A Translation Scheme for Domain Ontologies Based on Model Ontologies of KBS

Houria Mihoubi, Ana Simonet and Michel Simonet
Laboratoire TIMC-IMAG,
Faculté de Medecine de Grenoble
38706 La Tronche Cedex - France
e-mail : Houria.Mihoubi@imag.fr, (Ana, Michel).Simonet@imag.fr

Abstract

To ensure the re-use of domain ontologies the usual approach relies on a pivot language, e.g., Kif, into and from which the domain ontology is translated. This requires several distinct translators for the different target Knowledge Based System (KBS). We propose a solution which uses a unique translator whatever the target KBS, Description Logic, objects or frames. It relies on a Domain Ontology Description Language which allows writing domain ontologies in a declarative manner and a Model Ontology Description Language, a meta-language which allows a declarative description of any model based on frames, objects, relations, and Description Logic. Both languages are based on Kif binary and unary relations. Starting from these relations, we have identified a certain number of concepts which enable a user to perform explicitly a semantic enrichment of binary and unary relations. We introduce the concept of meta-relation which allows the expression of knowledge on the relations themselves. The set of meta-relation definitions describing the KBS primitives constitutes its model ontology. A unique translator associated to the meta-language generates a schema which describes a domain ontology only using representative terms of the target KBS.

1 Introduction

A domain ontology (DO) is a specification of the conceptual knowledge of an application domain. Domain Ontologies cannot be used unless translated into an implemented KBS. The need to re-use DOs by different KBS arose from the necessity to exploit them through different KBS, each one being specialized in a category of reasoning services. For example, a KBS based on a description logic can use a DO to perform concept classification. Another KBS will be able to use the same DO in the scope of a planification project. To ensure the re-use of DOs, the usual approach relies on a pivot language into and from which the DO is translated, which requires several distinct translators. For example, with the pivot language Kif, used by the Ontolingua system [3], to translate a DO to the systems Loom, Epikit and Algernon, one translator for each KBS is required. BRS [9] is an other pivot language proposed for DO re-use by object-based systems and in particular the RIO system [1]. BRS requires a translator for each system.

In this paper we show how DOs can be re-used using only one translator whatever the target KBS, in particular a Description Logic system. The approach we propose has two requirements:

A *Domain Ontology Description Language* which allows writing DOs in a declarative manner, free from implementation aspects of a specific KBS. So, it can be considered as a specification language for Description Logic systems.

A *Model Ontology Description Language*, a meta-language which allows a declarative description of any model based on frames, objects, or Description Logic. Each description is a Model Ontology.

Both languages are based on a subset of Kif [2] limited to relations with arity ≤ 2. Starting from these relations, we have identified a certain number of concepts which enable a user to perform explicitly a semantic enrichment of binary and unary relation.

The meta-language which we propose is defined as an extension of this Kif subset by the concept of meta-relation which expresses the knowledge on relations themselves. The set of meta-relation definitions specifying the model concepts which is implemented by a target system is a Model Ontology. To translate a DO into a specific KBS, one has to describe in this meta-language its model ontology. A translator associated to the meta-language generates a schema which describes a DO using only representative terms described in the Model Ontology.

This paper is organized as follows. First we give a brief description of Kif, which is central to our approach. In section 3 we present the concepts of the Domain Ontology Description Language. We then present the Model Ontology Description Language by giving successively the notion of model ontology (section 4.1), the meta-model concepts (section 4.2) and the syntax of the meta-language (section 4.3). The translation process will be described in section 5. To illustrate our translation approach we present a application to a MO of a system based on Description logics. We conclude by giving an evaluation of our work.

2 General overview of Kif

Kif (Knowledge Interchange Format) is considered as an interlanguage and plays a similar role to data exchange standards, but at the knowledge level. Given its ability to express conceptual knowledge, several projects such as SHADE [8] and Ontolingua use its notation. The formalization of knowledge in KIF requires a conceptualization of the world in terms of objects, functions, and relations. A Kif knowledge base is a finite set of sentences, definitions and rules, based on predicate logic extended to include function terms and equality.
A Kif sentence is constructed on the basis of three syntactic categories: variables, constants and operators. Three categories of constants are taken into consideration: object, relation and function constants.
Kif definitions provide a powerful tool for interpreting meanings attributed to objects of the conceptualization. Three definition operators, defobject, deffunction, and defrelation allow the complete or partial definition of new constants. A definition links a constant to a set of sentences thus limiting its possible interpretations. Such definitions are termed axiomatic.

3 Domain Ontology Description Language

In order that the specification of the conceptual knowledge of an application domain be translatable into any KBS, including Description Logic systems, via a unique translator, the language for DO description must be independent from any particular system while providing a common base for all KBS. We have identified unary and binary relations as being this common base. They constitute a pivot language which can be adapted to any representation based on DL, objects or frames. The KIF language has well defined syntax and semantics of relations. Based on predicate logic, its expressive power is now well known. However, since it is more expressive than most KBS, it cannot therefore be fully translated into every KBS. Such limitations are also present in Ontolingua where the translation is complete only for a set of some common idioms (Kif sentences) that are supported by most of the target systems [4]. In fact, we think that only a subset of Kif limited to unary and binary relations is necessary to describe conceptual knowledge of an application domain. For this reason the Domain Ontology Description Language is based on a subset of Kif limited to relations with arity ≤ 2.

3.1 Conceptualization of an application domain

Conceptualization of an application domain is not a simple task [5], [6]. In our approach unary relations represent concepts of the application domain. They correspond to sets of entities. Given two unary relations R1 and R2 (sets of entities), a binary relation with domain R1 and codomain R2 is an identified set of couples of entities of R1 and R2 respectively. Starting from these unary and binary relations we have identified a certain number of concepts which allow a user to perform their semantic enrichment explicitly.

3.2 Binary Relations

We distinguish the following categories of binary relations:
Attribute Binary Relations: they correspond to mandatory attributes of R1. For example, the relationship *Has-Name* between the sets of entities *Person* and *Name* (set of names) is an attribute of *Person*. All the entities of *Person* have the property of having a name, whether its value is known or not at a given moment. This would be translated as exist *Has-Name* in a DL.

Specialization Binary Relations: they correspond to specialization in abstract data types and mean that all the elements of R1 belong to R2. All R1 entities have all R2 attributes. An example of such a specialization relation is : *Man* specialize *Person*.

3.3 Unary Relations

Unary Relations can be considered according to two axes, classification and specialization. We consider four categories of unary relations.
Primitive Unary Relations: the concept description is partial, i.e., expressed as necessary conditions. A consequence is that an entity needs to be explicitly assigned to the relation in order to belong to it.
Non-primitive Unary Relations: the concept description is complete, i.e., expressed as necessary and sufficient conditions. An entity can be automatically classified relative to the necessary and sufficient conditions.
Specialized Unary Relations: the unary relation S1 specializes at least one other unary relation R2.
Non-specialized Unary Relations: the relation inherits no other relation.

3.4 Domain Ontology

A Domain Ontology consists of the specification of the set of Primitive Specialized, Primitive Non-specialized, Non-primitive Specialized, and Non-primitive Non-Specialized unary relations, along with the Attribute and Specialization binary relations.
The Domain Ontology Description Langage enables the description of a DO as a set of definitions of the above categories of unary and binary relations. Each definition associates a set of Kif sentences with the name of the relation.

Example of a Domain Ontology

```
(DefPrimUnaRel Person(?X)
    :Attribute
        (and (Has-name ?X ?N)
             (Name ?N)))
```

(*DefUnaRel* Male (?X)
  :Attribute
    (Has-Sex ?X M))

(*DefUnaRel* Man (?X)
  :from (and (Person ?X) (Male ?X)))

A binary relation is specified by the keyword ***DefRelation***, its domain (keyword ***:Dom***), its codomain (Keyword ***:Range***), and other constraints like restrictions on the codomain and cardinality constraints (keyword ***:constraints***).

(***DefRelation*** Husband (?X ?Y)
  :Dom (Man ?X)
  :Range (Person ?Y)
  :constraints (and (Not (Man ?Y))
    (= ***Cardinality*** 1)))

4 Model Ontology Description Language

4.1 The Model Ontology notion

We term Model Ontology (MO) the specification relating to the meaning of terms relative to a particular model. For every KBS targeted by the use or re-use of a DO, the MO consists of the specification of each representation primitive used by the KBS. The MO of a Description Logic system would consist in the specification of the primitives: Define-Concept, Primitive-Concept, the Subsomption relationship, etc.

4.2 The meta-model

The knowledge representation systems which our approach aims at are systems based on objects, frames, relations or description logic. A common element to these systems is that they are based on unary and binary relations. Therefore the descriptive language of the MO is founded on a meta-model (Fig. 1) whose basic concepts are: unary meta-relation for describing knowledge on unary or binary relations of a DO and binary meta-relation for specifying knowledge concerning the link between pairs of unary or binary relations. The objective of the meta-model is dual:

1 For a given model i, it must represent in terms of meta-relations, all the representation primitives of the KBS which implements i .

2 Each unary or binary relation described in the DO, used or re-used by a KBS, becomes an instance of a meta-relation described in the MO of the model i.

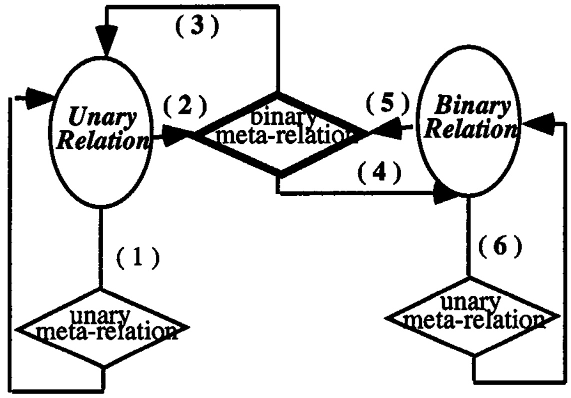

Fig. 1 : The meta-model

Fig.1 illustrates the meta-model in diagram form. ***UnaryRelation*** and ***BinaryRelation*** are two predefined meta-relations (represented by an oval). They represent sets of unary and binary relations. They enable any user (hereafter called MO designer) who wishes to describe a MO, to designate a binary or unary relation. The meta-relations which the MO designer can define are represented by a diamond drawn with normal strokes for unary meta-relations and bold strokes for binary meta-relations. It thus follows that a unary meta-relation could describe knowledge about unary (arrow 1) or binary (arrow 6) relations. A binary meta-relation will describe a specific link between two unary relations (arrows 2 and 3), a unary and a binary relation (arrows 2 and 4), a binary and a unary relation (arrows 5 and 3), or two binary relations (arrows 5 and 4).

4.3 The meta-language

The Model Ontology Description Language is a meta-language used for the description of **Model Ontologies**. It is defined from a subset of Kif restricted to the unary and binary relations but extended by the concept of meta-relation. It should be noted that at this description level the MO-designer, perhaps the same who describes a DO does not describe the content of a concept but the concept itself. The description of each concept is made in a declarative and explicit manner using a definition of a meta-relation which links a set of meta-language sentences to the name of the meta-relation. This definition represents a generic description of the meta-relation. The extension of a unary meta-relation is a set of unary relations. The extension of a binary meta-relation is a set containing couples of relations. The set of unary and binary meta-relation definitions describing the concepts of a representation model is a **Model Ontology**. The meta-language proposes a certain number of predefined meta-relations which all MO-designers can use in their description, among which: ***UnaryRelation***, which designates any unary relation, ***PrimitiveRelation***, which designates any unary relation partially defined (with the ***DefPrimUnaRel*** operator), ***DefinedRelation***, used to designate a completely specified unary relation (by a ***DefUnaRel*** operator), etc.
Using the meta-language, we can describe a part of a

Model Ontology of a description logic system as follows:

(***DefMetaRelation*** PrimitiveConcept (?X)
(and (***UnaryRelation*** ?X)
(***PrimitiveRelation*** ?X)))

(***DefMetaRelation*** DefineConcept (?X)
(and (***UnaryRelation*** ?X)
(***DefinedRelation*** ?X)))

(***DefMetaRelation*** is-Subsumed-by (?X ?Y)
(and (or (PrimitiveConcept ?X)
(DefineConcept ?X))
(or (PrimitiveConcept ?Y)
(DefineConcept ?Y))
(***SubRelation*** ?X ?Y)))

5 The translation process

The meta-level is the description level in which all the unary and binary relations in the DO become instances of meta-relations. The mapping between the meta-model and a model is done by a unique translator (Fig. 2).

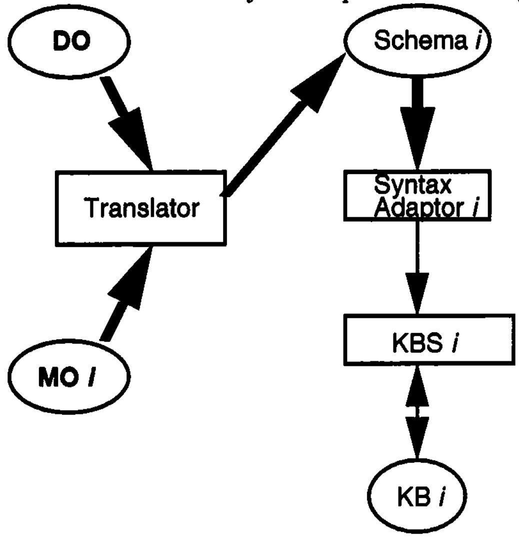

Fig. 2 : The translation process

Fig. 2 shows that whatever the target KBS, only one translator is used. To do this the meta-language translator uses the ontology of a model *i* (MO *i*), which is a model implemented by KBS *i*. The resulting translation is Schema *i*, which is a schema describing the DO solely in terms of representation primitives of KBS *i*. The translation consists in the instanciation of each meta-relation defined in the MO *i*, with the relations which verify its definition. Schema *i* is an intermediate form and can be transformed into accurate KBS *i* syntax by a relatively simple program. In Fig. 2 we have represented this program with the caption 'syntax adaptor'. Mapping is carried out in several steps.

Step 1
The translator pre-processes the DO. Pre-processing consists of :
1 Distinguishing between unary and binary relations.
2 Determining relations specified by means of necessary conditions (***DefPrimUnaRel*** operator).
3 Determining relations which are specified by means of a necessary and sufficient condition (***DefUnaRel*** operator).
4 For each binary relation, determining:
the unary relation upon which it is defined;
its range, restrictions and constraints on its range;
5 For each unary relation determining the super-relation (the relation in which it is included). If there is no super-relation, giving the special M-Kif relation, **root**.

Step 2
The translator instanciates the predefined unary and binary meta-relations. It then disposes of an initial set of meta-relations which we call *Base1*.

Step 3
For each meta-relation MR *i* defined in MO *i*:
1 Execute MR *i* on *Base1*;
2 Instanciate MR *i*;
3 Add MR *i* (the extension of MR *i*) to *Base1*;
4 Go to 1.
Executing MR*i* on *Base1* consists in searching through *Base1* for the relations which verify the meta-language sentences specified in the definition of MR*i*.

Final step
Eliminate from *Base1* all the meta-relations which are not predefined. The result obtained is Schema*i*, the descriptive schema of the application domain in the primitives of the target system.

6 Application to a MO

We consider the DO example described in section 3.3 and a part of the MO decribed in section 4.2. After the second step of mapping, *Base1* contains:
(UnaryRelation Person);
(UnaryRelation Name)
(UnaryRelation Male)
(UnaryRelation Sex)
(UnaryRelation Man);
(BinaryRelation Has-Name);
(BinaryRelation Has-Sex);
(BinaryRelation Hasband);
(PrimitiveRelation Person);
(DefinedRelation Male);
(DefinedRelation Man);
(SubRelation person Root);
(SubRelation Name Root);
(SubRelation Sex Root);
(SubRelation Man Person);
(SubRelation Man Male);
(Domain Has-Name Person);
(Domain Has-Sex Male);
(Domain Husband Man);
(Range Has-Name Name);

(Range Has-Sex Sex);
(Range Hasband Person);

After the third step of the mapping *Base1* is increased by:
(PrimitiveConcept Person);
(PrimitiveConcept Name);
(PrimitiveConcept Sex);
(DefinedConcept Male);
(DefinedConcept Man);
(Is-Subsumed Man Person);
(Is-Subsumed Man Male);
(Role Has-Name);
(Role Has-Sex);
(Role Husband);

Base1 without predefined meta-relations constitutes the schema generated by the translator. To acquire this schema with an accurate KBS syntax, a syntactic adaptor, limited solely to the role of primitive correspondence, is necessary. In the Loom system [7], the syntax adaptor will therefore generate:

```
(DefineConcept Person
    :is (:and :Primitive
            (:all Has-Name Name)))

(DefineConcept Male
    :is (:and :Defined
            (:all Has-Sex Sex)
            (:filled-by Sex M)))

(DefineConcept Man
    :is (:and :Defined
            Person
            Male))
```

7 Conclusion

In this paper we proposed a declarative approach for domain ontology re-use by different KBS, in particular description logic systems. This approach relies on a language for describing domain ontologies and a meta-language for describing model ontologies, i.e., representative terms of target KBS. In order to homogenize the description of DOs and Model Ontologies, both are defined as a layer above Kif binary and unary relations. Starting from these relations, we have identified a certain number of concepts which enable a user to perform explicitly their semantic enrichment. We introduce the concept of meta-relation which allows the expression of knowledge on the relations themselves. The set of meta-relation definitions describing the KBS primitives constitutes its model ontology. This approach requires only one translator no matter what KBS is used, provided that the latter is described in its model ontology. We believe that the task of writing a model ontology for a given KBS is simpler than writing a specific translator, and above all it is not situated at the same level : defining a model ontology is a specification task (hence more declarative) whereas writing a translator is a programming task. The specific part of the translation task is limited to a syntax adaptor which transforms an abstract form of a KBS into its external syntax.
Our approach is still in the definition phase and therefore has not been implemented yet. Evaluating its pros and cons would require a full scale study. To be truly complete, i.e., to allow faithful in-detail description of any target system, it should offer a greater choice of predefined primitives. Along these lines, a possible match-up with the Frame Ontology would certainly contribute towards enhancing the meta-language. The maturity of the Ontolingua system leads us to believe that the content of the Frame Ontology (axiomatization of relations, slot constraints, inverse relations, composition relations, etc.) would partly fulfil the meta-language's requirements concerning predefined meta-relations.

References

[1] Deloule, F., Roche, C., Ontologies & Knowledge Representation, In IEEE International Conference on Systems, Man and Cybernetics. Vancouver, British Columbia Canada, October, 1995.

[2] Genesereth, M., R. and Fikes, R., E. (1992) Knowldge Interchange Format Version 3.0. Reference Manuel. Logic Group, Report Logic-92-1.

[3] Gruber,T., R., Ontolingua: A Mechanism to Support Portable Ontologies, Stanford University, Knowledge Systems Laboratory, Technical Report KSL-91-66. Revision.

[4] Gruber, T., R. , A Translation Approch to Portable Ontology Specifications, In R. Mizoguchi (editor), Proceedings of the second Japanese Knowledge Acquisition for Knowledge-based Systems Workshop, Kobe, Japan, 1992.

[5] Guarino, N., Carrara, M., and Giaretta, P., An Ontology of Meta-Level Categories, LADSEB-CNR Int.Rep.6/93, Substantially revised version January 16, 1994.

[6] Guarino, N., The Ontological Level, Invited paper presented at the 16th Wittgenstein Symposium, Kirchberg, Austria, August 1994.

[7] MacGregor, R., M., A Deductive Pattern Matcher, In proceedings AAAI-87, St. Paul, Minnesota (1987) 403-408

[8] McGuire, J., G., Kuokka, D., R., Weber, J., C., Tenenbaum, J., M., Gruber, T., R., Olsen, G., R., SHADE: Technology for Knowledge-based Collaborative Engineering, Journal of Concurrent Engineering: Applications and Research, (CERA), 1(2), September 1993.

[9] Roche, C., Deloule, F., Policastro R., BRS: a generic binary relation system for knowledge-based systems, In Proceedings of Modelling Languages for Knowledge-based Systems. Amsterdam, 1995.