Ontology fusion in high-level-architecture-based collaborative engineering environments

Sun, Hongbo; Fan, Wenhui; Shen, Weiming; Xiao, Tianyuan

Publisher's version / Version de l'éditeur:
https://doi.org/10.1109/TSMCA.2012.2190138
IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems And Humans, PP, 99, pp. 1-12, 2012-05-01

Ontology Fusion in HLA-based Collaborative Engineering Environments

Hongbo Sun[1,2], *Member, IEEE*, Wenhui Fan[2], Weiming Shen[1], *Senior Member, IEEE*, and Tianyuan Xiao[2]

[1] Centre for Computer-assisted Construction Technologies, National Research Council, London, Ontario N6G 4X8, Canada
[2] National CIMS Engineering Research Centre, Tsinghua University, Beijing 100084, China

***Abstract*—In HLA (High Level Architecture)-based distributed heterogeneous collaborative engineering environments (CEEs), the construction of FOM (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 a 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 inter-ontology relations. In a typical collaborative engineering environment, 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 collaborative engineering environments. A case study has been conducted to validate the feasibility of the proposed approach.**

***Index Terms*—Collaborative Engineering Environments, Product Development, HLA - High Level Architecture, Ontology.**

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. But as a matter of fact, collaborative engineering environments (CEEs) are complicated and comprise various computer-aided engineering (CAE) systems for collaborative design, simulation, and optimization. It involves processes like CAD modeling, simulation and optimization, 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. It early development was sponsored by US 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.

Since federates exist within a federation in the form of data abstraction, federated integration keeps well the independency of its participants. This kind of integration is more suitable for and is widely used in distributed and loosely coupled simulation integration. The owner of each participant does not need to worry about exposing too much private information. The federation only defines the interesting domains for given objectives and the rules of inter operations. It is a real loosely-coupled integration solution. Within a federation, subsystems collaborate in an indirect way so that the context of interoperation can be taken into consideration.

Nowadays more and more simulation functions have been added into collaborative product development [4]. The design of product can be deemed as a multi-step process in which a set of design goals and requirements are transformed into a functional system. Simulation functions help these systems fulfill their design goals and add to their potential values.

When simulation is added into a collaborative product development environment, there always exist several subsystems in the same environment with independent design goals. These subsystems may follow different design or management rules in their respective engineering fields [5].

In an HLA-based CEE, FED (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 (Federation Object Model) needs multidisciplinary

Manuscript received October 20, 2010. Corresponding author: H. Sun (e-mail:hsun@ieee.org).
Digital Object Identifier inserted by IEEE

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 [7].Ontology-based approaches have been used to resolve the problem of heterogeneous data and information integration [8,9]. The target of this research is to explore a new FOM construction method which takes full advantage of ontology technologies.

From the viewpoint of reasoning ability, ontology can be briefly classified into two categories: light-weighted ontology and heavy-weighted ontology. Light-weighted ontology does not have the ability of reasoning. It is in fact a well-organized vocabulary. While heavy-weighted ontology has reasoning abilities, such as first-order predicate. It usually includes axiom or rule definitions for reasoning use. In this paper, a semi-automatic construction method of FOM files is proposed. Because it builds collaboration ontology from exchanging data ontologies of subsystems, and explores new bridge relations, it is deemed as a heavy-weighted ontology fusion algorithm. The rest of the paper is organized as follows. Section 2 reviews the related literature and analyses the requirements of this application problem. Section 3 discusses the theoretical foundations of the proposed method. Section 4 describes the algorithms supporting the proposed method. Three algorithms are introduced in this paper: ontology mapping, ontology alignment, and ontology merging. Section 5 depicts a typical collaborative engineering environment involving three systems to demonstrate the applicability of the proposed method. Section 6 provides conclusions and a further research plan concerning this research topic. A brief complexity analysis is also included in this section.

II. Related Work

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 semi-automatic construction of FOMs can be considered as an ontology integration problem. Ontology integration is the consequence of ontology heterogeneousness (syntax heterogeneousness and non-syntax heterogeneousness [9]). Ontology heterogeneousness can be classified into four layers: representation, 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. And the problems of semantics layer are hard to resolve [10]. 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:

- There is no well-established domain ontology to use [11].
- 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 meta-structure should be designed to support separated domain laws and bridge relations.
- There are significant differences among knowledge representation methods among the subsystems according to a different series of domain laws [12, 7]. These differences cannot be easily eliminated by means of existing ontology technologies. So 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.

All the factors mentioned above bring difficulties when applying existing methods to this problem. Most well-known ontology integration tools are not applicable here, including PROMPT [12], OntoMerge [13], MAFRA [14], GLUE [15] and OntoMap [16]. Some of them are built on literal-based similarity computing methods (e.g., OntoMerge, PROMPT, ONION, and Anchor-PROMPT), while others are too simple and weak in their description abilities (e.g., OntoMap). Some are instances-based merging (e.g., GLUE); others only adopt bridge axioms. There are also some methods that only take terms and structures of light-weighted ontologies into consideration [17]. When it comes to ontology construction, although formal concept analysis (FCA) [18] can be successfully applied to ontology construction, in the procedure of ontology fusion it is not convenient to use. If this method is adopted, the formal background must be recomputed before ontology merging. This computation brings too much work to CEE, and the computation of multidisciplinary formal backgrounds is difficult.

The main idea presented in this paper is to take full advantage of formalization to automatically discover implicit and explicit bridge relations among term pairs of different ontologies; then to establish the union of the equivalent term pairs as the collaborative ontology; and, at the same time, construct a well-defined meta-ontology in order to facilitate interoperations among independent subsystems. The process of building the output collaborative ontology from independent domain ontologies is called CEE ontology fusion.

The main challenges in ontology fusion include instances and concepts confusion, top concepts correspondences, modeling habits differences, synonyms, and coding formats [9]. In an HLA-based collaborative engineering environment, the instances are created by restrict definitions of exchange information format. The instances are not confused with concepts. The foundation of this method is not literal semantic distance, so the problem of using similar words has no effect here. Since the data types used are defined in meta-ontology, the coding format is unique. Because the top concept is the given product, there is no doubt that the top concept is unique. In the scope of one collaboration project, collaboration ontologies are built in the same way by the same group of people, so there are no differences in modeling habits. Lastly, no matter what approach is adopted, synonym problems always need to be addressed by domain experts.

III. Theoretical Foundation

Can we use ontology as a common understanding media for different disciplinary systems? Theoretically speaking, we can. Because, on one hand, the objective of CEE at one time is unique; every established model is describing some aspects of the same thing. On the other hand, the theory of concept lattice (Galois lattice) [19] provides solid ground for this method.

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 [20]. 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 a common meta-data, 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 ($\top$), and all the products have a common ancestor, *Thing*. Also the minimum original concept ($\perp$) defined under these partial orders is *binary characters*; and it is also the public descendants of these concepts. Based on this, the paper defines related concepts as follows:

Definition 1: CEE ontology

$$\mathrm{O} ::= (\mathrm{C}, \mathrm{H_C}, \mathrm{R_C}, \mathrm{H_R}, \mathrm{M}, \mathrm{R_M}, \mathrm{A})$$

CEE ontology O is defined as a seven tuple. C denotes a collaboration concept set of CEE. $\mathrm{H_C}$ 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) which source is the given model of collaborative product and whose sink is a binary fragment. $\mathrm{R_C}$ denotes a set of non-inherited partial order relations on concept set C, and these partial order relations correspond to concept attributes. $\mathrm{H_R}$ defines inherited relations on the partial order relation set $\mathrm{R_C}$. M is a series of collaborative product meta-ontology concepts that give a series of inheritable instances of $\mathrm{R_C}$. $\mathrm{R_M}$ denotes a set of partial order relations under M; they describe the relations among elements in the meta-ontology set, and they are also the basis for collaborative product ontology reasoning. A defines a set of axioms among the ontology concept set and meta-ontology relation set and they provide the major premises of CEE ontology reasoning.

Definition 2: CEE ontology fusion

$$\mathrm{fuse} ::= \mathrm{SET_O} \rightharpoonup \mathrm{O},$$
$$(\forall \mathrm{c}, \mathrm{c} \in \mathrm{O} \rightarrow \exists \mathrm{f}_1, \mathrm{f}_2, \mathrm{f}_1 \Leftrightarrow \mathrm{f}_2, \mathrm{f}_1 \in \mathrm{O}_{\mathrm{i}_1},$$
$$\mathrm{f}_2 \in \mathrm{O}_{\mathrm{i}_2}, \mathrm{O}_{\mathrm{i}_1} \subset \mathrm{SET_O}, \mathrm{O}_{\mathrm{i}_2} \subset \mathrm{SET_O}: \mathrm{fuse}(\mathrm{f}_1, \mathrm{f}_2, \mathrm{SET_O}) = \mathrm{c})$$

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, 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 $\mathrm{O}_{\mathrm{i}_1}$ 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 $\mathrm{O}_{\mathrm{i}_2}$ of the prepared ontology set. Even the term is unique in different ontologies, it will appears 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.

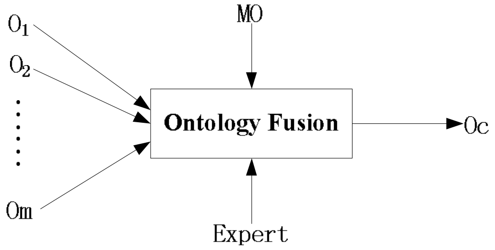


Fig.1. Collaborative engineering environments ontology fusion

The CEE ontology fusion procedure involves mapping from a set of ontologies $\{\mathrm{O}_1, \mathrm{O}_2, \ldots, \mathrm{O}_\mathrm{m}\}$ to one collaboration ontology $\mathrm{O_c}$. During this process, expert instructions work as the mechanism and meta-ontology MO controls the whole ontology fusion process.

The ontology fusion method introduced here includes three steps: mapping, alignment, and merging.

Definition 3: CEE ontology mapping

$$\mathrm{map} ::= \mathrm{E_O} \rightharpoonup \mathrm{E_O},$$
$$(\forall \mathrm{e} \in \mathrm{O}_1 \rightarrow \exists \mathrm{f} \in \mathrm{O}_2 :$$
$$\mathrm{map}(\mathrm{e}, \mathrm{O}_1, \mathrm{O}_2) = \mathrm{f} \vee \mathrm{map}(\mathrm{e}, \mathrm{O}_1, \mathrm{O}_2) = \perp).$$

It is a partial order mapping based on vocabulary E of one ontology to vocabulary E of another ontology. The term $\mathrm{e} \in \mathrm{E}$ may be concept, relation or instance. The mapping may be an equivalent relation, inherit relation or ownership. This mapping is not transmissible except that it describes equivalent relations. So, in the common sense, this mapping does not involve more than two heterogeneous ontologies.

Definition 4: CEE ontology alignment

$$\mathrm{align} ::= \mathrm{E_O} \times \mathrm{E_O} \rightarrow [0, 1],$$
$$\mathrm{align}(\mathrm{e}, \mathrm{f}) = 1 \Leftrightarrow \mathrm{e} = \mathrm{f}\text{: two concepts equal}$$
$$\mathrm{align}(\mathrm{e}, \mathrm{f}) = 0 \Leftrightarrow \mathrm{e} \neq \mathrm{f}\text{: two concepts not equal}$$

The CEE ontology alignment function involves an equivalent mapping based on the concept vocabulary E of two different ontologies.

Definition 5: CEE ontology merging

$$\mathrm{merge} ::= \mathrm{SET_O} \rightharpoonup \mathrm{O},$$
$$(\forall \mathrm{e}, \mathrm{e} \in \mathrm{O} \rightarrow \exists \mathrm{f}, \mathrm{f} \in \mathrm{O_{i,}}\, \mathrm{O_i} \subset \mathrm{SET_O}: \mathrm{merge}(\mathrm{f}, \mathrm{SET_O}) = \mathrm{e})$$

It represents partial order mapping from an ontology set ($\mathrm{SET_O}$) of CEE to one ontology O. Any term $\mathrm{e} \in \mathrm{E}$ in the output ontology, no matter whether it is a concept, a relation or an instance, it must have a corresponding term f in one ontology of the prepared merging ontology set.

Equivalent graph is a directed graph (may be an unconnected one). It represents the equivalent relations of concepts in an ontology. In an ontology it is not possible for two concepts to be equal (they should be one concept with alias). Equivalent graph describes equivalent relations between one concept and a group of concepts or attributes. These equivalent relations can be classified into two categories: structure equivalent and description equivalent relations. The structure equivalent relation is comprised of a group of one-to-many equivalent relations defined on the partial order relations (inheritance) of the concepts or their attributes. There are two sub categories of structure equivalent relations: inheritance equivalent relation and division equivalent relation (see definitions below). The description equivalent relation is comprised of a group of one-to-many equivalent relations defined on the partial order relations (ownership) between one concept and a set of its attributes. One-to-one equivalent relations are included here, that means an attribute of the concept can be deemed as its identifier. What is more, the description equivalent relation has quantifier constraints.

Definition 6: inheritance equivalent relation

$$\mathrm{C_i} \Leftrightarrow \mathrm{C_{j_1}} \cap \mathrm{C_{j_2}} \cap \dots \cap \mathrm{C_{j_m}}$$

The inheritance equivalent relation describes the relation of a given sub concept $\mathrm{C_i}$ that can be uniquely determined by a group of ancestor concepts $\{\mathrm{C_{j_1}}, \mathrm{C_{j_2}}, \dots, \mathrm{C_{j_m}}\}$. To a given concept $\mathrm{C_i}$, there may be no inheritance equivalent relation concept set, or many equivalent relation concept sets.

Definition 7: division equivalent relation

$$\mathrm{C_i} \Leftrightarrow \mathrm{C_{j_1}} \cup \mathrm{C_{j_2}} \cup \dots \cup \mathrm{C_{j_m}}$$

The division equivalent relation describes a group of complete divisions $\{\mathrm{C_{j_1}}, \mathrm{C_{j_2}}, \dots, \mathrm{C_{j_m}}\}$ of concept $\mathrm{C_i}$. That is to say, any instance of $\mathrm{C_i}$ can find a corresponding instance of $\{\mathrm{C_{j_1}}, \mathrm{C_{j_2}}, \dots, \mathrm{C_{j_m}}\}$. Specifically, the correspondence is not required to be unique here. To a given concept $\mathrm{C_i}$, the division equivalent relation may or may not exist.

All of these equivalent relations mentioned above can be denoted by an m-in-arc with one degree (m is greater than 1), as shown in Fig. 2.

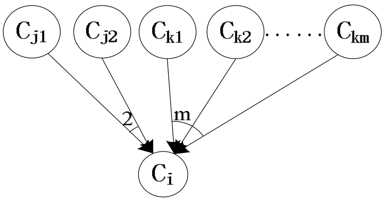


Fig. 2. Equivalent relations denotation

Definition 8: description equivalent relation

$$\mathrm{C_i} \Leftrightarrow \theta_1 \mathrm{R_{k_1}}.\mathrm{C_{j_1}} \cap \theta_2 \mathrm{R_{k_2}}.\mathrm{C_{j_1}} \cap \theta_3 \mathrm{R_{k_x}}.\mathrm{C_{j_y}} \dots \cap \theta_m \mathrm{R_{k_n}}.\mathrm{C_{j_m}},$$
$$\theta_o = \forall|\exists|\urcorner| = p| \le p| \ge p, p \in N^+, o \in 1 \dots \mathrm{m}$$

This gives the semantic equivalent relations between a concept $\mathrm{C_i}$ and a group of constrained attributes $\{\theta_1 \mathrm{R_{k_1}}.\mathrm{C_{j_1}}, \theta_2 \mathrm{R_{k_2}}.\mathrm{C_{j_1}}, \theta_3 \mathrm{R_{k_x}}.\mathrm{C_{j_y}} \dots, \theta_m \mathrm{R_{k_n}}.\mathrm{C_{j_m}}\}$. The constrained attribute group contains a set of attributes $\theta_o \mathrm{R_{k_x}}.\mathrm{C_{j_y}}$ constrained by description logic. $\mathrm{C_{j_y}}$ denotes a given concept. $\mathrm{R_{k_x}}$ denotes the partial order relation R from $\mathrm{C_i}$ to $\mathrm{C_{j_y}}$ and θ_o represents the constraints. The constraints include quantifier constraints and numeric constraints. The universal quantifier constraint $\forall$ represents the relation R value range and are all the instances of $\mathrm{C_{j_y}}$; the existential quantifier constraint $\exists$ means that relation R has at least one corresponding value in instances of concept $\mathrm{C_{j_y}}$; the negative constraint $\urcorner$ represents the fact that there is no correspondence of relation R in $\mathrm{C_{j_y}}$ instances, and numeric constraint $(= p| \le p| \ge p)$ says relation R owns a $\mathrm{C_{j_y}}$ instance number of $= p, \le p$ or $\ge p$; p is a non-zero natural number. As a matter of fact, description equivalent relations and inheritance equivalent relations are often used together. Hence, descriptions equivalent relations are described as follows:

$$\mathrm{C_i} \Leftrightarrow \mathrm{C_{j_1}} \cap \theta_2 \mathrm{R_{k_2}}.\mathrm{C_{j_1}} \cap \mathrm{C_{j_y}} \dots \cap \theta_m \mathrm{R_{k_n}}.\mathrm{C_{j_m}}$$

In an equivalent graph, the constrained attributes are denoted by an identifier on the m-in-arc, as shown in Fig. 3.

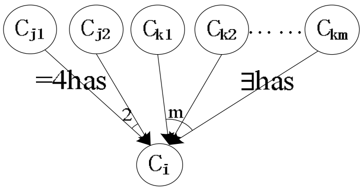


Fig. 3. Mixed Equivalent relation denotation

Equivalent and mutual exclusive graph is an enhanced graph G' based on equivalent graph G with the exclusive relations added (no longer a DAG). The mutually exclusive relation between concepts $(\mathrm{C_i}, \mathrm{C_j})$ in CEE ontology is a symmetrical relation, and any instance of $\mathrm{C_i}$ and its sub concepts will not be the instance of $\mathrm{C_j}$ and its sub concepts. The equivalent and mutual exclusive graph denotes these relations by ↔ between $\mathrm{C_i}$ and $\mathrm{C_j}$. One mutual exclusive relation may contain another one. In that case, two ancestor concepts of mutual exclusion imply descendant concepts mutual exclusion. This mutual exclusive relation is described as a trivial mutual exclusive relation.

Structure graph is a graphical representation of inheritance relations $\mathrm{H_C}$ in a CEE ontology $\mathrm{O} ::= (\mathrm{C}, \mathrm{H_C}, \mathrm{R_C}, \mathrm{H_R}, \mathrm{M}, \mathrm{R_M},$ $\mathrm{A})$. It is a DAG with only one source, and it denotes the inheritance relations among concepts.

Definition 9: equivalent relation bridge

$$\mathrm{C_k^{(O_i)}} \Leftrightarrow \mathrm{C_l^{(O_j)}}$$

An equivalent relation bridge describes the equivalent relation between one term $\mathrm{C_k^{(O_i)}}$ in ontology $\mathrm{O_i}$ and another term $\mathrm{C_l^{(O_j)}}$ of a different ontology $\mathrm{O_j}$. It presents itself in the form of a term pair ($\mathrm{C_k^{(O_i)}}$, $\mathrm{C_l^{(O_j)}}$). An equivalent relation bridge is the main concept relation to be found in CEE ontology mapping and ontology alignment.

Domain equivalent bridge axioms refer to a group of semantic equivalent relations $(C_k^{(O_i)}, C_l^{(O_j)})$ from a concept set $C^{(O_i)}$ of ontology O_i to a concept set $C^{(O_j)}$ of ontology O_j, and they are the foundation of inference or reasoning in CEE ontology mapping.

B. Mathematical Properties

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

Theorem 1: operations $\wedge$, $\vee$ on CEE ontology concept lattice $< C, \preccurlyeq >$ have properties as follows:

Idempotent law:

any $a \in C$, there exist $a \wedge a = a$, $a \vee a = a$

Commutative law:

any $a, b \in C$, there exist $a \wedge b = b \wedge a$, $a \vee b = b \vee a$

Associative law:

any $a, b, c \in C$, there exist $(a \wedge b) \wedge c = a \wedge (b \wedge c)$,

$(a \vee b) \vee c = a \vee (b \vee c)$

Absorption law:

any $a, b \in C$, there exist $a \wedge (a \vee b) = a$,

$a \vee (a \wedge b) = a$

It can be inferred from the above 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 $< C, \preccurlyeq >$ is a concept lattice of given CEE ontology, $\succcurlyeq$ is the reverse of relation $\preccurlyeq$. Any $a, b, c, d \in C$, there exist

- **reflexivity:** $a \preccurlyeq a$; $a \succcurlyeq a$
- **anti-symmetry:** $a \preccurlyeq b$ and $b \preccurlyeq a \Rightarrow a = b$;
  $a \succcurlyeq b$ and $b \succcurlyeq a \Rightarrow a = b$
- **transitivity:** $a \preccurlyeq b$ and $b \preccurlyeq c \Rightarrow a \preccurlyeq c$;
  $a \succcurlyeq b$ and $b \succcurlyeq c \Rightarrow a \succcurlyeq c$
- $a \wedge b \preccurlyeq a$; $a \vee b \succcurlyeq a$; $a \wedge b \preccurlyeq b$; $a \vee b \succcurlyeq b$
- $c \preccurlyeq a$ and $c \preccurlyeq b \Rightarrow c \preccurlyeq a \wedge b$;
  $c \succcurlyeq a$ and $c \succcurlyeq b \Rightarrow c \succcurlyeq a \vee b$
- $a \preccurlyeq b \Rightarrow a \wedge b = a \Rightarrow a \vee b = b$
- $a \preccurlyeq b$ and $c \preccurlyeq d \Rightarrow a \wedge c \preccurlyeq b \wedge d$;
  $a \preccurlyeq b$ and $c \preccurlyeq d \Rightarrow a \vee c \preccurlyeq b \vee d$
- **rank preservation:** $a \preccurlyeq b \Rightarrow a \wedge c \preccurlyeq b \wedge c$;
  $a \preccurlyeq b \Rightarrow a \vee c \preccurlyeq b \vee c$
- **distribution inequality:**
  $a \vee (b \wedge c) \preccurlyeq (a \vee b) \wedge (a \vee c)$;
  $a \wedge (b \vee c) \succcurlyeq (a \wedge b) \vee (a \wedge c)$
- **norm inequality:** $a \preccurlyeq c \Rightarrow a \vee (b \wedge c) \preccurlyeq (a \vee b) \wedge c$

It can be easily inferred from Theorem 2 that the concept lattice of CEE consists of two abelian monoids with the same basic elements. The theorems above are an important source of basic axioms required by CEE ontology reasoning.

IV. ONTOLOGY FUSION ALGORITHM

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

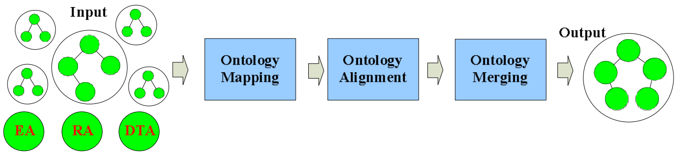


Fig. 4. CEE ontology fusion framework

The purpose of ontology mapping is to mark definite equivalent relations and mutual exclusive relations of any term pair from two different ontologies in a collaborative ontology set. These relations are defined at the semantic level, and they can only be acquired by axioms (universal axioms, user defined axioms and data-type transformation axioms) or deductions from these axioms. The overlapping relations will be inferred and confirmed by the experts in the next step. Different from Quick Ontology Matching (QOM, ontology mapping is a part of ontology matching) which is based on literal semantic distance, this paper introduces an equivalent (mutual exclusive) graph-based domain axiom mapping algorithm.

Because (semi)automatic ontology alignment is a key issue of interoperation among ontologies [21], equivalent relations among terms should be discovered before interoperations as many as possible. The objective of ontology alignment is to infer all the equivalent relations of term pairs in which two terms are from different ontologies. Because the ontology may not be constructed in a sufficiently complete way, this procedure needs domain expert instructions.

CEE collaboration ontology merging is an important basis for collaboration among several ontologies. It corresponds to the procedure of negotiations to form a collaborative FOM among several Simulation Object Models (SOMs) in an HLA framework. This paper introduces an equivalent structure and a concept equivalent, bridge-based CEE ontology merging algorithm. The main task is to reorganize the equivalent terms which are found in the procedure of ontology alignment and ontology mapping, and then form the collaboration ontology, Federation ONtology (FON, the same as FOM in HLA).

The first step of ontology fusion is ontology mapping. This creates a bridge equivalent concept pair list and the bridge mutual exclusive concept pair list of a given ontology set, based on domain axioms. It can be described by algorithm 1.

```
Algorithm 1. Ontology_mapping({$O^m$}, DA)
Input:      {$O^m$} candidate mapping ontology set
```

```
         DA   domain axiom set
Output:  EC   bridge equivalent concept pair list
         IC   bridge mutual exclusive concept pair list
foreach $(O_i,O_j)$ in $\{O^m\}$ do
  EC ← $\{(C_k^{(O_i)}, C_l^{(O_j)})\}$
        // find domain equivalent bridge axiom from DA
  IC ← $\emptyset$
  /*Extract equivalent (mutual exclusive) graphs */
  $\widetilde{G}_i$ ← Equivalent(Mutual_Exclusive)_Relation_Travel($O_i$)
  $\widetilde{G}_j$ ← Equivalent(Mutual_Exclusive)_Relation_Travel($O_j$)
  /*Simplify equivalent (mutual exclusive) graphs by deleting trivial
  equivalent (mutual exclusive) relations (Thing, data type equivalent,
  trivial mutual exclusive relations and independent concept nodes)*/
  $G_i$ ← Simplify($\widetilde{G}_i$)
  $G_j$ ← Simplify($\widetilde{G}_j$)
  /*According to $\{(C_k^{(O_i)}, C_l^{(O_j)})\}$, mark $\{C_k^{(O_i)}\}$ of $G_i$, and mark $\{C_l^{(O_j)}\}$
  in $G_j$, iteratively delete un-marked concepts of zero in-degree and
  their m-out-arc */
  $G'_i$ ← Bridge_simplify($G_i$)
  $G'_j$ ← Bridge_simplify($G_j$)
  /*Inferring bridge equivalent relations, only equivalent graphs of $G'_i$
  and $G'_j$, $G'^{=}_i$ and $G'^{=}_j$, are used here and all the discussions below are
  all based upon structure equivalent relations*/
  foreach unmarked concept $C_i$ in $G'^{=}_i$ do
  if(∃ one-one bridge equivalent relation between ancestor concepts of
     $C_i$ and any concept of $G'_j$, $C_j$) then
        EC← EC + ($C_i$, $C_j$)
        //duplicate elements eliminated
    elseif( ∃ one-one bridge equivalent relation between ancestor
    concepts of $C_i$ and any concept of $G'_j$, $C_j$, 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
      EC← EC + ($C_i$, $C_j$)
    end if
  end
  /*Inferring bridge mutual exclusive relations*/
 foreach $(C_p^{(O_I)},\ C_q^{(O_I)})$ in $\{(C_p^{(O_I)}, C_q^{(O_I)}), I \in (i,j)\}$ of $G_i$ or $G_j$ do
    foreach equivalent concept of $C_q^{(O_I)}, C_{q'}^{(O_{\tilde{I}})}$ in EC do
      IC← IC ∪ $\{(C_p^{(O_I)} \times C_{q'}^{(O_{\tilde{I}})})\}_\perp$
      //According to the structure graph $G_{O_I}$ of ontology
      //$O_I\{C_p^{(O_I)}|G_{O_I}\}_\perp$ is concept $C_p^{(O_I)}$ and all of its descendents, the
      //same as $\{C_{q'}^{(O_{\tilde{I}})}|G_{O_{\tilde{I}}}\}_\perp$.
    end
    foreach equivalent concept of $C_p^{(O_I)}, C_{p'}^{(O_{\tilde{I}})}$ in EC do
      IC← IC ∪ $\{(C_{p'}^{(O)} \times C_q^{(O_I)})\}_\perp$
    end
 end
end
return EC, IC
```

This algorithm adopts a knowledge representation and top-down inferring mechanism, based on equivalent (mutual exclusive) graphs and structure graphs. Its inferring ability is determined by the completeness of the domain equivalent bridge axioms. Compared with most mapping methods used to date, the main advantage of this algorithm is that most of the description features of heavy-weight ontology are taken into consideration, and this algorithm can find all the explicit and derived bridge relations.

The second step of ontology fusion is ontology alignment, which is developed to search implied bridge relations. After confirmation by the domain experts, these bridge relations are added into EC for further use.

```
Algorithm 2. Ontology_alignment($\{O^m\}$, EC, IC, DTA)
Input:   $\{O^m\}$  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
foreach $(O_i,O_j)$ in $\{O^m\}$ do  //Extract structure graphs
   $G_i$ ← Travel($O_i$)
   $G_j$ ← Travel($O_j$)
   $\{(SC^{(O_i)}, SC^{(O_j)})\} \leftarrow O^C{}_i \times O^C{}_j$
      // Cartesian product of concept set in $O_i$ and $O_j$

    /* According to mutual exclusive bridge relations simplify
      $\{(SC^{(O_i)}, SC^{(O_j)})\}$ */
   if($\exists\{(IC_m^{(O_I)}, IC_m^{(O_{\tilde{I}})})\}$ in IC and $IC_m^{(O_I)} \equiv SC_k^{(O_I)}$) then
      //$\{IC_m^{(O_I)}|G_{O_I}\}_\perp$ is concept $IC_m^{(O_I)}$ and all of its descendants
      //according to the structure graph $G_{O_I}$ of $O_I$, the same as
      //$\{IC_m^{(O_{\tilde{I}})}|G_{O_{\tilde{I}}}\}_\perp$.
      $\{(R_IC^{(O_i)}, R_IC^{(O_j)})\} \leftarrow \{(SC^{(O_i)}, SC^{(O_j)})\} - \{(IC_m^{(O_I)} \times IC_m^{(O_{\tilde{I}})})\}_\perp$
   end if
     /*According to equivalent bridge relations simplify
      $\{(R_IC^{(O_i)}, R_IC^{(O_j)})\}$ */
   if($\exists\{(EC_m^{(O_I)}, EC_m^{(O_{\tilde{I}})})\}$ in EC and $EC_m^{(O_I)} \equiv R_IC_k^{(O_I)}$) then
      //$\{EC_m^{(O_I)}|G_{O_I}\}^\top$ is concept $EC_m^{(O_I)}$ and all its ancestors according
      //to the structure graph $G_{O_I}$ of $O_I$, the same as $\{EC_m^{(O_{\tilde{I}})}|G_{O_{\tilde{I}}}\}_\perp$
      $\{(R_EC^{(O_i)}, R_EC^{(O_j)})\} \leftarrow$
          $\{(R_IC^{(O_i)}, R_IC^{(O_j)})\} - \{(IC_m^{(O_I)})^\top \times (IC_m^{(O_{\tilde{I}})})_\perp\}$
    end if
   /*Inferring equivalent bridge relations.*/
   $\{(R_MC^{(O_i)}, R_MC^{(O_j)})\} \leftarrow \{(R_EC^{(O_i)}, R_EC^{(O_j)})\}$
   foreach $(R_EC_m^{(O_I)}, R_EC_n^{(O_{\tilde{I}})})$ in $\{(R_EC^{(O_i)}, R_EC^{(O_j)})\}$ do
     if(data type construction is different according to data type meta
       class definition of meta ontology) then
       //the difference of data type construction include data type unit
       //number inconsistency and data type inheritable
       $\{(R_MC^{(O_i)}, R_MC^{(O_j)})\} =$
           $\{(R_MC^{(O_i)}, R_MC^{(O_j)})\} - (R_EC_m^{(O_I)}, R_EC_n^{(O_{\tilde{I}})})$
     end if
   end
 $\{(R_UC^{(O_i)}, R_UC^{(O_j)})\}$ ←Confirmed($\{(R_MC^{(O_i)}, R_MC^{(O_j)})\}$)
      // domain experts confirmation
 end
 return EC ← EC ∪ $\{(R_UC^{(O_i)}, R_UC^{(O_j)}), i < j\}$
```

This algorithm is defined on the structure graph-based knowledge representation and an attribute set comparison-based bottom-up inferring mechanism. The inferring capability relies on the comparison ability between attribute sets. The output of ontology alignment algorithms are term relations; a set of term pairs in different ontologies. This paper proposes the heuristic information-based (attributes equal), semi-automatic, 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. It can reach the active upper bound of implicit equivalent bridge relations through searching, which greatly enhances the ability of CEE and remarkably reduces manpower. But ontology alignment does not generate new ontologies; it only establishes a mapping set to support interoperations among ontologies, so ontology merging is used here to generate new ontology, based on the existing ones.

In a typical CEE system, there are always two or more subsystems. As mentioned above, the concept lattice of CEE consists of two abelian monoids. So the final collaboration ontology can be constructed by one-one integration in turns.

Algorithm 3. Ontology_merging($\{O^m\}$, EC)

```
Input:    {O^m}   candidate mapping ontology set
          EC      bridge equivalent concept pair list
Output:   FON
foreach $O_i$ in $\{O^m\}$ do
    //simplify structure graph of $O_i$ according to bridge equivalent
    //concept pair list EC
    $G_i \leftarrow$ Equivalent_travel(EC, $O_i$)
end
$FON_{\le 1} \leftarrow G_1 + Thing$ (as top concept)
/* Merge $G_j \in \{\overline{G}_1^{m-1}\}$ and $FON_{\le j-1}$ in turns*/
foreach $G_j$ in $\{\overline{G}_1^{m-1}\}$ do
    let $EC_j = \{(C^{(O_{\le j-1})}, C^{(O_j)})\}$ in EC
    let $\overline{C}_t^{G_j}$= top node of $G_j$
    if($\exists(C_x^{G_{\le j-1}}, \overline{C}_t^{G_j})$ in $EC_j$)then
        Add $\overline{C}_t^{G_j}$ into bridge equivalent concept chain of $C_x^{G_{\le j-1}}$
        $EC_j \leftarrow EC_j - (C_x^{G_{\le j-1}}, \overline{C}_t^{G_j})$
    else
        Add $\overline{C}_t^{G_j}$ into $FON_{\le j-1}$ as a direct child of Thing
    end if
    foreach node $C_t^{G_j}$ in Breadth_first_travel($G_j$, $\overline{C}_t^{G_j}$) do
        if($\exists(C_x^{G_{\le j-1}}, C_t^{G_j})$ in $EC_j$) then
            Add $\overline{C}_t^{G_j}$ into bridge equivalent concept chain of $C_x^{G_{\le j-1}}$
            $EC_j \leftarrow EC_j - (C_x^{G_{\le j-1}}, \overline{C}_t^{G_j})$
        else
            Add $\overline{C}_t^{G_j}$ into $FON_{\le j}$ as a brother node of $G_j$, and add all the in-
            arc of $G_j$
        end if
    end
end
return FON($FON_{\le m}$)
```

This algorithm is defined on the equivalent structure graph-based knowledge representation and an attribute group comparison-based merging mechanism. Compared with light-weight 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.

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 important 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 Scene Creator, localization information supported by Localization Server and instructions from the Decision Support System. In this paper, the Localization Server is supposed to be a real time tracking system, so it does not need any instructions.

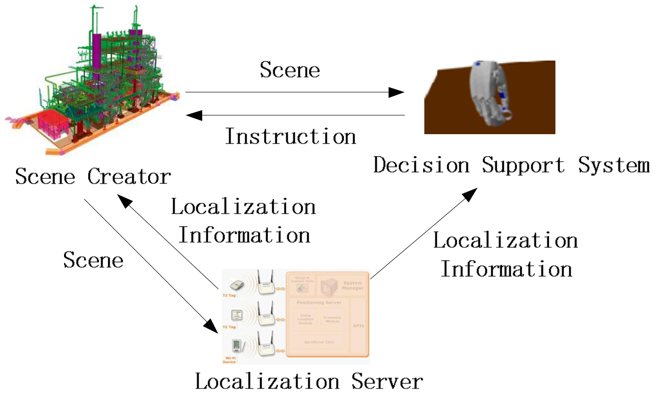
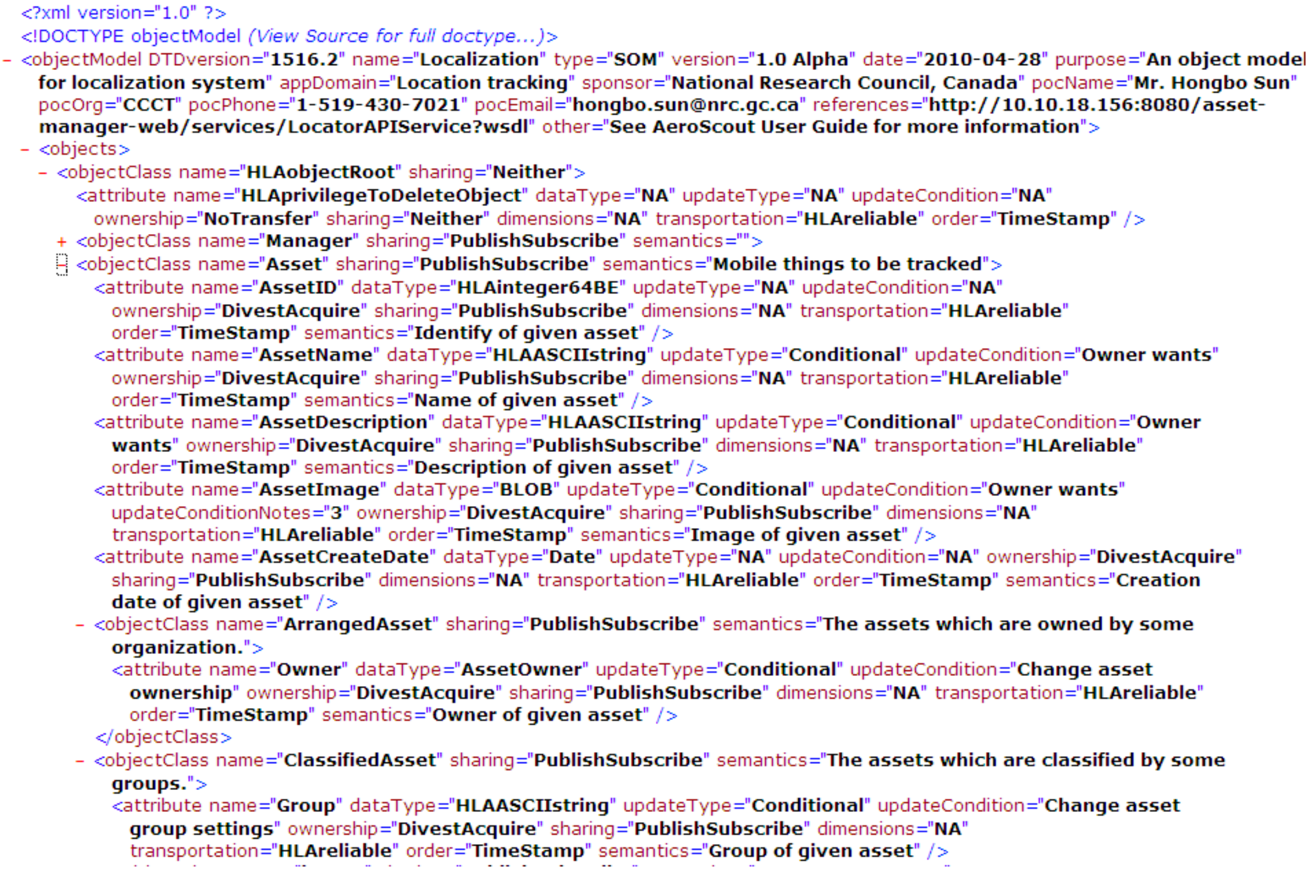


Fig. 5. A typical collaborative engineering environments unit

Before an HLA-based collaborative job started, the exchange data formats of each subsystem should be claimed first as SOM files (Fig. 6). Then domain experts sit together to negotiate about the general collaborative data formats. This processes is time consuming and expensive. Then a general collaborative data format will be defined and a FOM file (looks similar to SOM files) will come into being. After these have been finished, every subsystem changes its interfaces according to the new data exchange format. Finally the collaborative job can be performed. When the collaborative job changes, this process (negotiation and changing interfaces) repeats again.

When the proposed method is adopted, the workload for negotiation and interface changing is significantly reduced, though some preparations still need to be done. First a meta-ontology is built to provide templates of capability description ontologies, data type definitions and axioms. Then capability description ontologies of these systems are constructed separately according to the ability and requirements of the related engineering domains.



```
<?xml version="1.0" ?>
<!DOCTYPE objectModel (View Source for full doctype...)>
- <objectModel DTDversion="1516.2" name="Localization" type="SOM" version="1.0 Alpha" date="2010-04-28" purpose="An object model
  for localization system" appDomain="Location tracking" sponsor="National Research Council, Canada" pocName="Mr. Hongbo Sun"
  pocOrg="CCCT" pocPhone="1-519-430-7021" pocEmail="hongbo.sun@nrc.gc.ca" references="http://10.10.18.156:8080/asset-
  manager-web/services/LocatorAPIService?wsdl" other="See AeroScout User Guide for more information">
  - <objects>
    - <objectClass name="HLAobjectRoot" sharing="Neither">
        <attribute name="HLAprivilegeToDeleteObject" dataType="NA" updateType="NA" updateCondition="NA"
          ownership="NoTransfer" sharing="Neither" dimensions="NA" transportation="HLAreliable" order="TimeStamp" />
      + <objectClass name="Manager" sharing="PublishSubscribe" semantics="">
        <objectClass name="Asset" sharing="PublishSubscribe" semantics="Mobile things to be tracked">
          <attribute name="AssetID" dataType="HLAinteger64BE" updateType="NA" updateCondition="NA"
            ownership="DivestAcquire" sharing="PublishSubscribe" dimensions="NA" transportation="HLAreliable"
            order="TimeStamp" semantics="Identify of given asset" />
          <attribute name="AssetName" dataType="HLAASCIIstring" updateType="Conditional" updateCondition="Owner wants"
            ownership="DivestAcquire" sharing="PublishSubscribe" dimensions="NA" transportation="HLAreliable"
            order="TimeStamp" semantics="Name of given asset" />
          <attribute name="AssetDescription" dataType="HLAASCIIstring" updateType="Conditional" updateCondition="Owner
            wants" ownership="DivestAcquire" sharing="PublishSubscribe" dimensions="NA" transportation="HLAreliable"
            order="TimeStamp" semantics="Description of given asset" />
          <attribute name="AssetImage" dataType="BLOB" updateType="Conditional" updateCondition="Owner wants"
            updateConditionNotes="3" ownership="DivestAcquire" sharing="PublishSubscribe" dimensions="NA"
            transportation="HLAreliable" order="TimeStamp" semantics="Image of given asset" />
          <attribute name="AssetCreateDate" dataType="Date" updateType="NA" updateCondition="NA" ownership="DivestAcquire"
            sharing="PublishSubscribe" dimensions="NA" transportation="HLAreliable" order="TimeStamp" semantics="Creation
            date of given asset" />
        - <objectClass name="ArrangedAsset" sharing="PublishSubscribe" semantics="The assets which are owned by some
            organization.">
            <attribute name="Owner" dataType="AssetOwner" updateType="Conditional" updateCondition="Change asset
              ownership" ownership="DivestAcquire" sharing="PublishSubscribe" dimensions="NA" transportation="HLAreliable"
              order="TimeStamp" semantics="Owner of given asset" />
          </objectClass>
        - <objectClass name="ClassifiedAsset" sharing="PublishSubscribe" semantics="The assets which are classified by some
            groups.">
            <attribute name="Group" dataType="HLAASCIIstring" updateType="Conditional" updateCondition="Change asset
              group settings" ownership="DivestAcquire" sharing="PublishSubscribe" dimensions="NA"
              transportation="HLAreliable" order="TimeStamp" semantics="Group of given asset" />
```

Fig. 6. SOM File of Localization System

Meta-ontology (MO) is the ontology which stores the fundamental knowledge of inferences. It contains object templates, basic transformation data types and necessary general axioms. This approach is compatible with the definition of HLA OMT (Object Model Template) object classes and interaction classes.

Scene creator adopted here is a Building Information Management System. All the files which can be seen in an IFC (Industry Foundation Classes) viewer are deemed as an IFC building. Every element of IFC building has a unique UUID (Unique Universal IDentifier). IFC building has five sub categories: Building Element Proxy, IFC Building Elements, Single Storey Building, Multi-Storey Building and Space. Building Element Proxy is the monitor of building Elements, such as hydrometer and thermostat. Building Element is a part of Building. Single Storey Building is a Building that only owns one Storey. Multi-Storey Building is a building that has at least two floors. Space is somewhere which is isolated. IFC Building Elements include Basic wall, Furnishing Elements, Opening, Roof and Stair. Basic wall with openings is called Wall. Opening includes Window, Door and Door Entry. Furnishing Elements include Fixed Furniture and Movable Furniture. The Movable furniture is the target of localization system.

The Decision Support System adopted here is a Facility Maintenance Management System. The concepts involved here include Point, Building Description, Building Element and Building. Point exists in some space, when it has special meaning it becomes Physical location. Building description includes attributes of Buildings, such as Location and Map. Location can be specialized into Building Location, Level Location, and Physical Location. In this system, Assets are deemed as Building Elements. Among these Assets, assets with locations are looked as Trackable Assets. Building is a closed space in the management scope, which can be specialized into Frame, Level or Space and Room. The main objective of this system is asset tracking and maintenance.

Localization system adopted here is called AeroScout. The main concepts used in this system are Tag, Map, Coordinate, Asset, Corrective Action and Building. Tag can be divided into Active tag and Inactive tag according to the tracking state. Map can be specialized into Building Location, Level Location, Area Location and Zone Location or Physical Location. Physical Location is also a subclass of Coordinate. Asset can be classified into Arranged Asset and Live Asset. Arranged Asset belongs to some Department. Live Asset is a movable Asset. Among Live Assets, Trackable Assets are binding with Active Tags. Corrective Actions include Event and Internal Event, such as Invoking and Moving. Corresponding to the map, Building can also be specialized into Level, Area and Zone. The main targets of tracking are Active Assets.

The final preparation task is to abstract graphs from these ontologies. Because these systems are too complicated for a demonstration, only part of the concepts and relations are described in this paper. The bridge relation graphs are shown in Fig. 7, 8, 9 and structure graphs in Fig. 10,11,12.

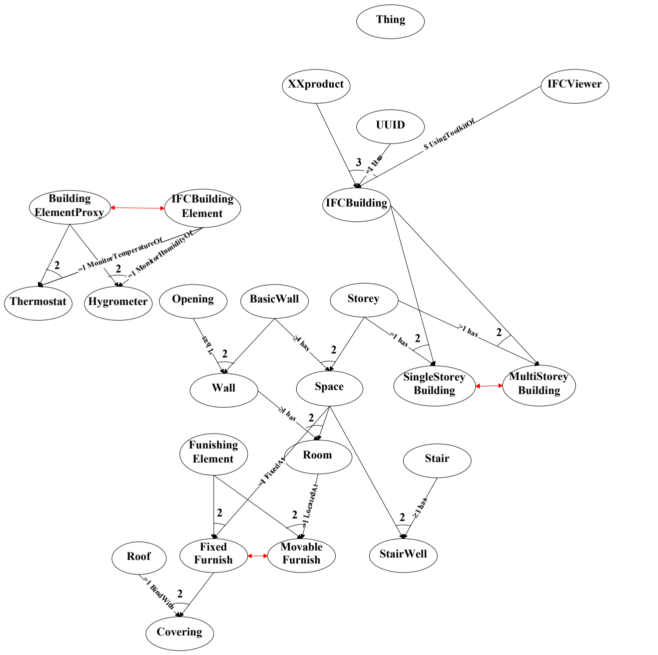


Fig. 7. Scene Creator Equivalent (mutual exclusive) Graph Demo

In Scene Creator Equivalent (mutual exclusive) Graph Demo, there are 12 equivalent relations and three mutual exclusive relations.

$$\text{IFC Building} \Leftrightarrow \text{product} \cap (= 1\ \mathit{has}\,.\,\text{UUID}) \cap (\exists \mathit{UsingToolkitOf}.\,\text{IFCViewer})$$

$$\text{SingleStorey Building} \Leftrightarrow \text{IFC Building} \cap (= 1\ \mathit{has}\,.\,\text{Storey})$$

$$\text{MultiStorey Building} \Leftrightarrow \text{IFC Building} \cap (> 1\ \mathit{has}\,.\,\text{Storey})$$

$$\text{Thermostat} \Leftrightarrow \text{Building ElementProxy} \cap (= 1\ \mathit{MonitorTemperatureOf}\,.\,\text{IFC Building Element})$$

$$\text{Hygrometer} \Leftrightarrow \text{Building ElementProxy} \cap (= 1\ \mathit{MonitorHumidityOf}\,.\,\text{IFC Building Element})$$

$$\text{Wall} \Leftrightarrow \text{Basic Wall} \cap (> 1\ \mathit{has}\,.\,\text{Opening})$$

$$\text{Space} \Leftrightarrow \text{Storey} \cap (\geq 4\ \mathit{has}\,.\,\text{Basic Wall})$$

$$\text{Room} \Leftrightarrow \text{Space} \cap (\geq 1\ \mathit{has}\,.\,\text{Wall})$$

$$\text{StairWell} \Leftrightarrow \text{Space} \cap (\geq 1\ \mathit{has}\,.\,\text{Stair})$$

$$\text{Fixed Furniture} \Leftrightarrow \text{Furnishing Element} \cap (= 1\mathit{FixedAt}\,.\,\text{Space})$$

$$\text{Movable Furniture} \Leftrightarrow \text{Furnishing Element} \cap (= 1\mathit{LocatedAt}\,.\,\text{Room})$$

$$\text{Covering} \Leftrightarrow \text{Fixed Furniture} \cap (= 1\mathit{FixedAt}\,.\,\text{Roof})$$

$$\text{Single Storey Building} \leftrightarrow \text{Multi} - \text{Storey Building}$$

$$\text{Fixed Furniture} \leftrightarrow \text{Movable Furniture}$$

$$\text{Building Element Proxy} \leftrightarrow \text{IFC Building Element}$$

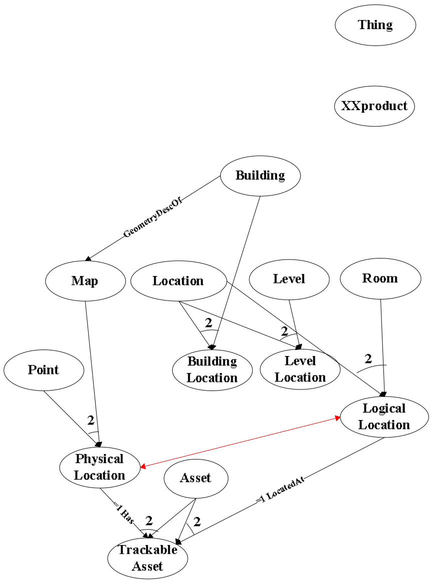


Fig. 8. Decision Support System Equivalent (mutual exclusive) Graph Demo

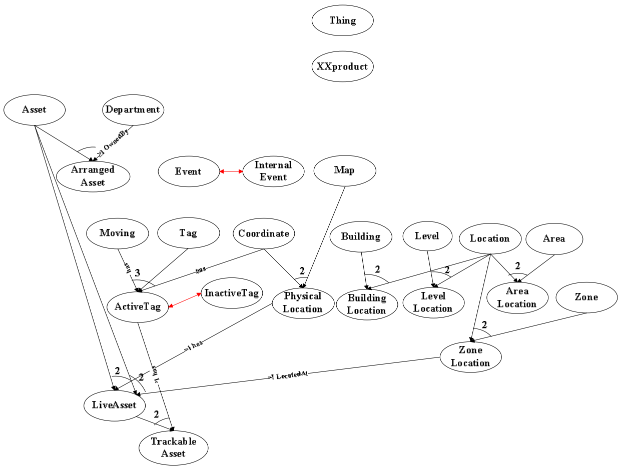


Fig. 9. Localization System Equivalent (mutual exclusive) Graph Demo

Similarly, in Decision Support System Equivalent (mutual exclusive) Graph Demo, there are seven equivalent relations and one mutual exclusive relation. In Localization System Equivalent (mutual exclusive) Graph Demo, there are 10 equivalent relations and two mutual exclusive relations.

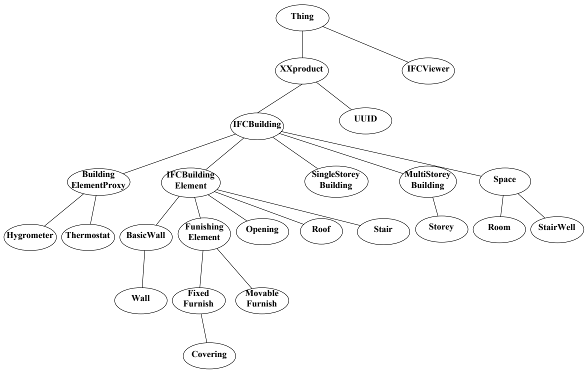


Fig. 10. Scene Creator Structure Graph Demo

Let S denote the scene creator, L represent the localization system and D stand for the decision support system. The bridge equivalent relations given by domain experts are as follows, and no new mutual exclusive bridge relation is inferred.

D.Building=S.IFCBuilding
D.Asset=S.FinishingElement
D.Map=L.Map
D.Level=L.Level
D.Asset=L.Asset
D.Level=S.Storey
D.LogicalLocation=S.Room
D.Location=L.Location
D.Point=L.Point
D.Room=L.Zone

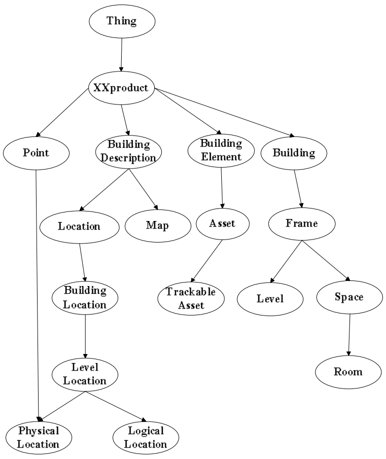


Fig. 11. Decision Support System Structure Graph Demo

By the ontology mapping algorithm mentioned above, new bridge equivalent relations are inferred as follows:

D.TrackableAsset=S.MovableFurnish
D.PhysicalLocation=L.PhysicalLocation
D.BuildingLocation=L.BuildingLocation
D.LevelLocation=L.LevelLocation
D.LogicalLocation=L.ZoneLocation
D.TrackableAsset=L.LiveAsset

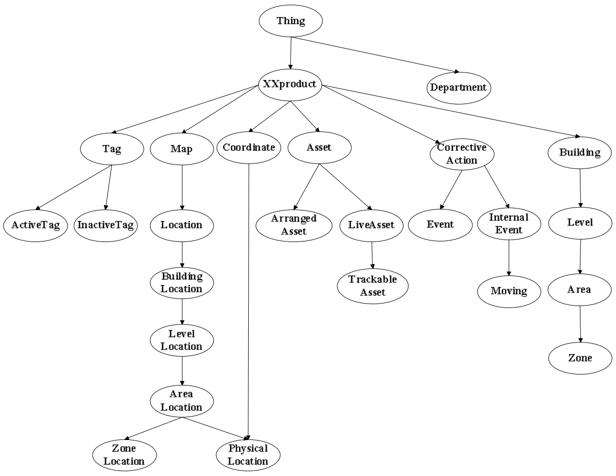


Fig. 12. Localization System Structure Graph Demo

The candidate equivalent bridge relations inferred by the ontology alignment algorithm are all denied by the domain experts, and no candidate mutual exclusive bridge relations are generated from this algorithm.

After ontology merging, the collaboration ontology is reached as shown in Fig. 13.

By means of the ontology fusion algorithms described above, the collaboration ontology is easily established. What the domain experts need to do is to give well known equivalent (mutual exclusive) bridge relations and to judge the inferred candidate relations, which significantly reduces the workload of domain experts and improves the efficiency of collaboration preparations. At the same time, because ontologies have expandability and equivalent (mutual exclusive) bridge relations are also stored in meta-ontology for future use, the ability of resources reuse is also remarkably enhanced in the CEEs.

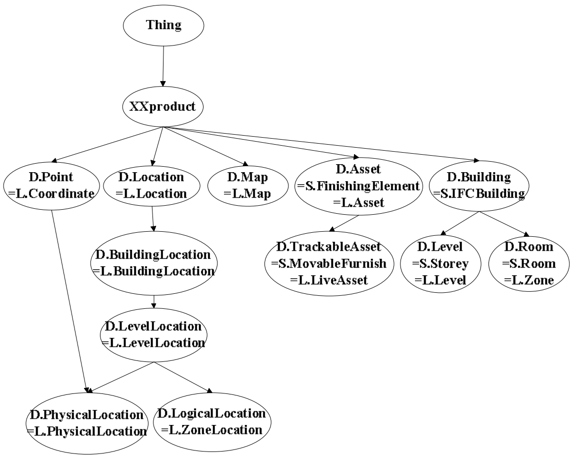

Thing
XXproduct
D.Point =L.Coordinate
D.Location =L.Location
D.Map =L.Map
D.Asset =S.FinishingElement =L.Asset
D.Building =S.IFCBuilding
D.BuildingLocation =L.BuildingLocation
D.TrackableAsset =S.MovableFurnish =L.LiveAsset
D.Level =S.Storey =L.Level
D.Room =S.Room =L.Zone
D.LevelLocation =L.LevelLocation
D.PhysicalLocation =L.PhysicalLocation
D.LogicalLocation =L.ZoneLocation

Fig. 13. Output Collaboration Ontology Demo

VI. Conclusion

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 proposes a semi-automatic ontology fusion method to establish a collaborative ontology as a 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.

Based on complexity analysis, this approach may not be the best choice. Because this approach is based on bridge axioms and equivalent relations, it is not easy to compare with other ontology integration methods. For the same reason, OAEI (Ontology Alignment Evaluation Initiative) is not applicable here. Here is a brief complexity comparison. Let n denote the concept number in the ontology, and l represent the length of DA (Domain Axiom set). When integrating two ontologies, the complexities of existing algorithms are: NOM - Naive Ontology Mapping $O(n^2 \cdot \log^2 n)$, PROMPT $O(n \cdot \log n)$, Anchor-PROMPT $O(n^2 \cdot \log^2 n)$, GLUE $O(n^2)$ and QOM - Quick Ontology Mapping $O(n \cdot \log n)$ [21]. While the complexity of our ontology mapping algorithm is $O(n \cdot l)$, and the complexity of our ontology alignment algorithm is $O(n^2)$.

Despite its complexity, it enjoys several remarkable advantages which are more suitable for HLA-based multidisciplinary collaborations:

- This method has firm theoretical foundations and starts from strict definitions.
- It reduces the workload of domain experts to prepare collaborations among independent engineering domains by automatic inference and deduction. For the same reason, it improves the efficiency in the preparation of future collaborations.
- 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.
- The proposed algorithms can find all the explicit and derived bridge relations.
- Since ontology is used in this method, the reuse of resources and the expandability of existing systems are greatly enhanced.
- The construction method of collaborative ontology also helps to accumulate knowledge and furthermore to build relatively complete models of the same objective from different aspects.

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.

ACKNOWLEDGMENT

This work was partially supported by Chinese National High-Tech Research and Development Program (863 Program, Grant No. 2009AA110302) and Chinese Nature Science Foundation (Grant No. 60874066).

REFERENCES

[1] W. Fan, W. Wang and T. Xiao, "Multidisciplinary Collaboration Simulation Optimization Platform for complex product design", Proceedings of the 2nd International Conference on Pervasive Computing and Applications, 2007 (ICPCA 2007). Birmingham, UK, 174 -178, Jul. 2007.

[2] H. Sun, T. Xiao and S. Tang, "Research on Federation-Based Pragmatic Integration Framework, Proceedings of 2009 World Congress on Computer Science and Information Engineering (CSIE 2009)". Los Angeles/Anaheim, USA, Apr. 2009.

[3] Simulation Interoperability Standards Committee (SISC) of the IEEE Computer Society, "IEEE Standard for Modeling and Simulation(M&S) High Level Architecture (HLA)-Framework and Rules", The Institute of Electrical and Electronics Engineers, Inc. New York, 2000.

[4] Y. Xu, T. Xiao, C. Liang and L. Zhang, "Federated integration of networked manufacturing service platforms", Advanced Engineering Informatics. vol. 22, 317-327, Jul. 2008.

[5] S. Tang, T. Xiao and W. Fan, "A collaborative platform for complex product design with an extended HLA integration architecture". Simulation Modelling Practice and Theory. vol. 18(8),1048-1068, Sep. 2010.

[6] IEEE Computer Society, "IEEE standard for modeling and simulation (M&S) high level architecture (HLA)-object model template (OMT) specification(IEEE Std 1516.2- 2000)", NewYork: The Institute of Electrical and Engineers, 2001.

[7] W. Zhou, Z. Liu, and H. Chen, "A Survey of the Research about Both FCA and Ontology", Computer Science, China, 33(2) 8-12,2006.

[8] J. Yu, and Y. Dang, "Review on Ontology Integration, Computer Science", China, 35 (7) 9-14, 2008.

[9] F. Fürst and F. Trichet, "Axiom-based ontology matching: a method and an experiment", research report, Mar. 2005.

[10] C. Bock, X. Zha, H. Suh and J. Lee, "Ontological product modeling for collaborative design", Advanced Engineering Informatics. in Press, 2010.

[11] A. Umar and A. Zordan, "Enterprise Ontologies for Planning and Integration of Business: A Pragmatic Approach", IEEE Transactions on Engineering Management, 56 (2) 352-371, May. 2009.

[12] N. F. Noy and M. A. Musen, "The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping", International Journal of Human-Computer Studies, 59 (6), 983-1024, Dec. 2003.

[13] D. Dou, D. McDermott, and P. Qi, "Ontology Translation on the Semantic Web", Journal on data semantics II, 3360 (2005) 35-57, 2005.

[14] A. Maedche, B. Motik, N. Silva and R. Volz, "MAFRA —An Ontology MApping FRAmework in the Context of the Semantic Web", In Workshop on Ontology Transformation at ECAI - 2002, Lyon, France, July 2002.

[15] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, "Learning to map between ontologies on the semantic web", Proceedings of the 11th international conference on World Wide Web, Honolulu, Hawaii, USA, 662-673, 2002.

[16] H. Schnurr and J. Angele, "Do not use this gear with a switching lever!Automotive industry experience with semantic guides", 4th International semantic web conference, Galway, IRLANDE, 3729 (2005) 1029-1040, 2005.

[17] Y. Huang, H. Zhou, J. Li, L. Tan, and Z. Li, "Survey of Ontology Mapping Approaches", Computer Engineering and Applications, China, 18 (2005) 27-29, 2005.

[18] B. Ganter, G. Stumme, R. Wille, "Formal concept analysis: foundations and applications", Springer-Verlag Berlin Heidelberg, Germany, 2005.

[19] R. Wille, "Concept lattices and conceptual knowledge systems". Computers & Mathematics with Applications, 23 (1992) 493-515, 1992.

[20] K. Qu, J. Liang, J. Wang and Z. Shi, "The algebraic properties of Concept Lattice", Journal of Systems Science and Information, 2(2) 271-277, 2004.

[21] M. Ehrig and S. Staab, "QOM - Quick Ontology Mapping, Proceedings of the International Semantic Web Conference(ISWC)", Hiroshima, Japan, 683-697, Nov. 2004.

[22] B. Li, X. Chai, G. Xiong, W. Zhu, C. Quan, H. Zhang and X. Wang, "Research and Primary Practice on Virtual Prototyping Engineering of Complex Product", Journal of System Simulation, 14 (3) 336-341, 2002.

H. Sun received his Bachelor degree in Information Science from Beijing Institute of Technology, China in 1998, his Master's degree in Software Engineering from Tsinghua University, China in 2005, and his PhD degree in Control Science and Engineering from Tsinghua University in 2011.

He is currently a Postdoctoral Researcher at National CIMS ERC, Tsinghua University, Beijing, China. He was visiting researcher at National Research Council Canada from November 2009 to November 2010. Between 2005 and 2006, he served as a Software Engineer at National CIMS ERC, Tsinghua University, Beijing, China. From 1998 to 2002, he was an Assistant Professor at Shenyang Institute of Technology, Liaoning, China. His research interests include system integration, artificial intelligence, algorithms, large-scale simulation and e-commerce.