Semi-automatic Construction of Ontology-based CBR System for Knowledge Integration

Junjie Gao and Guishi Deng

Abstract—In order to integrate knowledge in heterogeneous case-based reasoning (CBR) systems, ontology-based CBR system has become a hot topic. To solve the facing problems of ontology-based CBR system, for example, its architecture is nonstandard, reusing knowledge in legacy CBR is deficient, ontology construction is difficult, etc, we propose a novel approach for semi-automatically construct ontology-based CBR system whose architecture is based on two-layer ontology. Domain knowledge implied in legacy case bases can be mapped from relational database schema and knowledge items to relevant OWL local ontology automatically by a mapping algorithm with low time-complexity. By concept clustering based on formal concept analysis, computing concept equation measure and concept inclusion measure, some suggestions about enriching or amending concept hierarchy of OWL local ontologies are made automatically that can aid designers to achieve semi-automatic construction of OWL domain ontology. Validation of the approach is done by an application example.

Keywords—OWL ontology, Case-based Reasoning, Formal Concept Analysis, Knowledge Integration

I. INTRODUCTION

ONTOLOGY is formal, explicit specification of a shared conceptualization, and has been applied in many fields, such as Semantic Web, information integration and knowledge management, etc. More and more researchers begin to pay attention to ontology research.

At the same time, case-based reasoning (CBR) as a traditional research domain of artificial intelligence is a problem-solving paradigm that in many respects is fundamentally different from logic-based approaches. CBR is able to utilize the specific knowledge gained from previously experienced, similar problem situations (case) to solve a new problem [1, 2]. Instead of relying on exact reasoning in a well-ordered world, CBR focuses on inexact reasoning by similarity measure between objects. And there are abundant research productions and enterprise applications about CBR. However, CBR systems just like most of the knowledge-based systems (KBS) have some reusable ontological content but it is often influenced by the specific task, the restrictions of the

representation language, and the specific inference procedures employed. So it is difficult to achieve integration between heterogeneous CBR systems.

Compared with so many researches that have been done to achieve integration between heterogeneous CBR systems, ontology-based CBR system has become a hot topic and the trend of development [3, 4]. The reasons are as follows: the structure of domain knowledge is clarified by ontology that lays foundation of knowledge acquisition and knowledge expression and avoids analyzing domain knowledge repeatedly; Unifying domain terminologies and concepts by ontology make knowledge sharing possible. However, research about ontology-based CBR is at the initial stage, which still faces many problems, for example, its architecture is nonstandard, reusing knowledge in legacy CBR is deficient, the so called knowledge acquisition bottleneck (common for every KBS) is still there, and it is time-consuming and labor-intensive even by the aid of ontology editor to construct ontology, etc. To solve the problems above, we propose an approach for semi-automatic construct ontology-based CBR system whose architecture is based on two-layer ontology.

This paper is organized as follows: Section 2 discusses the architecture of ontology-based CBR system. Section 3 illustrates the approach for semi-automatic construction of ontology-based CBR system in detail. Section 4 introduces an application example. Finally, section 5 gives conclusions and the future work.

II. ARCHITECTURE

In nearly all ontology–based integration approaches, ontologies are used for the explicit description of the information source semantics. But there are different ways of how to employ the ontologies. In general, three gradually developing directions can be identified: single ontology approaches, multiple ontologies approaches and hybrid ontology approaches [5]. Single ontology approaches use one global ontology providing a shared vocabulary for the specification of the semantics (see fig.1a). All information sources are related to the global ontology. But if one information source has a different view on a domain, finding the minimal ontology commitment becomes a difficult task. In multiple ontology approaches (see fig.1.b), each information source is described by its own local ontology, and inter-ontology mapping identifies semantically corresponding terms of different local ontologies. When a local ontology

changed, only need to change relevant parts affected. But in practice the lack of a common vocabulary makes it extremely difficult to compare different local ontologies. The inter-ontology mapping is very difficult to define because the many semantic heterogeneity problems may occur. To overcome the drawbacks of the single or multiple ontology approaches, hybrid ontology approaches were developed (see fig.1.c). In order to make the local ontologies comparable to each other they are built upon one global shared vocabulary which contains basic terms (the primitives) of a domain. But the drawback of hybrid ontology approaches is that existing ontologies cannot be reused easily and have to be re-developed from scratch, because all source ontologies have to refer to the shared vocabulary.

In order to make up for the deficiencies of the traditional architectures, we improve hybrid ontology approaches and introduce the architecture based on two-layer ontology approach (see fig.1.d).
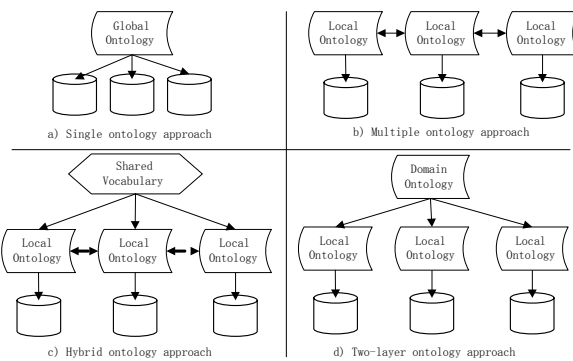

Fig. 1 Architecture of ontology-based integration approach

In the process of CBR, case retrieval is vital. In traditional view, CBR systems provide a search functionality that makes use of similarity measures for ranking results according to their utility with respect to a given query. And the similarity measure is of critical importance during the retrieval of knowledge items for a new query and it assesses the utility of a knowledge item only based on the characterization. However, based on the architecture of two-layer ontology, the measure of concept similarity can be applied to similarity model of CBR, which makes case retrieval more reasonable and assesses the utility of a knowledge item not only based on the characterization but also based on the concept hierarchy of OWL domain ontology. The practical case retrieval can be achieved by implementation of the case retrieval algorithm that we proposed in reference [6].

III. THE APPROACH FOR SEMI-AUTOMATIC CONSTRUCTION OF ONTOLOGY-BASED CBR SYSTEM

The approach flow of semi-automatic construction of ontology-based CBR system is shown as fig.2, which comprises mainly two parts: OWL local ontologies learned from case bases and semi-automatic construction of OWL domain ontology.
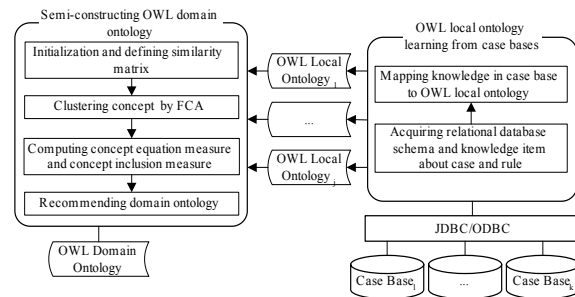

Fig. 2 The semi-automatic construction approach flow

A. OWL Local Ontology Learned from Case Base

1) Acquiring relational database schema and knowledge item about case and rule

The main elements of relational database schema consist of the base table structure design and declaration of integrality constraint. They being the current status message of database are stored in data dictionary. The general relational database schema extraction approach is to use API such as ODBC API, JDBC API to eliminate heterogeneity of different RDBMS. The development of this technology is mature; so we don't discuss it any more. The knowledge items about case and rule stored in tuples of relational tables can be acquired and operated by SQL sentence directly.

2) Mapping knowledge in case base to OWL local ontology automatically

Source data of mapping is based relational model, which is comprised of relational database schema information and tuples of relational table. The goal of mapping is OWL ontology that is a rich semantic model with more complicated structure.

Definition 1 Relational database schema is comprised of a set of name and a set of constraint. Thereinto, the name set is comprised of table name set, data type name set and column name set; the constraint set is comprised of primary key constraint, foreign key constraint, domain constraint, unique constraint and not null constraint.

Definition 2 OWL ontology is comprised of an optional ontology identifier set and a set of axiom. Thereinto, ontology identifier set is comprised of class identifier, object property identifier, data type property identifier and individual identifier; Axiom set is comprised of class axiom, property axiom and individual axiom.

Based on the formal definitions above and reference from correlative researches [7-10], tables are classified as entity tables and relationship tables that are mutually exclusive. There is single primary key in entity table, and there are multiple primary keys in relationship table. Then mapping relationships about source data and mapping goal are as follows: a entity table maps to a OWL class identifier and a class axiom; a relationship table maps to two object property identifiers and two property axioms indicating that they are inversive; a column who is not a foreign key and its domain constraint maps to a data type property identifier and a property axiom; a

column who is a foreign key and its foreign key constraint maps to a object property identifier and its property axioms; data type name maps to data type identifier which is a predefine XML Schema data type identifier used in OWL ontology; each of unique constraints, not null constraints and primary key constraints maps to a property restriction about cardinality constraint; if the primary key of a table is a foreign key referring to another table, then maps to a class axiom which describe that one class is subclass of another; a tuple of table

maps to a individual identifier and its individual axiom set.

Based on the mapping relationship above, an automatic mapping algorithm is proposed in this paper; its flow is shown in fig.3. From the angle of function, domain knowledge implied in relational database schema of case base is mapped to the elements of OWL local ontology by step 1 and step 2, the domain knowledge items about cases and rules are mapped to the elements of OWL local ontology by step 3.
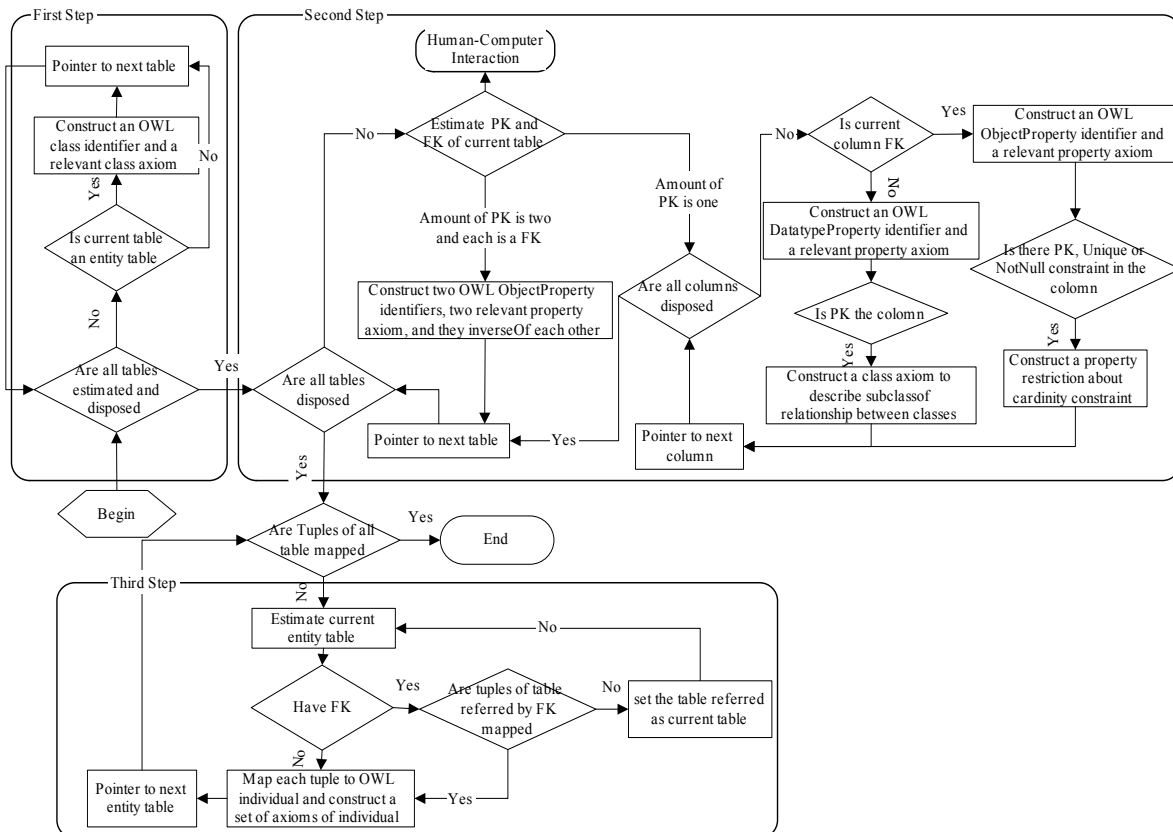

Fig. 3 The Flow of Automatic Mapping Algorithm

Theoretical analysis of this algorithm's time performance is discussed as follows:

All operations about identifiers construction can be involved in the operations about relevant axioms construction. So the basic operation of this algorithm can be considered an axiom construction. Define the scale of a CBR system as $N=N_T+N_A+N_i$, where N_T is the amount of all tables, N_A is the amount of all columns and N_i is the amount of tuples of all tables. For the first step, the extreme scenario is that all tables in database are entity table, and then the amount of the basic operation about class axioms construction is less than N_T. In the second step, there are two different situations; for entity table, the amount of basic operation is less than $4N_A$, and for relationship table, the amount of basic operation is less than N_T. For the third step, the amount of basic operation about

individual axioms construction is less than $N_A \times N_i$. So in the worst case scenarios, the total amount of basic operations is T, where $T=N_T+4N_A+N_T+N_A \times N_i<N^2$, then the time-complexity is lower than $O(N^2)$.

A. Semi-automatic constructing OWL domain ontology

The OWL local ontologies acquired by the approach above are lightweight and concept hierarchies are too horizontal. It is very difficult to construct domain ontology based on them manually. So we propose an approach for semi-automatic construct OWL domain ontology by using FCA technology and computing concept equation measure and concept inclusion measure.

The semantic relations between concepts are established based on similarity measure. Based on the definitions of similarity in references [11], concept similarity measure is

categorized as concept equation measure and concept inclusion measure in this paper. The definitions are introduced as follows.

Definition 3 Given two classes C_1 and C_2, a concept equation measure is defined as a real-valued function as follows:

$$Equ : (C_1) \times (C_2) \rightarrow [0,1] ; \qquad (1)$$

The value of Equ (c_1, c_2) indicates the probability that c_1 and c_2 represent the same concept. The function of Equ ought to be reflexive and symmetric.

Definition 4 Given two classes C_1 and C_2, a concept inclusion measure is defined as a real-valued function as follows:

$$Sub : (C_1) \times (C_2) \rightarrow [0,1] ; \qquad (2)$$

The value of Sub (c_1, c_2) indicates the probability that c1 is a subclass of c_2. In other words, the concept represented by class c1 is a sub-concept of represented by class c_2.

The approach is described in detail as follows.

1) Initialization and defining similarity matrix

Not all concepts (OWL classes) in OWL local ontologies are similar, or have semantic relationship. In order to reduce computing concept equation measure and concept inclusion measure and improve the efficiency of recommending concept hierarchy of domain ontology, a set of similar ontology concepts and relevant property sets should be chosen as initial information.

The process of choosing similar concepts is described as follows: To some class c_1 in OWL local ontology, firstly, find out the most similar OWL class set C_i according to the name similarity of OWL classes; secondly, find out the super-class and subclass set about class c_1 based on concept hierarchy assertion about class c_1 in OWL local ontology; finally, define the initial class set $C=C_i \cup C_j$. Now the initial information is determined as the class set C and relevant property sets. Thereinto, we adopt the Levenstein edit distance to compute string similarity measure, because all class names are string.

Construction of the similarity matrix is based on computing similarity between properties. So in the following, computing similarity between properties will be discussed, then an example of similarity matrix will be introduced.

There are some primary elements of OWL property such as name, range and cardinality constraint. So these elements must be considered when the similarity of two properties is computed. The following is the formula of property similarity measure in this paper:

$$ASim(a_1, a_2) = \omega_1 \times ASim_{name} + \omega_2 \times ASim_{range} + \omega_3 \times ASim_{cardinality} \quad (3)$$

where ω_1, ω_2, ω_3 are weights, which represent the importance degrees for property similarity measure about name, range, cardinality constraint respectively; $\omega_1 + \omega_2 + \omega_3 = 1$ and $\omega_1 > \omega_2 > \omega_3$. $ASim_{name}$ measured by Levenstein edit distance is the similarity of property name. $ASim_{range}$ is the similarity of property range. There are two types of property in OWL ontology. Based on the different property types, different strategies are launched as follows: For two properties a_1 and a_2, if types of a_1 and a_2 are different, then $ASim_{range} = 0$. If both a_1 and a_2 are objectproperty, then $ASim_{range} = Sim_{str}(rang(a_1)$, range($a_2$)), in other words, $ASim_{range}$ is the name similarity

between OWL classes which are ranges of a_1 and a_2. If a1 and a_2 are datatypeproperty, then $ASim_{range}$ is computed according to data types of a_1 and a_2, The match degree between different data types is based on TABLE I . $ASim_{card}$ is the similarity of property cardinality constraint. If the cardinality constraints of a_1 and a_2 are same, then $ASim_{card}=1$, else $ASim_{card}=0$.

Example 1 There are two similar classes c_1 and c_2, the relevant property sets are $A_1 = \{a_{11}, a_{12}, a_{13}, a_{14}\}$, $A_2 = \{a_{21}, a_{22}, a_{23}, a_{24}, a_{25}\}$. The similarity matrix, *Matrix M_{12}* of c_1 and c_2 is given in table II .

2) Clustering concept by FCA

FCA was introduced as a mathematical theory modeling the

TABLE I
DATA TYPE MATCH DEGREE

Match degree	Float	Int	String	Datetime
Float	1	0.9	0.1	0.
Int	0.9	1	0.1	0.8
String	0.1	0.1	1	0.1
Datetime	0.7	0.8	0.1	1

concept of "concepts" in terms of lattice theory. The basics of FCA theory needed for this paper can be found in references [12, 13]. The process of concept lattice generated from a formal context is a process of concept clustering. In this paper, clustering concept by FCA mainly includes constructing formal

TABLE II
SIMILARITY MATRIX OF C_1 AND C_2

Similarity	a_{21}	a_{22}	a_{23}	a_{24}
a_{11}	0.9	0.3	0.6	0.5
a_{12}	0.2	0.4	0.2	0.1
a_{13}	0.1	0.6	0.7	0.2
a_{14}	0.2	0.4	0.3	0.5
a_{15}	0.1	0.8	0.2	0.6

context and generating relevant concept lattice.

Different properties of classes can indicate different correlative relations between these classes, so we should construct different formal context for different classes and properties. The process of construction formal context includes two steps:

a. Based on the initialization, construct initial formal context in which the set of classes C as objects, the set of properties A as characteristics; The initial formal context is K=(G, M, I), where G = C, M = A and I = C×A.

b. Based on the similarity matrixes between classes, *Matrixes* amend the formal context that has been constructed. In order to determine the relationship of OWL class c_i and the properties of another OWL class c_j, in this step, the possibility of class c_i having the characteristics described by the properties of c_j should be acquired, and then the formal context should be amended.

For Example 1: In order to decide the relations between class c1 (or c_2) and A_2 (or A_1), the formal context is amended based on the similarity matrix M_{12}. The sub-steps of this process include:

Firstly, for each i and j, with $i \in \{1,...,p\}$, $j \in \{1,...,q\}$, acquire the similarity between a_{1i} and a_{2j} from *Matrix M_{12}*, where $a_{1i} \in A_1$ and $a_{2j} \in A_2$.

Secondly, for each $a_{1i} \in A_1$, find the property from A_2 which has the highest similarity with a_{1i}, named as MaxSimwith_a_{1i}. And MaxSimwith_a_{1i}

$$= \{a_{2j} \mid j = \arg \max_{j \subset \{1,...,q\}} \{Similarity(a_{1i}, a_{2j}) \neq 0\}\} \subseteq A_2 \quad (4)$$

And for each $a_{2j} \in A_2$, find the property from A_1 which has the highest similarity with a_{2j}, named as MaxSimwith_a_{2j}. And MaxSimwith_a_{2j}=

$$\{a_{1i} \mid i = \arg \max_{i \subset \{1,...,p\}} \{Similarity(a_{1i}, a_{2j}) \neq 0\}\} \subseteq A_1 \quad (5)$$

For *Matrix M_{12}*, MaxSimto_a_{11} = a_{21}, MaxSimto_a_{12} = a_{22}, MaxSimto_a_{13} = a_{23}, MaxSimto_a_{14} = a_{24}, MaxSimto_a_{15} = a_{22}, MaxSimto_a_{21} = a_{11}, MaxSimto_a_{22} = a_{15}, MaxSimto_a_{23} = a_{13}, MaxSimto_a_{24} = a_{15}.

Finally, let $P(c, a)$ means the possibility of class c having the characteristics described by the property a, and according to the possibility, construct amended formal context.

For Example 1, $P(c_1, a_{2j})$=Similarity(MaxSimto_a_{2j}, a_{2j}), $P(c_2, a_{1i})$= Similarity(MaxSimto_a_{1i}, a_{1i}), and if $(c, a) \in I$, then (c, a)=1. The amended formal context is given in Table Ⅲ.

Based on the amended formal context, we use the mark "×" to replace every real number that is more than threshold ε, and then acquire a simple value formal context. Here, ε is determined by users according the practical situations. So the existing methods and algorithms can still be used to generate concept lattice [12,13]. We use the tool of ConExp to generate relevant concept lattice [14], and an example will be introduced in section Ⅳ.

TABLE III
THE AMENDED FORMAL CONTEXT FOR EXAMPLE 1

	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{21}	a_{22}	a_{23}	a_{24}
c_1	1	1	1	1	1	0.9	0.8	0.7	0.6
c_2	0.9	0.4	0.7	0.5	0.8	1	1	1	1

3) Computing concept equation measure and concept inclusion measure

Compared with FCA theory, intention of concept is more important for ontology, so we compute concept similarity according to properties of OWL ontology in this paper. In other words, if the properties of two concepts are same, then the two concepts are same; if the properties of two concepts are similar, then the two concepts are similar. Based on the concept lattice generated above, we can compute the equation measure and inclusion measure. The detailed steps are as follows:

For OWL class $c \in G$, compute the probability that c belongs to the formal concept represented by the node which c is associated with in the foregoing concept lattice, denoted by $P(c)$:

$$P(c) = \prod_{i=1}^{n} P(c, a_i) \quad \text{where}$$

$$a_j \in \{c\}', \{c\}' = \{a \in M \mid (c, a) \in I\} \quad （6）$$

Based on the generated concept lattice, analyze OWL class c_1 and c_2, if $\{c_1\}' = \{c_2\}'$, then $Sub(c_1, c_2) = P(c_1) \times P(c_2)$; analyze OWL class c_1 and c_2, if $\{c_1\}' \subset \{c_2\}'$, then $Sub(c_1, c_2) = P(c_1) \times P(c_2)$.

4) Recommending domain ontology

The semantic relationships between concepts of OWL ontology are mainly described by axioms about subClassOf(), equivalentClasses(), disjointClasses() and disjointUnion()[15]. We mainly discuss automatic recommending axioms about subClassOf() and equivalentClasses() in this paper. The approach is introduced as follows: automatic recommend that the two concepts whose concept equation measure is more than the threshold δ are same; automatic recommend that the two concepts whose concept inclusion measure is more than the threshold η have super-sub relationship; then users determine to weather accept the recommendations of not. For example: if the concept equation measure between OWL class c_1 and c_2 is more than δ, then the axiom of equivalentClasses(c_1,c_2) will be recommended to users; if the concept inclusion measure between OWL class c_1 and c_2 is more than δ, then the axiom of subClassOf (c_1,c_2) will be recommended to users.
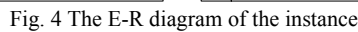
At the same time, generated concept lattice can show the super-sub relationship between concepts as Hasse diagram vividly and compactly. It can help designer to analyze the relationship between concepts and find new concept, and then aid designer to construct OWL domain ontology and rich the concept hierarchy of OWL domain ontology.

IV. IMPLEMENTATION

In order to evaluate the approach proposed above, we take a legacy CBR system about tooling man-hour rationing as an example, and experimental study semi-automatic construction of relevant ontology-based CBR system.

The main task of this legacy CBR system is to aid engineers to rapidly ration a suit of tooling whose structure is complex according to existed experiments, which is empirical and predictive. There are two types of knowledge in this system: tooling cases and tooling characteristic rules. They are stored in relational database. Tooling cases are facts knowledge that were rationed by engineer and proved right. Every tooling case consists of qualitative characteristic description (for example type of accessory processed), quantitative characteristic description (for example the size of die core), relevant quotiety and man-hour value. Tooling characteristic rule knowledge mainly provides extra domain knowledge and vocabulary. They consist of typical descriptions about tooling characteristic information and relevant quotiety that represents the influence degree contrasting current description to the benchmark description about tooling characteristic.

The mapping algorithm has been implemented based on J2SE v 1.4.2 and Jena2 API (mainly based on the interfaces and relevant methods in the packet com.hp.hpl.jena.ontology). In order to validate the automatic mapping algorithm, five sets of cases and relevant rules were mapped to OWL local ontology, and the result show that the relation between the practical running time of the algorithm and the problem scale N is

consistent with the theoretical analysis (better than N^2 curve). Now take the reduced forging-lancing die case table and relevant five rule tables as an example, E-R diagram of them is shown as fig.4.

Fig. 4 The E-R diagram of the instance

The amount of relevant tuples in forging-cutting die case table, structure type rule table, accessory processed type rule table, type of parting surface rule table, depth and width ratio of cavity rule table and amount of module rule table are 6, 3, 5, 6, 3, 4 respectively. The mapping result is shown as fig.5.

Fig. 5 The relevant result of automatic mapping

Axioms in the result consist of 6 class axioms, 17 data type property axioms, 5 object property axioms, 22 axioms about cardinality constraint and 124 individual axioms. It can be seen that it is consistent with the theoretical result, and the implementation is correct.

In order to assess the recommendation about concept hierarchy of domain ontology got by the approach of semi-automatic construction OWL domain ontology, we use standard information retrieval metrics [8].

recall: $r = \dfrac{correct\_recommendation}{existing}$

precision: $p = \dfrac{correct\_reommendation}{all\_recommendation}$

F-measure: $f = 2pr/(p + r)$

Here take a set of concepts about metal die cases as an example. There are 31 OWL classes in this set of concepts. According to this set of OWL classes and relevant properties, formal context was constructed as fig.6, and then relevant concept lattice was generated which is shown as fig. 7. Concept count is 32, edge count is 59 and lattice height is 8. Then concept hierarchy was recommended and OWL domain ontology was constructed. It is found: r=0.571, p=0.693, f=0.626.

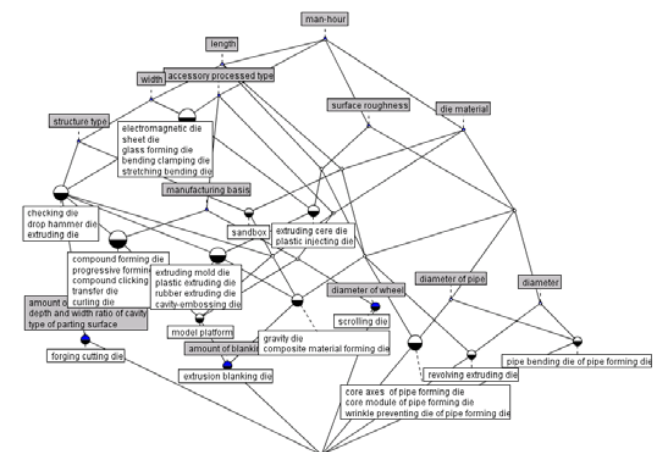Fig. 6 The formal context instance

Fig. 7 The Hasses diagram of concept lattice instance

It can be seen that the approach of FCA-based semi-automatic domain ontology construction is valid and satisficating. Applying it can aid designer to construct OWL domain ontology effectively.

V. CONCLUSION

We have presented a practical approach for constructing ontology-based CBR system semi-automatically. Its advantages are summarized as follows: its architecture is based on two-layer ontology which can make up for the deficiencies of existing methods; knowledge in legacy CBR can be reused by the greatest extent; OWL local ontology can be mapped

from case base automatically; semi-automatic construction of OWL domain ontology can be aided by recommendations and Hasses diagram of concept lattice that are generated by FCA and computing concept equation measure and concept inclusion measure.

In the next work, we will focus on the research about ontology evolution.

REFERENCES

[1] Agnar Aamodt, Enric Plaza, "Case-Based Reasoning:Foundational Issues, Methodological Variations, and System Approaches," *AI Communications.* IOS Press.1994, Vol.7, pp. 39-59

[2] David W. Aha, "The omnipresence of case-based reasoning in science and application," *Knowledge-Based System.* 1998, Vol.11, pp. 261-273.

[3] B. Diaz-Agudo and P. A. Gonzalez-Calero, "An Architecture for Knowledge In-tensive CBR Systems," In *Proceedings of the 5th European Workshop on Advances in Case-Based Reasoning (EWCBR'00)*, volume 1898 of LNCS, pages Springer-Verlag, 2000. , pp. 37-48.

[4] B. Diaz-Agudo and P. A. Gonzalez-Calero, " Knowledge Intensive CBR through Ontologies," In *Procs of the 6ht UK CBR Workshop.* 2001.

[5] Wache, H., V ogele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hubner, S, "Ontology-based integration of information - a survey of existing approaches," In *IJCAI-01 Workshop: Ontologies and Information Sharing,* 2001, pp. 108--117.

[6] Junjie Gao, Guishi Deng. , "The research of applying domain ontology to case-based reasoning system," In *Proceedings of International Conference on Services Systems and Services Management,* Beijing, 2005, pp.1113 - 1117

[7] Man Li, Xiao-yong Du, "Learning ontology from relational database," *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics,* Guangzhou, 18-21 August 2005. pp.3410-3415

[8] I. Astrova, "Extracting Ontologies from Relational Databases," In *Proceedings of the 22nd IASTED International Conference on Databases and Applications (DBA)*, 2004, pp. 56–61

[9] Habegger,B, "Mapping a database into an ontology: an interactive relational learning approach," In *2007 Proceedings of the 23rd International Conference on Data Engineering* pp.1443-1447

[10] Zhuoming Xu, Shichao Zhang,Yisheng Dong, "Mapping between Relational Database Schema and OWL Ontology for Deep Annotation," In *2006 Proceedings of International Conference on Web Intelligence.* pp. 548-552

[11] Ehrig, M., Staab, S, "QOM - quick ontology mapping," In *The Semantic Web – ISWC 2004.* LNCS 3298. Springer-Verlag, Berlin Heidelberg New York pp.683-696

[12] Ganter, B., Wille, R., *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag, New York (1999)

[13] Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L, "Fast computation of concept lattices using data mining techniques," In *2000 Proceedings of 7th International Workshop on Knowledge Representation Meets Databases.* Berlin, Germany pp.129-139.

[14] http://soureforge.net/projects/conexp

[15] http://www.w3.org/TR/2008/WD-owl11-syntax-20080108/

Junjie Gao is a Ph D. student in the Institute of Systems Engineering, Dalian University of Technology. His research interests mainly include knowledge management, semantic web, intelligent information processing etc.

Guishi Deng is a professor, Ph.D. supervisor with the Institute of Systems Engineering, Dalian University of Technology. His research interests mainly include knowledge management, artificial intelligence etc.