Institution Morphisms for Relating OWL and Z

Dorel Lucanu
Faculty of Computer Science
"A.I.Cuza" University
Iaşi, Romania
dlucanu@info.uaic.ro

Yuan Fang Li and Jin Song Dong
School of Computing
National University of Singapore
Singapore
{liyf,dongjs}@comp.nus.edu.sg

Abstract

Checking for properties of Web ontologies is important for the development of reliable Semantic Web systems. Software specification and verification tools can be used to complement the Knowledge Representation tools in reasoning about Semantic Web. The key to this approach is to develop sound transformation techniques from Web ontology to software specifications so that the associated verification tools can be applied to check the transformed specification models. Our previous work has demonstrated a practical approach to translating Web ontologies to Z specifications. However, from a sound engineering point of view, the translation is lacking the theoretical work that can formally relate the respective underlying logical systems of OWL and Z. In this paper, we take the advantage that the logics underlying OWL and Z can be represented as institutions and we show that the institution comorphism provides a formal semantic foundation for the transformation from OWL to Z.

1. Introduction

The development of Semantic Web (SW) takes a layered approach where ontology languages such as RDF [10], DAML+OIL [14] and OWL [11] provide semantic markups for describing resources on the Web and they form the foundation for development of upper-layer technologies in SW. Therefore, it is utterly important to ensure the correctness of Web ontologies.

Various SW reasoning engines have been developed to facilitate reasoning about Web ontologies. Fully automated are these tools, they have certain disadvantages. Firstly, as they are automated tools, reasoning tasks involving complex undecidable ontologies cannot be carried out very effectively. Secondly, although inconsistencies can be detected by these tools, the source of the inconsistency cannot be located accurately. This makes debugging fairly difficult.

In our previous works [2, 3], we have proposed to use software engineering tools with complementing power to RACER [7] in a combined approach to checking DAML+OIL ontologies. As the first step, we constructed the DAML+OIL semantics in formal languages Z [15] and Alloy. By transforming ontologies to models in these languages, we are then able to verify properties inexpressible in DAML+OIL (and OWL) and to find the source of inconsistencies detected by RACER.

Our previous works focused on the practical aspects of the approach. The formal proof of the soundness of the Z semantics of DAML+OIL language was not shown. As ontology languages such as DAML+OIL and formal methods such as Z are based on different logic systems, such proof requires a high-level device that represents and reasons about software models without assumption of the underlying logical systems.

The notion of institutions [6] was introduced to formalize the concept of "logical system". Institutions provide a means of reasoning about software specifications regardless of the logical system. Institutions are suitable for proving the soundness of our approach as the underlying logical systems of DAML+OIL (OWL) and Z can be represented as institutions and by applying institution comorphisms [5], we can prove the soundness of the Z semantics for OWL.

Based on our previous works [3], we have constructed the Z semantics for OWL DL[1]. The semantics is defined for OWL DL as it is more expressive than OWL Lite and still decidable, unlike OWL Full. Some changes have been made from the semantics for DAML+OIL to reflect more faithfully the model-theoretic semantics of OWL [12].

In this paper, we show the formal proof of the soundness of the Z semantics for OWL DL[2] using institution comorphisms [5]. The rest of the paper is organized as follows. In Section 2, we give a brief account of institutions and institution morphisms. In Sections 3 and 4, we present the OWL and Z institutions, respectively. The proof of soundness of

[1] The full specification can be found at `http://www.comp.nus.edu.sg/~liyf/OWL2Z.tex`

[2] OWL DL is very similar to DAML+OIL, only a small number of language constructs are changed.

the transformation is given in Section 5. Finally, Section 6 concludes the paper.

2. Institutions & Institution Morphisms

Institutions were introduced by J. Goguen and R. Burstall [6] to formalize the notion of logical system and to provide a basis for reasoning about software specification independently of the underlying logical system chosen. The basic components of a logical system are models and sentences, related by the satisfaction relation. The compatibility between models and sentences is provided by signatures, which formalizes the notion of vocabulary used to build sentences. Modeling the signatures of a logical system as a category, we get the possibility to translate sentences and models across signature morphisms. The consistency between the satisfaction relation and this translation is given by the *satisfaction condition* which intuitively means that *the truth is invariant under the change of notation*.

Formally, an *institution* is a quadruple $\Im = (\mathsf{Sign}, \mathsf{sen}, \mathsf{Mod}, \models)$ where Sign is a category whose objects are called *signatures*, sen is a functor $\mathsf{sen} : \mathsf{Sign} \to \mathsf{Set}$ which associates with each signature Σ a set whose elements are called Σ-*sentences*, $\mathsf{Mod} : \mathsf{Sign}^{op} \to \mathsf{Cat}$ is a functor which associates with each signature Σ a category whose objects are called Σ-*models*, and $\models$ is a family of binary relations $\models_\Sigma$ between Σ-models and Σ-sentences, called *satisfaction relations*, such that for each morphism $\phi : \Sigma \to \Sigma'$, the *satisfaction condition*

$$\mathsf{Mod}(\phi)(M') \models_\Sigma e \Leftrightarrow M' \models_{\Sigma'} \phi(e)$$

holds for each model $M' \in \mathsf{Mod}(\Sigma')$ and each sentence $e \in \mathsf{sen}(\Sigma)$.

The functor sen abstracts the way the sentences are constructed from signatures (vocabularies) and extends the signature morphisms to translations between sentences. The functor Mod is defined over the opposite category Sign^{op} because a translation between two signatures $\phi : \Sigma \to \Sigma'$ defines a forgetful functor $\mathsf{Mod}(\phi^{op}) : \mathsf{Mod}(\Sigma') \to \mathsf{Mod}(\Sigma)$ such that for each Σ'-model M', $\mathsf{Mod}(\phi^{op})(M')$ is M' viewed as a Σ-model. The satisfaction condition may be read as "M' satisfies the ϕ-translation of e iff M' viewed as a Σ-model satisfies e", i.e., the meaning of e is not changed by the translation ϕ. We often use $\mathsf{Sign}(\Im)$, $\mathsf{sen}(\Im)$, $\mathsf{Mod}(\Im)$, $\models_\Im$ to denote the components of the institution $\Im$.

The migration from one logical system to another is captured by institution morphism or institution comorphism. An institution morphism expresses a relationship from a "richer" logical system to a "poorer" one, and an institution comorphism expresses how a "poorer" logical system is encoded in a "richer" one. In this paper we use only the later one. An *institution comorphism* $(\Phi, \alpha, \beta) : \Im \to \Im'$ consists of a functor $\Phi : \mathsf{Sign} \to \mathsf{Sign}'$, a natural transformation $\alpha : \mathsf{sen} \Rightarrow \Phi\,;\mathsf{sen}'$, and a natural transformation $\beta : \Phi^{\mathrm{op}}\,;\mathsf{Mod}' \Rightarrow \mathsf{Mod}$ such that the following satisfaction condition holds:

$$M' \models'_{\Phi(\Sigma)} \alpha_\Sigma(e) \text{ iff } \beta_\Sigma(M') \models_\Sigma e$$

for any $\Phi(\Sigma)$-model M' from $\Im'$ and Σ-sentence e from $\Im$. The functor Φ translates the signatures in $\Im$ to signatures in $\Im'$.

The natural transformation α consists of a function $\alpha_\Sigma : \mathsf{sen}(\Sigma) \to \mathsf{sen}'(\Phi^{\mathrm{op}}(\Sigma))$, translating Σ-sentences to $\Phi(\Sigma)$-sentences, for each signature Σ in $\Im$. The natural transformation β consists of a functor $\beta_\Sigma : \mathsf{Mod}'(\Phi^{\mathrm{op}}(\Sigma)) \to \mathsf{Mod}(\Sigma)$, associating a Σ-model to each $\Phi^{\mathrm{op}}(\Sigma)$-model, for each signature Σ in $\Im$. If $\Im'$ is a theoroidal institution, i.e., an institution whose signatures are theories of other institution, then (Φ, α, β) is a *theoroidal comorphism*.

We recommend [5, 13] for systematic investigations of the relationships between institutions.

3. The OWL Institution $\mathfrak{O}$

We recall from [9] the definition of the institution formalizing the logic underlying the Web Ontology Language OWL DL. We note that in OWL DL we have the mutual disjointness between classes, properties, and individuals.

We suppose that all the OWL specifications share the same data types. Therefore we consider given a set $\mathbb{D}$ of *data type names*, a set $\mathcal{V}$ of *data values*, and a function $[\![\_]\!]$ which associates a subset $[\![D]\!] \subseteq \mathcal{V}$ with each data type name D. The set of *data expressions* is defined as follows:

$$\mathcal{D} ::= D \mid \{v_1, \ldots, v_n\}$$

where D ranges over data type names and v_i ranges over data values. We extend the definition of $[\![\_]\!]$ over data expressions by setting $[\![\{v_1, \ldots, v_n\}]\!] = \{v_1, \ldots, v_n\}$. In OWL definition [12] a data type D is characterized by a lexical space, $L(D)$, a value space, $V(D)$, and a mapping $L2V(D) : L(D) \to V(D)$. We represent a data type in a more abstract way by forgetting the lexical space. $V(D)$ is denoted here by $[\![D]\!]$. For instance, $(\mathbb{D}, [\![\_]\!])$ might be the set of the XML data types and/or the set of the OWL built-in types. We separate the data world from the world over which we define ontologies. A first reason for this separation is that the specification of the data types is quite different from that of ontologies. Another reason is that we get more flexibility in relating web ontologies with various formalisms. For instance, we may use directly the built-in implementations of the data types in these formalisms and focus only on the translation of the taxonomy and its sentences.

An *OWL signature* consists of a quadruple $\mathcal{O} = (\mathbb{C}, \mathbb{R}, \mathbb{U}, \mathbb{I})$, where $\mathbb{C}$ is the set of the *concept (class) names*, $\mathbb{R}$ is the set of the *individual-valued property names*, $\mathbb{U}$ is the set of the *data-valued property names*, and $\mathbb{I}$ is the set of *individual names*. We denote by $\mathcal{N}(\mathcal{O})$ the set $\mathbb{C} \cup \mathbb{R} \cup \mathbb{U} \cup \mathbb{I}$. An *OWL signature morphism* ϕ :

$(\mathbb{C}, \mathbb{R}, \mathbb{U}, \mathbb{I}) \rightarrow (\mathbb{C}', \mathbb{R}', \mathbb{U}', \mathbb{I}')$ consists of a quadruple of functions $\phi = (\phi_{co}, \phi_{op}, \phi_{dp}, \phi_{in})$ where $\phi_{co} : \mathbb{C} \rightarrow \mathbb{C}'$, $\phi_{op} : \mathbb{R} \rightarrow \mathbb{R}'$, $\phi_{dp} : \mathbb{U} \rightarrow \mathbb{U}'$, and $\phi_{in} : \mathbb{I} \rightarrow \mathbb{I}'$. We denote by $\mathsf{Sign}(\mathfrak{O})$ the category of the OWL signatures. Given an OWL signature $\mathcal{O} = (\mathbb{C}, \mathbb{R}, \mathbb{U}, \mathbb{I})$, an *$\mathcal{O}$-structure (model)* is a tuple $A = (\Delta_A, [\![\_]\!]_A, Res_A, res_A)$ consisting of a set of *resources* Res_A, a subset $\Delta_A \subseteq Res_A$ called *domain*, a function $res_A : \mathcal{N}(\mathcal{O}) \cup \mathbb{D} \rightarrow Res_A$ associating a resource to each name in $\mathcal{O}$ or $\mathbb{D}$, and an interpretation function $[\![\_]\!]_A : \mathbb{C} \cup \mathbb{R} \cup \mathbb{U} \rightarrow \mathcal{P}(Res) \cup P(Res) \times P(Res)$ such that the following conditions hold:

$\mathcal{V} \subseteq Res_A$,
$\Delta_A \cap \mathcal{V} = \emptyset$,
$[\![C]\!]_A \subseteq \Delta_A$ for each $C \in \mathbb{C}$,
$[\![R]\!]_A \subseteq \Delta_A \times \Delta_A$ for each $R \in \mathbb{R}$,
$[\![U]\!]_A \subseteq \Delta_A \times \mathcal{V}$ for each $U \in \mathbb{U}$,
$res_A(o) \in \Delta_A$ for each $o \in \mathbb{I}$.

We often write $[\![o]\!]_A$ for $res_A(o)$ to have a uniform notation.

The definition above corresponds to that of abstract interpretation of an OWL vocabulary given in [12]. In particular we have $\Delta_A = O$, $[\![\_]\!]_A\ |_{\mathbb{C}} = EC$, $[\![\_]\!]_A\ |_{\mathbb{R}\cup\mathbb{U}} = ER$, and $res_A = S$. Here $[\![\_]\!]_A\ |_X$ denotes the restriction of the function $[\![\_]\!]_A$ to the subset X. The meaning of the inclusion $\mathcal{V} \subseteq Res_A$ should be read as "there is a total relation $\rho \subseteq \mathcal{V} \times Res$" and the condition $\Delta_A \cap \mathcal{V} = \emptyset$ should be read as $\Delta_A \cap \operatorname{ran} \rho = \emptyset$. It is of worth to have a look over the semantics of the empty OWL signature $\emptyset = (\emptyset, \emptyset, \emptyset, \emptyset)$. A $\emptyset$-structure is of the form $A = (\emptyset, [\![\_]\!]_A, Res_A, res_A)$, where $[\![\_]\!]_A$ is the unique function $\emptyset \rightarrow Res_A$ and $res_A : \mathbb{D} \rightarrow Res_A$; i.e., A consists only of the data types.

Given two $\mathcal{O}$-structures $A = (\Delta_A, [\![\_]\!]_A, Res_A, res_A)$ and $A' = (\Delta_{A'}, [\![\_]\!]_{A'}, Res_{A'}, res_{A'})$, an *$\mathcal{O}$-homomorphism* $h : A \rightarrow A'$ is a function $h : Res_A \rightarrow Res_{A'}$ such that:

1. $h(\Delta_A) = \Delta_{A'}$;
2. $res_{A'} = res_A;\ h$;
3. for each $C \in \mathbb{C}$ and $x \in \Delta_A$, if $x \in [\![C]\!]_A$ then $h(x) \in [\![C]\!]_{A'}$;
4. for each $R \in \mathbb{R}$ and $x, y \in \Delta_A$, if $(x, y) \in [\![R]\!]_A$ then $(h(x), h(y)) \in [\![R]\!]_{A'}$;
5. for each $U \in \mathbb{U}$, $x \in \Delta_A$, and $v \in \mathcal{V}$, if $(x, v) \in [\![U]\!]_A$ then $(h(x), v) \in [\![U]\!]_{A'}$.

We denote by $\mathsf{Mod}(\mathfrak{O})(\mathcal{O})$ the category of the $\mathcal{O}$-models. If $\phi : \mathcal{O} \rightarrow \mathcal{O}'$ is an OWL signature morphism and $A' = (\Delta_{A'}, [\![\_]\!]_{A'}, Res_{A'}, res_{A'})$ an $\mathcal{O}'$-structure, then the ϕ-reduct $A'\!\upharpoonright_\phi$ is the $\mathcal{O}$-structure $A = (\Delta_A, [\![\_]\!]_A, Res_A, res_A)$, where $Res_A = Res_{A'}$, $\Delta_A = \Delta_{A'}$, $res_A(N) = res_{A'}(\phi(N))$ for each name $N \in \mathcal{N}(\mathcal{O})$, and the interpretation function $[\![\_]\!]_A$ is defined as follows:

$[\![C]\!]_A = [\![\phi_{co}(C)]\!]_{A'}$ for each $C \in \mathbb{C}$;
$[\![R]\!]_A = [\![\phi_{op}(R)]\!]_{A'}$ for each $R \in \mathbb{R}$;
$[\![U]\!]_A = [\![\phi_{dp}(U)]\!]_{A'}$ for each $U \in \mathbb{U}$.

We may consider now the functor $\mathsf{Mod}(\mathfrak{O}) : \mathsf{Sign}(\mathfrak{O})^{op} \rightarrow \mathsf{Cat}$ mapping each OWL signature $\mathcal{O}$ to the category of its models $\mathsf{Mod}(\mathfrak{O})(\mathcal{O})$ and each OWL signature morphism $\phi : \mathcal{O} \rightarrow \mathcal{O}'$ to the forgetful functor $\mathsf{Mod}(\mathfrak{O})(\phi^{op}) : \mathsf{Mod}(\mathfrak{O})(\mathcal{O}') \rightarrow \mathsf{Mod}(\mathfrak{O})(\mathcal{O})$ by $\mathsf{Mod}(\mathfrak{O})(\phi^{op})(A') = A'\!\upharpoonright_\phi$ and $\mathsf{Mod}(\mathfrak{O})(\phi^{op})(h') = h'\!\upharpoonright_\phi$.

The set of the *$\mathcal{O}$-expressions* is defined by:

$$\begin{aligned}
\mathcal{C} ::= &\ \bot \mid \top \mid C \mid \mathcal{C} \sqcap \mathcal{C} \mid \mathcal{C} \sqcup \mathcal{C} \mid \neg\, \mathcal{C} \\
&\mid \forall\, \mathcal{R}.\mathcal{C} \mid \exists\, \mathcal{R}.\mathcal{C} \mid \leqslant n\, \mathcal{R} \mid \geqslant n\, \mathcal{R} \mid R : o \\
&\mid \forall\, U.\mathcal{D} \mid \exists\, U.\mathcal{D} \mid \leqslant n\, U \mid \geqslant n\, U \mid U : v \\
&\mid \{o_1, \ldots, o_n\} \\
\mathcal{R} ::= &R \mid \mathsf{Inv}(R)
\end{aligned}$$

where n ranges over natural numbers, C ranges over concepts names, R ranges over individual-valued properties names, U over data-valued properties, v over $\mathcal{V}$, and o_i over individuals names.

The set of *$\mathcal{O}$-sentences* is defined by:

$$\begin{aligned}
F ::= &\ \mathcal{C} \sqsubseteq \mathcal{C} \mid \mathcal{C} \equiv \mathcal{C} \mid \mathsf{Disjoint}(\mathcal{C}, \ldots, \mathcal{C}) \\
&\mid \mathsf{Tr}(R) \mid \mathcal{R} \sqsubseteq \mathcal{R} \mid \mathcal{R} \equiv \mathcal{R} \\
&\mid U \sqsubseteq U \mid U \equiv U \\
&\mid o : \mathcal{C} \mid (o, o') : \mathcal{R} \mid (o, v) : U \mid o \equiv o' \mid o \not\equiv o'
\end{aligned}$$

where o and o' range over individuals names, and v over data values. We denote by $\mathsf{sen}(\mathfrak{O})(\mathcal{O})$ the set of the $\mathcal{O}$-sentences. If $\phi : \mathcal{O} \rightarrow \mathcal{O}'$ is an OWL signature morphism, then $\mathsf{sen}(\mathfrak{O})(\phi) : \mathsf{sen}(\mathfrak{O})(\mathcal{O}) \rightarrow \mathsf{sen}(\mathfrak{O})(\mathcal{O}')$ is the function translating the $\mathcal{O}$-sentences in $\mathcal{O}'$-sentences in the standard way; for instance,
$\mathsf{sen}(\mathfrak{O})(\phi)(\forall\, R.C \sqcap C') = \forall\, \phi_{op}(R).\phi_{co}(C) \sqcap \phi_{co}(C')$.

The semantics of the $\mathcal{O}$-expressions is given by:

$[\![\bot]\!]_A = \emptyset$,
$[\![\top]\!]_A = \Delta_A$,
$[\![\mathsf{Inv}(R)]\!]_A = \{(y, x) \mid (x, y) \in [\![R]\!]_A\}$,
$[\![\mathcal{C} \sqcap \mathcal{C}']\!]_A = [\![\mathcal{C}]\!]_A \cap [\![\mathcal{C}']\!]_A$,
$[\![\neg\, \mathcal{C}]\!]_A = \Delta_A \setminus [\![\mathcal{C}]\!]_A$,
$[\![\forall\, \mathcal{R}.\mathcal{C}]\!]_A = \{x \mid (\forall y)(x, y) \in [\![\mathcal{R}]\!]_A \Rightarrow y \in [\![\mathcal{C}]\!]_A\}$,
$[\![\leqslant n\, \mathcal{R}]\!]_A = \{x \mid \#(\{y \mid (x, y) \in [\![\mathcal{R}]\!]_A\}) \leqslant n\}$,
$[\![R : o]\!]_A = \{x \mid (x, [\![o]\!]_A) \in [\![R]\!]_A\}$,
$[\![\forall\, U.\mathcal{D}]\!]_A = \{x \mid (\forall v)(x, v) \in [\![U]\!]_A \Rightarrow v \in [\![\mathcal{D}]\!]\}$,
$[\![\leqslant n\, U]\!]_A = \{x \mid \#(\{v \mid (x, v) \in [\![U]\!]_A\}) \leqslant n\}$,
...

The satisfaction relation between $\mathcal{O}$-structures and $\mathcal{O}$-sentences is defined as follows:

$A \models_{\mathfrak{O},\mathcal{O}} \mathcal{C} \sqsubseteq \mathcal{C}'$ iff $[\![\mathcal{C}]\!]_A \subseteq [\![\mathcal{C}']\!]_A$,
$A \models_{\mathfrak{O},\mathcal{O}} \mathcal{C} \equiv \mathcal{C}'$ iff $[\![\mathcal{C}]\!]_A = [\![\mathcal{C}']\!]_A$,
...

The OWL institution $\mathfrak{O}$ is given by $\mathfrak{O} = (\mathsf{Sign}(\mathfrak{O}), \mathsf{sen}(\mathfrak{O}), \mathsf{Mod}(\mathfrak{O}), \models_{\mathfrak{O}})$.

The definition of the institution $\mathfrak{O}$ follows mainly the lines described in [12] and [8]. The use of the institution theory offers several significant advantages: ability to work with structured ontologies, use of constraints to distinguish between OWL DL and OWL Full ontologies, and a solid foundation for tools extending and linking OWL languages with other formalisms similar to those presented in [2, 4]. In the next section we will show that the semantics of OWL ontologies in Z (based on that of DAML+OIL presented in [3]) defines in fact an institution comorphism. This proves that that encoding is correct.

4. The Institution $\mathfrak{Z}$

Z [15] is a formal specification language based on first-order logic and ZF set theory. It is well suited for modeling system data and states. Z has a rich set of language constructs including given type, abbreviation type, axiomatic definition, schema definitions, etc.

We briefly recall from [1] the institution $\mathfrak{Z}$, denoted by $\mathcal{S}$ in [1], formalizing the logic underlying the specification language Z.

A *Z signature* $\mathcal{Z}$ is a triple (G, Op, τ) where G is the set of the *given-sets names*, Op is a set of the *identifiers*, and τ is a function mapping the names in Op into types $\mathcal{T}(G)$, where $\mathcal{T}(G)$ is inductively defined by:

1. $G \subseteq \mathcal{T}(G)$,
2. $T_1 \times \cdots \times T_n \in \mathcal{T}(G)$ for $T_i \in \mathcal{T}(G)$, $i = 1, \ldots, n$,
3. $\mathcal{P}(T) \in \mathcal{T}(G)$ for $T \in \mathcal{T}(G)$,
4. $\langle x_1 : T_1, \ldots, x_n : T_n \rangle \in \mathcal{T}(G)$ for $T_i \in \mathcal{T}(G)$ and x_i is a variable name, $i = 1, .., n$, such that $i \neq j \Rightarrow x_i \neq x_j$.

A *Z signature morphism* $\phi : (G, Op, \tau) \to (G', Op', \tau')$ is a pair of functions $\phi_{gs} : G \to G'$ and $\phi_{op} : Op \to Op'$ such that $\tau ; \mathcal{T}(\phi_{gs}) = \phi_{op} ; \tau'$. $\mathcal{T}(\phi_{gs})$ is the standard extension of ϕ_{gs} to $\mathcal{T}(\phi_{gs}) : \mathcal{T}(G) \to \mathcal{T}(G')$. We denote by $\mathsf{Sign}(\mathfrak{Z})$ the category of Z signatures. Given a Z signature $\mathcal{Z} = (G, Op, \tau)$, a *$\mathcal{Z}$-structure (model)* is a pair (A_G, A_{Op}) where A_G is a functor from G, viewed as a discrete category, to Set, and A_{Op} is a set $\{(o, v) \mid o \in Op\}$ where $v \in \overline{A}_G(\tau(o))$. The functor $\overline{A}_G$ is the standard extension of A_G to $\overline{A}_G : \mathcal{T}(G) \to \mathsf{Set}$. A *$\mathcal{Z}$-homomorphism* $h : (A_G, A_{Op}) \to (B_G, B_{Op})$ is a natural transformation $h : A_G \Rightarrow B_G$ given by $\overline{h}_{\tau(o)}(v) = v'$, where $(o, v) \in A_{Op}$ and $(o, v') \in B_{Op}$; again, $\overline{h}$ is the usual extension of h to $\overline{h} : \overline{A}_G \Rightarrow \overline{B}_G$. We denote by $\mathsf{Mod}(\mathfrak{Z})(\mathcal{Z})$ the category of $\mathcal{Z}$-structures. Given a Z signature morphism $\phi : \mathcal{Z} \to \mathcal{Z}'$ and a $\mathcal{Z}'$-structure $A' = (A'_{G'}, A'_{O'})$, the ϕ-reduct $A'\upharpoonright_\phi$ is the $\mathcal{Z}$-structure $A = (A_G, A_{Op})$ given by $A_G = \phi_{gs} ; A'_{G'}$ and $A_{Op} = \{(o, v) \mid (\phi_{op}(o), v) \in A'_{Op'}, o \in Op\}$.
Given a Z signature $\mathcal{Z}$, the sets of *$\mathcal{Z}$-expressions E*, *$\mathcal{Z}$-schema-expressions S*, and (part) of *$\mathcal{Z}$-formulas P* are defined by:

$$
\begin{aligned}
E ::= {} & id \mid x \mid (E, \ldots, E) \mid E.i \mid \langle x_1 \mapsto E, \ldots, x_n \mapsto E \rangle \\
& \mid E.x \mid E(E) \mid \{E, \ldots, E\} \mid \{S \bullet E\} \mid \mathcal{P}(E) \\
& \mid E \times \cdots \times E \mid S
\end{aligned}
$$

$$
\begin{aligned}
S ::= {} & x_1 : E;\ \ldots;\ x_n : E \mid (S \mid P) \mid \neg\, S \mid S \vee S \mid S \wedge S \\
& \mid S \Rightarrow S \mid \forall S.S \mid \exists S.S \mid S \setminus [x_1, \ldots, x_n] \\
& \mid S[x_1/y_1, \ldots, x_n/y_n] \mid S\, Decor \mid E \\
P ::= {} & \texttt{true} \mid \texttt{false} \mid E \in E \mid E = E \mid \neg\, P \mid P \vee P \mid P \wedge P \\
& \mid P \Rightarrow P \mid \forall S.P \mid \exists S.P
\end{aligned}
$$

The *$\mathcal{Z}$-sentences* are $\mathcal{Z}$-formulas well defined, i.e., all the operators and quantifiers are given over expressions having the types compatible with their definition.

Example 1 *The following simple Z specification:*

$[Class, Resource]$

ClassesAsResources

$instances : Class \to \mathbb{P}\, Resource$
$res : Class \rightarrowtail Resource$

$\forall\, c, c' : Class;\ r : Resource;\ pr : \mathbb{P}\, Resource \bullet$
$\quad c \mapsto r \in res \Rightarrow \neg(r \in pr \wedge c' \mapsto pr \in instances$

is described in the terms of the institution $\mathfrak{Z}$ as $\mathit{CR} = ((G, Op, \tau), P)$ where $G = \{Class, Resource\}$, $Op = \{instances, res\}$, $\tau(instances) = \mathcal{P}(Class \times \mathcal{P}(Resource))$, $\tau(res) = \mathcal{P}(Class \times Resource)$, and P includes the formulas expressing the functionality of the relation instances, the functionality and the injectivity of the relation res, together with the invariant of the state schema ClassesAsResources. It is easy to see, e.g., that $c \mapsto r \in res$ is a CR-expression and $c, c' : Class;\ r : Resource;\ pr : \mathbb{P}\, Resource$ is a CR-schema-expression.

The interpretation of the $\mathcal{Z}$-formulas by $\mathcal{Z}$-structures and the *satisfaction relation* between $\mathcal{Z}$-structures and $\mathcal{Z}$-sentences are defined as expected; e.g., $A \models c \mapsto r \in res$ iff for all variable bindings $\{(c, v_c), (r, v_r)\}$, $(v_c, v_r) \in w$ and $(res, w) \in A_{Op}$.

The institution $\mathfrak{Z}$ is given by $\mathfrak{Z} = (\mathsf{Sign}(\mathfrak{Z}), \mathsf{sen}(\mathfrak{Z}), \mathsf{Mod}(\mathfrak{Z}), \models_{\mathfrak{Z}})$, where $\mathsf{Sign}(\mathfrak{Z})$ is the category of Z signatures, the functor $\mathsf{sen}(\mathfrak{Z})$ maps each Z signature $\mathcal{Z}$ to its set of $\mathcal{Z}$-sentences, the functor $\mathsf{Mod}(\mathfrak{Z})$ maps each Z signature $\mathcal{Z}$ to the category of $\mathcal{Z}$-structures, and $\models_{\mathfrak{Z},\mathcal{Z}}$ is defined as above.

5. Encoding $\mathfrak{O}$ in $\mathfrak{Z}$

In our previous works [2, 3], we developed the semantics for DAML+OIL language in formal language Z as an extension of the standard library. This semantic library was later on revised for the new ontology language OWL, incorporating changes incurred in OWL from DAML+OIL. In this section, we will demonstrate, through institutions comorphisms, that the Z encoding of OWL is indeed sound.

The main idea is to associate a Z specification $\Phi(\mathcal{O}, F)$ with each OWL specification $(\mathcal{O}, F)$ such that an $(\mathcal{O}, F)$-model can be extracted from each $\Phi(\mathcal{O}, F)$-model. The construction of $\Phi(\mathcal{O}, F)$ is given in two steps: we first associate a Z specification $\Phi(\mathcal{O})$ with each OWL signature $\mathcal{O}$ and then we add to it the sentences F translated via a natural transformation.

Since $\Phi(\mathcal{O}, F)$ can be seen as a Z semantics of $(\mathcal{O}, F)$, it includes a distinct subspecification $(\mathcal{Z}^\emptyset, P^\emptyset)$ defining the main OWL concepts and the operations over sets. More precisely, we consider $(\mathcal{Z}^\emptyset, P^\emptyset)$ as being the vertex of the colimit having as base the standard library, the specification of the data types, together with the following specification:

given sets:

$\quad\texttt{Resource};$

identifiers:

- ✓ corresponding to OWL signatures:
  $\texttt{Class}, \texttt{Property}, \texttt{Thing}, \texttt{Nothing}, \texttt{ObjectProperty}, \texttt{DatatypeProperty}, \texttt{Individual}$
- ✓ giving Z semantics to OWL signatures:
  $\texttt{instances}, \texttt{subVal}$
- ✓ corresponding to OWL class axioms:
  $\texttt{disjointClasses}, \texttt{equivalentClasses}, \texttt{subClassOf}$
- ✓ corresponding to OWL descriptions and restrictions:
  $\texttt{unionOf}, \texttt{intersectionOf}, \texttt{complementOf}, \texttt{oneOf}, \ldots$
- ✓ corresponding to OWL property axioms:
  $\texttt{domain}, \texttt{range}, \texttt{functional}, \texttt{inverseOf}, \texttt{symmetric}, \ldots$

$\tau^\emptyset$ for the new identifiers:

- ✓ corresponding to OWL signatures:
  $\tau^\emptyset(\texttt{Class}) = \tau^\emptyset(\texttt{Property}) = \tau^\emptyset(\texttt{ObjectProperty}) = \tau^\emptyset(\texttt{DatatypeProperty}) = \mathcal{P}(\texttt{Resource})$
  $\tau^\emptyset(\texttt{Thing}) = \tau^\emptyset(\texttt{Nothing}) = \texttt{Resource}$
- ✓ giving Z semantics to OWL signatures:
  $\tau^\emptyset(\texttt{instances}) = \mathcal{P}(\texttt{Resource} \times \mathcal{P}(\texttt{Resource}))$
  $\tau^\emptyset(\texttt{subVal}) = \mathcal{P}(\texttt{Resource} \times \mathcal{P}(\texttt{Resource} \times \texttt{Resource}))$
- ✓ corresponding to OWL class axioms:
  $\tau^\emptyset(\texttt{disjointClasses}) = \tau^\emptyset(\texttt{Class} \times \texttt{Class}) = \mathcal{P}(\texttt{Resource} \times \texttt{Resource})$
  $\tau^\emptyset(\texttt{subClassOf}) = \mathcal{P}(\texttt{Resource} \times \texttt{Resource})$
  $\ldots$
- ✓ corresponding to OWL descriptions, restrictions
  $\tau^\emptyset(\texttt{unionOf}) = \tau^\emptyset((\texttt{Class} \times \texttt{Class}) \times \texttt{Class})$
  $\quad = \mathcal{P}((\texttt{Resource} \times \texttt{Resource}) \times \texttt{Resource})$
  $\ldots$
- ✓ corresponding to OWL property axioms:
  $\tau^\emptyset(\texttt{domain}) = \mathcal{P}(\texttt{Resource} \times \texttt{Resource})$
  $\tau^\emptyset(\texttt{range}) = \mathcal{P}(\texttt{Resource} \times \texttt{Resource})$
  $\ldots$

sentences :

- ✓ corresponding to OWL signatures:
  $\texttt{Class} \cap \texttt{Property} = \emptyset$
  $\texttt{Class} \cap \texttt{Individual} = \emptyset$
  $\texttt{Property} \cap \texttt{Individual} = \emptyset$
  $\ldots$
- ✓ giving Z semantics to OWL signatures:
  $\texttt{instances}(\texttt{Thing}) = \texttt{Individual}$
  $\texttt{instances}(\texttt{Nothing}) = \emptyset$
  $\forall c : \texttt{Class} \bullet \texttt{instances}(c) \subseteq \texttt{Individual}$
  $\ldots$
- ✓ corresponding to OWL class axioms:
  $\forall c_1, c_2 : \texttt{Class} \bullet c_1 \mapsto c_2 \in \texttt{disjointClasses} \Leftrightarrow \texttt{instances}(c_1) \cap \texttt{instances}(c_2) = \emptyset$
  $\forall c_1, c_2 : \texttt{Class} \bullet c_1 \mapsto c_2 \in \texttt{subClassOf} \Leftrightarrow \texttt{instances}(c_1) \subseteq \texttt{instances}(c_2)$
  $\ldots$
- ✓ corresponding to OWL descriptions, restrictions:
  $\forall c, c_1, c_2 : \texttt{Class} \bullet (c_1, c_2) \mapsto c \in \texttt{unionOf} \Leftrightarrow \texttt{instances}(c) = \texttt{instances}(c_1) \cup \texttt{instances}(c_2)$
  $\ldots$
- ✓ corresponding to OWL property axioms:
  $\forall p_1, p_2 : \texttt{Property} \bullet p_1 \mapsto p_2 \in \texttt{subPropertyOf} \Leftrightarrow \texttt{subVal}(p_1) \subseteq \texttt{subVal}(p_2)$
  $\forall p : \texttt{Property};\ c : \texttt{Class} \bullet p \mapsto c \in \texttt{domain} \Leftrightarrow \operatorname{dom} \texttt{subVal}(p) \subseteq \texttt{instances}(c)$
  $\ldots$

We define $\Phi^\diamond : \mathsf{Sign}(\mathfrak{O}) \to \mathsf{Sign}(\mathfrak{Z})$ as follows. Let $\mathcal{O} = (\mathbb{C}, \mathbb{R}, \mathbb{U}, \mathbb{I})$ be an OWL signature. Then $\Phi^\diamond(\mathcal{O}) = (G, Op, \tau)$ is defined as follows:

$G = G^\emptyset;$
$Op = Op^\emptyset \cup \mathbb{C} \cup \mathbb{R} \cup \mathbb{U} \cup \mathbb{I};$
$\tau(C) = \texttt{Resource}$ for each $C \in \mathbb{C}$,
$\tau(R) = \texttt{Resource}$ for each $R \in \mathbb{R}$,
$\tau(U) = \texttt{Resource}$ for each $U \in \mathbb{U}$,
$\tau(o) = \texttt{Resource}$ for each $o \in \mathbb{I}$.

If $\varphi : \mathcal{O} \to \mathcal{O}'$ is an OWL signature morphism and $\Phi^\diamond(\mathcal{O}) = (G^\emptyset, Op, \tau)$ and $\Phi^\diamond(\mathcal{O}') = (G^\emptyset, Op', \tau')$, then $\Phi^\diamond(\varphi) : \Phi(\mathcal{O}) \to \Phi(\mathcal{O}')$ is the Z signature morphism $(\mathrm{id} : G^\emptyset \to G^\emptyset, \Phi^\diamond(\varphi)_{op} : Op \to Op')$ such that $\Phi^\diamond(\varphi)_{Op}$ is the identity over the subset $Op^\emptyset$ and $\Phi^\diamond(\varphi)_{op}(N) = \varphi(N)$ for each name N in $\mathcal{O}$. It is easy to check that $\tau ; \mathcal{T}(\mathrm{id}) = \Phi^\diamond(\varphi)_{op} ; \tau'$.

We extend $\Phi^\diamond$ to $\Phi : \mathsf{Sign}(\mathfrak{O}) \to \mathsf{Th}(\mathfrak{Z})$ by defining $\Phi(\mathcal{O}) = (\Phi^\diamond(\mathcal{O}), P)$, where P is $P^\emptyset$ together with the following sentences:

$\{C \in \texttt{Class}) \mid C \in \mathbb{C}\} \cup$
$\{R \in \texttt{ObjectProperty} \mid R \in \mathbb{R}\} \cup$
$\{U \in \texttt{DatatypeProperty} \mid U \in \mathbb{U}\} \cup$
$\{o \in \texttt{Individual} \mid o \in \mathbb{I}\}.$

If $\mathcal{O}$ is an OWL signature, then

$$\alpha_{\mathcal{O}} : \mathsf{sen}(\mathfrak{O})(\mathcal{O}) \to \mathsf{sen}(\mathfrak{Z})(\Phi(\mathcal{O}))$$

is defined by:

$\alpha_{\mathcal{O}}(\bot) = \texttt{Nothing}, \alpha_{\mathcal{O}}(\top) = \texttt{Thing},$
$\alpha_{\mathcal{O}}(C_1 \sqcap C_2) = \texttt{intersectionOf}(\alpha_{\mathcal{O}}(C_1), \alpha_{\mathcal{O}}(C_2)),$
$\ldots$
$\alpha_{\mathcal{O}}(\forall R.C) = \texttt{allValuesFrom}(\alpha_{\mathcal{O}}(R), \alpha_{\mathcal{O}}(C)),$
$\ldots$
$\alpha_{\mathcal{O}}(\leqslant n R) = \texttt{maxCardinality}(\alpha_{\mathcal{O}}(R), n), \ldots$
$\alpha_{\mathcal{O}}(C_1 \sqsubseteq C_2) = \alpha_{\mathcal{O}}(C_1) \mapsto \alpha_{\mathcal{O}}(C_2) \in \texttt{subClassOf},$
$\ldots$
$\alpha_{\mathcal{O}}(E) = \{\alpha_{\mathcal{O}}(e) \mid e \in E\}.$

Lemma 1 $\alpha = \{\alpha_{\mathcal{O}} \mid \mathcal{O} \in \mathsf{Sign}(\mathfrak{O})\}$ *is a natural transformation* $\alpha : \mathsf{sen}(\mathfrak{O}) \Rightarrow \Phi^{\diamond}; \mathsf{sen}(\mathfrak{Z})$[3].

If $\mathcal{O} = (\mathbb{C}, \mathbb{R}, \mathbb{U}, \mathbb{I})$ is an OWL signature and $A' = (A'_G, A'_{Op})$ a $\Phi^{\diamond}(\mathcal{O})$-model, then $\beta_{\mathcal{O}}(A')$ is the $\mathcal{O}$-model $A = (\Delta_A, [\![\_]\!]_A, Res_A, res_A)$ defined as follows:

$Res_A = A'_G(\texttt{Resource})$,
$res_A(N) = v$ where $(N, v) \in A'_{Op}$ for each name $N \in \mathcal{O}$,
$\Delta_A = v$ where $(\texttt{Thing}, v) \in A'_{Op}$,
if $C \in \mathbb{C}$, then $[\![C]\!]_A = v_C$ where $(\texttt{instances}, v) \in A'_{Op}$ and $(C, v_C) \in v$,
if $R \in \mathbb{R}$, then $[\![R]\!]_A = v_R$ where $(\texttt{subVal}, v) \in A'_{Op}$ and $(R, v_R) \in v$,
if $U \in \mathbb{U}$, then $[\![U]\!]_A = v_U$ where $(\texttt{subDVal}, v) \in A'_{Op}$ and $(U, v_U) \in v$.

We extend $\beta_{\mathcal{O}}$ to a functor $\beta_{\mathcal{O}} : \mathsf{Mod}'(\Phi^{\diamond}(\mathcal{O})) \to \mathsf{Mod}(\mathcal{O})$ as follows: if $h : A' \to B'$ is a $\Phi^{\diamond}(\mathcal{O})$-homomorphism, then $\beta_{\mathcal{O}}(h)$ is the $\mathcal{O}$-homomorphism $\beta_{\mathcal{O}}(h) : \beta_{\mathcal{O}}(A') \to \beta_{\mathcal{O}}(B')$ given by $\beta_{\mathcal{O}}(h) = h_{\texttt{Resource}}$.

Lemma 2 $\beta = \{\beta_{\mathcal{O}} \mid \mathcal{O} \in \mathsf{Sign}(\mathfrak{O})\}$ *is a natural transformation* $\beta : \Phi^{\diamond op}; \mathsf{Mod}(\mathfrak{Z}) \Rightarrow \mathsf{Mod}(\mathfrak{O})$.

Theorem 1 $(\Phi, \alpha, \beta) : \mathfrak{O} \to \mathfrak{Z}$ *is a simple theoroidal comorphism.*

6. Conclusion

The complementary power of Semantic Web and software engineering tools have been shown to verify ontology-related properties more efficiently and effectively. The overall correctness of the combined approach largely depends on the soundness of the transformation from OWL to Z, two languages of different underlying logical systems.

In this paper, we demonstrated the soundness of the above transformation through the use of institution morphisms. This allows us to use Z reasoners for proving properties of OWL ontologies. If e is a property of the OWL ontology $(\mathcal{O}, F)$ and we prove that the Z-encoding of $(\mathcal{O}, F)$ satisfies the translation of e, $\alpha_{\mathcal{O}}(e)$, then $(\mathcal{O}, F)$ satisfies e by the satisfaction condition from the definition of the comorphism.

The method we used can be applied to any translation of OWL ontologies into institution-based formalism and many of the formalisms we know can be formalized as institutions.

[3]The proofs of this and following lemma/theorem can be found in [9].

Acknowledgement

The second author would like to thank Singapore Millennium Foundation (`http://www.smf-scholar.org/`) for the financial support.

References

[1] H. Baumeister. Relating abstract datatypes and Z-schemata. In D. Bert and C. Choppy, editors, *Recent Trends in Algebraic Development Techniques - Selected Papers*, volume 1827 of *Lect. Notes in Comput. Sci.*, pages 366–382, Bonas, France, 2000. Springer-Verlag.

[2] J. S. Dong, C. H. Lee, Y. F. Li, and H. Wang. A Combined Approach to Checking Web Ontologies. In *Proceedings of 13th World Wide Web Conference (WWW'04)*, pages 714–722, New York, USA, May 2004.

[3] J. S. Dong, C. H. Lee, Y. F. Li, and H. Wang. Verifying DAML+OIL and beyond in Z/EVES. In *Proceedings of 26th International Conference on Software Engineering (ICSE'04)*, pages 201–210, Edinburgh, Scotland, May 2004.

[4] J. S. Dong, Y. F. Li, and H. Wang. TCOZ Approach to Semantic Web Services Design. In *Proceedings of 13th World Wide Web Conference (WWW'04)*, pages 442–443, New York, USA, May 2004.

[5] J. Goguen and G. Roşu. Institution morphisms. Formal Aspects of Computing, 2002.

[6] J. A. Goguen and R. M. Burstall. Introducing institutions. In *Proc. Logics of Programming Workshop*, number 164 in Lect. Notes in Comput. Sci., pages 221–256. Springer-Verlag, 1984.

[7] V. Haarslev and R. Möller. *RACER User's Guide and Reference Manual: Version 1.7.6*, Dec. 2002.

[8] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. of Web Semantics*, 1(1):7–26, 2003.

[9] D. Lucanu, Y. F. Li, and J. S. Dong. Web Ontology Verification and Analysis in the Z Framework. Technical Report TR 05-01, University "Alexandru Ioan Cuza" of Iaşi, Romania, Jan. 2005. `http://thor.info.uaic.ro/~tr/tr05-01.ps`.

[10] F. Manola and E. M. (editors). RDF Primer. `http://www.w3.org/TR/rdf-primer/`, Feb. 2004.

[11] D. L. McGuinness and F. van Harmelen (editors). OWL Web Ontology Language Overview. `http://www.w3.org/TR/2003/PR-owl-features-20031215/`, Dec. 2003.

[12] P. F. Patel-Schneider and I. Horrocks (editors). OWL: Direct Model-Theoretic Semantics. `http://www.w3.org/TR/owl-semantics/direct.html`.

[13] Andrzej Tarlecki. Moving between logical systems. In *COMPASS/ADT*, pages 478–502, 1995.

[14] F. van Harmelen, P. F. Patel-Schneider, and I. H. (editors). Reference description of the DAML+OIL ontology markup language. March, 2001.

[15] J. Woodcock and J. Davies. *Using Z: Specification, Refinement, and Proof.* Prentice-Hall International, 1996.