Frame-based ontological view for semantic integration

Xue, Yunjiao; Ghenniwa, Hamada H.; Shen, Weiming

Publisher's version / Version de l'éditeur:
https://doi.org/10.1016/j.jnca.2011.02.010
Journal of Network and Computer Applications, 35, 1, pp. 121-131, 2012-01-01

Frame-based Ontological View for Semantic Integration

Yunjiao Xue [a, *], Hamada H. Ghenniwa [a], Weiming Shen [a, b]

[a] *Department of Electrical and Computer Engineering, The University of Western Ontario, London, Ontario, Canada*

[b] *Centre for Computer-assisted Construction Technologies, National Research Council, London, Ontario, Canada*

Abstract

Semantic integration is crucial for successful collaboration between heterogeneous information systems. Traditional ontology-driven approaches rely on the availability of explicit ontologies. However, in most application domains, this prerequisite cannot be met. In order to address this issue, this paper investigates the theoretical foundation of ontologies and extends the traditional ontology concept to an ontological view concept. To explicitly and formally specify the ontological views, a Frame-based Ontological view Specification Language (FOSL) is proposed. This language is based on the *frame* knowledge representation paradigm and uses XML as the encoding. The ontological view-driven semantic integration can be achieved based on the specifications. A proof-of-concept prototype environment has been implemented to achieve semantic integration based on ontological views specified with FOSL.

Keywords:

Semantic integration

Ontology

Ontological view

Frame

Specification language

---

[*] Corresponding author. Address: Department of Electrical and Computer Engineering, The University of Western Ontario, London, Ontario, Canada N6A 5B9
Tel.: +1 519-661-2111 Ext. 80344 Email address: yxue24@uwo.ca

1. Introduction

An information system is a combination of infrastructures, organizations, personnel, and software components within a specific boundary for collecting, processing, persisting, transferring, presenting, delivering, and exchanging information. In the past several decades, a great number of information systems have been developed and deployed. More systems are under design and development.

Information systems are usually deployed in an open environment. An open environment is a computing environment where various platforms, technologies, protocols, and standards coexist, and decentralized applications collaborate through interoperability.

In the open environments, various information systems, such as different collaborative design and manufacturing systems (Shen et al., 2001), are expected to collaborate to support information exchange, processing, and decision-making. With interactions and interoperations among them, the systems are able to achieve common goals collaboratively, working like a single system. Such a virtually single system is an integration of multiple ones, and such integration requires the underlying systems to understand, communicate, and cooperate with each other. Among these three goals, the most fundamental one is to make the systems understand each other and achieve common agreements on domain concepts and relationships managed by different systems.

However, as the information systems are growing larger, more complex, and more distributed, it becomes increasingly difficult for anyone to effectively organize and work with the information and systems. As a case, semantics-based information integration in various organizations has been hindered by differences in the software applications and by the structural and semantic heterogeneity of the different information sources (De Bruijn et al., 2003). Due to the nature of being independently designed and built, existing information systems, even for the same domain, are often heterogeneous in terms of their: (1) supporting infrastructures (hardware platforms, operating systems, communication facilities, etc.); (2) syntactic representations of information; (3) schematic designs of information models, and (4) semantics of information. These heterogeneities present major practical and research challenges. This problem has made information retrieval and collaboration among information systems extremely difficult. As such, there is a requirement to integrate these information sources and applications to provide consistent services to global users.

There have been many proposed solutions for the first three areas of concerns (Sheth, 1999). The fourth one, also known as the *semantic integration* problem (Noy, 2004), is attracting growing attention from today's research and development communities.

Semantic integration intends to resolve *semantic incompatibility / heterogeneity* among various information systems. The major reason for semantic incompatibility / heterogeneity is the lack of specifications for the semantics of information. Ontology-driven semantic integration is a category of solutions for the semantic integration problem (Hakimpour and Timpf, 2001). The traditional solutions are based on available ontologies. *Ontology mapping* (Kalfoglou and Schorlemmer, 2003) / *alignment* (Sowa, 1997) can be applied to discover semantic correspondences among a set of formal ontologies. *Ontology integration* (Wache et al., 2001) / *fusion* (Sun et al., 2010) will result in a more complete ontology, given that multiple source ontologies are available. However, in most application domains there are no pre-established explicit ontologies available. Instead, the "ontologies" are implied in different formats, such as the underlying information representations. For example, a database-centralized information system may work based on a relational database schema. The schema is not a formal ontology but to some extent it specifies the semantics of information that it manages. For instance, a relational database schema contains multiple tables and each table may represent a concept. Accordingly, data rows in a table represent instances of the concept. In fact, each system reflects a specific conceptual view of the domain conceptualization, which, in this work, is defined as an *ontological view*. The formal definition of *ontological view* will be provided in Section 3. As a type of concept model, where a specification language is required, an ontological view will be explicitly and formally specified.

This paper proposes to use ontological views to explicate the semantics of information models of information systems in open environments where no explicit “ontologies” are available. It discusses the modeling of ontological views with a frame and proposes a frame-based language for specifying the ontological views. An ontological view acts as a vehicle that carries semantics and serves various approaches that must be applied on explicit specifications/representations for resolving semantic heterogeneities and achieving semantic integration. The rest of the paper is organized as follows: Section 2 reviews some literature with topics related to ontology and semantic integration; Section 3 defines the fundamental terms and discusses the use of the *frame* knowledge representation paradigm; Section 4 proposes an ontological view specification language; Section 5 presents the implementation and analysis of the proposed language; Section 6 concludes the paper with some discussion on future work.

2. Literature Review

Ontology plays an important role in understanding and dealing with information semantics. Ontology has been recognized as a fundamental infrastructure for advanced approaches to knowledge management (Arroyo, 2007). Ontologies are useful for many different applications that can be classified into several areas (Jasper and Ushold, 1999). The common idea for all of these applications is to use ontologies to reach a common understanding of a particular domain (Stuckenschmidt and van Harmelen, 2005), that may be reused and shared across applications and groups (Chandrasekaran et al., 1999). The use of ontologies also helps to reach a common understanding of the meaning of terms. In contrast to syntactic standards, the understanding is not restricted to a common representation or a common structure. Therefore, ontologies are a promising candidate that can support semantic interoperability and information retrieval, especially in the Semantic Web (Berners-Lee et al., 2001).

Many definitions of "ontology" have been proposed. Basically, ontology is a formal and explicit specification of a shared conceptualization (Gruber, 1995), used to help computer programs and humans share knowledge (Gruber, 1993). Here,

- "explicit" means that "the type of concepts used and the constraints on their use are explicitly defined";
- "formal" refers to the fact that "it should be machine readable";
- "shared" refers to the fact that "the knowledge represented in an ontology is agreed upon and accepted by a group";
- "conceptualization" refers to an abstract model that consists of the relevant concepts and relationships that exist in a certain situation. In another sense, a conceptualization is an abstract and simplified view of the world that we wish to model for a particular purpose.

Ontology can also be understood as a model that defines the concepts, properties of concepts, and the relations between concepts of a domain of discourse (Crubzy et al., 2003). Some people view ontology, in the simplest case, as a hierarchy of concepts related by subsumption relationships, such as things, events, and a set of relations that are specified in some way to create an agreed-upon vocabulary for exchanging information. As described in (Tan et al., 2006), ontologies are used for communication between people and organizations by providing a common terminology over a domain. They provide the basis for interoperability between systems. Ontology establishes a joint terminology between members of a community of common interests and these members can be humans or software agents. It can be viewed as a semantic substrate for information integration and aggregation processes, providing explicit semantics which may be useful for information exchange between heterogeneous sources.

Formal ontologies are considered more than schema definitions in databases. Schemas are mainly concerned with organizing data in databases, but formal ontologies are concerned with the understanding of the members of a community and help to reduce ambiguity in communications.

The ability to exchange information at run time, also known as interoperability, is an important topic. Ontologies are often used as interlinguas for providing interoperability (Uschold and Gruninger, 1996): they serve as a common format for data interchange. Each system that wants interoperatability with other systems has to transfer its information into this common framework.

Ontologies provide machine-readable semantics of information sources that can be communicated between applications and humans. The use of ontologies as semantic translators is a viable approach to overcome the problem of semantic heterogeneity (Hakimpour and Timpf, 2001).

Ontologies are expressed in languages that are machine processable and can be used for reasoning (Noy, 2003). The expressed artifact is also called an "ontology model", given that the ontology itself is abstract. The description of information semantics (local ontologies or conceptual models of information sources) may be represented in ER [1], UML [2], RDF [3], or other logic models. Many ontology languages have been proposed. Some are based on description logics (DL) (Badder and Sattler, 2001), such as OWL [4] and LOOM (Arens et al., 1996), and some are frame-based, such as Frame Representation Language (FRL) (Robert and Goldstein, 2002).

In ontology-driven semantic integration approaches, the integration is obtained by sharing common ontologies among different information sources. Mappings are created between the ontologies and local information models. Traditionally, such integration is based on available explicit ontologies provided in a domain. However, this is not always true in an open computing environment. Instead, the "ontologies" are

---

[1] http://en.wikipedia.org/wiki/Entity-relationship_model
[2] http://www.uml.org/
[3] http://www.w3.org/RDF/
[4] http://www.w3.org/TR/owl-features/

implied in different formats, such as the underlying information representations. This situation reveals a gap between the traditional solutions and the actual open environments. New research is required to bridge such a gap.

Minsky's *frame* theory (Minsky, 1975) is a major milestone in the history of knowledge representation. Proposed in the 1970s, this theory suggests the idea of using object-oriented groups to define a frame which is the data structure that represents stereotypical situations (Brachman and Levesque, 2004). It can represent the world meaningfully and naturally, and, for domain experts, it is cognitively simple, intuitive, and understandable. Frames have been widely used in artificial intelligence and knowledge-based systems. Frame-like structures, in combination with rules, are used extensively in expert systems (Aikins, 1993). Some recent examples of applying frames to knowledge representation can be found in (Kiatisevi et al., 2006; Marinov, 2008).

As defined in the Open Knowledge-Base Connectivity (OKBC) specification (OKBC, 1995), *frame* is one of the most widely-used ontology modeling paradigms. It is implemented in the core Protégé, a widely used tool for creating, editing, browsing, and maintaining ontologies (Protégé, 2010).

Some researchers view *frame* itself as a modeling language, such as (Brachman and Levesque, 1984), compared to other modeling paradigms such as production rules, description logics, and semantic networks. We view *frame* as a modeling paradigm at the conceptual level. From the system's perspective, there should be a specification language that provides structures and semantics to encode frames. However, as yet, there is no standard frame specification language (Wang et al., 2006).

3. Theoretical Foundation

3.1 Ontology and Ontological View

In the following paragraphs, we present a set of definitions based on the work of Guarino (1998) and Genesereth and Nilsson (1987) that are necessary for formally defining *ontology* from a traditional perspective.

The *World* is the entire aggregation of everything that exists anywhere. The existing things in the world are perceived as *Concepts*. A *Domain* is a portion of the world that is related to a problem to be solved. Formally, domain D is defined as a set of *concepts* that exist in the domain, i.e., D = $\{C_1, C_2, \ldots, C_n\}$ where each C_i is a concept.

A *state of affairs* describes a possible situation about how concepts are related to each other. A state of affairs is a certain type of proposition. Its existence depends on whether the proposition is said to be true or false (Menzel, 2008). A state of affairs is said to *include* a second state of affairs if it is impossible for the former to exist and the latter to fail to exist. A state of affairs is said to *preclude* a second state of affairs if it is impossible for them both to exist. A state of affairs is called *maximal* if, for every other state of affairs, it either includes or precludes that other state of affairs (Plantinga and Davidson, 2003). A *maximal state of affairs* is also called a *possible world*. The set of possible worlds (maximal state of affairs) of a domain is denoted as W, W = $\{w_1, w_2, \ldots, w_n\}$ where each w_i is a possible world.

A *domain space* is a structure <D, W>, where D is a domain and W is a set of possible worlds of the domain. Given a domain space <D, W>, a *conceptual relation* ρ^n of arity *n* is a function from a set W of possible worlds to the set of all *n*-ary relations on D, 2^{D^n}, $\rho^n : W \rightarrow 2^{D^n}$.

A *conceptualization* of domain D is defined as an ordered triple **C** = <D, W, $\Re$>, where $\Re$ is a set of conceptual relations on the domain space <D, W>.

For each possible world $w \in W$, the *intended structure* of *w* according to a conceptualization **C** = <D, W, $\Re$> is the structure $\mathbf{S}_{w\mathbf{C}}$ = <D, $\mathbf{R}_{w\mathbf{C}}$>, where $\mathbf{R}_{w\mathbf{C}} = \{\rho(w) \mid \rho \in \Re\}$ is the set of extensions (relative to *w*) of the elements of $\Re$. We use $\mathbf{S_C} = \{\mathbf{S}_{w\mathbf{C}} \mid w \in W\}$ to denote all the intended structures of **C** (also called *world structure*).

A *logical language* **L** is a composition of a vocabulary V and a set of models of the language. V contains constant symbols and predicate symbols. Given a logical language **L** with a vocabulary V, a *model* of **L** is a structure $\langle \mathbf{S}, I \rangle$, where $\mathbf{S} = \langle D, \mathbf{R} \rangle$ is a world structure and $I: V \rightarrow D \cup \mathbf{R}$ is an interpretation function assigning elements of D to constant symbols of V, and elements of **R** to predicate symbols of V.

An *intensional interpretation* of a logical language **L** with a vocabulary V is a structure $\langle \mathbf{C}, \Im \rangle$, where $\mathbf{C} = \langle D, W, \Re \rangle$ is a conceptualization and $\Im: V \rightarrow D \cup \Re$ is a function assigning elements of D to constant symbols of V, and elements of $\Re$ to predicate symbols of V. This intensional interpretation is called *ontological commitment* for **L**, denoted as $\mathbf{K} = \langle \mathbf{C}, \Im \rangle$. We also say that **L** commits to **C** by means of **K**, where **C** is the underlying conceptualization of **K**. **K** constrains the intensional interpretation of **L**, i.e., the language is used in an intended way for a domain instead of an arbitrary way.

Given a language **L** with a vocabulary V and an ontological commitment $\mathbf{K} = \langle \mathbf{C}, \Im \rangle$ for **L**, a model $\langle \mathbf{S}, I \rangle$ is *compatible* with **K** if: i) $\mathbf{S} \in \mathbf{S_C}$; ii) for each constant symbol $c \in V$, $I(c) = \Im(c)$; iii) there exists a world w such that for each predicate symbol $p \in V$, I maps the predicate into an admissible extension of $\Im(p)$, i.e., there exists a conceptual relation ρ such that $\Im(p) = \rho \wedge \rho(w) = I(p)$.

Given a language **L** and an ontological commitment **K**, the set $\mathbf{I_K}(\mathbf{L})$ of all models of **L** that are compatible with **K** is called the *set of intended models* of **L** according to **K.** Given a language **L** with an ontological commitment **K**, an *ontology* for **L** is a set of axioms designed in a way such that the set of its models approximates as much as possible the set of intended models of **L** according to **K**.

According to the above definition, "ontology" is a "designed" artifact that is committed to a conceptualization by means of an ontological commitment. It reflects the designer's view of the conceptualization. Actually, there is not merely one unique "ontology" for a conceptualization in an open environment since it can be viewed in various ways. Instead, various views of the conceptualization may exist. When different designers are facing the same conceptualization, it is natural that multiple views will be created. Since each view can be formally and explicitly specified, we define the corresponding specification as an *ontological view*. Accordingly, its intensional interpretation is called an *ontological commitment of view*. There can be multiple ontological views for a single conceptualization. As for information systems, each system implies an ontological view of the conceptualization of the domain for which it is built.

Different languages can be employed to design ontological views. Further, if two languages are employed for ontological views with partially overlapping intended models, it is possible for the corresponding ontological views to be semantically integrated. Formally, given one ontological view O with intended models $\mathbf{I_K(L)}$ and another ontological view O' with intended models $\mathbf{I_{K'}(L')}$, O and O' are integrateable (denoted by $\Diamond$) if and only if $\mathbf{I_K(L)}$ overlaps with $\mathbf{I_{K'}(L')}$. That is,

$$(\mathbf{I_K(L)} \neq \mathbf{I_{K'}(L')}) \wedge (\mathbf{I_K(L)} \cap \mathbf{I_{K'}(L')} \neq \emptyset) \leftrightarrow (O \Diamond O')$$

This can be illustrated through Fig. 1.

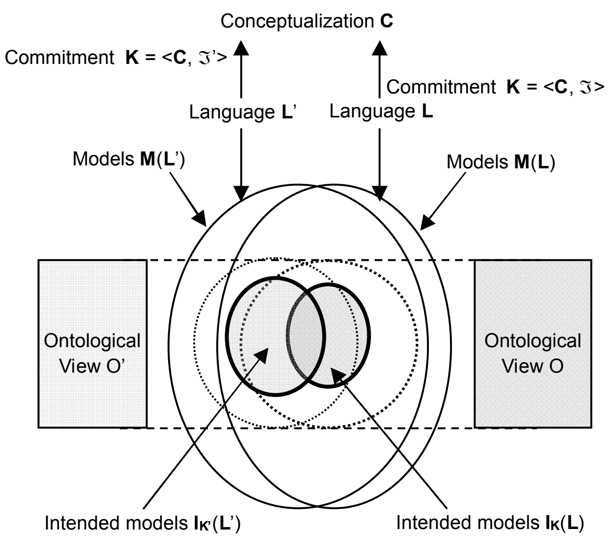

Fig. 1. Different ontological views with different languages which sets of intended models overlap. This figure is an adoption of a similar figure from (Guarino, 1998).

3.2 Semantic Integration

Information systems are built based on *information models* which are explicit models specifying information in the systems, such as a database schema. Given a set of information models IM_1, IM_2, …, IM_n, the semantic integration upon them includes two aspects:

(1) For any two elements e_i and e_j from IM_i and IM_j, if they refer to the same concept in terms of the domain of discourse, no matter how they are represented, such a fact can be discovered.

(2) For any element e_i from IM_i, if it is required to be communicated to IM_j (if applicable), it can be converted into another element (referring to the same concept) that is correct in both representation and semantics in IM_j such that IM_j can handle it in a semantically reasonable manner.

Ontology view-driven semantic integration is a mechanism to integrate information at the semantic level using the semantics carried by ontological views in a way that the same concept references of the sets of intended models of multiple ontological views, are identified, modeled, persisted, and reused when performing information access and exchange.

3.3 Hypothesis for Semantic Integration

Based on how information systems are designed for a domain, we can safely assume that there are explicit information models available and the models are committed to the intended models that overlap. The information models are not restricted to a particular language or paradigm, such as relational, XML, or Objected-Oriented (OO). Further, the modeling languages of the information models adopt symbols based on a natural language foundation, such as English. The constant symbols, such as English words, refer to concepts under an ontological commitment.

An ontological view, as an explicitly represented model, can be created from an information model. The ontological views provide a common base that eliminates structural and syntactical heterogeneities among information models, therefore semantic integration can be conducted at the ontological view level.

Before we present the hypothesis of our work, we formally define the *semantically equivalent mapping* (or *equivalence mapping*) between languages:

Given a source language $\mathbf{L}_S$ with the ontological commitment of view $\mathbf{K}_S = <\mathbf{C}, \Im_S>$ and vocabulary V_S, a target language $\mathbf{L}_T$ with the ontological commitment of view $\mathbf{K}_T = <\mathbf{C}, \Im_T>$ and vocabulary V_T, the two languages share the same conceptualization $\mathbf{C} = <D, W, \Re>$, a semantically equivalent mapping is a function from V_S to V_T, m: $V_S \rightarrow V_T$, assigning symbols in V_T to the ones in V_S which share the same intensional interpretation, i.e., i) for constant symbols $c_S \in V_S$ and $c_T \in V_T$, $m(c_S) = c_T$ if and only if $\exists d \in D$, such that $\Im_S(c_S) = \Im_T(c_T) = d$ and ii) for predicate symbols $p_S \in V_S$ and $p_T \in V_T$, $m(p_S) = p_T$ if and only if $\exists \rho \in \Re$ such that $\Im_S(p_S) = \Im_T(p_T) = \rho$.

In this context, we base our work on the following hypothesis:

If the semantically equivalent relationships between concepts (specified by symbols in languages) in multiple ontological views can be discovered, then these ontological views, as well as the information models from which the ontological views are developed, can be semantically integrated.

To support this hypothesis, we introduce the following two propositions.

(1) A concept in a conceptualization can be externalized by a constant symbol in a language under an ontological commitment.

Proof: According to the definition of the intended model, given a language **L** with an ontological commitment **K**, the set $\mathbf{I_K}(\mathbf{L})$ of all models of **L** that are compatible with **K** is defined as the set of intended models of **L** according to **K**. Therefore, for any two models m_1 and m_2 in $\mathbf{I_K}(\mathbf{L})$, m_1 and m_2 are compatible with **K**. That is, for each constant symbol c in the vocabulary of **L**, there is $\mathbf{I}_1(c) = \mathfrak{I}(c)$ for m_1 where $\mathbf{I}_1$ is the interpretation function of m_1, and $\mathbf{I}_2(c) = \mathfrak{I}(c)$ for m_2, where $\mathbf{I}_2$ is the interpretation function of m_2, and $\mathfrak{I}$ is the interpretation function in **K**. That is, under the given ontological commitment **K,** a constant symbol c is always interpreted as a concept in the domain of discourse.

On the other hand, since **I** is a function in any model, it is guaranteed that c is interpreted as only one concept, say C, under **K**. In other words, it is explicit in concept C. Therefore, if even C is implicit, c can be taken as its representative. Since c is explicit, it can be used for processing the concept that it represents. □

(2) The semantically equivalent relationship between symbols under an ontological commitment implies the same concept reference.

Proof: Given symbols v_1 and v_2 from two ontological views such that v_1 maps to a concept C_1 in a conceptualization and v_2 maps to a concept C_2 in the same conceptualization (Proposition 1), if v_1 and v_2 have a semantically equivalent relationship, then they have the same semantics, i.e., the same concept reference. Therefore, it can be concluded that C_1 and C_2 are actually the same concept in the conceptualization. Consequently, information models corresponding to v_1 and v_2 are semantically equivalent. □

The first proposition indicates that each ontological view has a specific *representation* based on a language since the ontological view is an explicit model. The second proposition shows that the *semantic similarity* between representations of models can be used to approximate the semantically equivalent relationships between the models themselves. Semantic similarity is a metric upon explicitly represented models computed from either a linguistic or structural perspective. Such a metric implies that two models may have the same semantics because their representations are linguistically or structurally similar. This also shows that a representation formalism, on which further analysis can be conducted, is necessary for the ontological views. In the following sections we discuss how the *frame* paradigm can be adopted for modeling ontological views and then we propose a specification language.

3.4 Frame

In the *frame* theory, a frame models a concept which represents a collection of instances. Each frame has an associated collection of slots which can be filled by values or other frames. The slots define different characteristics of the objects or relations through other objects. In particular, frames can have an IS-A slot which allows the assertion of a concept taxonomy.

Structurally, a frame has the following four-level structure:

- The highest level is FRAME, which is a primitive object that represents a concept in the domain of discourse.
- SLOT level captures the properties associated with the concept and relationships to other concepts (frames).
- Within a SLOT, there is FACET level which captures the details of each SLOT. The FACET level contains multiple facets, with each specifying one aspect of the slot, such as data type, cardinality, and value range.
- Finally, DATA level (or INSTANCE level) provides specific information about each property for an instance of the concept. This level is provided to build a complete knowledge base. When modeling concepts, usually the DATA level is not used if the major focus is on the concept itself and does not concern the instances of the concept.

Brachman and Levesque introduced a simple formal representation formulism to express the frame's structure as follows (Brachman and Levesque, 2004):

```
(Frame-name
    <:IS-A frame-name>
    <slot-name1 filler1>
    <slot-name2 filler2>
    ...
)
```

According to this structure, a frame owns a list of slots into which values can be dropped. The items that go into them are called fillers. The fillers of slots that represent relationships are the names of other frames. The frames can have a ":IS-A" slot whose filler is the name of a more generic frame, meaning that the former frame is a specialization of the latter one.

The frame and slot names are atomic symbols (like numbers or strings without further structures). The fillers are either atomic values or the names of other frames.

The frame paradigm covers five characteristics regarding knowledge representation (Jurafsky and Martin, 2008): verifiability, unambiguity, canonical form, reasoning capability, and expressiveness, where there is always a trade-off between good expressiveness and the ease of inference (Brachman and Levesque, 2004).

3.5 Modeling Ontological View with Frame

Support for logical inference is one of the most valued aspects for some knowledge representation paradigms. For example, the OWL DL provides the description-logic reasoning capabilities that enable a reasoning engine to infer knowledge that is not explicitly represented, including subsumption, equivalence, consistency, and instantiation testing.

Different from the knowledgebase systems where logical inference is an essential requirement, the information models within information systems mainly focus on modeling concepts and the characteristics of the concepts in the domain of discourse. Each concept is specified by its own characteristics (even other concepts can be involved to specify its characteristics), not defined by other concepts. The fundamental modeling concept of a DL is the axiom - a logical statement relating to roles and/or concepts (Grau et al., 2008). This is a key difference from the frames paradigm where a *frame specification* declares and completely defines a class (Grau et al., 2008).

Furthermore, the models focus on the stereotype instead of the individual instances. Therefore, the reasoning capability as provided by DL is not an essential element for modeling the ontological views based on the information models, and the instances can usually be ignored.

The concepts are the fundamental elements in the information models. Concepts can be related to each other through relationships. A concept can be modeled as a structure of *C* = <*P*, *hasProperty*>, where *P* is a set of dependent concepts and *hasProperty* is a semantic relationship which associates *P* to *C*. A dependent concept is a concept that is semantically dependent on an extrinsic (independent) concept. Usually, a dependent concept is not being processed solely by itself. Instead, it is always considered along with another concept that it depends on. In this sense a *property* is treated as a dependent concept. Therefore, it can also be claimed that a concept is modeled by a set of properties.

Many of the paradigms used to build information models, such as relational and object-orientation, are following the *concept-property* construct. In this work we believe it will be natural to adopt the *concept-property* construct for modeling ontological views. To sum up, the paradigm for specifying ontological views should support modeling:

(1) Concepts: extrinsic concept is a structure of dependent concepts with a *hasProperty* relationship.

(2) Properties: dependent concepts.

(3) Relationships between concepts such as *is-a* and *part-of*.

As a knowledge modeling paradigm, frame provides a clear and explicit structure that is adequate for modeling the proposed ontological view model, in particular in describing the properties of concepts, which makes frame an ideal candidate for modeling ontological views.

4. A Frame-based Ontological View Specification Language (FOSL)

4.1 Specification of Ontological Views

The ontological views must be explicitly specified in order to be used with information systems, i.e., delivered using some concrete representation.

The specification of an ontological view is composed of: (1) symbols mapped to concepts (as an explicit representation of the intended model), (2) symbols mapped to properties and their associated characteristics, (3) symbols mapped to relationships between concepts, and (4) symbols that logically connect (1), (2), and (3) with specific semantics.

Note that the *language* specifying the ontological views and the *language* specifying the conceptualizations belong to different categories. The former contains the basic elements, syntactical rules upon the elements, and the semantics to specify meaningful models. It is guaranteed that these elements and rules are commonly agreed to by any semantic integration service within a computing environment. The latter refers to the vocabulary that is used to denote the concepts as well as the interpretation of the vocabulary. This language contains symbols that map to concepts, properties, and relationships. This paper focuses on the former.

An information model does not always explicitly describe concepts, properties, or relationships. However, some of its constructs usually imply these elements. For example, in a relational database schema (which is a type of information model), a table can be used to represent a concept; in an XML document, a node can represent a concept. Given that an information model M is specified by language $\mathbf{L}_M = \langle \mathbf{S}_M, \mathrm{I}_M \rangle$ with vocabulary V_M and the ontological view model is specified by language $\mathbf{L}_O = \langle \mathbf{S}_O, \mathrm{I}_O \rangle$ with vocabulary V_O, the creation of an ontological view is to find a mapping m between $\mathbf{L}_O$ and $\mathbf{L}_M$ such that $m(\mathrm{I}_O) \subseteq \mathrm{I}_M$. The mapping requires a set of rules for each modeling paradigm to identify:

- what constructs in the information model can be mapped to concepts;
- what constructs in the information model can be mapped to properties;
- what constructs in the information model can be mapped to facets of the properties;
- what constructs in the information model can be mapped to values of facets;
- what constructs in the information model can be mapped to relationships between concepts.

For example, as to a relational database schema:

- a table which has a primary key is a candidate of a concept;
- each column in the table is a candidate of a property;
- the attributes of the column, such as data type, size, default value, null-able, are candidates of facets;
- the value of the attributes, such as *Integer* and *NULL*, are candidates of values of facets;
- a foreign key column implies a relationship to a concept indicated by the referred table;
- a table that has a combined primary key and each of which column is a foreign key implies a relationship between two concepts indicated by the referred tables.

The explicit specification of ontological views following a specific modeling paradigm provides a common foundation that eliminates the heterogeneities residing in the underlying information models in terms of technical platform, modeling paradigm, specification syntax, etc. Later work such as semantic integration can just focus on the semantic aspect, i.e., the difference between various views of the domain conceptualization, based on a single modeling paradigm without regard for the different ways of modeling and specifying the models.

4.2 Definition of FOSL

We propose a Frame-based Ontological view Specification Language (FOSL) to support specification of the above aspects. It is a logical language created from the following vocabulary:

(1) Constant symbols: the set of $FR \cup S \cup F \cup V$, where FR is a set of symbols referring to frames (concepts), S is a set of symbols referring to slots (properties), F is a set of symbols referring to facets, and V is a set of values that the facets can take.

(2) Variable symbols: there are four sets V_{FR}, V_S, V_F, V_V of variable symbols which ranges are FR, S, F, and V, respectively.

(3) Predicate symbols: the following predicate symbols are defined:

(a) A binary predicate *hasProperty* applied on $FR \times S$. $hasProperty(fr, s)$ refers to a frame $fr \in FR$ with a slot $s \in S$.

(b) A triple predicate *hasFacet* applied on $FR \times S \times F$. $hasFacet(fr, s, f)$ indicates that slot $s \in S$ has a facet $f \in F$ in a frame $fr \in FR$.

(c) A quad predicate *hasValue* applied on $FR \times S \times F \times V$. *hasValue*($fr$, s, f, v) indicates that the slot $s \in S$'s facet $f \in F$ has a value $v \in V$ in a frame fr$\in FR$.

(d) A binary predicate *isA* applied on $FR \times FR$. *isA*(fr_1, fr_2) indicates that frame $fr_1 \in FR$ is a type of frame $fr_2 \in FR$, i.e., the concept modeled by fr_1 is a specialization of the concept modeled by fr_2.

(e) A binary predicate *partOf* applied on $FR \times FR$. *partOf*(fr_1, fr_2) indicates that frame $fr_1 \in FR$ is a part of frame $fr_2 \in FR$, i.e., the concept modeled by fr_1 is a part of the concept modeled by fr_2.

The predicates *isA* and *partOf* specify two types of relationships between concepts selected to be defined in FOSL. The reasoning behind the choice is that these two types provide strict semantics that can be commonly agreed upon among multiple parties. Other relationships are rather arbitrary, resulting in unpredictable semantics. For instance, a frequently used example is "*Student* takes *Course*" where *Student* and *Course* are two concepts and take is a relationship. Here take does not provide inferable semantics but only a human reader can understand its meaning.

Even the predicate *hasFacet* implies *hasProperty* because when *hasFacet*(fr, s, f) holds we also have *hasProperty*(fr, s) (a similar case applies to predicate *hasValue* and *hasFacet*), the individual *hasProperty* predicate is still necessary since it is not guaranteed that every information model is complete. That is, in some models maybe only properties of a concept are listed but details of the properties are missing.

This redundancy also increases the readability of a specification written in FOSL in a way that a layered structure of the concept specification is presented and different reader interests can be well satisfied. For example, given a set of statements with *hasProperty* predicate, it is easy to grasp a general view of a concept, i.e., "this concept is described by this set of properties", without any unnecessary information being involved. If a reader is interested in what a property is like, a set of statements with the *hasFacet* predicate will help. Furthermore, the statements with the *hasValue* predicate provide the lowest level of details for the facets.

4.3 Inference Rules

Now we define the inference rules that can be expressed by the language.

Inheritance Rule:

- *isA*(*subfr*, *superfr*) ← *isA*(*subfr*, *fr*) & *isA*(*fr*, *superfr*), i.e., a frame *subfr* specialized from another frame *fr* is also a specialization of that frame's generalized frame *superfr*.
- *hasProperty*(*subfr*, *s*) ← *isA*(*subfr*, *fr*) & *hasProperty*(*fr*, *s*), i.e., a generic frame's slots are inherited by its specialized frames.
- *hasFacet*(*subfr*, *s*, *f*) ← *isA*(*subfr*, *fr*) & *hasProperty*(*fr*, *s*) & *hasFacet*(*fr*, *s*, *f*), i.e., the facets of a slot of a generic frame are inherited by the same slot of its specialized frames.
- *hasValue*(*subfr*, *s*, *f*, *v*) ← *isA*(*subfr*, *fr*) & *hasProperty*(*fr*, *s*) & *hasFacet*(*fr*, *s*, *f*) & *hasValue*(*fr*, *s*, *f*, *v*), i.e., the value of a facet of a slot of a generic frame is inherited by the same facet of the same slot of its specialized frames.

Composition Rule:

- $\exists$ *fr*$\in$*FR* ← $\exists$ *partialfr*$\in$*FR* & *partOf*(*partialfr*, *fr*), i.e., there must exist a frame where another frame is a part of it.
- *partOf*(*partialfr*, *wholefr*) ← *partOf*(*partialfr*, *fr*) & *partOf*(*fr*, *wholefr*), i.e., if a frame *partialfr* is a part of another frame *fr*, it is also a part of a larger frame *wholefr* which has that other frame as a part of it.

4. 4 Encoding

To explicitly encode ontological views, we propose a human readable and machine processable representation which enables:

(1) The ontological view created from an information model to be verified and refined by human experts;

(2) The semantic integration to be executed in an automated manner based on the analysis applied on the representations.

To this end we adopt an XML-based representation for FOSL. An ontological view can be modeled as a set of frames and represented in an XML document. The document is supported with multiple <*concept*> tags for concepts (frames), respectively. Under a <*concept*> tag the slots are divided into two categories and specified by <*relationships*> and <*properties*>. Under each category there is a collection of individuals, namely <*relationship*> and <*property*>. The *isA* and *partOf* predicates are represented as specific <*relationship*> nodes with pre-defined semantics.

The facets of each slot are tagged as <*facet*> which is described by two attributes: *name* and *value*. To uniquely identify each concept, there is also a sub-tag <*name*> under each <*concept*> tag denoting the name of each concept.

Fig. 2 shows the schema of the XML document derived from FOSL.

```xml
<?xml version="1.0" encoding="utf-16"?>
<xsd:schema attributeFormDefault="unqualified" elementFormDefault="qualified" version="1.0" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="ontological_view" type="ontological_viewType" />
 <xsd:complexType name="ontological_viewType">
  <xsd:sequence>
   <xsd:element maxOccurs="unbounded" name="concept" type="conceptType" />
  </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="conceptType">
  <xsd:sequence>
   <xsd:element name="name" type="xsd:string" />
   <xsd:element name="properties" type="propertiesType" />
   <xsd:element name="relationships" type="relationshipsType" />
  </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="relationshipsType">
  <xsd:sequence>
   <xsd:element name="relationship" type="relationshipType" />
  </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="relationshipType">
  <xsd:sequence>
   <xsd:element name="name" type="xsd:string" />
   <xsd:element name="target_concept" type="xsd:string" />
  </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="propertiesType">
  <xsd:sequence>
   <xsd:element maxOccurs="unbounded" name="property" type="propertyType">
  </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="propertyType">
  <xsd:sequence>
   <xsd:element name="name" type="xsd:string" />
   <xsd:element name="facets" type="facetsType" />
  </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="facetsType">
  <xsd:sequence>
   <xsd:element maxOccurs="unbounded" name="facet" type="facetType" />
  </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="facetType">
  <xsd:attribute name="name" type="xsd:string" />
  <xsd:attribute name="value" type="xsd:string" />
 </xsd:complexType>
</xsd:schema>
```

Fig. 2. XML schema of FOSL.

5. Implementation and Analysis

5.1 Mapping to a Relational Model

In Section 4, we propose to use *frame* as the paradigm to model ontological views. The fundamental elements within *frame* include:

- Frames representing concepts.

- Slots representing properties of concepts or relationships with other concepts.
- Facets representing characteristics of properties.
- Data representing instances of a concept.

These elements should be mapped to constructs provided by an implementation technology (except the data element that may be unnecessary in some modeling situations) to guarantee that the solution can be supported by the technology. We mainly consider the relational model (Codd, 1990) since it is well supported by the relational databases and is widely used in today's information management systems.

Mapping between the frame and the relational model enables the adoption of a relational model as an implementing technology. According to the definition of frame and relational model, the following mapping rules are defined:

- A frame is mapped to a relation.
- A slot of a frame representing a property is mapped to an attribute of a relation.
- A facet of a slot is mapped to a domain.
- A set of values on all slots representing an instance is mapped to a tuple.
- The set of one or more slots that uniquely identify an instance is mapped to a primary key.
- A slot representing a relationship to another frame is mapped to a foreign key or a relation, all of which attributes are foreign keys.

The relational model is the foundation of relational database systems. To apply the relational model,

- *Type* is used to implement a domain. A type may be the set of integers, the set of character strings, the set of dates, or the two boolean values *true* and *false*, and so on. The corresponding type names for these types might be the strings "int", "char", "date", "boolean", etc.

- Attribute is the term used in the relational theory for what is commonly referred to as a *column* in a relational database.
- The database systems provide rich characteristics, besides name and type, for attributes, e.g., value range, null-able, default value, etc.
- *Table* is commonly used in place of the theoretical term *relation*. A table structure is specified as a list of column definitions, each of which specifies a unique column name and the type of values that are permitted for that column.
- A tuple is basically the same thing as a *row*.

Based on these rules, the frame model is mapped to the relational model, which is further implemented by the relational database technology.

5.2 Mapping to XML-based Models

XML is a standard for specifying data on the Web in a structured manner. Strictly speaking, XML is a formalism of encoding information. An XML document is a flat file with a rigid structure to specify concepts. It may follow the concept-property paradigm and be compatible with a relational model.

An XML schema is helpful for defining the valid structure of an XML document.

A concept can be mapped to an element within an XML document. The element may have multiple attributes, each of which is corresponding to a property of the concept. Another way is to map a property to a sub-element of an element within an XML document. The different situations show that the structure of XML can be quite arbitrary in terms of how the concepts are modeled. For example, each of the following two XML fragments shows a valid modeling of the concept *product*:

```
<product>
    <name>Donut</name>
    <price>1.99</price>
</product>
```

```
<product name="donut" price="1.99" />
```

The frame model is compatible with either case. However, creating a frame-based ontological view from such arbitrary models may pose a significant challenge. In our solution, we assume that the XML documents follow a given format:

```
<concept-group>
   <concept-name attribute-list />
   <concept-name attribute-list />
<…>
</concept-group>
<relationship-group>
   <relationship-name>
      <subject-concept-name identifier-attribute-list />
      <object-concept-name identifier-attribute-list />
   </relationship-name>
</relationship-group>
```

It still follows the concept-property structure. A concept in a frame is mapped to an element in an XML document. The properties of a concept are mapped to attributes of an element. A relationship is denoted by a specific element that has multiple sub-elements indicating the subject concept and object concept in the relationship. Note that in an XML document the instance data is embedded. A schema can be extracted from a valid XML document. The facets of a property are not directly specified in an XML document but can be specified in the schema as further attributes of an attribute within the document.

In the current stage we only consider binary relationships but the solution can be extended to support multi-arity relationships.

A wrapper is necessary to convert the XML-based model into an ontological model. The wrapper can be enhanced to support more formats.

5.3 Analysis

Traditionally, evaluation of modeling focuses on the degree to which a model accurately represents the real world from the perspective of intended uses of the model (AIAA, 1998). In our work, we assume that the accuracy of the original modeling is guaranteed. That is, evaluation of the modeling itself is not necessary. Also, we do not worry about the expressiveness of the original modeling paradigm since we assume that it is expressive enough to model the scenarios which the application cares about. Therefore, we only focus on the generated ontological view model and evaluate the degree to which it reflects the original model.

The FOSL language is based on the *frame* modeling paradigm. The completeness of the language lies in three aspects:

- The completeness of the modeling paradigm, i.e., if the modeling paradigm is able to model all mandatory elements, i.e., extrinsic concepts, dependent concepts (properties), characteristics of properties, and relationships.
- The vocabulary and syntax of the language, i.e., if the above elements can be specified by the constructs of the language.
- The transferability with other modeling languages, i.e., if all necessary constructs of another modeling language can be mapped to the constructs of the language.

Generally speaking, an information model is an abstraction and formal representation of a domain of discourse. An information model is developed following a specific modeling paradigm (also referred to as a knowledge representation paradigm). The information implied by a model is relies heavily on how the symbolic system is interpreted. Considering the availability of an interpretation, any formal or informal representation can express some information. In other words, any data structure in a computer system can be a specific representation of a model. Some representations can express specific worlds, but too much information is implied by the simple formalism and a complicated interpretation is required. A good specification language should make the implied information as explicit as possible.

In terms of the elements to model, many of the modeling paradigms model the world around the notion of *concept*. These include first order logic, description logic, production rules, conceptual graph, semantic network, F-logic, entity-relationship model, object-oriented model, and RDF. That is, these paradigms are able to specify individual elements that can be mapped to concepts. Some other paradigms do not have the explicit notion of *concept*. For example, the state-space paradigm represents the structure of a problem in terms of the alternatives available at each possible state of the problem. It uses specific forms to represent the states that involve objects. Explicit interpretation is necessary to explain how the objects and relationships are arranged in the states. Specific applications are required to decide how the transitions can occur between the states.

In a procedural representation, knowledge about the world is contained in procedures—small programs that know how to do specific things, how to proceed in well-specified situations. For instance, in a parser for a natural language understanding system, the knowledge that a *noun phrase* may contain articles, adjectives, and nouns is represented in the program by calls to routines that know how to process articles, nouns, and adjectives. In this paradigm concepts are not stated explicitly and thus are neither typically extractable in a form that humans can easily understand, nor reusable by other programs.

Many expert systems use decision-making rules that can be represented using the **IF…THEN** format, that is

IF <*situation*> **THEN** <*action*>

Other clauses such as **OR** and **ELSE** can also be used with this construct to show alternative situations or different courses of action. Rules in a knowledgebase system (KBS) stand alone as statements of truth or fact and can be used by an inference engine to reach other true conclusions. This representation does not provide a standard way to specify concepts in the *situation* and *action* part.

Propositional logic is one approach for representing knowledge in many expert systems. In this approach, the elementary building blocks, propositions, are *atomic statements* that cannot be further decomposed, e.g., “It is raining”, “Tom is a student”. Logical connectives like “and”, “or”, “not” can be used to build propositional formulas. Similarly, there is no standard way to specify concepts in the propositions.

Among the paradigms that have the notion of concept, first order logic and production rules do not differentiate concepts and instances of concepts. Others can specify concepts and instances separately. For example, in conceptual graphs, each concept has a concept type and referent such as [Person: Tom].

All the paradigms that have the notion of *concept* also support the notion of *relationship* that associates concepts. For example, in first order logic a relationship can be represented as *sell*(Store, Product).

Most of the paradigms do not provide facilities to model further details such as properties of concepts as well as characteristics of properties. Properties and the further characteristics actually refer to relationships with specific meanings. The entity-relationship model, object-oriented model, and frame provide means to model all these aspects. The production rule has the *entity-attribute-value triple* structure, which can be viewed as a form to represent properties of concepts. The entity-relationship and object-oriented paradigm can model characteristics of properties but the capability is not complete. It is completed at the supporting technology level such as the relational database and application written with specific OO languages, but not at the modeling level.

Many of the modeling paradigms also model the behavioral/logical aspects besides the informational aspects. The exceptions are state space, conceptual graph, and semantic network. In the implementations, usually the informational aspects are supported by persistence technologies and the behavior aspects are supported by applications. In the modeling of ontological views the behaviors of concepts are not required.

The degree of structure of a modeling paradigm indicates how different elements are represented separately so each one of them can be differentiated from others and treated individually. The procedural representations embed the model of the world within programs so it is hard to extract the individual elements. Similarly, the rule-based methods and propositional logic do not define internal structures for the sentences. First order logic is more structured in a sense that atomic formulas are interpreted as statements about relationships between objects. Other modeling paradigms are quite structured since they provide separate structures for different types of elements.

Model Implication means the degree to which the model requires interpretation for humanunderstanding. A well structured paradigm is usually explicit in terms of the meaning of the internal constructs, which makes the models easier to understand. An exception is the state space which can be highly structured but how each state represents the world requires lots of interpretation.

Table 1 presents a summary of the features of various modeling paradigms. It shows that frame provides the most complete features for our modeling purpose.

Table 1. Comparison of various modeling paradigms

Features / **Modeling paradigms**	**Has Notion of Concept**	**Differentiate Concepts and Instances**	**Has Notion of Relationship**	**Has Notion of Property**	**Has Notion of Property Characteristics**	**Has Notion of Behavior / Logic**	**Structured**	**Model Implication**	**General-purpose Supporting Technology**
State space	No	No	No	No	No	No	High	High	No (interpreted by applications)
Procedural Representation	No	No	No	No	No	Yes	Low	High	No (interpreted by applications)
Rule-based methods	No	No	No	No	No	Yes	Low	High	No (implemented by specialized systems)
Propositional logic	No	No	No	No	No	Yes	Low	Medium	No (implemented by specialized systems)
First order logic	Yes	No	Yes	No	No	Yes	Medium	Medium	Prolog
Description Logic	Yes	Yes	Yes	No	No	Yes	High	Low	OWL
Production rules	Yes	No	Yes	Yes	No	Yes	High	Low	Prolog
Conceptual Graph	Yes	Yes	Yes	No	No	No	High	Low	No (implemented by specialized systems)
Semantic Network	Yes	Yes	Yes	No	No	No	High	Low	No (implemented by specialized systems)
F-Logic	Yes	Yes	Yes	No	No	Yes	High	Low	No (implemented by specialized systems)
Entity-Relationship	Yes	Yes	Yes	Yes	Yes (not complete)	No	High	Low	Relational database

Object-Oriented	Yes	Yes	Yes	Yes	Yes (not complete)	Yes	High	Low	Object-Oriented languages
Frame	Yes	Yes	Yes	Yes	Yes	No	High	Low	Not required

According to the definition of ontological views, a complete specification language should provide constructs to denote concepts, properties of concepts, characteristics of properties, and relationships to specify the objects to be modeled. We examined two languages that are practically used in specifying information models since our work is based on the existing information systems: relational (implemented by SQL) and XML schema. They are well supported by mature persistence technologies.

Table 2 presents the comparison between FOSL, SQL, and XML schema elements. It shows that FOSL has the complete set of constructs for modeling the expected elements and all the constructs can be mapped to the counterparts within SQL and XML schema.

Table 2. Comparison of FOSL, SQL, and XML

Representation Language / **Language Construct** / **Modeling Object**		**FOSL**	**SQL**	**XML Schema**
World		Ontological_View	database	schema
Concept		Concept	table	element
Property		Property	column	attribute
Relationship		Relationship	foreign key	embedded element (complexType)
Property/Relationship Characteristics		Facet	column attribute	element attribute
Characteristics	Name	Name	column name	*name* attribute
	Identity	(Not necessary)	primary key, unique key	*key* element, *unique* element
	Auto-Increment	Auto_Increment	auto_increment/identity	
	Data type	Data_Type	type	*type* attribute
	Default value	Default_Value	default	*default* attribute
	Fixed value	Fixed_Value		*fixed* attribute
	Optional	Nullable	null/not null	*use* attribute
	Restriction on	(Not necessary)	check	*restriction* element, *minInclusive*

	values			elemnt, *maxInclusive* element
	Restriction on a set of values	(Not necessary)	check	*restriction* element, enumeration element
	Restriction on a series of values	(Not necessary)	check	*restriction* element, *pattern* element
	Restriction on string length	Size	column length	*restriction* elemment, *length* element, *minLength* element, *manLength* elemnt
	Restriction on data types	Decimal_Size	column length	*restriction* element, *fractionDigits* elemnt, *totalDigits* element
	Relationship cardinality	Cardinality	(Implicit by model)	*maxOccur* attribute, *minOccur* attribute

6. Conclusion

Semantic integration, as an important factor for successful information integration, has grown into one of the most active research areas. Our work fits into the evolvement by extending the traditional ontology-driven approaches to an ontological view-driven approach to overcome the grand challenges that were not thoroughly addressed by the traditional approaches. The most significant advancement is the removal of the assumption about the availability of explicit ontologies.

We adopt *frame* as the modeling paradigm of the ontological view. An ontological view can be created from the information model of an information system. In an open environment, the frame-based ontological views create a common level that eliminates the structural and syntactic heterogeneities among the information models. With this commonness only semantic heterogeneities should be considered in the following integration. We propose a frame-based ontological view specification language (FOSL) and use XML to explicitly encode the ontological views. With FOSL we provide a formal way to explicitly and formally represent the concepts in ontological views with rich details. This work establishes a solid foundation for semantic integration in open computing environments.

Our future work will focus on automatic ontological view generation based on richer information models such as richer XML document formats, visual editing of ontological views, and efficient model validation to ensure the consistency of ontological views. Based on analysis of the ontological views, e.g., discovering various semantic relationships between the ontological views, the semantic integration can be achieved.

References

Aikins JS. Prototypical knowledge for expert systems: a retrospective analysis. Artificial Intelligence, Vol. 59, 1993; p.207-211.

American Institute of Aeronautics and Astronautics (AIAA). Guide for the Verification and Validation of Computational Fluid Dynamics Simulations. AIAA-G-077-1998, Reston, VA, 1998.

Arens Y, Knoblock CA, Hsu CN. Query processing in the SIMS information mediator. In: A. Tate (ed), Advanced Planning Technology. Menlo Park: AAAI Press, 1996.

Arroyo S. Ontology and grammar of the SOPHIE choreography conceptual framework – an ontological model for knowledge management. Journal of Universal Computer Science, Vol. 13, No. 9, 2007; p.1157-1183.

Badder F, Sattler U. An overview of tableau algorithms for description logics. Studia Logica, Vol. 69, No. 1, 2001; p. 5-40.

Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Scientific American, 2001(5).

Brachman RJ, Levesque HJ. The tractability of subsumption in frame-based description languages. In: Proceedings of the 4th National Conference on Artificial Intelligence (AAAI-84), 1984; p. 34-37.

Brachman RJ, Levesque HJ. Knowledge representation and reasoning. Morgan Kaufmann publishers, 2004.

Chandrasekaran B, Josephson JR, Benjamins VR. Ontologies: what are they? Why do we need them? IEEE Expert (Intelligent Systems and Their Applications), 1999; 14(1): 20-26.

Codd EF. The Relational Model for Database Management: Version 2. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1990.

Crubzy M, Pincus Z, Musen MA. Mediating knowledge between application components. In: Proceedings of the Semantic Integration Workshop of the Second International Semantic Web Conference (ISWC-03), Sanibel Island, Florida, 2003.

De Bruijn J, Ding Y, Arroyo S, Fensel D. Semantic information integration in the COG project. Technical Report, Digital Enterprise Research Institute (DERI), Universiity of Innsbruck, Austria; 2003.

Genesereth MR, Nilsson NJ. Logical foundation of artificial intelligence. Morgan Kaufmann, Los Altos, California, 1987.

Grau BC, Horrocks I, Motik B, Parsia B, Patel-Schneider P, Sattler U. OWL 2: The next step for OWL. Journal of Web Semantics, 6(4):309-322, November 2008.

Gruber TR. A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2): 199-220, 1993.

Gruber TR. Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, 43(5/6): 907-928, 1995.

Guarino N. Formal ontology and information systems. In: Proceedings of FOIS'98, Trento, Italy, June 6-8, 1998, Amsterdam, IOS Press; p. 3-15.

Hakimpour F, Timpf S. Using ontologies for resolution of semantic heterogeneity in GIS. In: Proceedings 4th AGILE Conference on Geographic Information Science, Brno, Czech Republic, 2001; p. 385-395.

Jasper R, Ushold M. A framework for understanding and classifying ontology applications. In: Proceedings of the 12th Banff Knowledge Acquisition for Knowledge-based Systems Workshop, 1999; p. 16-21.

Jurafsky D, Martin JH. Speech and language processing: an introduction to natural language processing. Computational Linguistics, and Speech Recognition, Second Edition, Prentice Hall, 2008.

Kalfoglou, Y. and Schorlemmer, M. Ontology mapping: the state of the art. The Knowledge Engineering Review, 18(1): 1–31, Cambridge University Press, 2003.

Kiatisevi P, Ampornaramveth V, Ueno H. A frame-based knowledge software tool for developing interactive robots. Artificial Life and Robotics, vol. 10, No. 1, July 2006; p. 18-28.

Marinov M. Using frames for knowledge representation in a CORBA-based distributed environment. Knowledge-Based Systems, vol. 21, issue 5, July 2008; p. 391-397.

Menzel C. An account of abstract possible worlds. Stanford Encyclopedia of Philosophy, http://plato.stanford.edu/entries/actualism/possible-worlds.html, 2008.

Minsky MA. Framework for representing knowledge. Winston PH (ed.), The Psychology of Computer Vision, McGraw-Hill Book Company, New York, 1975; p. 211-277.

Noy FN. What do we need for ontology integration on the semantic web, position statement. In: Proceedings of the Workshop on Semantic Integration, jointed held with the 2nd International Semantic Web Conference, Sanibal Island, Florida, USA, 2003. p. 175-176.

Noy FN. Semantic integration: a survey of ontology-based approaches. SIGMOD Record, Special Issue on Semantic Integration, 2004; 33(4): 65-70.

OKBC <http://www.ai.sri.com/~okbc/>, 1995.

Plantinga A, Davidson M. Essays in the metaphysics of modality. Oxford University Press US, 2003.

Protégé <http://protege.stanford.edu/index.html>, 2010.

Robert RB, Goldstein PI. The FRL primer. Technical Report. Massachusetts Institute of Technology Artificial Intelligence Laboratory, 2002.

Shen W, Norrie DH, Barthes JA. Multi-agent systems for concurrent intelligent design and manufacturing. Taylor & Francis; 2001.

Sheth AP. Changing focus on interoperability in information systems: from system, syntax, structure to semantics. In: Goodchild MF, Egenhofer M, Fegeas R, Kottman CA (eds), Interoperating Geographic Information System, Vol. 47, 1999; p. 5-29.

Sowa JF. Electronic communication in the onto-std mailing list, December 4, 1997.

Stuckenschmidt H, van Harmelen F. Information sharing on the semantic web. Springer-Verlag Berlin Heidelberg; 2005.

Sun H, Fan W, Shen W, Xiao T. Ontology fusion in HLA-based collaborative product development. In: Proceedings of 2010 International Conference on Systems Man and Cybernetics (SMC 2010), 2010; p. 2526-2532.

Tan H, Jakoniene V, Lambrix P, Aberg J, Shahmehri N. Alignment of biological ontologies using life science literature. In: Proceedings of the International Workshop on Knowledge Discovery in Life Science, Singapore, 2006; p. 1-17,.

Uschold M, Gruninger M. Ontologies: principles, methods and applications. Knowledge Engineering Review, 11(2): 93-155, 1996.

Wache H, Vogele T, Visser U, Stuckenschmidt H, Schuster G, Neumann H, Hubner S. Ontology-based integration of information – a survey of existing approaches. In: Proceedings of the IJCAI-01 Workshop on Ontologyies and Information Sharing, Seattle, USA, 4-5 August, 2001; p. 108-117.

Wang H, Noy N, Rector A, Musen M, Redmond T, Rubin D, Tu S, Tudorache T, Drummond N, Horridge M, Seidenberg J. Frames and OWL side by side. In: Proceedings of the 9th International Protégé Conference, July 24-26, 2006, Stanford University, USA.