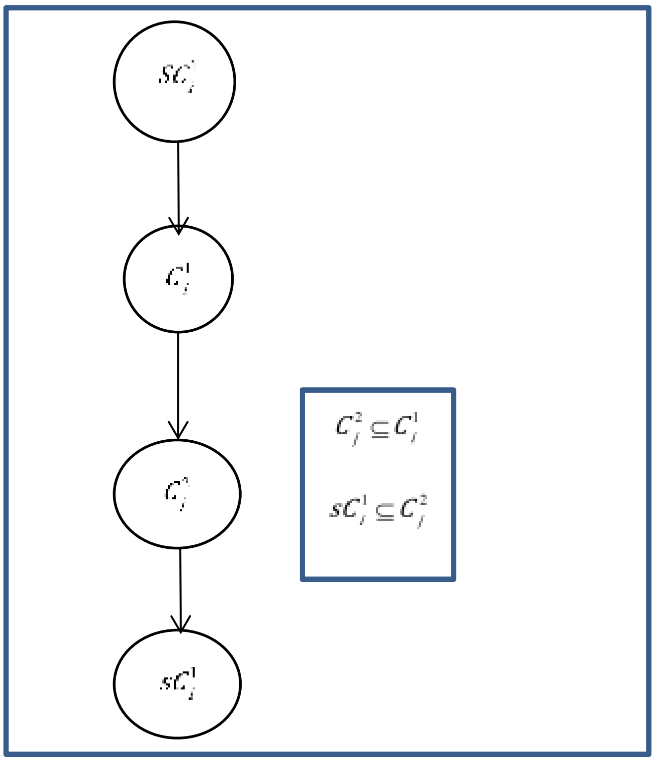


Figure 6: Case 6

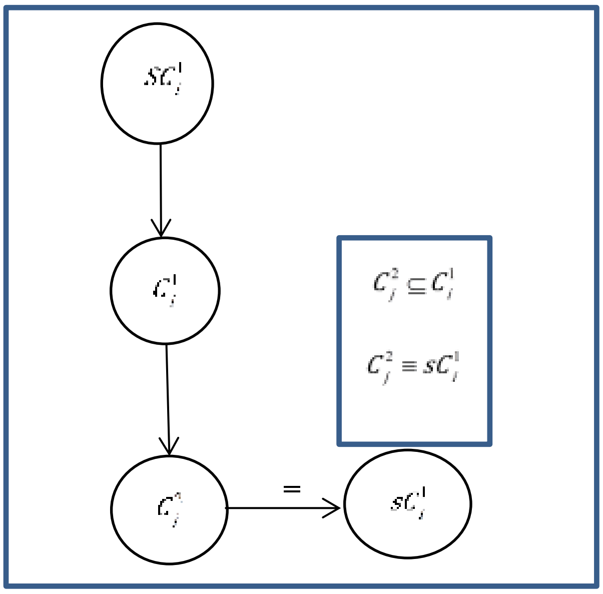
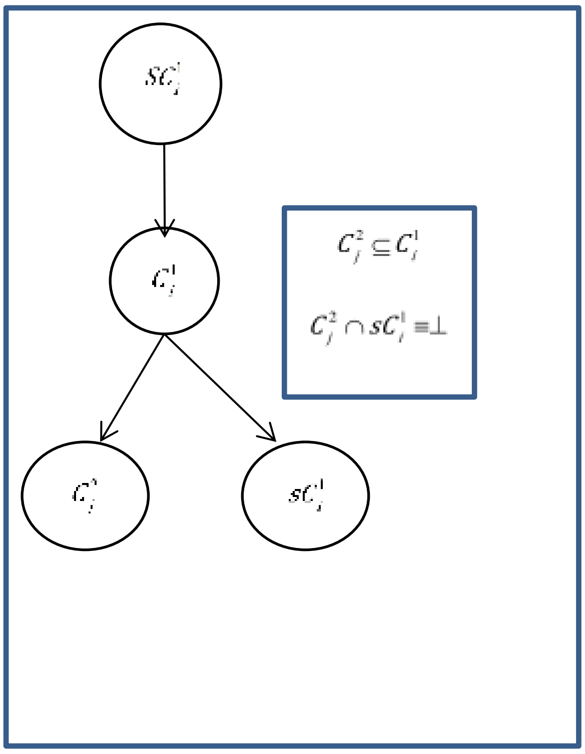


Figure 7: Case 7

The last condition (case 9) describes the situation when C_j^2 is a subclass of sC_i^1 (fig 9). In this condition, to get the exact position of C_j^2, it must be compared with sC_i^1 and further evaluated for conditions 6-8.

Although this approach has considered all possible conditions for equivalence, sub and super relations, it may also be possible that the new concept has no defined position or possibly has no relation with existing concepts (including case 5 and 9). In this scenario merging and reconfiguration is carried out using the super concept of the new concept. Let SC_j^2 be the super concept of C_j^2 and the relational matrix is obtained in the same manner as in the case of C_i^1 and C_j^2. The following conditions can be obtained in line with the previous explanations *(Prefix* 'Super' (Ṡ) *has been used to emphasise that super concept of a new concept is compared to get the relational matrix)*:



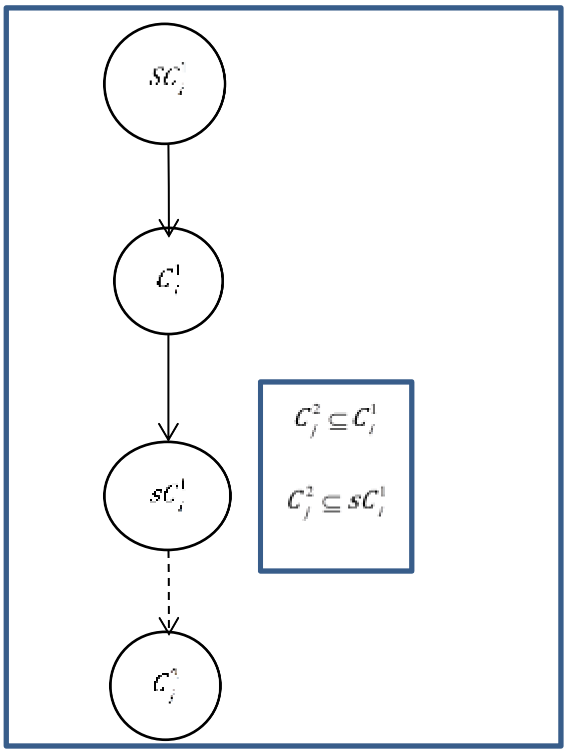
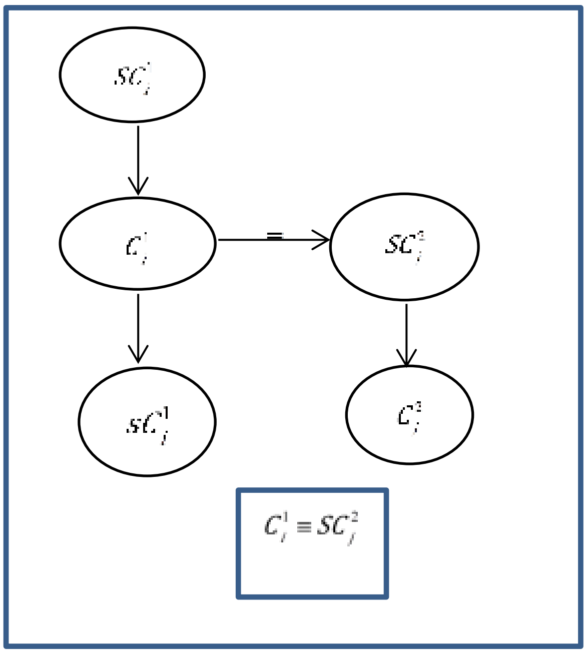


Figure 8: Case 8

Figure 9: Case 9

Super equivalence relation: In a super-equivalence relation, the mapping of a new concept (C_j^2) in terms of its super-concept (SC_j^2) with respect to C_i^1 follows the same procedure as the mapping between C_i^1 and C_j^2. The simplest condition is when SC_j^2 is equivalent to C_i^1 (Condition Ṡ1). This is depicted in figure (10). As no relation is found between C_j^2 and the existing ontology, this is simply added as a sub concept of SC_j^2 in the merged ontology.

Similar to condition 2, condition Ṡ2 arises (fig 11) when a new concept SC_j^2 is added in the ontology as a sub-concept of SC_i^1 and C_j^2 is added in the ontology accordingly.



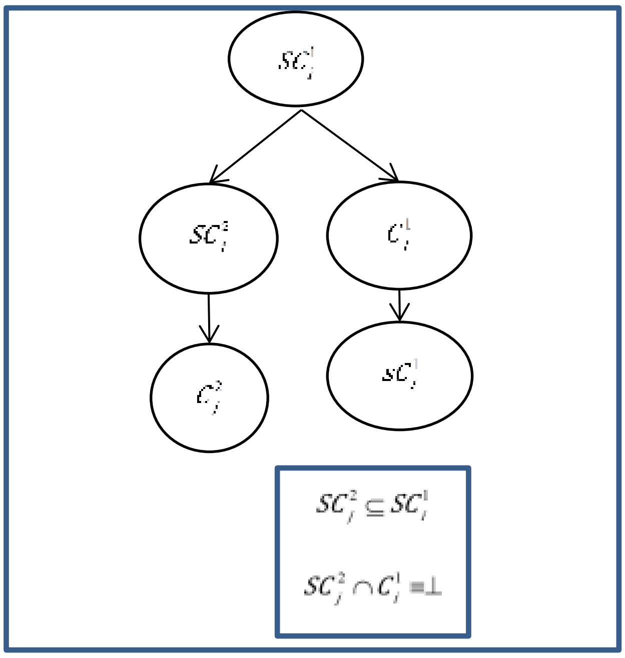
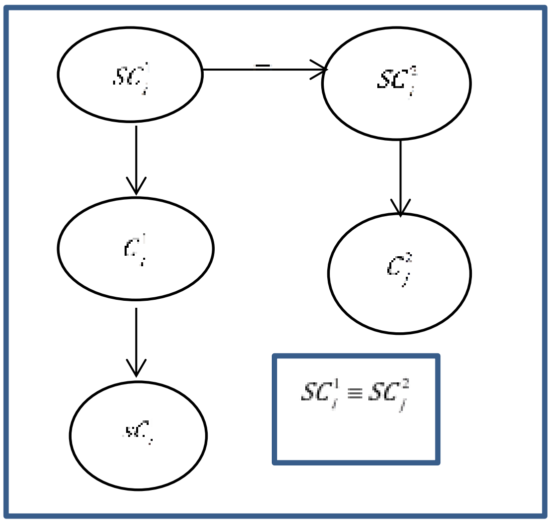


Figure 10 : Case Ṡ1

Figure 11: Case Ṡ2

Super-Sub-relation: Super-Sub-relation mapping is carried out when the relational matrix entails that SC_j^2 is closer to the sub concept of the C_i^1. Condition Ṡ3 (fig 12) and Ṡ4 (fig 13) have same logical base as cases 3 and 4 respectively. Similar to condition 5, in the case of condition Ṡ5 (fig. 14), the super concept of SC_j^2 is checked with C_i^1 for conditions Ṡ3 and Ṡ4.



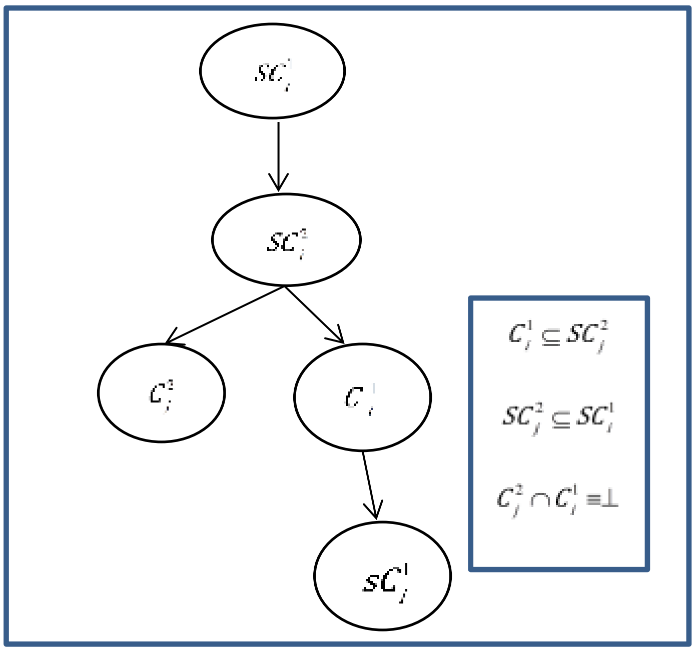


Figure 12: Case Ṡ3 Figure 13: Case Ṡ4

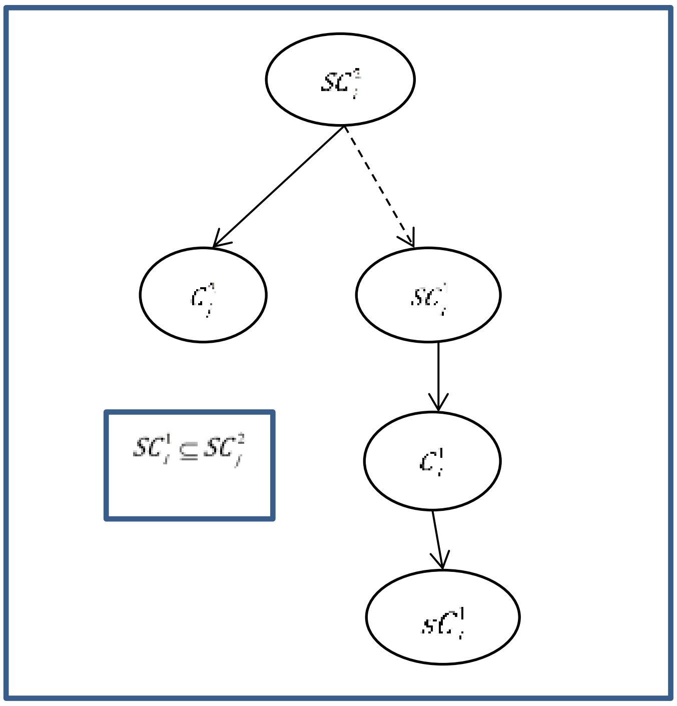


Figure 14: Case Ṡ5

Super-super relation: Super-super relation mapping occurs when the relational matrix intimates that C_i^1 is the super class of SC_j^2. All possible places where SC_j^2 and C_j^2 can fit have been shown in the figures 15-18. Conditions Ṡ6, Ṡ7 and Ṡ8 have similar logical explanations as conditions 6, 7 and 8 respectively.

Similar to condition 9, in condition Ṡ9 the exact position of SC_j^2 cannot be determined. It is compared with sC_i^1 and condition 6,7 and 8 are checked with respect to sC_i^1 and SC_j^2.

If none of the conditions (Case 1 to Case 9 and Case Ṡ1 to Case Ṡ9) are satisfied in this process then it is clear that a new concept needs to be added in the ontology. For this, the process finds the super most concepts, unrelated to the existing ontology, as a new concept and adds its sub-concepts accordingly.

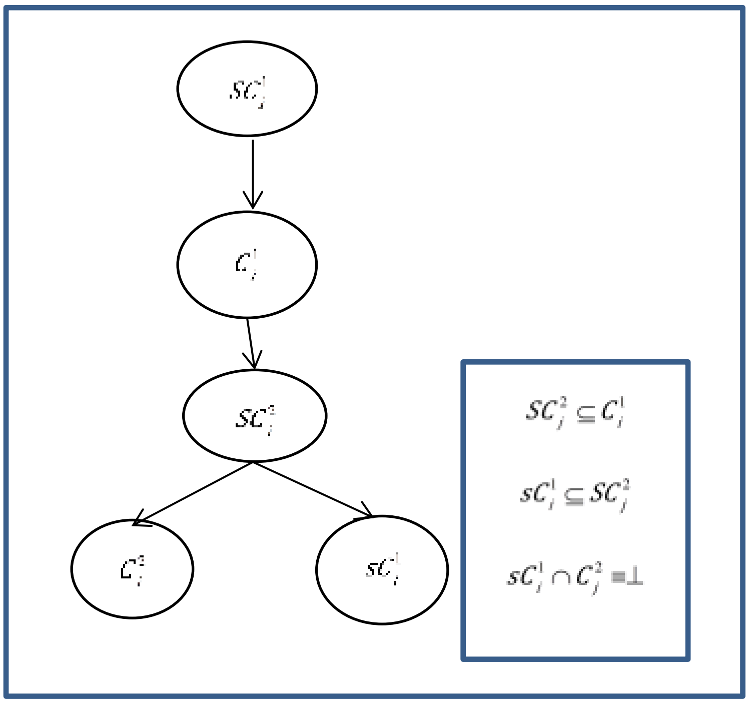


Figure 15: Case Ṡ6

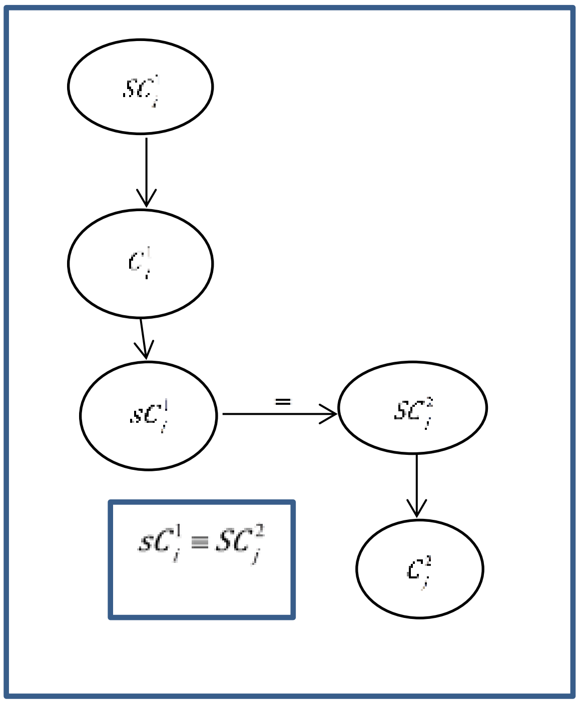


Figure 16: Case Ṡ7

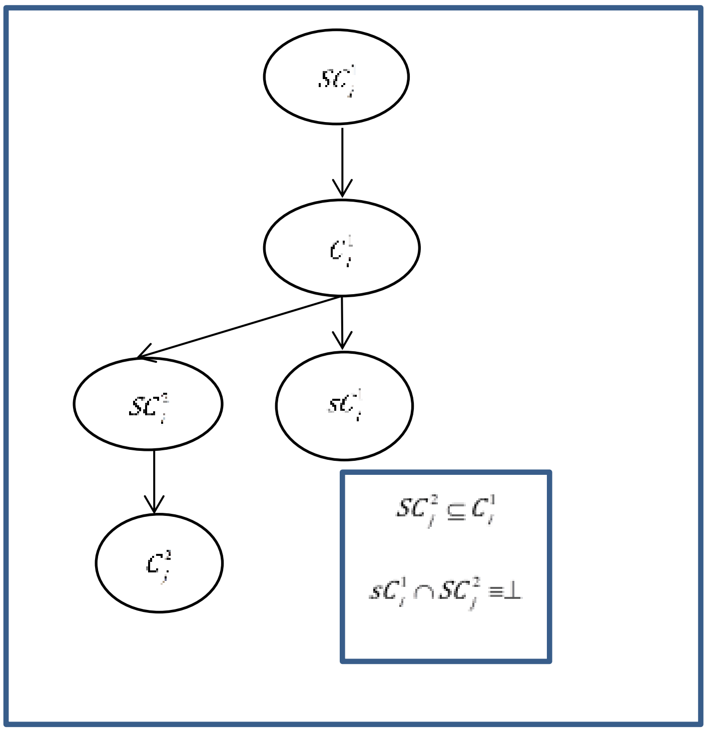


Figure 17: Case Ṡ 8

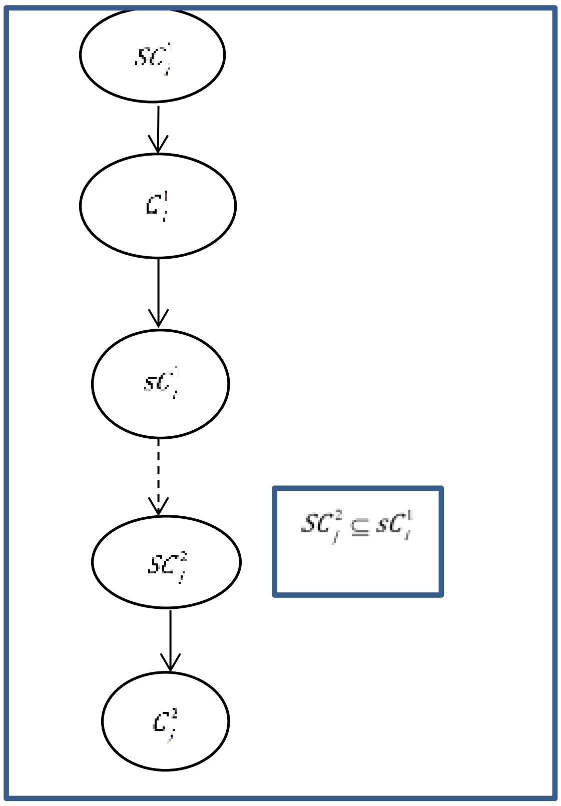


Figure 18: Case Ṡ9

Fuzzy domain:

In fuzzy-DL two concrete concepts or even an assertion of individual in concrete concepts and roles are related with the fuzzy value or truth value as described in the section 3. Unlike the concrete domain, as explained earlier, a fuzzy domain ontology not only requires mapping and reconfiguration of the ontology but also requires a fuzzy value or truth value to be calculated for the merged or reconfigured concept with the concepts of the existing ontology. The first step for fuzzy domain ontology mapping and reconfiguration is similar to the concrete domain (i.e. finding the maximum). The second step is implemented in two stages: the first stage determines the position of the new ontology within the existing one (similar to case 1 to case Ṡ 9) and the second stage recalculates the fuzzy value or truth value for different relations among the concepts.

Considering the case (4) as shown in figure (4), following fuzzy values are available:

$$\text{SubR}(C_i^1, C_j^2) \equiv C_i^1 \sqsubseteq C_j^2 = \mu \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \text{(A.1)}$$

SupR$(SC_i^1, C_i^1) \equiv C_j^2 \sqsubseteq SC_i^1 = \omega$ .... (A.2)

From A.1 and A.2, it is clear that $C_i^1 \sqsubseteq C_j^2 \sqsubseteq SC_i^1$. This step merges or reconfigures the ontologies and introduces the intermediate concept C_j^2. This reconfiguration or introduction of the new concept needs to identify the fuzzy sub-concept value between C_j^2 and SC_i^1. Assuming,

$C_j^2 \sqsubseteq SC_i^1 = \lambda$ .... (A.3)

Now the sub-concept relation $C_i^1 \sqsubseteq SC_i^1$ is the implication of two sub-concept relations $C_i^1 \sqsubseteq C_j^2$ and $C_j^2 \sqsubseteq SC_i^1$, i.e.

$(C_i^1 \sqsubseteq SC_i^1) \equiv (C_i^1 \sqsubseteq C_j^2) \rightarrow (C_j^2 \sqsubseteq SC_i^1)$................................ (A.4)

Using the Lukasiewicz implication function and A.1 – A.4 ,

$\omega = \min(1, 1 - \mu + \lambda)$

The method shown above for truth value recalculation has considered only truth values equal to some constant, but a similar approach can be used in cases where the truth value is greater than ($\geq$) or less than ($\leq$) some constant value. The rest of the ontology mapping and reconfiguration in fuzzy case can be obtained for all cases (case 1 to case Ṡ9) in a similar way.

IMPLEMENTATION METHOD:

The proposed methodology for ontology merging and reconfiguration for both concrete and fuzzy domains can be created in an OWL API such as Protege[35] and its plugin fuzzyDL[36]. Merging and reconfiguration is carried out in Java. The overall implementation method is summarized in figure (19). Jena parser, a Java API is used as the Ontology API to get the concept names without the namespace. Wordnet API[33] (Wordnet) is used to get the synonym, hyponym and hypernym of the concepts for carrying out the word similarity and finally calculating the Lexicon similarity matrix. A Structural similarity matrix is calculated as previously described. Pellet-reasoner[37] is used to find the relationship between concepts (i.e. equivalence, super, sub etc.). As the two ontologies considered here are from the same domain, the assumption that they are built on same base ontology is valid. This assumption has been used for building the Tbox and Abox for reasoners.

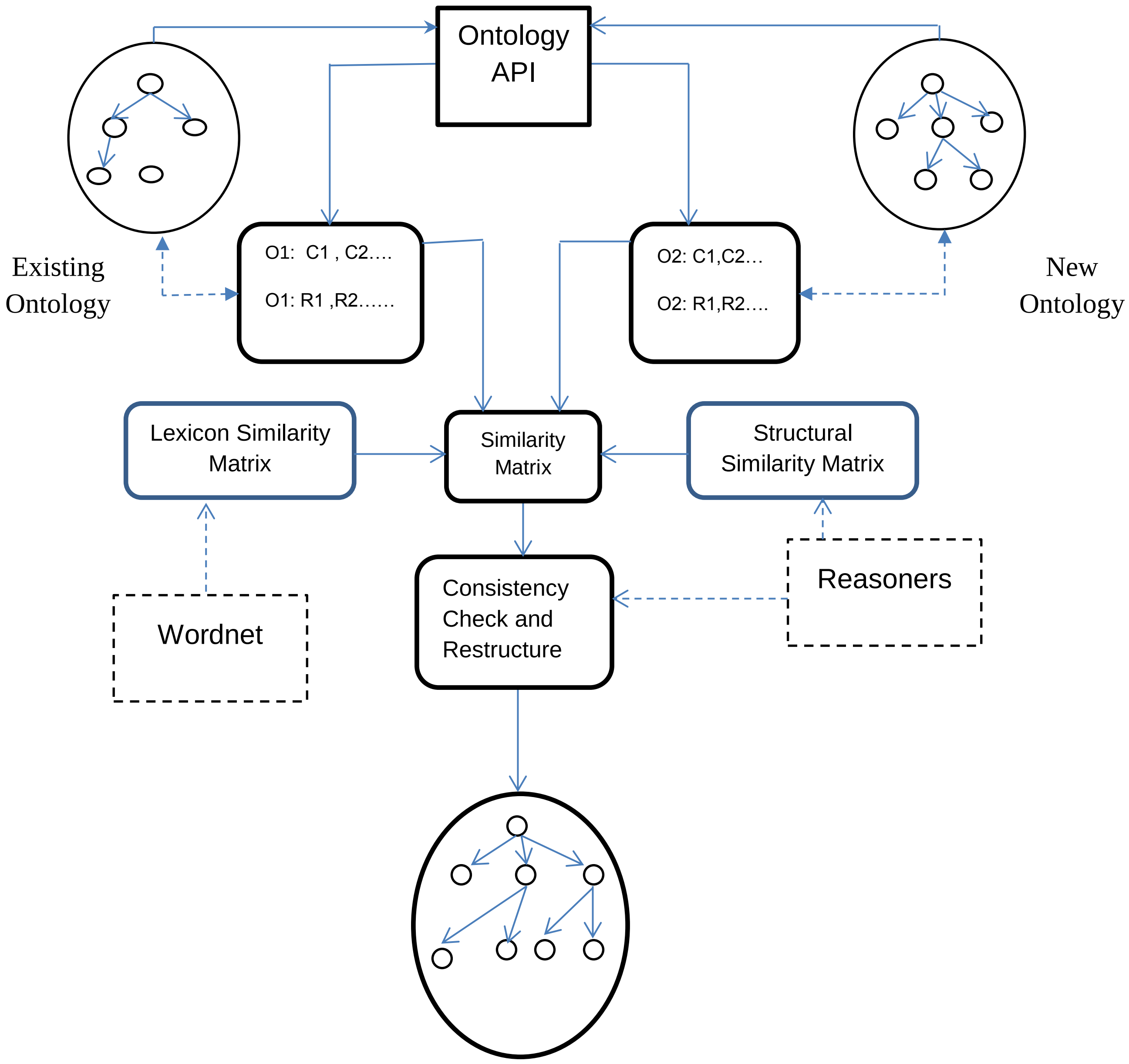


Figure 19. Merged/Reconfigured Ontology

The Semantic similarity matrix and Structural similarity matrix are used to calculate the relational similarity matrix. The next steps, merging, reconfiguration and consistency checks are illustrated in figure (20).

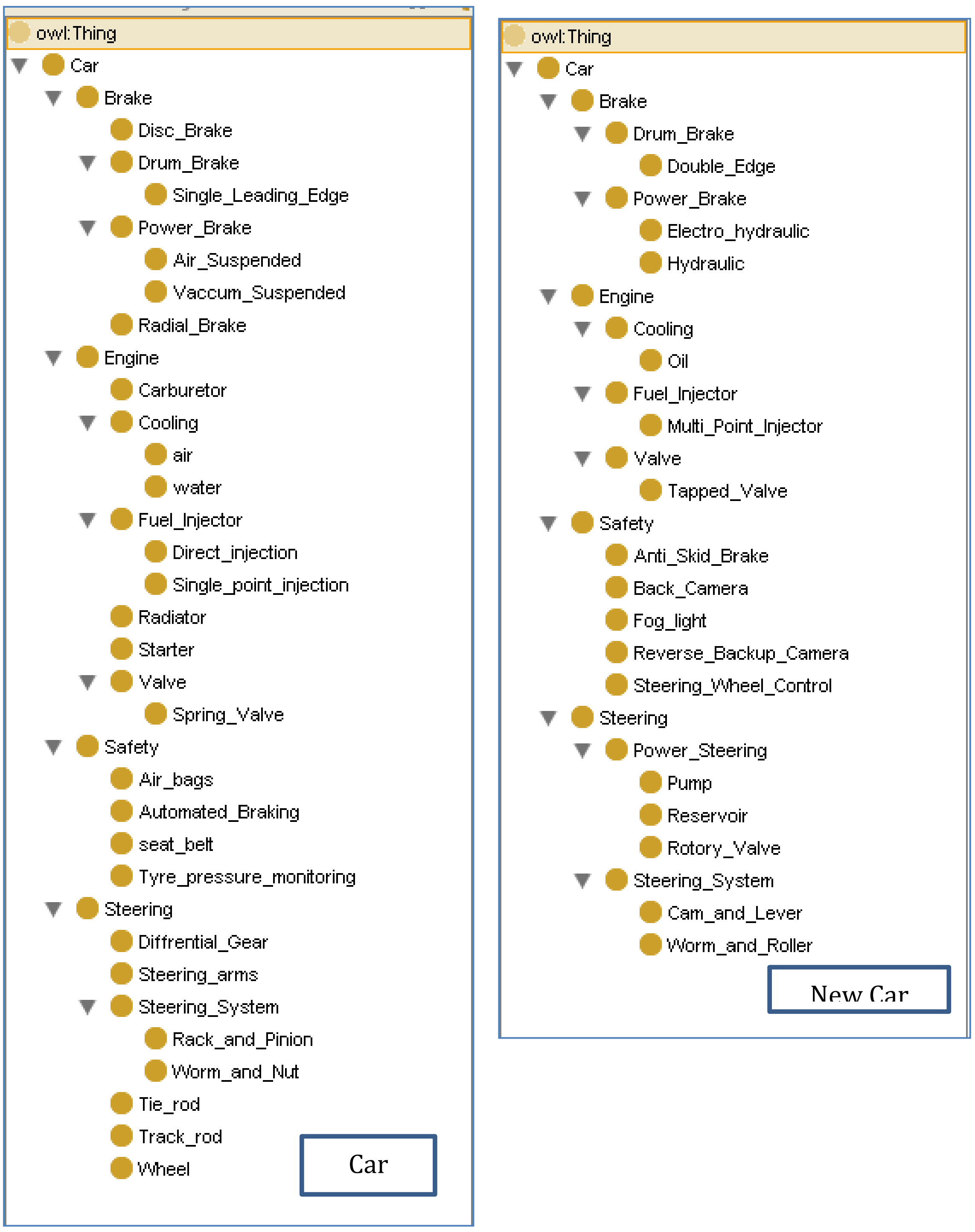
```
// Start
{
Step 1: Input two ontologies
Step 2: Get Concept name
Step 3: Get synonyms, hypernyms, hyponyms
Step 4: Calculate Semantic, structural and relational matrix
Step: 5  for int  j = 1 to J ( j Є $O_2$) s
    {
        k = $\text{argmax}_i$ $ERM(C_j^2), SRM(C_j^2),\ sRM(C_j^2)$ }
        check case 1 to case Ṡ9 for $(C_k^1, C_j^2)$
        if ! satisfied with all super-concept
        add as new concept
    }
end //
}
```

Figure 20. Procedure for Ontology merging and Reconfiguration

Example:

Concrete domain

In order to illustrate the overall procedure, two ontologies in the concrete domain of car manufacturing have been developed from different car parts website available on the internet. As shown in figure (21), The Car ontology describes the current knowledge of the field, whereas the New Car ontology represents new knowledge in the field of car manufacturing. In order to merge the two ontologies, reconfigure an existing ontology (Car Ontology), or incorporate the new knowledge (New Car ontology) the process as described in the previous section is carried out.



- owl:Thing
  - Car
    - Brake
      - Disc_Brake
      - Drum_Brake
        - Single_Leading_Edge
      - Power_Brake
        - Air_Suspended
        - Vaccum_Suspended
      - Radial_Brake
    - Engine
      - Carburetor
      - Cooling
        - air
        - water
      - Fuel_Injector
        - Direct_injection
        - Single_point_injection
      - Radiator
      - Starter
      - Valve
        - Spring_Valve
    - Safety
      - Air_bags
      - Automated_Braking
      - seat_belt
      - Tyre_pressure_monitoring
    - Steering
      - Differntial_Gear
      - Steering_arms
      - Steering_System
        - Rack_and_Pinion
        - Worm_and_Nut
      - Tie_rod
      - Track_rod
      - Wheel

Car

- owl:Thing
  - Car
    - Brake
      - Drum_Brake
        - Double_Edge
      - Power_Brake
        - Electro_hydraulic
        - Hydraulic
    - Engine
      - Cooling
        - Oil
      - Fuel_Injector
        - Multi_Point_Injector
      - Valve
        - Tapped_Valve
    - Safety
      - Anti_Skid_Brake
      - Back_Camera
      - Fog_light
      - Reverse_Backup_Camera
      - Steering_Wheel_Control
    - Steering
      - Power_Steering
        - Pump
        - Reservoir
        - Rotory_Valve
      - Steering_System
        - Cam_and_Lever
        - Worm_and_Roller

New Car

Figure 21. Input Ontologies

With the help of the Jena parser, concept names are identified. The next step involves calculating the similarity matrix. This step comprises of calculating the Lexicon similarity matrix and Structural similarity. Consider the two concepts: Water from the Car ontology (C: Water) and Oil from the New Car ontology (NC: Oil). As there is no similarity between the two concepts in terms of synonyms, hypernyms and hyponyms, their lexicon similarity:

$\psi(C:Water, NC:Oil) = 0.$

For calculating the structural similarity, super and sub concepts need to be identified.

$S(C:Water) = \{\text{Cooling, Engine, Car}\}$

$s(C:Water) = \phi$

$S(NC:Oil) = \{\text{Cooling, Engine, Car}\}$

$s(NC:Oil) = \phi$

As both child concepts are empty sets,

$$ER(C:Water, NC:Oil) = 0.5\left\{\frac{Sim(SC_i^1, SC_j^2)}{\left|SC_i^1 \cup SC_j^2\right|}\right\}^2 + 0.5\left\{\frac{Sim(sC_i^1, sC_j^2)}{\left|sC_i^1 \cup sC_j^2\right|}\right\}^2$$

$$= \left(\frac{3}{3}\right)^2 = 1$$

As both child concepts are empty sets it follows that:

$Sup(C:Water, NC:Oil) = 0$ and

$Sub(C:Water, NC:Oil) = 0$

Now, considering equal weightage for semantic and structural similarity

$$\mathfrak{E}\mathfrak{R}(C:Water, NC:Oil) = 0.5\ \psi(C:Water, NC:Oil) + 0.5\ ER(C:Water, NC:Oil)$$

$$= 0.5$$

Similarly,

$\mathcal{S}\Re(C:Water, NC:Oil)$ =0 and $\mathbf{s}\Re$ $(C:Water, NC:Oil)$ = 0.

In a similar manner the similarity matrix is calculated between ‘ $NC:Oil$ ‘and all the concepts of Car Ontology. The non-zero values obtained are:

$\mathcal{E}\Re$ $(C:water, NC:Oil)$ = 0.5 and $\mathcal{E}\Re$ $(C:air, NC:Oil)$ =0.5.

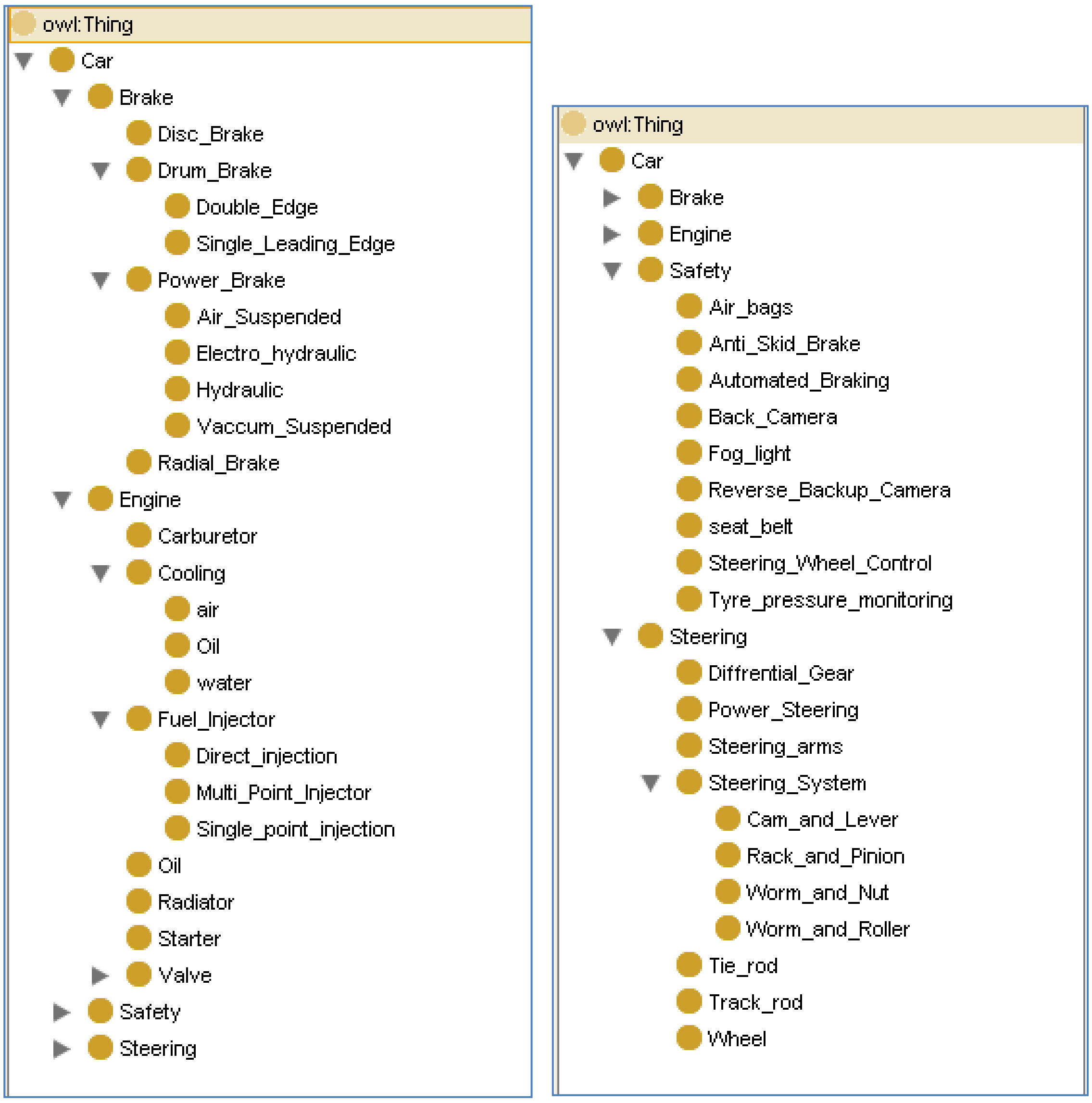
The next step involves the merging of the new concept into the existing concept. As both the similarity matrix indexes have the same value i.e. 0.5, the algorithm arbitrarily selects one of them and tries to merge it logically into the existing concept as explained in section 5. Taking ‘C: water’ the logical relations obtained are:

$C:Cooling \equiv NC:Cooling$ (From TBox similarity) ……………………… (a)

$NC:Oil \subseteq \{NC:Cooling \equiv C:Cooling\}$ ………………………………………. (b)

$NC:Oil \cap C:Water \equiv \perp$ ………………………………………………………… (c)

Clearly, conditions (a), (b) and (c) lead to the case (2) and ‘ $NC:Oil \subseteq C:Cooling$ ’ is established as shown in the figure 22. Table 2 depicts the overall result obtained in this process. In the case of the same similarity index occurring between more than one concept, random selection process has been adopted.



- owl:Thing
  - Car
    - Brake
      - Disc_Brake
      - Drum_Brake
        - Double_Edge
        - Single_Leading_Edge
      - Power_Brake
        - Air_Suspended
        - Electro_hydraulic
        - Hydraulic
        - Vaccum_Suspended
      - Radial_Brake
    - Engine
      - Carburetor
      - Cooling
        - air
        - Oil
        - water
      - Fuel_Injector
        - Direct_injection
        - Multi_Point_Injector
        - Single_point_injection
      - Oil
      - Radiator
      - Starter
      - Valve
    - Safety
    - Steering

- owl:Thing
  - Car
    - Brake
    - Engine
    - Safety
      - Air_bags
      - Anti_Skid_Brake
      - Automated_Braking
      - Back_Camera
      - Fog_light
      - Reverse_Backup_Camera
      - seat_belt
      - Steering_Wheel_Control
      - Tyre_pressure_monitoring
    - Steering
      - Diffrential_Gear
      - Power_Steering
      - Steering_arms
      - Steering_System
        - Cam_and_Lever
        - Rack_and_Pinion
        - Worm_and_Nut
        - Worm_and_Roller
      - Tie_rod
      - Track_rod
      - Wheel

Figure 22. Merged / Reconfigured Ontology

New Concept	Max similarity with Existing Concept	Type of similarity	Logical Case	Final relation
NC : Car	C: Car	TBox	-	Equivalent
NC: Brake	C: Brake	TBox	-	Equivalent
NC: Drum_Brake	C: Drum_Brake	TBox	-	Equivalent
NC: Double_Edge	C:Single_Leading_edge	Equivalent	Case 2	ô C:Drum_Brake
NC: Power_Brake	C: Power_Brake	TBox	-	Equivalent
NC: Electro_Hydraulic	C: Air_Suspended	Equivalent	Case 2	ô C: Power_Brake
NC: Hydraulic	C: Vaccumm_Suspended	Equivalent	Case 2	ô C: Power_Brake
NC: Engine	C: Engine	TBox	-	Equivalent
NC: Cooling	C: Cooling	TBox	-	Equivalent
NC: Oil	C: air	Equivalent	Case 2	ô C: Cooling
NC: Fuel_Injector	C: Fuel_Injector	TBox	-	Equivalent
NC: Multi_Point_Injector	C: Direct_Injection	Equivalent	Case 2	ô C: Fuel_Injector
NC: Valve	C: Valve	TBox	-	Equivalent
NC: Tapped_Valve	C: Spring_valve	Equivalent	Case 2	ô C: Valve
NC: Safety	C: Safety	TBox	-	Equivalent
NC: Anti_Skid_Brake	C: Seat_belt	Equivalent	Case 2	ô C: Safety
NC: Back_Camera	C: Air_bags	Equivalent	Case 2	ô C: Safety
NC: Fog_Light	C: Seat_belt	Equivalent	Case 2	ô C: Safety
NC: Reverse_Backup_Camera	C: Automated_Braking	Equivalent	Case 2	ô C: Safety
NC: Steering_Wheel_Control	C:Air_bags	Equivalent	Case 2	ô C: Safety
NC: Steering	C: Steering	TBox	-	Equivalent
NC: Power_Steering	C: steering	Sub-class	Case Ṡ1	ô C: Steering
NC: Pump	C: steering	Sub-class	Case Ṡ1	ô C: Power_Steering
NC: Reservoir	C: steering	Sub-class	Case Ṡ1	ô C: Power_Steering
NC: Rotary_Valve	C: steering	Sub-class	Case Ṡ1	ô C: Power_Steering
NC: Steering_System	C: Steering_System	TBox	-	Equivalent
NC: Cam_and_Lever	C: Rack_and_pinion	Equivalent	Case 2	ô C:Steering_System
NC: Worm_and_Roller	C:Worm_and_Nut	Equivalent	Case 2	ô C:Steering_System

Table 2: Merging process outcome

Although, example presented here describes the knowledge merging process within an enterprise but this process can be extended in case of merging of two different enterprises with different ontology based data bases built on same domain.

<u>**Fuzzy Domain:**</u>

In the case of a fuzzy ontology, the process of finding the similarity index and determining the position of merging of a new concept with the existing ontology has been explained in sections 4 and 5. In this process Fuzzy domain ontologies differ from concrete domain ontologies only in the step comparing the fuzzy values of the new concept with the existing ones.

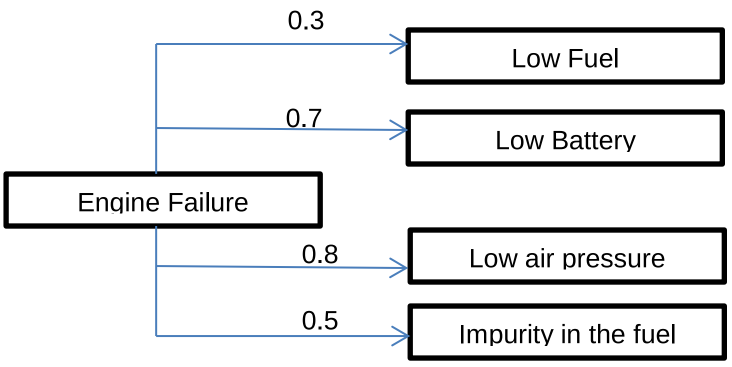


Figure 23: Fuzzy existing knowledge

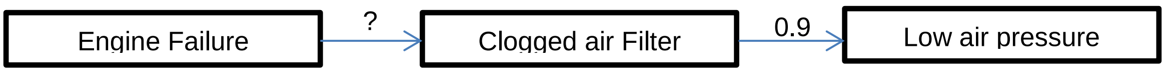


Figure 24: Fuzzy new knowledge

Taking an example as shown in figure 23, a concept 'Engine failure' has different sub-concepts with different fuzzy values e.g. 'Low Fuel $ô_{0.3}$ Engine Failure', and in the new knowledge as shown in figure 24, 'Low air pressure $ô_{0.9}$ Clogged air filter'. These two ontologies are first compared for similarity and then to determine the position of the new concept within the new merged ontology, the same process is followed as described earlier. The results obtained identify 'Low air pressure ô Clogged air filter ô Engine Failure'. Regarding the fuzzy values the following information is available:

Low air pressure $ô_{0.8}$ Engine Failure ……………………………………………….(x)

Low air pressure $ô_{0.9}$ Clogged air filter ……………………………………………. (y)

The unified ontology needs to find the fuzzy value of 'Clogged air filter $ô_{?}$ Engine failure' (assume β). As explained in section 5,

Low air pressure $ô_{0.8}$ Engine failure = (Low air Pressure $ô_{0.9}$ Clogged air filter) ⟶ (Clogged air filter $ô_{\beta}$ Engine failure).

Using the Luasiewicz implication function: 0.8 = min (1, 1-0.9 +β) = min (1, 0.1+β),

clearly, β = 0.7 as shown in the 25. Using this process all other fuzzy values for the fuzzy relation can be defined.

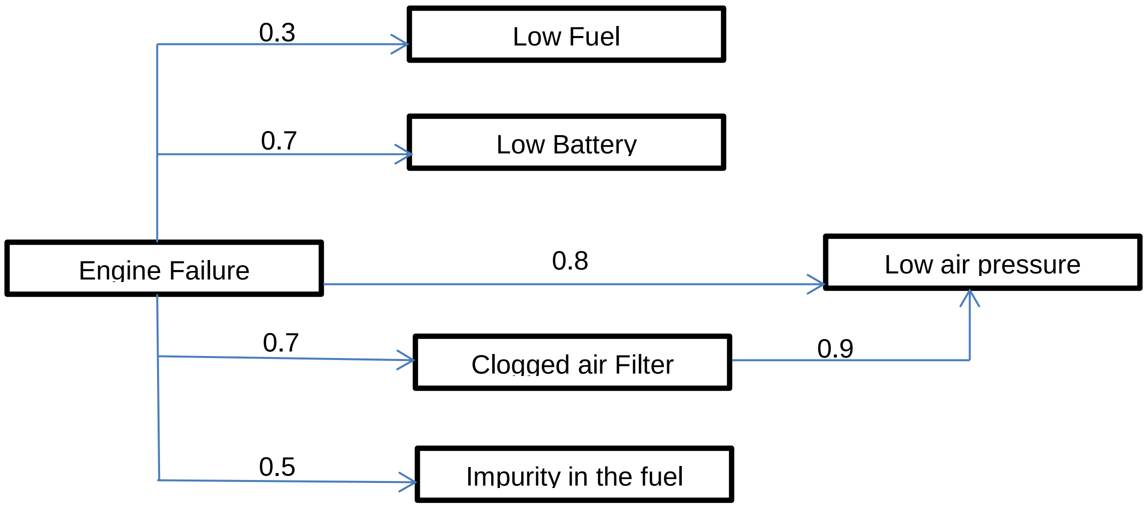


Figure 25: Fuzzy merged knowledge

CONCLUSION:

This paper addresses issues related to combining new and existing knowledge in the form of an ontology. The advantage of the proposed method is that enterprises need not reconstruct their existing ontology to accommodate newly acquired knowledge. This methodology merges the new knowledge into an existing ontology using ontology merging and reconfiguration. It also checks for any inconsistencies. The process of merging and reconfiguration first identifies the similarity between the concepts of two ontologies and then, with the help of reasoning, identifies the positions where the new concepts will fit in to the existing ontology without any inconsistencies. This approach not only considers the concrete domain but also the fuzzy domain. The proposed approach can also be used in any ontology based knowledge base with different application domains such as product data management (PDM), product lifecycle management (PLM)[12] and product development[10]. OWL representation of knowledge has been shown by many researchers[10,38-39] and in all such cases new knowledge can be merged in the existing database in the form of an ontology, build in the same domain, using this approach.

REFERENCES:

1. Kim TY, Lee S, Kim K, Kim CH. A modeling framework for agile and interoperable virtual enterprises. *Comput Ind.* 2006 Apr;57(3):204–217.

2. Singh J. Collaborative Networks as Determinants of Knowledge Diffusion Patterns. *Manage Sci.* 2005 May;51(5):756–770.

3. Gold-Bernstein B, Ruh W. Enterprise integration: The essential guide to integration solutions. Boston, MA: Addison–Wesley; 2005.

4. Pollalis YA, Dimitriou NK. Knowledge management in virtual enterprises: A systemic multi-methodology towards the strategic use of information. *Int J Inf Manage.* 2008 Aug;28(4):305–321.

5. Kumar SK, Harding JA. Ontology mapping using description logic and bridging axioms. *Comput Ind.* 2013 Jan;64(1):19–28.

6. Mo JPT, Zhou M. Tools and methods for managing intangible assets of virtual enterprise. *Comput Ind.* 2003 Jun;51(2):197–210.

7. Ku K, Wensley A, Kao H. Ontology-based knowledge management for joint venture projects. *Expert Syst Appl.* 2008 Jul;35(1-2):187–197.

8. Das A, Wu W, Mcguinness DL. Industrial Strength Ontology Management. In: In Proceedings of the First Semantic Semantic Web Working Symposium, SWWS-01. IOS Press; 2001.

9. Zhang ZN, Liu ZL, Chen Y, Xie YB. Knowledge flow in engineering design: An ontological framework. *Proc Inst Mech Eng Part C J Mech Eng Sci.* 2012 Jul 31;227(4):760–770.

10. Sun H, Fan W, Shen W, Xiao T. Ontology-based interoperation model of collaborative product development. *J Netw Comput Appl.* 2012 Jan;35(1):132–144.

11. Ivanović M, Budimac Z. An overview of ontologies and data resources in medical domains. *Expert Syst Appl.* 2014 Sep;41(11):5158–5166.

12. Raza MB, Kirkham T, Harrison R, Reul Q. Knowledge Based Flexible and Integrated PLM System at Ford. 2011;1(1):8–16.

13. Gunasekaran A, Ngai EWT. Knowledge management in 21st century manufacturing. *Int J Prod Res.* 2007 Jun;45(11):2391–2418.

14. Kebede G. Knowledge management: An information science perspective. *Int J Inf Manage.* 2010 Oct;30(5):416–424.

15. Liao S. Knowledge management technologies and applications—literature review from 1995 to 2002. *Expert Syst Appl.* 2003 Aug;25(2):155–164.

16. Brockman BK, Morgan RM. The Role of Existing Knowledge in New Product Innovativeness and Performance. *Decis Sci.* 2003 May;34(2):385–419.

17. Sun C, Xu X, Li X, Deng S. Notice of Violation of IEEE Publication Principles, Knowledge Discovery from Virtual Enterprise Model Based on Semantic Annotation. *2008 Fifth Int Conf Fuzzy Syst Knowl Discov.* 2008 Oct;:546–551.

18. Korposh D, Lee YC, Wei CC. Modeling the Effects of Existing Knowledge on the Creation of New Knowledges. *Concurr Eng.* 2011 Oct 9;19(3):225–234.

19. Huang N, Diao S. Ontology-based enterprise knowledge integration. *Robot Comput Integr Manuf.* 2008 Aug;24(4):562–571.

20. Ling L, Hu Y, Wang X, Li C. An ontology-based method for knowledge integration in a collaborative design environment. *Int J Adv Manuf Technol.* 2007 Aug 31;34(9-10):843–856.

21. Rajsiri V, Lorré J-P, Bénaben F, Pingaud H. Knowledge-based system for collaborative process specification. *Comput Ind.* 2010 Feb;61(2):161–175.

22. Ho CT, Chen YM, Chen YJ, Wang CB. Developing a distributed knowledge model for knowledge management in collaborative development and implementation of an enterprise system. *Robot Comput Integr Manuf.* 2004 Oct;20(5):439–456.

23. Pirró G, Mastroianni C, Talia D. A framework for distributed knowledge management: Design and implementation. *Futur Gener Comput Syst.* 2010 Jan;26(1):38–49.

24. Chen RC, Bau CT, Yeh CJ. Merging domain ontologies based on the WordNet system and Fuzzy Formal Concept Analysis techniques. *Appl Soft Comput.* 2011 Mar;11(2):1908–1923.

25. Raunich S, Rahm E. ATOM: Automatic target-driven ontology merging. *2011 IEEE 27th Int Conf Data Eng.* 2011 Apr;:1276–1279.

26. Noy NF, Musen MA. PROMPT : Algorithm and Tool for Automated Ontology Merging and Alignment. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence. AAAI Press; 2000. p. 450–455.

27. Dou D, Mcdermott D, Qi P. Ontology Translation by Ontology Merging and Automated Reasoning. In: EKAW2002 Workshop on Ontologies for Multi-Agent Systems (OMAS 2002). 2002. p. 3–18.

28. Stoilos G, Stamou G, Pan JZ. Fuzzy extensions of OWL: Logical properties and reduction to fuzzy description logics. *Int J Approx Reason.* 2010 Jul;51(6):656–679.

29. Bobillo F, Straccia U. Reasoning with the finitely many-valued Łukasiewicz fuzzy Description Logic SROIQ. *Inf Sci (Ny).* 2011 Feb 15;181(4):758–778.

30. Lu J, Li Y, Zhou B, Kang D. Reasoning within extended fuzzy description logic. *Knowledge-Based Syst.* 2009 Jan;22(1):28–37.

31. Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider P, editors. The Description Logic Handbook. Cambridge University Press; 2003.

32. Zadeh LA. Fuzzy Sets. *Inf Control.* 1965;8(3):338–353.

33. Wordnet. : http://wordnet.princeton.edu/ .

34. Gruber TR. A translation approach to portable ontology specifications. *Knowl Acquis.* 1993;5(2):199–220.

35. Protege. : http://protege.stanford.edu/ .

36. FuzzyDL Systems. : http://gaia.isti.cnr.it/~straccia/software/fuzzyDL .

37. Pellet:OWL 2 Reasoner for Java. : http://clarkparsia.com/pellet/ .

38. Sarigecili MI, Roy U, Rachuri S. Interpreting the semantics of GD&T specifications of a product for tolerance analysis. *Comput Des.* 2014 Feb;47:72–84.

39. Borsato M. Bridging the gap between product lifecycle management and sustainability in manufacturing through ontology building. *Comput Ind.* 2014 Feb;65(2):258–269.