International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.3 Issue No.5, pp : 689-693 1 May 2014

High-Level-Architecture-Based Collaborative Engineering Environments By Ontology Fusion

L.Seethala Devi, P.H.Gauthami, P.S.Divya, Mr.K.Kulunthan, Mrs.R.Vijayalatha
Vel Tech Multitech Dr.Rangarajan Dr.Sakunthala Engineering College
seetha279@gmail.com, phgautha@gmail.com, divipemmasanni@gmail.com, kulunthan@veltechmultitech.org, rvijayalatha@gmail.com

Abstract—In high-level-architecture (HLA)-based distributed heterogeneous collaborative engineering environments (CEEs), the construction of federation object model files is time consuming. This paper presents an ontology fusion approach aiming at establishing a common understanding in such collaborative environments. The proposed approach has three steps: ontology mapping, ontology alignment, and ontology merging. Ontology mapping employs a top-down approach to explore all bridge relations between two terms from different ontologies based on bridge axioms and deduction rules. Ontology alignment adopts a bottom-up approach to discover implicit bridge relations between two terms from different domain ontologies based on equivalent inference. Ontology merging generates new collaboration ontology from discovered equivalent bridge relations. It adopts an axiom-based ontology fusion strategy and takes heavy-weighted ontologies into consideration. It can find all the explicit and derived interontology relations. In a typical CEE, the proposed approach has a great potential to improve the efficiency of preparation for HLA-based collaborative engineering processes, reduce the work load for adaptive adjustment of existing platforms, and enhance the reusability and flexibility of CEEs.

Index Terms—Collaborative engineering environments (CEEs), high-level architecture (HLA), ontology, product development.

I. INTRODUCTION

IN INCREASINGLY saturated markets, innovation and product development are essential conditions for the sale of products. Adopting collaborative engineering makes full use of several independent product development systems and enhances their abilities at the same time. However, as a matter of fact, collaborative engineering environments (CEEs) are complicated and comprise various computer-aided engineer- ing (CAE) systems for collaborative design, simulation, and optimization. It involves processes like computer-aided design (CAD) modeling, simulation, and optimization and requires data and information like CAD digital models, CAE analysis, and optimization results [1], [2]. When several independent systems need to be integrated, common understanding among these systems is always a challenge.

High-level architecture (HLA) is a general-purpose architecture for distributed computer simulation systems. Its early development was sponsored by the U.S. Defense Modeling and Simulation Office. In 2000, it was adopted by IEEE as an international standard IEEE 1516 [3]. In its definition, federation is a named set of federate applications and a common federation object model (FOM) that are used as a whole to achieve some specific objectives.

In an HLA-based CEE, federation execution description files describe the data and information exchange standard of a given simulation. They are essential to common understanding among collaborative systems. Within these files, the construction of FOM needs multidisciplinary professional knowledge and technologies [6]. It is always time consuming and expensive.

Fortunately, ontology in knowledge engineering is the semantic basis of communication among domain entities. It is applicable to automatic reasoning, knowledge representation, and reuse. Ontology-based approaches have been used to resolve the problem of heterogeneous data and information integration.The target of this research is to explore a new FOM construction method which takes full advantage of ontology technologies.

II.RELATED WORKS

Although the target of this research is to develop a new FOM construction method in order to take full advantage of ontology technologies, this task is far from just applying existing ontology technologies into CEEs. The semiautomatic construction of FOMs can be considered as an ontology integration problem. Ontology integration is the consequence of ontology heterogeneousness(syntax heterogene -ousness and nonsyntax heterogeneousness). Ontology heterogeneousness can be classified into four layers: terminology, conceptualization, and semantics. In the representation layer, different representation forms are used, and the representation differences can be resolved by formalization. In the terminology layer, different terms are adopted, and the term differences can be resolved by term mapping. In the conceptualization layer, ontology theory always takes effect here. Furthermore, the problems of the semantics layer are hard to resolve . When it comes to CEE, because collaboration participants adopt the same ontology construction tools and language, there is no difference in representation at all. However, because of multidisciplinary-coupled resolutions, regional distribution of organizations, and various participants, heterogeneousness on the terminology, conceptualization, and semantics layers cannot be ignored. That leads to several challenges in applying ontology technologies to CEE.

1) There is no well-established domain ontology to use .

2) Every subsystem is totally equal in position. There is no kernel subsystem. The merging order of capability ontologies should not influence the final merging results. A metastructure should be designed to support separated domain laws and bridge relations.

3) There are significant differences among knowledge representation methods among the subsystems according to a different series of domain laws. These differences cannot be easily eliminated by means of existing ontology technologies. Thus, bridge relations (the relations between related concepts from different representation systems) need to be preset by domain experts. Therefore, the light-weighted ontology approach is not applicable here.

III.THEORETICAL FOUNDATION

A. Definitions

Because the algebraic system defined on the concept set of CEE and the partial-order relations of these concepts have the same upper bound and lower bound, it can be deemed as a concept lattice. The common understanding models of CEE have a common source, a given product model, and any model involved is specialized in some aspects. They also share common metadata, binary stream, and any datum collaboration required is a given parse of a binary fragment. That is to say, the partial-order relations such as part-of or inherit-from have a common ancestor, the product ( ), and all the products have a common ancestor, *Thing*.

Definition 1: CEE Ontology:

$O ::= (C, H_C, R_C, H_R, M, R_M, A)$.

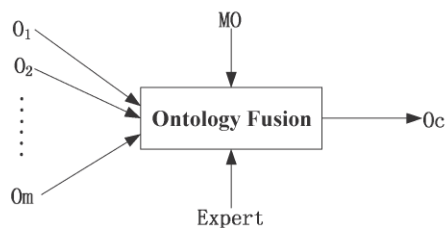
CEE ontology O is defined as seven tuple. C denotes a collaboration concept set of CEE. H_C defines a set of partial orders on concept set C. They are inheritance relations among the concepts involved. The concept sets and the inheritance relations form a directed acyclic graph (DAG) whose source is the given model of collaborative product and whose sink is a binary fragment. R_C denotes a set of non inherited partial-order relations on concept set C, and these partial-order relations correspond to concept attributes. H_R defines inherited relations on the partial-order relation set R_C. M is a series of collaborative product MO concepts that give a series of inheritable instances of R_C. R_M denotes a set of partial-order relations under M; they describe the relations among elements in the MO set, and they are also the basis for collaborative product ontology reasoning. A defines a set of axioms among the ontology concept set and MO relation set, and they provide the major premises of CEE ontology reasoning.

Definition 2: CEE Ontology Fusion:

$fuse ::= SET_O\ O, (\forall c, c \in O \rightarrow \exists f_1, f_2, f_1 \Leftrightarrow f_2, f_1 \in O_{i1}, f_2 \in O_{i2}, O_{i1} \subset SET_O, O_{i2} \subset SET_O : fuse(f_1, f_2, SET_O) = c)$.

CEE ontology fusion is a partial-order mapping from an ontology set of CEE to one ontology (as Fig. 1 shows). Because the fused ontology is for collaboration, only more than one ontology (representing the exchange information needed by a subsystem) employs the same concept, and it is useful for future use. To any term c in the resultant ontology, it can find a corresponding term f_1 in one ontology O_{i1} which can be found in the prepared ontology set. At the same time, to use the term f_1, there must be an equivalent term f_2 in another ontology O_{i2} of the prepared ontology set. Even if the term is unique in different ontologies, it will appear in the fused ontology in a different name (with prefix of the original ontology). The only results of ontology fusion can be one ontology or null.

The CEE ontology fusion procedure involves mapping from a set of ontologies $\{O_1, O_2, \ldots, O_m\}$ to one collaboration ontology O_c. During this process, expert instructions work as



B. Mathematical Properties

This paper uses $a_1 \wedge a_2 \wedge \cdots \wedge a_n$ to denote the maximum lower bound of set $\{a_1, a_2, \ldots, a_n\}$ and $a_1 \vee a_2 \vee \cdots \vee a_n$ to represent its minimum upper bound.

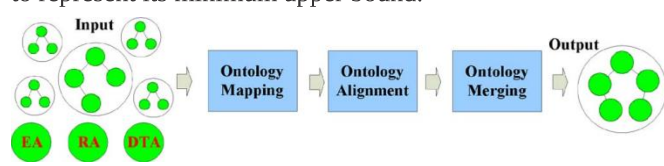


Fig.4.CEEontology fusion framework.

***Theorem 1:* Operations $\wedge$ and $\vee$ on CEE ontology concept lattice C, have properties as follows.**

1) **Idempotent law**: For any $e \in C$, there exist $a \wedge a = a$ and $a \vee a = a$.

2) **Commutative law**: For any $a, b \in C$, there exist $a \wedge b = b \wedge a$ and $a \vee b = b \vee a$.

3) **Associative law**: For any $a, b, c \in C$, there exist $(a \wedge b) \vee c = a \wedge (b \wedge c)$ and
$(a \vee b) \vee c = a \vee (b \vee c)$.

4) **Absorption law**: For any $a, b \in C$, there exist $a \wedge (a \vee b) = a$ and
$a \vee (a \wedge b) = a$.

It can be inferred from the aforementioned laws that the minimum upper bound and the maximum lower bound of any concept in CEE are themselves the concept. This is one of the most effective ways to align ontologies. When seeking the minimum upper bound and the maximum lower bound, the same operation is irrelevant to the order. The maximum lower bound of any concept with its ancestors is itself, and the minimum upper bound of any concept with its descendant is also itself.

Theorem 2: Suppose that C, is a concept lattice of given CEE ontology; is the reverse of relation . For any $a, b, c, d \in C$, the following exist.

1) **Reflexivity**: aa; aa.
2) **Antisymmetry**: ab and $ba \Rightarrow a = b$; ab and $b\ a \Rightarrow a = b$.
3) **Transitivity**: ab and $bc \Rightarrow ac$; $a \quad b$ and $bc \Rightarrow a \quad c$.
4) $a \wedge ba$, $a \vee b \quad a$, $a \wedge bb$, and $a \vee bb$.
5) ca and $ab \Rightarrow ca \wedge b$; ca and $cb \Rightarrow c$ $a \vee b$.
6) $ab \Rightarrow a \wedge b = a \Rightarrow a \vee b = b$.
7) ab and $c \quad d \Rightarrow a \wedge c \quad b \wedge d$; $a\ b$ and $c\ d \Rightarrow a \vee c \_ b \vee d$.
8) **Rank preservation**:
$a \_ b \Rightarrow a \wedge c \_ b \wedge c$;
$a \_ b \Rightarrow a \vee c \_ b \vee c$.
9) **Distribution inequality**:
$a \vee (b \wedge c) \_ (a \vee b) \wedge (a \vee c)$;
$a \wedge (b \vee c) \_ (a \wedge b) \vee (a \wedge c)$.
10) **Norm inequality**:
$a \_ c \Rightarrow a \vee (b \wedge c) \_ (a \vee b) \wedge c$.

IV.ONTOLOGY FUSION ALGORITHM

As Fig. 4 shows, the main parts of ontology fusion include three algorithms: ontology mapping, ontology alignment, and ontology merging.

Algorithm 1. Ontology mapping(*{Om}*, DA)
Input: *{***Om***}* candidate mapping ontology set
DA domain axiom set
Output: EC bridge equivalent concept pair list
IC bridge mutual exclusive concept pair list
1 **foreach** (*Oi*,*Oj*) **in** *{Om}* **do**
2 *EC* ← *{*(*C*(*Oi*)*k* , *C*(*Oj*)*l*)*}*//find domain equivalent bridge axiom from DA
3 *IC* ← ∅/∗Extract equivalent (mutual exclusive) graphs∗/
4*Gi*←Equivalent(Mutual_Exclusive)_Relation_Travel(*Oi*)
5*Gj*←Equivalent(Mutual_Exclusive)_Relation_Travel(*Oj*)
/∗Simplify equivalent (mutual exclusive) graphs by deleting trivial equivalent (mutual exclusive) relations (Thing, data-type equivalent, trivial mutual exclusive relations, and independent concept nodes)∗/
6 *Gi* ← Simplify(*Gi*)
7 *Gj*←Simplify(*Gj*)/∗According to *{*(*C*(*Oi*)*k* ,*C*(*Oj*)*l*)*}*, mark*{*(*C*(*Oi*)*k*)*}* of *Gi*, and mark *{*(*C*(*Oj*)*l*)*}* in *Gj* , iteratively deleteunmarked concepts of zero in-degree and their *m*-out-arc ∗/
8 *Gi*← Bridge_simplify(*Gi*)
9 *Gj*← Bridge_simplify(*Gj*)/∗Inferring bridge equivalent relations, only equivalent graphs of *Gi* and *G_j* , *G*=*i* and *G*=*j* , are used here, and all the discussions below are all based upon structure equivalent relations∗/
10 **foreach** unmarked concept *Ci* **in** *G*=*i* **do**
11 **if** (∃ one–one bridge equivalent relation between ancestor concepts of *Ci* and any concept of *Gj, Cj*) **then**
12 *EC*←*EC*+(*Ci, Cj*)//duplicate elements eliminated
13 **elseif** (∃ one–one bridge equivalent relation between ancestor concepts of *Ci* and any concept of *Gj, Cj* , and the attributes, constraints, partial order relations between concept and its attributes are also equal, the concepts in constraint path also have corresponding equivalent bridge concepts.) **then**
14 *EC* ← *EC* + (*Ci, Cj*)
15 **end if**
16 **end**/∗Inferring bridge mutual exclusive relations∗/
17 **foreach** (*C*(*OI*)*p* , *C*(*OI*)*q*) **in** *{*(*C*(*OI*)*p* , *C*(*OI*)*q*)*I* ∈ (*i, j*)*}* of *Gi* or *Gj* **do**
18 **foreach** equivalent concept of *C*(*OI*)*q* , *C*(*O˜I*)*q*_**in** EC **do**
19 *IC* ← *IC* ∪*{*(*C*(*OI*)*p* × *C*(*O˜I*)*q*_)*}*
⊥//**A**ccording to the structure graph *GOI* of ontology//*OI*{*C*(*OI*)*p* |*GOI*}⊥ is concept *C*(*OI*)*p* and all of its descendents, the//same as *{C*(*O˜I*)*q*_ |*GO˜I}*⊥.
20 **end**
21 **foreach** equivalent concept of *C*(*OI*)*p* , *C*(*O˜I*)*p*_ **in** EC **do**
22 *IC* ← *IC* ∪*{C*(*O*)*p*_ , *C*(*OI*)*q* }⊥
23 **end**
24 **end**
25 **end**
26 **return** EC, IC

This algorithm is defined on the equivalent-structure-graph-based knowledge representation and an attribute- group-comparison-based merging mechanism. Compared with lightweight ontology merging which is based only on structure and terms, the main advantage of this algorithm is that, when merging ontologies, the heuristic information, such as the equivalent structure graph and semantic equivalence of the attribute, is also taken into consideration. The efficiency and accuracy of ontology merging have been greatly improved.

Algorithm 2. Ontology_alignment(*{Om}*, EC, IC, DTA)
Input: *{***Om***}* candidate mapping ontology set
EC bridge equivalent concept pair list
IC bridge mutual exclusive concept pair list
DTA equivalent data type axiom set
Output: EC
1 **foreach** (*Oi*,*Oj*) **in** *{Om}* **do**//Extract structure graphs
2 *Gi* ← Travel(*Oi*)
3 *Gi* ← Travel(*Oj*)
4 *{*(*SC*(*Oi*), *SC*(*Oj*))*}* ← *OCi*× *OCj* //Cartesian product of concept set in *Oi* and *Oj* /∗According to mutual exclusive bridge relations, simplify *{*(*SC*(*Oi*), *SC*(*Oj*))*}* ∗/
5 **if** (∃*{*(*IC*(*OI*)*m* , *IC*(*O˜I*)*m*)*}* in IC and *IC*(*OI*)*m*≡ *SC*(*OI*)*k*)**then**//*{IC*(*OI*)*m*|*GOI}*⊥ is concept *IC*(*OI*)*m* and all of its descendants//according to the structure graph *GOI* of *OI*, the same as//*{IC*(*O˜I*)*m* |*GO˜I}*⊥.
6 *{*(*RIC*(*Oi*),*RIC*(*Oj*))*}* ← *{*(*SC*(*Oi*), *SC*(*Oj*))*}* −*{*(*IC*(*OI*)*m*× *IC*(*O˜I*)*m*)*}*⊥

International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.3 Issue No.5, pp : 689-693 1 May 2014

7 **end if**/∗According to equivalent bridge relations, simplify{(*RIC*(*Oi*),*RIC*(*Oj*))} ∗/
8 **if** (∃{(*EC*(*OI*)*m* ,*EC*(*O˜I*)*m*)} in EC and *EC*(*OI*)*m*≡*RIC*(*OI*)*k*) **then**//{*EC*(*OI*)*m*|*GOI*}_
is concept *EC*(*OI*)*m* and all its ancestors according//to the structure graph *GOI* of *OI*, the same as {*EC*(*O˜I*)*m* |*GO˜I*}⊥
9 {(*REC*(*Oi*),*REC*(*Oj*))} ←{(*RIC*(*Oi*),*RIC*(*Oj*))} − {(*IC*(*OI*)*m*)_× (*IC*(*O˜I*)*m*)⊥}
10 **end if**/∗Inferring equivalent bridge relations.∗/
11 {(*RMC*(*Oi*),*RMC*(*Oj*))} ← {(*REC*(*Oi*),*REC*(*Oj*))}
12 **foreach** (*REC*(*OI*)*m* ,*REC*(*O˜I*)*n*) **in** {(*REC*(*Oi*),*REC*(*Oj*))} **do**
13 **if** (data-type construction is different according to data-type meta class definition of meta ontology) **then**//the difference of data-type construction includes data type unit//number inconsistency and data type inheritable
14 {(*RMC*(*Oi*),*RMC*(*Oj*))} ={(*RMC*(*Oi*),*RMC*(*Oj*))} − (*REC*(*OI*)*m* ,*REC*(*O˜I*)*n*)
15 **end if**
16 **end**
17{(*RUC*(*Oi*),*RUC*(*Oj*))} ← Confirmed({(*RMC*(*Oi*),*RMC*(*Oj*))})//domain expert confirmation
18 **end**

19 **return** *EC* ← *EC* ∪{(*RUC*(*Oi*),*RUC*(*Oj*)), *i* < *j*}
This algorithm is defined on the structure-graph-based knowledge representation and an attribute-set-comparison based bottom-up inferring mechanism.The output of ontology alignment algorithms is term relations, a set of term pairs in different ontologies. This paper proposes the heuristic information-based (attributes equal) semiautomatic bottom-up ontology alignment method. Because of heuristic information involved, the algorithm has a risk of making wrong judgments, so it needs a domain expert to confirm the candidate-equivalent bridge relations.
Algorithm 3. Ontology_merging({*Om*}, EC)
Input: {**Om**} candidate mapping ontology set
EC bridge equivalent concept pair list
Output: FON
1 **foreach** *Oi* **in** {*Om*} **do**//simplify structure graph of *Oi* according to bridge equivalent//concept pair list EC
2 *Gi* → Equivalent_travel(*EC*,*Oi*)
3 **end**
4 *FON*≤1 → *G*1 + *Thing* (as top concept)/∗Merge *Gj* ∈{*Gm*−1} and *FON*≤*j*−1 in turns∗/
5 **foreach** *Gj* **in** {*Gm*−11} **do**
6 **let** *ECj* = {(*C*(*O*≤*j*−1), *C*(*Oj*))} in EC
7 **let** *CGjt* = top node of *Gj*
8 **if** (∃(*CG*≤*j*−1*x* ,*CGjt*) in *ECj*) **then**
9 Add *CGj t* into bridge equivalent concept chain of *CG*≤*j*−1*x*
10 *ECj* → *ECj* − (*CG*≤*j*−1*x* ,*CGjt*)
11 **else**
12 Add *CGjt* into *FON*≤*j*−1 as a direct child of *Thing*
13 **end if**
14 **foreach** node *CGj t* **in** Breadth_first_travel (*Gj* ,*CGjt*)**do**
15 **if**(∃(*CG*≤*j*−1*x* , *CGjt*) in *ECj*) **then**
16 Add *CGjt* into bridge equivalent concept chain of *CG*≤*j*−1*x*
17 *ECj* → *ECj* − (*CG*≤*j*−1*x* ,*C***G***jt*)
18 **else**
19 Add *CGjt* into *FON*≤*j* as a brother node of *Gj* ,and add all the in-arc of *Gj*
20 **end if**
21 **end**
22 **end**
23 **return** *FON*(*FON*≤*m*)
V.CASE STUDY

A typical CEE unit can be described as having three main parts (Fig. 5): scene creator, localization server, and decision support system. The scene creator generates scenes for collaboration jobs. The localization server keeps tracking im- portant entities, and the decision support system makes the final decision or suggestions about a future collaboration step. The collaboration information among these three independent systems lies in the scene provided by the scene creator, local- ization information supported by the localization server, and of inferences. It contains object templates, basic transformation data types, and necessary general axioms. This approach is compatible with the definition of HLA object model template object classes and interaction classes.

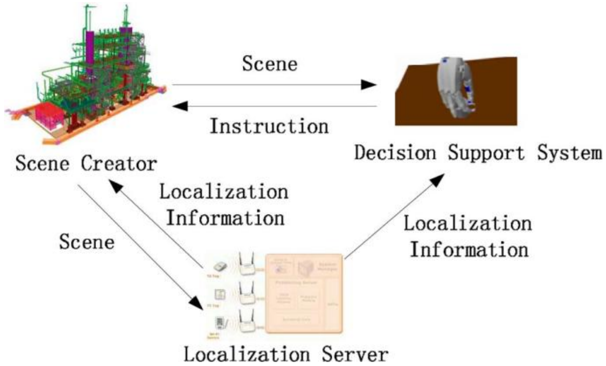


Fig. 5. Typical CEE unit
In HLA-based CEEs, it is difficult to establish a collaborative system, but it is even more difficult to adaptively adjust interface codes of existing systems and to negotiate among multidisciplinary domains [22]. This paper has proposed a semiautomatic ontology fusion method to establish a collaborative ontology as media in HLA-based CEEs, which is constructed by several independent subsystems. These subsystems may be working in different domains. The main part of this method includes three algorithms: ontology mapping, ontology alignment, and ontology merging.
Despite its complexity, it enjoys several remarkable advantages which are more suitable for HLA-based multidisciplinary collaborations.

1) This method has firm theoretical foundations and starts from strict definitions.
2) Different from most other ontology integration methods using literature distance, this method employs axioms, bridge axioms, equality rules, and attribute set equality conditions as the basis for reasoning.

3) The proposed algorithms can find all the explicit and derived bridge relations.
4) Since ontology is used in this method, the reuse of resources and the expandability of existing systems are greatly enhanced.

The applicability of the proposed method has been demonstrated through a case study. More efforts are still required in order to improve the proposed method for real industrial applications.

References

i. *W. Fan, W. Wang, and T. Xiao, "Multidisciplinary collaboration simu- lation optimization platform for complex product design," in Proc. 2nd ICPCA, Birmingham, UK, Jul. 2007, pp. 174–178.*

ii. *H. Sun, T. Xiao, and S. Tang, "Research on federation-based prag- matic integration framework," in Proc. World Congr. CSIE, Los Angeles/Anaheim, CA, Apr. 2009, pp. 535–539.*

iii. *IEEE Standard for Modeling and Simulation High Level Architecture (HLA)-Framework and Rules, IEEE Std. 1516-2000.*

iv. *Ontology matching By Jeþro^me Euzenat, Jérôme. Euzenat, Je´ro^me Euzenat, Pavel Shvaiko*

v. *S. Tang, T. Xiao, and W. Fan, "A collaborative platform for complex product design with an extended HLA integration architecture," Simul.Modelling Pract. Theory, vol. 18, no. 8, pp. 1048–1068, Sep. 2010.*

vi. *Handbook on Ontologies by Steffen Staab, Rudi Studer*