Ontological view based semantic transformation for distributed systems
Wang, Y. D.; Ghenniwa, H.; Shen, W.
Ontological view based semantic transformation for distributed systems

NRCC-51371

Wang, Y.D.; Ghenniwa, H.; Shen, W.

October 2009

A version of this document is published in / Une version de ce document se trouve dans: 2009 IEEE International Conference on Systems, Man, and Cybernetics (Special Session on Collaborative Commerce), San Antonio, Texas, October 11-14, 2009, pp. 4107-4112.
Ontological View Based Semantic Transformation for Distributed Systems

Ying Daisy Wang
Southwestern University of Finance and Economics, Chengdu, P.R.
Chinawangying_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, \( K = (C, \mathcal{A}) \) is an ontological commitment of view for \( L \), where \( C = (D, W, \mathcal{R}) \) is a conceptualization, the intended model \( I_k \) is a subset of the view ontological commitment assignment function \( \exists' \), the view assignment function \( \exists' \) is a subset of the possible ontology’s ontological commitment assignment function \( \exists \). The view vocabulary \( V' \) is a subset of the possible ontology vocabulary \( V \). \( \forall V \subseteq V' \), \( \exists: V \rightarrow D \cup \mathcal{R} \) function assigning elements of \( D \) to constant symbols of \( V' \), and element of \( \exists' (\mathcal{R} \subseteq \mathcal{R}) \) to predicate symbols of \( V' \).

[Definition 2] Assuming the intended world structure is fixed, given a language \( L \) with ontological commitment \( K \), an ontological view for \( L \) 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 \( L \) according to \( K' \).

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 $\Phi_s$ with intended model $I_k(\Phi_s)$ and a target ontological view $\Phi_t$ with intended model $I_k(\Phi_t) - \Phi_s$ is transformable (denoted by $\infty$) to $\Phi_t$ if and only if $I_k(\Phi_t)$ is equal to $I_k(\Phi_t) - \Phi_s$.

**B. Problem Formulation**

![Figure 3: Relationship between Source Ontological View Vocabulary, Target Ontological View Vocabulary and Intended Model Vocabulary](image)

As shown in Fig. 3, the intended model vocabulary $V_I$ is constituted by constant symbols $v_C$ and predicate symbols $v_P$:

$$V_I = \{v_C, v_P\}$$  (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 $L$, $\Phi_S$, and $\Phi_T$. The following theorem holds.

**Theorem 1** Given a set of intended models $I_k(L)$ of $L$ according to $K$, a source ontological view $\Phi_S = \{L, K\}$ and a target ontological view $\Phi_T = \{L, K\}$ if language $L$ has vocabulary $V$, ontological commitment $K$ maps the conceptualization to vocabulary $V_S$, where $V_S \subseteq V$, the view ontological commitment $K_S = \{C, A\}$ maps the conceptualization to vocabulary $V_S$, where $V_S \subseteq V$, and the view ontological commitment $K_T = \{C, A\}$ maps the conceptualization to vocabulary $V_T$, where $V_T \subseteq V$, then $I_k(L) \subseteq \Phi_S \cap \Phi_T$.

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 $L$ and the ontological commitment $K^*$.

**Definition 4** Given a source ontological view $\Phi_S$ and a target ontological view $\Phi_T$, a transformation function $\mu: \Phi_S \rightarrow \Phi_T$ is a function that maps elements of $\Phi_S$ to elements of $\Phi_T$.
And from the definition given above, we can have the following proposition.

**[Proposition 2]** Given a conceptualization $C=\langle D, W, R \rangle$, 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 $K_s=\langle C, \emptyset \rangle$ and a target model $I_k(L)$ through $K_t=\langle C, \emptyset \rangle$. As for each ontological view $\Phi_t=\langle L, K_t \rangle$, it's target ontological view vocabulary's anonymous constant symbols $V_s$ uniquely identifies the target $\Phi_t$. Therefore, under such an assumption, the vocabulary and ontological view has a one-to-one relationship. The source view vocabulary $\Phi_s$ is equal to $\Phi_t$ if and only if $V_s$ is equivalent to $V_t$. Thus, the following theorem holds.

**[Theorem 3]** For a given conceptualization $C=\langle D, W, R \rangle$, $\Phi_s=\langle L, K_s \rangle$ 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 $\Phi_s$ and $\Phi_t$ both approximate a common intended model $I_k(L)$, $\Phi_s$ is a subset of $\Phi_t$, and $\Phi_s=\Phi_t \cap \Phi_s$, $\Phi_s$ is a subset of $\Phi_t$, and $\Phi_s=\Phi_t \cap \Phi_s$ represents the target vocabulary of $L$, denoted by $V_t$, commits to $C$ by $K_t$, and $\Phi_s$ represents by the subset vocabulary of $L$, denoted by $V_s$, commits to $C$ by $K_s$, then $V_s$ is transformable to $V_t$, $V_s \subseteq V_t$.

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 $C=\langle D, W, R \rangle$, $\Phi=\langle L, K \rangle$ is the source ontological view, represented by language $L$, with a vocabulary $V_c \subseteq V \subseteq \{V_c, V_p\}$ commits to the conceptualization $C$ by approximating the intended model $I_k(L)$ through $K=\langle C, \emptyset \rangle$, where $V_c$ is the set of constant symbols; $V_p$ 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 $K_t=\langle C, \emptyset \rangle$, 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 $\Phi_s$ to the target ontological view $\Phi_t$ is to find a function (mapping) $\mu=\langle \mu_c, \mu_p \rangle$, where $\mu_c$ assigns the constant symbols $V_{cs}$ of the target ontological view to the constant symbols $V_{ct}$ of the source ontological view or empty set $\emptyset$, $\mu_p: V_{cs} \rightarrow V_{ct}$, and $\mu_p$ 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 $\emptyset$.

**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.

**Algorithm ViewTrans**

```plaintext
function ViewTrans(sV, tV, nsm) {
    \{sCS,sPS\} = PARSE(sV)
    \{tCS,tPS\} = PARSE(tV)
    consMapping = CONSTANTTrans(sCS, tCS, nsm)
    isaMapping = isaMapping + consMapping
    \{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\}
}
```

SMC 2009
1) Constant Symbol Transformation

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

[Definition 6] Given a conceptualization \( C = \langle D, W, \rho \rangle \), an intended model \( I_{\text{c}}(L) \) where \( K = \langle c, \lambda \rangle \): language \( L \) has a vocabulary \( \forall c, V : V \rightarrow D \) function assigns domain concepts \( d \in D \) to constant symbols \( V_c \) of \( V \), Domain Concept Correspondence, denoted by \( \text{Cor} \), is the set of corresponding constant symbols that function \( \exists \) assigned to for \( d \).

[Definition 7] Given a conceptualization \( C = \langle D, W, \rho \rangle \), an intended model \( I_{\text{c}}(L) \) where \( K = \langle c, \lambda \rangle \): language \( L \) has a vocabulary \( \forall c, V : V \rightarrow D \) function assigns domain concepts \( d \in D \) to constant symbols \( V_c \) of \( V \), for each domain concept \( d \in D \), \( \text{Cor} \) is the domain concept correspondence of it. Namespace Mapping \( \Phi = \langle \text{V}, \text{D} \rangle \) of vocabulary \( \forall \) on domain \( D \) is the set of domain concept correspondence \( \text{Cor} \) of domain \( D \), \( \Phi = \{ \text{Cor}, \{ d \in D \} \} \), so that for each \( d \in D \), there exists one and only one domain concept correspondence \( \text{Cor} \) of it.

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

[Proposition 3] Given a conceptualization \( C = \langle D, W, \rho \rangle \), an intended model \( I_{\text{c}}(L) \) where \( K = \langle c, \lambda \rangle \): if language \( L \) has a vocabulary \( \forall \), \( \Phi = \langle L, K \rangle \) is the source ontological view, represented by language \( L \) with a vocabulary \( \forall c, V \rightarrow V \), \( \forall c = \{ V_c, V_{ps} \} \) commits to the conceptualization \( C \) by approximating the intended model \( I_{\text{c}}(L) \) through \( K = \langle c, \lambda \rangle \), where \( V_c \) is the set of constant symbols, while \( V_{ps} \) is the set of predicate symbols. \( \Phi = \langle L, K \rangle \) is the target ontological view, represented by language \( L \) with a vocabulary \( \forall c, V \rightarrow V \), \( \forall c = \{ V_c, V_{ps} \} \) commits to the conceptualization \( C \) by approximating the intended model \( I_{\text{c}}(L) \) through \( K = \langle c, \lambda \rangle \), where \( V_{cs} \) is the set of constant symbols, while \( V_{ps} \) is the set of predicate symbols. Then the constant symbol transformation from the source of ontological view constant symbols \( V_{cs} \) to the set of target ontological view constant symbols \( V_{ct} \) is the union of:

1) The intersection of \( V_{cs} \) and \( V_{ct} \);
2) The subset of the source ontological view constant symbols that is not in the target ontological view constant symbols, yet has a mapping with the target ontological view constant symbols, according to namespace mapping \( \Phi = \langle L, K \rangle \):

\[
\left\{ x \mid \exists y \left( \begin{array}{c}
\text{cor} \left( x, y \right) \\
\left( x \in V_{cs} \setminus V_{ct} \right)
\end{array} \right) \right\}
\]

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 \( \rho \) 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 \( \rho \).

The propositional calculus has many useful properties, such as reflection, 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 \( \rho \) 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

\( (sPS, tPS, \cdot, V, \cdot, consMapping) \) { 
  \( \{\text{isaMapping}, sPS\} = \) 
  ISATrans\((sPS, V, \cdot, consMapping)\) 
  \( \{\text{hasaMapping}, g, sPS\} = \) 
  HASATrans\((sPS, V, \cdot, consMapping, \cdot, \text{isaMapping})\) 
  \( \{\text{anonymousMapping}, g, \cdot, \text{isaMapping}\} = \) 
  ANONYMOUSTrans\((sPS, tPS, \cdot, consMapping, g, \cdot, \text{isaMapping})\) 
  return \( \{\text{isaMapping}, \text{hasaMapping}, \text{anonymousMapping}\} \)
}

### 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.

![Logical Structure of Semantic Services](image)

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**


