Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Ontological View Based Semantic Transformation for Distributed Systems

Ying Daisy Wang
Southwestern University of Finance and Economics, Chengdu, China
wangying_t@swufe.edu.cn

Hamada Ghenniwa
University of Western Ontario
London, Canada
hghenniwa@eng.uwo.ca

Weiming Shen
National Research Council Canada
London, Canada
weiming.shen@nrc.gc.ca

***Abstract*—This paper presents a novel ontological view-based semantic transformation method to address the semantic heterogeneity design issue of open distributed information systems. After carefully reviewing the traditional ontology definitions, this work extends the ontological view concept to represent the partial knowledge about the same business domain in an open environment where common ontology does not exist or is not explicitly represented. Solutions in mathematical formulation and corresponding application algorithms, based on the definition of an ontological view, are proposed. By dealing with structural constant and predicate heterogeneity, respectively, the solution enables the ontological views to be transformed, one to another, automatically.**

***Keywords*—ontology, semantic integration, distributed systems, open environment**

I. Introduction

Globalization is the key to success in today's business world. Each organization depends on knowledge for efficiently and effectively accomplishing its business mission. Knowledge becomes a property as well as a resource of a business. This motivates the so-called information society's demand for complete access to available information, which is often heterogeneous. This heterogeneity is due to the presence of an information source that is unpredictable without a common representation structure of knowledge that is predefined in the open environment. Semantic heterogeneity occurs when there is a disagreement about the meaning, interpretation, or intended use of the same or related data [17]. This requires the information to be integrated for a unique external appearance.

In the past, semantics, which played an important role during the integration task, came into focus leading to so-called ontology-based integration approaches. In Artificial Intelligence, ontology refers to an engineering artifact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words [9]. Ontology for a given logical language is a set of axioms designed in such a way that the set of its models approximates, as best as possible, the set of intended models of the language, according to the ontological commitment. Therefore, ontology can "specify" a conceptualization only in a very indirect way. Ontology, for a logical language, approximates a conceptualization if an ontological commitment exists, according to which the intended models of the language are included in the models of the ontology. Thus, ontology can be viewed as a logical theory accounting for the intended meaning of a formal vocabulary. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. Ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models. Based on the definition given above, this work further defines *ontological view* as the engineering artifact, constituted by a specific subset vocabulary used to describe a certain subset of reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words subset. Entities in an open environment might exhibit heterogeneous ontological views of the same business domain. After carefully investigated various heterogeneities occurring in constant symbols and predicate symbols of the vocabulary of the logical language, corresponding algorithms are proposed to resolve the ontological view transformation issues, respectively.

The rest of this paper is organized as follows: Section II provides a brief literature review; Section III introduces the Ontological View based semantic transformation; Section IV discusses prototype implementation issues; and Section V concludes the paper with some perspectives.

II. Literature Review

Researchers and developers have been working on resolving such heterogeneities within the traditional database context for many years. The solutions are categorized into several classes: (1) schema level integration [12]; (2) multi-level matches [16]; (3) schema merging [1]. These solutions require an explicitly predefined schema.

In the generic information integration research, some try to solve it by establishing semantic correspondences (also called mappings) between vocabularies of different data sources. The techniques include linguistic analysis of terms [10], mapping to common reference ontology [3] and use of heuristics that look for specific patterns in the concept definitions [13]. These solutions require either a global knowledge representation or human interruption.

On the other hand, some researches specifically focus on various aspects of ontology. Firstly, the definition and mathematical representation has been widely studied. Based on the definition of *conceptualization* given by [11], [9] further defined the term "*ontology*" as an engineering artifact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary symbols. Ontology is a logical theory accounting for the *intended meaning* of a formal vocabulary, i.e. its *ontological commitment* to a particular *conceptualization* of the world. Ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models as shown in Fig. 1. Secondly, some researches focus on ontology classification [8]. Thirdly, some researches focus on ontology languages, such as XML, RDF/RDFS [2] that capture and represent ontology; some of them also take future inference and reasoning into consideration. Reference [4] compared these ontology languages and their usage. Fourthly, a lot of effort has also been in the area of ontology construction [6]. Other research projects focus on ontology reuse [7]. Lastly, some researches directly focus on ontology integration methodology. Some typical resolutions include: (1) using multiple models [15]; (2) transforming into a formal representation [5]**Error! Reference source not found.**; (3) building new ontology, merging existing ontology [14].

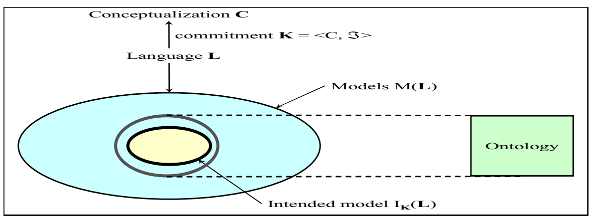


Figure 1. Conceptualization, Intended Model and Ontology

Considering the nature of open distributed systems, an explicitly expressed global ontology does not exist. Most of the time, human interruption is also not realistic. Thus, the fact that global ontology and automation do not exist are vital characteristics. A review of the existing solutions based on these requirements, indicates that they were built on the premise of the existence of a global ontology, the existence of human interruption, or both. It is necessary to further our study considering the open distributed systems' requirements.

The definition and mathematical representation of the partial knowledge of the same business domain is a foundation for this study. Since the definition of ontology given by [9] is most suitable, our work extended this concept from closed environment ontology to an open environment ontological view. As our study is at the conceptual level rather than the representation level where ontology languages are located, this work only utilized the available language (OWL) that best suits the representational needs. The development of the ontology language is out of our scope and we are not taking into account the construction of ontology due to the premise that global ontology does not exist.

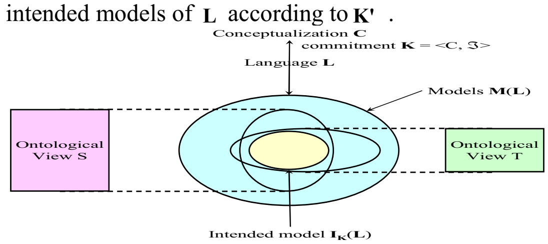
III. Ontological View Transformation

A. *Ontological View*

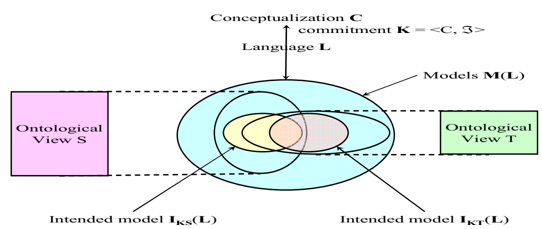
For a close centralized information system, the semantics of the system is implicitly expressed in the software component. However, in an open distributed/decentralized information system environment, it is not realistic to have a common ontology that everyone agrees to before each individual system is developed. Thus, for a given business domain, the semantics of each individual system become an ontological view of the same conceptualization. Adopting the definition of conceptualization from [9], formal definitions of the related terms are given as follows:

[Definition 1] Assuming the intended world structure is fixed, $\mathbf{K}' = \langle \mathbf{C}, \Im' \rangle$ is an *ontological commitment of view* for $\mathbf{L}$, where $\mathbf{C} = \langle D, W, \Re \rangle$ is a conceptualization, the intended model $\mathbf{I}_{\mathbf{K}}$ is a subset of the view ontological commitment assignment function $\Im'$, the view assignment function $\Im'$ is a subset of the possible ontology's ontological commitment assignment function $\Im$, $\mathbf{I}_{\mathbf{K}} \subseteq \Im' \subseteq \Im$, the view vocabulary V' is a subset of the possible ontology vocabulary V, $V' \subseteq V$, $\Im' : V' \rightarrow D \cup \Re'$ function assigning elements of D to constant symbols of V', and element of $\Re'$ ($\Re' \in \Re$) to predicate symbols of V'.

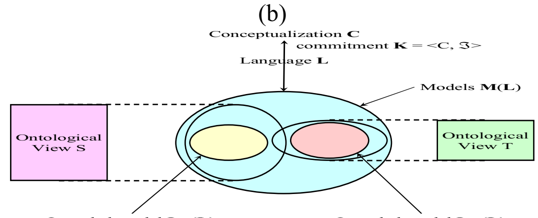
[Definition 2] Assuming the intended world structure is fixed, given a language $\mathbf{L}$ with ontological commitment $\mathbf{K}$, an *ontological view* for $\mathbf{L}$ is a set of axioms designed in a way such that the set of its models approximates as best as possible the intended models of $\mathbf{L}$ according to $\mathbf{K}'$.



(a)



(b)



(c)

Figure 2. Relationship between Conceptualization and Ontological View

From the above definition, we can use Fig. 2 to illustrate the relationship between conceptualization, the intended model and the ontological view. We classify the ontological view transformation problem into three scenarios as shown in (a), (b) and (c). In scenario (a), the source and target ontological view share the same intended model. In other words, the intersection of different ontological views must include the intended model within it. In scenario (b), the intended models of the source and target ontological view overlap. In other words, the intersection of different ontological views is not empty. In scenario (c), the intended models of the source and target ontological view have no intersection. However, the two ontological views do have an intersection. Scenario (a) is the only case among the three scenarios that the source ontological view can be completely transformed into a target ontological view through the shared intended model, yet the transformation completeness is guaranteed. In this work, we focus on the transformable case (scenario (a)) and propose our approach.

[Definition 3] Given a source ontological view Φ_S with intended model $\mathbf{I}_{K_S}(\mathbf{L})$ and a target ontological view Φ_T with intended model $\mathbf{I}_{K_T}(\mathbf{L})$, Φ_S is *transformable* (denoted by ∞) to Φ_T if and only if $\mathbf{I}_{K_S}(\mathbf{L})$ is equal to $\mathbf{I}_{K_T}(\mathbf{L})$. $(\mathbf{I}_{K_S}(\mathbf{L}) = \mathbf{I}_{K_T}(\mathbf{L})) \leftrightarrow (\Phi_S \infty \Phi_T)$.

B. Problem Formulation

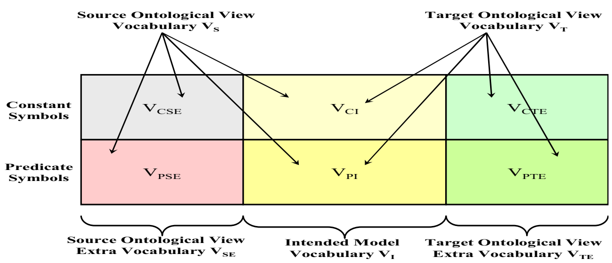


Figure 3. Relationship between Source Ontological View Vocabulary, Target Ontological View Vocabulary and Intended Model Vocabulary

As shown in Fig. 3, the intended model vocabulary V_I is constituted by constant symbols V_{CI} and predicate symbols V_{PI}:

$$V_I = \{V_{CI}, V_{PI}\} \quad (1)$$

The source ontological view vocabulary is denoted by V_S. Taking the shared intended model assumption into consideration, we may further explore V_S 's property by introducing the following theorem regarding the relationship between $\mathbf{L}$, Φ_S and Φ_T. The following theorem holds.

[Theorem 1] Given a set of intended models $\mathbf{I}_K(\mathbf{L})$ of $\mathbf{L}$ according to $\mathbf{K}$, a source ontological view $\Phi_S = \langle \mathbf{L}, \mathbf{K}_S \rangle$ and a target ontological view $\Phi_T = \langle \mathbf{L}, \mathbf{K}_T \rangle$, if language $\mathbf{L}$ has vocabulary V, ontological commitment $\mathbf{K}$ maps the conceptualization to vocabulary V_I, where $V_I \subseteq V$, the view ontological commitment $\mathbf{K}_S = \langle \mathbf{C}, \Im_S \rangle$ maps the conceptualization to vocabulary V_S, where $V_S \subseteq V$, and the view ontological commitment $\mathbf{K}_T = \langle \mathbf{C}, \Im_T \rangle$ maps the conceptualization to vocabulary V_T, where $V_T \subseteq V$, then $\mathbf{I}_K(\mathbf{L}) \subseteq \Phi_S \cap \Phi_T$.

However, as [9] has addressed, ontology indirectly reflects the ontological commitment (and the underlying conceptualization) by approximating theses intended models. That is to say, the intended model can not be represented accurately, explicitly and directly. Instead, the intended models can only be approximated by ontology. Based on Definition 2, the ontological view is a subset of ontology. Therefore, Guarino's statement also stands for the ontological view. That is,, in the open environment, the intended models can also be approximated by each ontological view. Furthermore, Theorem 1 even shows that the intended model is contained in the intersection of different ontological views which intend to approximate the same intended model and this intersection approximates the intended model better than each of the ontological views.

According to Definition 2, an ontological view is identified by language $\mathbf{L}$ and the ontological commitment $\mathbf{K'}$. Language $\mathbf{L}$ is identified by its vocabulary V'. According to Definition 1, an ontological commitment $\mathbf{K'}$ is identified by the conceptualization $\mathbf{C} = \langle D, W, \Re \rangle$ and assignment function $\Im'$. The assignment function is decided by the language's vocabulary V' and domain D and conceptual relation $\Re$. Therefore, the five factors $\langle D, W, \Re, \Im', V' \rangle$ uniquely identify an ontological view. Obviously, the following proposition holds.

[Proposition 1] Given a conceptualization $\mathbf{C} = \langle D, W, \Re \rangle$, a source ontological view $\Phi_S = \langle \mathbf{L}, \mathbf{K}_S \rangle$ that commits to the conceptualization $\mathbf{C}$ by $\mathbf{K}_S = \langle \mathbf{C}, \Im_S \rangle$ and a target ontological view $\Phi_T = \langle \mathbf{L}, \mathbf{K}_T \rangle$ that commits to the conceptualization $\mathbf{C}$ by $\mathbf{K}_T = \langle \mathbf{C}, \Im_T \rangle$, if language $\mathbf{L}$ has a vocabulary V, the view ontological commitment $\mathbf{K}_S = \langle \mathbf{C}, \Im_S \rangle$ maps the conceptualization $\mathbf{C}$ to the vocabulary V_S, where $V_S \subseteq V$, and the view ontological commitment $\mathbf{K}_T = \langle \mathbf{C}, \Im_T \rangle$ maps the conceptualization $\mathbf{C}$ to the vocabulary V_T, where $V_T \subseteq V$. If $\Im_S'$ is the subset of $\Im_S$, $\Im_T'$ is the subset of $\Im_T$, and $\Im_S' \leftrightarrow \Im_T' \leftrightarrow I_K$, then the source ontological view Φ_S is equal to the target ontological view Φ_T if and only if V_S is equivalent to V_T.

$$(V_S \leftrightarrow V_T) \xleftrightarrow{(C_S = C_T) \wedge (\Im_S' = \Im_T')} (\Phi_S = \Phi_T) \quad (2)$$

Thus, we notice that it is only a subset of the source ontological view that can be transformed to the target ontological view while the rest is not, which is expressed in the following theorem:

[Theorem 2] For a given conceptualization $\mathbf{C} = \langle D, W, \Re \rangle$, $\Phi_S = \langle \mathbf{L}, \mathbf{K}_S \rangle$ is the source ontological view, represented by language $\mathbf{L}$ with a vocabulary V_S commits to $\mathbf{C}$ by $\mathbf{K}_S$, and $\Phi_T = \langle \mathbf{L}, \mathbf{K}_T \rangle$ is the target ontological view, represented by language $\mathbf{L}$ with a vocabulary V_T commits to $\mathbf{C}$ by $\mathbf{K}_T$. If Φ_S and Φ_T both approximate a common intended model $\mathbf{I}_K(\mathbf{L})$, Φ_S' is a subset of Φ_S, and $\Phi_S' = \Phi_S \cap \Phi_T$, Φ_T' is a subset of Φ_T, and $\Phi_T' = \Phi_S \cap \Phi_T$, then Φ_S' is transformable to Φ_T'.

Up to this point, we can conclude that the ontological view transformation problem is within the intersection of Φ_S and Φ_T ($\Phi_S \cap \Phi_T$). The symmetric difference ($\Phi_S \oplus \Phi_T$) is not of interest to us at this time. Before we explore the view transformation problem, we formally define the related term below:

[Definition 4] Given a source ontological view Φ_S and a target ontological view Φ_T, a *transformation function* $\mu: \Phi_S \to \Phi_T$ is a function that maps elements of Φ_S to elements of Φ_T.

And from the definition given above, we can have the following proposition.

[Proposition 2] Given a conceptualization $C = \langle D, W, \Re \rangle$, a source ontological view $\Phi_S = \langle L, K_S \rangle$ commits to the conceptualization C by approximating the intended model $I_K(L)$ through $K_S = \langle C, \Im_S \rangle$ and a target ontological view $\Phi_T = \langle L, K_T \rangle$ commits to the conceptualization C by approximating the intended model $I_K(L)$ through $K_T = \langle C, \Im_T \rangle$, language L has a vocabulary V, the view ontological commitment $K_S = \langle C, \Im_S \rangle$ maps the conceptualization C to the vocabulary V_S, where $V_S \subseteq V$, and the view ontological commitment $K_T = \langle C, \Im_T \rangle$ maps the conceptualization C to the vocabulary V_T, where $V_T \subseteq V$. There is a transformation function μ, that maps the source ontological view Φ_S to the target ontological view $\Phi_T \cdot \{\Phi_S, \Phi_T\} \xrightarrow{C \wedge I_K(L)} \exists \mu (\mu : \Phi_S \rightarrow \Phi_T)$

As the ontological view transformation problem is under the assumption of the same conceptualization $C = \langle D, W, \Re \rangle$ and shared intended model $I_K(L)$, then V_S uniquely identifies Φ_S and V_T uniquely identifies the target Φ_T. Therefore, under such an assumption, the vocabulary and ontological view has a one-to-one relationship.. The source ontological view Φ_S is equal to Φ_T if and only if V_S is equivalent to V_T. Thus, the following theorem holds.

[Theorem 3] For a given conceptualization $C = \langle D, W, \Re \rangle$, $\Phi_S = \langle L, K_S \rangle$ is the source ontological view, represented by language L with a vocabulary V_S commits to C by K_S, and $\Phi_T = \langle L, K_T \rangle$ is the target ontological view, represented by language L with a vocabulary V_T commits to C by K_T. If Φ_S and Φ_T both approximate a common intended model $I_K(L)$, $\Phi_S^{'}$ is a subset of Φ_S, and $\Phi_S^{'} = \Phi_S \cap \Phi_T$, $\Phi_T^{'}$ is a subset of Φ_T, and $\Phi_T^{'} = \Phi_S \cap \Phi_T$. $\Phi_S^{'}$ represented by the subset vocabulary of language L, denoted by $V_S^{'}$, commits to C by K_S, and $\Phi_T^{'}$ represented by the subset vocabulary of language L, denoted by $V_T^{'}$, commits to C by K_T, then $V_S^{'}$ is transformable to $V_T^{'}$, $V_S^{'} \infty V_T^{'}$.

That is to say, the intersection ($\Phi_S \cap \Phi_T$) of the source and the target ontological view corresponding vocabulary is the transformation space, while the symmetric difference ($\Phi_S \oplus \Phi_T$) corresponding vocabulary is not of interest to us at this time.

As for each ontological view $\Phi = \langle L, K \rangle$, its language vocabulary is always constituted by two parts – the set of constant symbols V_C and the set of predicate symbols V_P, therefore, the ontological view transformation problem can be formulated as below:

[Definition 5] For a given conceptualization $C = \langle D, W, \Re \rangle$, $\Phi_S = \langle L, K_S \rangle$ is the source ontological view, represented by language L with a vocabulary $V_S \subseteq V, V_S = \{V_{CS}, V_{PS}\}$ commits to the conceptualization C by approximating the intended model $I_K(L)$ through $K_S = \langle C, \Im_S \rangle$, where V_{CS} is the set of constant symbols; V_{PS} is the set of predicate symbols. $\Phi_T = \langle L, K_T \rangle$ is the target ontological view, represented by language L with a vocabulary $V_T \subseteq V, V_T = \{V_{CT}, V_{PT}\}$ commits to the conceptualization C by approximating the intended model $I_K(L)$ through $K_T = \langle C, \Im_T \rangle$, where V_{CT} is the set of constant symbols, while V_{PT} is the set of predicate symbols.

The *Ontological View Transformation Problem* from the source ontological view Φ_S to the target ontological view Φ_T is to find a function (mapping) $\mu = \{\mu_C, \mu_P\}$, where μ_C assigns the constant symbols V_{CT} of the target ontological view to the constant symbols V_{CS} of the source ontological view or empty set ϕ, $\mu_C : V_{CS} \rightarrow V_{CT} \cup \phi$, and μ_P assigns the predicate symbols V_{PT} of the target ontological view to the predicate symbols V_{PS} of the source ontological view or empty set ϕ, $\mu_P : V_{PS} \rightarrow V_{PT} \cup \phi$.

C. Proposed Solution

As the predicate symbols are related to the constant symbols, their transformation should be in the sequence of the constant symbols transformation and the predicate symbols transformation as shown in the ViewTrans algorithm, where sV is the source ontological view vocabulary, tV is the target ontological view vocabulary, nsm is the namespace mapping, sCS is the source ontological view vocabulary's constant symbols, sPS is the source ontological view vocabulary's predicate symbols, tCS is the target ontological view vocabulary's constant symbols, tPS is the target ontological view vocabulary's predicate symbols, $consMapping$ is the transformed source ontological view vocabulary's constant symbol mapping table, $isaMapping$ is the transformed source ontological view vocabulary's ISA predicate symbols, $hasaMapping$ is the transformed source ontological view vocabulary's $HASA$ predicate symbols, $anonMapping$ is the transformed source ontological view vocabulary's anonymous predicate symbols, v is any vocabulary, cs is the set of constant symbols, ps is the set of predicate symbols, ConstantTrans is the constant symbol transformation algorithm discussed later, and PredicateTrans is the predicate symbol transformation algorithm, discussed later.

Algorithm ViewTrans

```
function VIEWTRANS ( sV , tV , nsm ) {
    ⟨sCS , sPS⟩ = PARSE( sV )
    ⟨tCS , tPS⟩ = PARSE( tV )
    consMapping = CONSTANTTRANS( sCS , tCS , nsm )
    ⟨isaMapping, hasaMapping, anonMapping⟩ =
        PREDICATETRANS( sPS , tPS , tV , consMapping )
    return ⟨consMapping, isaMapping,
            hasaMapping, anonMapping⟩
}
function PARSE ( v ) {
        cs = constant symbols of v
        ps = predicate symbols of v
        return ⟨cs , ps⟩     }
```

1) Constant Symbol Transformation

Some constant symbols may have one or more synonyms. In order to make assignment function $\Im$ an invertible function and to utilize its convenient properties, we consider introducing a new concept – Domain Concept Correspondence - so that $\Im$ is a one to one function between this new concept and domain D .

[Definition 6] Given a conceptualization $C = \langle D, W, \Re \rangle$, an intended model $I_K(L)$ where $K = \langle C, \Im \rangle$, language L has a vocabulary V , $\Im : V_C \rightarrow D$ function assigns domain concepts $d \in D$ to constant symbols V_C of V , *Domain Concept Correspondence*, denoted by Cor_d , is the set of corresponding constant symbols that function $\Im$ assigned to for d .

[Definition 7] Given a conceptualization $C = \langle D, W, \Re \rangle$, an intended model $I_K(L)$ where $K = \langle C, \Im \rangle$, language L has a vocabulary V , $\Im : V_C \rightarrow D$ function assigns domain concepts $d \in D$ to constant symbols V_C of V , for each domain concept $d \in D$, Cor_d is the domain concept correspondence of it, *Namespace Mapping* M_V^D of vocabulary V on domain D is the set of domain concept correspondence Cor_d on domain D, $M_V^D = \{Cor_d | d \in D\}$, so that for each $d \in D$, there exists one and only one domain concept correspondence Cor_d of it.

The Invertible Function Solution for Constant Symbol Transformation can be described below:

[Proposition 3] Given a conceptualization $C = \langle D, W, \Re \rangle$, an intended model $I_K(L)$ where $K = \langle C, \Im \rangle$, if language L has a vocabulary V , M_V^D is the namespace mapping of V on domain D , $\aleph : M_V^D \rightarrow D$ function assigns domain concepts $d \in D$ to domain concept correspondence Cor_d of namespace mapping M_V^D · $\Phi_S = \langle L, K_S \rangle$ is the source ontological view, represented by language L with a vocabulary $V_S \subseteq V$, $V_S = \{V_{CS}, V_{PS}\}$ commits to the conceptualization C by approximating the intended model $I_K(L)$ through $K_S = \langle C, \Im_S \rangle$, where V_{CS} is the set of constant symbols, while V_{PS} is the set of predicate symbols. $\Phi_T = \langle L, K_T \rangle$ is the target ontological view, represented by tlanguage L with a vocabulary $V_T \subseteq V$, $V_T = \{V_{CT}, V_{PT}\}$ commits to the conceptualization C by approximating the intended model $I_K(L)$ through $K_T = \langle C, \Im_T \rangle$, where V_{CT} is the set of constant symbols, while V_{PT} is the set of predicate symbols. Then the constant symbol transformation from the set of source ontological view constant symbols V_{CS} to the set of target ontological view constant symbols V_{CT} is the union of:

i) The intersection of V_{CS} and V_{CT}: $V_{CS} \cap V_{CT}$;

ii) The subset of the source ontological view constant symbols is not in the target ontological view constant symbols, yet has a mapping with the target ontological view constant symbols, according to namespace mapping M_V^D:

$$\left\{ x \middle| \exists cor \begin{pmatrix} (cor \in M_V^D) \wedge (x \in (V_{CS} - V_{CT})) \\ \wedge (y \in V_{CT}) \wedge (x \in cor) \wedge (y \in cor) \end{pmatrix} \right\}$$

$$\mu_C : V_{CS} \rightarrow \begin{pmatrix} (V_{CS} \cap V_{CT}) \cup \\ \left\{ x \middle| \exists cor \begin{pmatrix} (cor \in M_V^D) \\ \wedge (x \in (V_{CS} - V_{CT})) \\ \wedge (y \in V_{CT}) \\ \wedge (x \in cor) \wedge (y \in cor) \end{pmatrix} \right\} \end{pmatrix}$$

Therefore, the constant symbol transformation problem can be simplified as finding a mapping constant symbol for the given constant symbol from the namespace mapping. The algorithm, ConstantTrans, takes the set of source ontological view constant symbols, the set of target ontological view symbols and namespace mapping as input. It then returns a constant symbol mapping table that contains the overlapping constant symbols with the namespace conflict constant transformed, and the source constant symbol that contains the non-transformable ones remain.

2) Predicate Symbol Transformation

Each conceptual relation ρ relates a set of objects by an axiom. The nature of these objects leads to the categorization of a conceptual relational axiom. There are three classes of conceptual relational axiom: (1) ISA; (2) HASA; and (3) Anonymous conceptual relation ρ .

The propositional calculus has many useful properties, such as reflexivity, irreflexivity, symmetry, asymmetry, antisymmetry, transitivity, inverse, cyclic constrain, composition and partition. Since the 3-ary logical predicate is ordered, most properties are excluded. Of the above, the transitivity and cyclic constrain propositional properties are the most meaningful and useful for inference. We have a series of deductive propositions for ISA , $HASA$ and ρ conceptual relations and corresponding symbols as inference rules.

The ISATrans algorithm takes the set of source ontological view predicate symbols, the set of the target ontological view vocabulary symbols and mapping table, returned by the ConstantTrans algorithm as input, and returns an updated mapping table with the transformable source ontological view, the ISA predicate symbol and corresponding target ontological view vocabulary symbol and an updated set of source ontological view predicate symbols, excluding those transformable ISA symbols.

Since only those transformable ISA symbols' $HASA$ predicates are worthy of investigation, the HASATrans algorithm takes the set of source ontological view predicate symbols, the set of target ontological view vocabulary symbols, the mapping table returned by the ConstantTrans algorithm and the mapping table returned by the ISATrans algorithm, as input. It then returns a mapping table $hasaMapping$ with tuples of the transformable source ontological view $HASA$ predicate symbol and corresponding target ontological view vocabulary symbol and an updated set of source ontological view predicate symbols, excluding those transformable $HASA$ symbols.

The AnonymouTrans algorithm takes the sets of source and target ontological view predicate symbols, the mapping table returned by the ConstantTrans algorithm and the mapping table returned by the ISATrans algorithm, as input. It then returns the mapping table $predMapping$ with tuples of the transformable source ontological view anonymous predicate symbol, the corresponding transformed target ontological view predicate symbol and an updated set of source ontological view predicate symbols, excluding those transformable anonymous predicate symbols.

Finally, the PredicateTrans algorithm guides the whole procedure of predicate symbol transformation.

Algorithm PredicateTrans

function PREDICATETRANS (sPS , tPS , tV , $consMapping$) {

$\langle isaMapping, sPS \rangle =$
ISATRANS(sPS , V_T , $consMapping$)

$\langle hasaMapping, sPS \rangle =$
HASATRANS
(sPS , V_T , $consMapping$, $isaMapping$)

$\langle anonMapping, sPS \rangle =$
ANONYMOUSTRANS
(sPS , tPS , $consMapping$, $isaMapping$)

return $\langle isaMapping, hasaMapping, anonMapping \rangle$ }

IV. IMPLEMENTATION

The proposed Ontological View Transformation algorithms have been implemented and tested on the OpenCDMSS (Open Cooperative Distributed Manufacturing Scheduling System) prototype, as shown in Fig. 4.

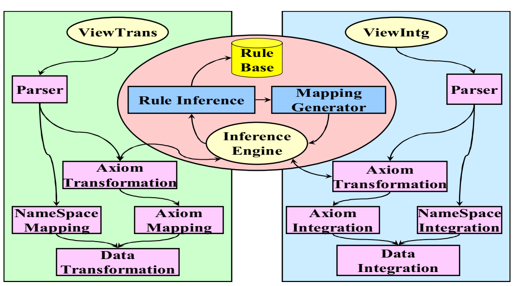


Figure 4. Logical Structure of Semantic Services

The proposed algorithms have been validated in the open distributed environment by different ontological views about the manufacturing scheduling domain in four aspects: (1) constant symbol heterogeneity; (2) ISA symbol heterogeneity; (3) HASA symbol heterogeneity; and (4) anonymous predicate symbol heterogeneity.

V. CONCLUSIONS

This work tackled the semantic heterogeneity problem raised from the nature of open distributed systems. Our main contributions include: (1) selection of a proper ontology definition as our research base; (2) proposal of the novel ontological view concept; (3) proposal of the ontological view transformation algorithms based on the above concept definition. The advantages of our approach are: (1) it is based on none global ontology assumption; and (2) it transforms the ontological view from one to another completely automatically. These advantages assure the proposed algorithms-based semantic services survive in a true open environment without a pre-agreed global ontology, and without human interruption.

This work can be extended in three directions: (1) further study the ontological view integration, based on the algorithms proposed in this work; (2) add a language translation layer on top of current semantic services in case different ontological views use different representation languages; (3) extend the proposed algorithms by considering an additional type of heterogeneity other than structural differences.

Ontological view transformation and its theoretical foundation can benefit current or future information retrieval and data mining technologies, such as Web 2.0, Facebook, and Google. This research would back up these technologies with semantic services and empower these technologies for real world applications.

REFERENCES

[1] Batini, C., Lenzerini, M., Navathe, S., "A comparative analysis of methodologies for database schema integration", ACM Computing Survey, 18(4), 323-364, December 1986.

[2] Broekstra, J., Klein, M., Decker, S., Fensel, D., van Harmelen, F., Horrocks, I.: "Enabling knowledge representation on the Web by extending RDF Schema", World Wide Web, pp. 467-478, 2001.

[3] Calvanese, D., De Giacomo, G., "Data integration a logic-based perspective", AI magazine, 26(1), 59-70, Spring 2005.

[4] Corcho, O., Gomez-Perez, A., "Evaluating knowledge representation and reasoning capabilities of ontology specification languages", ECAI-00, Workshop of Applications of Ontologies and Problem-Solving Methods, 2000.

[5] Fensel, D., Angele, J., Decker, S., Erdmann, M., Schnurr, H.P., Studer, R., "Lessons learned from applying AI to the Web", International Journal of Cooperative Information Systems, 9(4), 361-382, 2000.

[6] Farquhar, A., Fikes, R., Rice J., "The Ontolingua server: a tool for collaborative ontology construction", Technical report, Stanford KSL 96-26, 1996.

[7] Fikes, R., Farquhar, A., "Distributed repositories of highly expressive reusable ontologies", IEEE Intelligent Systems, 14(2), 73-79, 1999.

[8] Gangemi, A., Pisanelli, D. M., Steve G., "An overview of the onions project: applying ontologies to the integration of medical terminologies", Data Knowledge Engineering, 31(2), 183-220, 1999.

[9] Guarino, N., "Formal ontology and information systems", Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998. pp. 3-15.

[10] Gruninger, M., Kopena, J.B., "Semantic integration through invariants", AI magazine, 26(1), 11-20, Spring 2005.

[11] Genesereth, M. R., Nilsson, N. J., "Logical foundation of artificial intelligence", Morgan Kaufmann, Los Altos, California, 1987.

[12] Kashyap, V., Sheth, A., "Semantic heterogeneity in global information systems: the role of metadata, context and ontologies", Inn Cooperative Information Systems: Current Trends and Directions, Papazoglou, M., Schlageter, G., (editors), pp. 139-178, 1998.

[13] Li, X., Morie, P., Roth, D., "Semantic integration in text from ambiguous names to identifiable entities", AI magazine, 26(1), 45-58, 2005.

[14] Pinto, H.S., "Some issues on ontology integration", Proceedings of the IJCAI-99 Workshop: Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, August 2, 1999.

[15] Pisanelli, D. M., Gangemi, A., Steve G., "Ontologies and information systems: the marriage of the century? ", Proceedings of Lyee Workshop, Paris 2002

[16] Rahm, E., Bernstein P. A., "A survey of approaches to automatic schema matching", VLDB Journal 10, 334-350, 2001.

[17] Sheth, A. P., Larson, J. A., "Federated database systems for managing distributed heterogeneous, and autonomous databases", ACM Computing Surveys, 22(3), 183-236, 1990.