A frame-based ontological view specification language

Xue, Y.; Ghenniwa, H.; Shen, W.

Publisher's version / Version de l'éditeur:

2010 14th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD 2010) [Proceedings], pp. 228-233, 2010-04-14

A frame-based ontological view specification language

NRCC-52699

Xue, Y.; Ghenniwa, H.; Shen, W

May 2010

A version of this document is published in / Une version de ce document se trouve dans:
2010 14th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD 2010), Shanghai, P.R. China, April 14-16, 2010, pp. 228-233
.

A Frame-based Ontological View Specification Language

Yunjiao Xue [1], Hamada H. Ghenniwa [1], Weiming Shen [1, 2]
[1] *Department of Electrical and Computer Engineering, The University of Western Ontario, London, Ontario, Canada*
[2] *Centre for Computer-assisted Construction Technologies, National Research Council, London, Ontario, Canada*
yxue24@uwo.ca, hghenniwa@eng.uwo.ca, weiming.shen@nrc.gc.ca

Abstract

Semantic integration is crucial for successful collaboration between heterogeneous information systems in an open environment. Traditional ontology-driven approaches rely on the availability of explicit ontologies. However, in many 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. A prototype environment that supports semantic integration, based on ontological views specified with FOSL is introduced.

Keywords: Semantic Integration, Collaboration, Ontological View, Frame, Specification Language.

1. Introduction

In an open, dynamic, and distributed environment, various information systems, such as different collaborative design and manufacturing systems [4], are expected to collaborate to support information exchange, processing, and decision making.

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 supporting infrastructures, syntactic representations of information, schematic designs of information models, and semantics of information, all of which will significantly hinder their collaboration. There have been many proposed solutions for the first three areas of concern [1]. The fourth one, also known as the *semantic integration* problem [2], is attracting growing attention from today's research communities.

Ontology-driven semantic integration is a category of solutions for the semantic integration problem [3]. The traditional solutions are based on available ontologies. However, in many domains there are no pre-established explicit ontologies available. Instead, the "ontologies" are implied in different formats, such as the underlying information representations. In fact, each system reflects a specific view of the domain conceptualization, and in our work such a view is defined as an *ontological view*. As a type of concept model, where a specification language is required, an ontological view will be explicitly and formally specified..

This paper proposes a *frame*-based ontological view specification language for semantic integration in an open environment where information models are available but explicit ontologies are not provided. The rest of the paper is organized as follows: Section 2 reviews some related work; 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 a prototype environment that utilizes the proposed language; Section 6 concludes the paper with some discussion on future work.

2. Related Work

Ontology plays an important role in understanding and dealing with information semantics. An ontology is a formal and explicit specification of a conceptualization [5]. Simply, an ontology specifies the concepts and relationships between the concepts in a domain [6]. Ontology-based semantic integration approaches assume that multiple explicit ontologies are provided in a domain. However, this is not always true in an open computing environment.

Minsky's *frame* theory [7] 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 will represent stereotypical situations [8]. It can represent the world meaningfully and naturally, and is cognitively simple, intuitive, and understandable for domain experts. 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 [9]. Some recent examples of applying frames to knowledge representation can be found in [10, 11].

As defined in the Open Knowledge-Base Connectivity (OKBC) specification [12], *frame* is one of the most widely-used ontology modeling paradigms. It is implemented in the core Protégé, a cutting-edge tool for creating, editing, browsing, and maintaining ontologies [13].

Some researchers view *frame* itself as a modeling language, 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 isn't a single standard frame specification language. [14].

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 [15] and Genesereth and Nilsson [17] 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, a 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. It is said to obtain (exist) or not where the proposition is said to be true or false. A state of affairs will be said to *include* a second state of affairs if it is impossible for the former to obtain (exist) and the latter to fail to obtain. A state of affairs will be said to *preclude* a second state of affairs if it is impossible for them both to obtain. A state of affairs is called *maximal* if, for every other state of affairs, it either includes or precludes that other state of affairs. 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 $\mathbf{C} = \langle D, W, \Re \rangle$, 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 $\mathbf{C} = \langle D, W, \Re \rangle$ is the structure $\mathbf{S}_{w\mathbf{C}} = \langle D, \mathbf{R}_{w\mathbf{C}} \rangle$, 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 $\mathbf{S}_\mathbf{C} = \{\mathbf{S}_{w\mathbf{C}} \mid w \in W\}$ 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) $\mathbf{S} \in \mathbf{S}_\mathbf{C}$; 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 best as possible the set of intended models of **L** according to **K**.

According to the above definition, an 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 isn't merely one unique "ontology" for a conceptualization since it can be viewed in various ways. Instead, different views of the conceptualization may exist. 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 $\mathbf{I_K}(\mathbf{L})$ and another ontological view O' with intended models $\mathbf{I_{K'}}(\mathbf{L'})$, O and O' are integrate-able (denoted by ◊) if and only if $\mathbf{I_K}(\mathbf{L})$ overlaps with $\mathbf{I_{K'}}(\mathbf{L'})$. That is,

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

This can be illustrated by the following Figure 1:

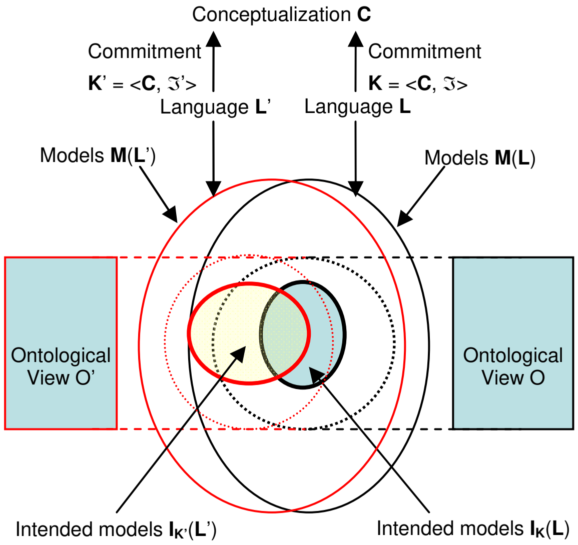

Figure 1. Different ontological views with different languages for one conceptualization which sets of intended models overlap.

3.2 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 doesn't concern the instances of the concept.

Brachman and Levesque [8] introduced a simple formal representation formulism to express the frame's structure as follows:

(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 slot "*: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 [16]: verifiability, unambiguity, canonical form, reasoning capability, and expressiveness, where there is always a trade-off between the good expressiveness and the ease of inference [8].

3.3 Modeling Ontological Views with Frame

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

Different from the knowledge base 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 (even other concepts can be involved to specify its characteristics), not defined by other concepts. 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 $C = \langle P, hasProperty \rangle$, where P is a set of intrinsic concepts and *hasProperty* is a semantic relationship which associates P to C. An intrinsic concept is a concept that is semantically dependent on an extrinsic (contrary to intrinsic) concept. Usually, an intrinsic 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 an intrinsic concept. Therefore, it can be also 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 our work we believe it will be normal 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 intrinsic concepts with a *hasProperty* relationship.

(2) Properties: intrinsic 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 the ontological views.

4. 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 by any semantic integration service within an 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 is focused on the former language.

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.

By applying these rules, an ontological view can be constructed from a corresponding information model. These rules reveal the key requirements for the specification language, including the symbols and syntax indicating concepts, properties, facets, facet values, and relationships.

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) (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 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.

5. Encoding and Implementation

5.1 XML-based Encoding

To explicitly encode ontological views we propose a human readable and machine process-able 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 a 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 are 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.

Following is the schema of the XML document derived from FOSL.

```
<?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>
```

5.2 Implementation

A prototype environment that supports defining ontological views with FOSL and processing the concepts defined in the ontological views has been developed. The environment contains multiple information systems. Each system has an ontological view written with FOSL which is stored as an XML document. A set of Java classes are created to represent different components of the *frame* model, including the class for general Concept, Property, Facet, Relationship, etc. The instances of the classes represent the actual conceptual elements modeled within the system. For example, an instance of the Concept class represents a concept modeled in the system.

The concepts can be exchanged to other systems through Web services. The system can proceed with semantic integration then, for example, by comparing a concept from another system and the concepts within its own concept model to find out which one matches that concept the best. A layer of supporting services provides other fundamental functionality such as data storage and exception handling.

Figure 2 shows the architecture of one system in the environment.

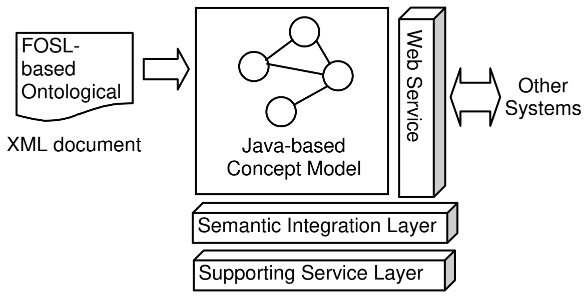

Figure 2. Architecture of an information system in the open environment.

6. Conclusion

Semantic integration, as an important factor for successful information integration, has grown into one of the most active research areas. Our work on semantic integration 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. 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 an open environment.

Our future work will focus on automatic ontological view generation based on regular information models, visual editing of ontological views, efficient model validation to ensure the consistency of ontological views. Based on these efforts the semantic integration service layer can continually be improved.

References

[1] A. P. Sheth, "Changing Focus on Interoperability in Information Systems: from System, Syntax, Structure to Semantics", In: M.F. Goodchild, M. Egenhofer, R.

Fegeas, and C. A. Kottman, (eds), *Interoperating Geographic Information System*, Vol. 47, 1999, pp. 5-29.

[2] F. N. Noy, "Semantic Integration: A Survey of Ontology-based Approaches", *SIGMOD Record, Special Issue on Semantic Integration*, 2004, 33(4), pp. 65-70.

[3] F. Hakimpour and S. Timpf, "Using Ontologies for Resolution of Semantic Heterogeneity in GIS", *Proceedings 4th AGILE Conference on Geographic Information Science*, Brno, Czech Republic, 2001, pp. 385-395.

[4] W. Shen, D. H. Norrie and J. A. Barthes, *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*, Taylor & Francis, 2001.

[5] T. R. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", *KSL-93-04*, Knowledge Systems Laboratory, Standard University, http://ksl-web.stanford.edu/.

[6] M. Crubzy, Z. Pincus and M. A. Musen, "Mediating Knowledge between Application Components", *Proceedings of the Semantic Integration Workshop of the Second International Semantic Web Conference (ISWC-03)*, Sanibel Island, Florida, 2003.

[7] M. A Minsky, "Framework for Representing Knowledge", *The Psychology of Computer Vision*, ed. Patrick Henry Winston, McGraw-Hill Book Company, New York, 1975, pp. 211-277.

[8] R. J. Brachman and H. J. Levesque, *Knowledge Representation and Reasoning*, Morgan Kaufmann publishers, 2004.

[17]

[9] J. S. Aikins, "Prototypical Knowledge for Expert Systems: a Retrospective Analysis", *Artificial Intelligence*, Vol. 59, 1993, Elsevier, pp.207-211.

[10] P. Kiatisevi, V. Ampornaramveth and H. Ueno, "A Frame-based Knowledge Software Tool for Developing Interactive Robots", *Artificial Life and Robotics*, Vol. 10, No. 1, July 2006, pp. 18-28.

[11] M. Marinov, "Using Frames for Knowledge Representation in a CORBA-based Distributed Environment", *Knowledge-Based Systems*, Vol. 21, Issue 5, July 2008, pp. 391-397.

[12] OKBC, http://www.ai.sri.com/okbc/spec.html.

[13] Protégé, http://protege.stanford.edu/index.html.

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

[15] N. Guarino, "Formal Ontology and Information Systems", *Proceedings of FOIS'98*, Trento, Italy, June 6-8, 1998, Amsterdam, IOS Press, pp. 3-15.

[16] D. Jurafsky and J. H. Martin, Speech and Language Processing: An Introduction to Natural Language Procesing, Computational Linguistics, and Speech Recognition, Second Edition, Prentice Hall, 2008.

~~[17]~~ M. R. Genesereth and N. J. Nilsson, Logical Foundation of Artificial Intelligence, Morgan Kaufmann, Los Altos, California, 1987.