An use case for DAML+OIL: a knowledge base in a clinical domain ⋆

M. Taboada[1], M. Argüello[1], D. Martínez[2], J. Des[3], and J. Mira[4]

[1] Dpto. de Electrónica e Computación. Universidade de Santiago de Compostela.
15706 Santiago de Compostela. Spain.
Author to contact: María Taboada. E-mail: elchus@usc.es
Telephone number: 34-981-563100
Fax number: 34-981-599412
[2] Dpto. de Física Aplicada. Universidade de Santiago de Compostela. Spain
[3] Hospital Comarcal de Monforte. Monforte. Spain
[4] Dpto. de Inteligencia Artificial. UNED. Madrid. Spain

Abstract. This paper describes how we have developed a knowledge base for a heuristic application in a clinical domain. Firstly, the knowledge base has been modelled using CommonKADS. Later, the knowledge base has been represented in the language DAML+OIL. We will illustrate in some depth how objects, classes and relationships of a medical domain (modelled following an object-oriented approach) can be specified in the Web language DAML+OIL.

Keywords: knowledge modelling, knowledge representation, DAML+OIL.

1 Introduction

Currently, the development of knowledge-based applications is viewed as a modelling activity oriented to obtain structured knowledge models. It is advisable that the modelling initial step includes a very detailed ontological analysis [3], as the latter clarifies the structure of the domain knowledge and so, the semantics of its representation. In addition, with the aim of sharing and reusing ontologies, the Artificial Intelligence community is very interested in representing them on the Semantic Web [7]. Current standard Web languages, such as XML [1] or RDF [2], have been developed in order to express Web information in such a way as to be easily processable by machines. Other languages, such as DAML+OIL [6, 12], have arisen in order to provide a greater inter-operability on the semantic level. In this way, DAML+OIL extends the RDF Schema basic primitives for providing a more expressive ontology modelling language and some simple terms for creating inferences.

In this paper we describe how we have represented a knowledge base for a heuristic application using DAML+OIL. This knowledge base has been built by reusing pre-existing knowledge theories, and it has been developed in two phases:

⋆ This work has been funded by the Secretaría Xeral de Investigación e Desenvolvemento da Xunta de Galicia, through the research project PGIDT01-PXI20608PR

– Firstly, the underlying ontology has been modelled using the domain knowledge schema specification of CommonKADS [10], in order to represent graphically the objects, class and relations of the domain knowledge.
– Later, the knowledge base has been represented in DAML+OIL .

We will revise several types of common relations in an object-oriented approach and we will show how these types of relations can be represented in DAML+OIL. We will see that these basic relations are represented in DAML+OIL unambiguously, so there is a direct connection from a domain model to its representation in DAML+OIL. But, the direct connection from DAML+OIL to the domain model is lost and it cannot be easily reconstructed. However, for each particular domain, we will see that it is very easy to extend the set of DAML+OIL modelling primitives with the needed modelling primitives. These extensions facilitate the re-engineering of the domain model from just the representation in DAML+OIL.

The rest of the paper is organized as follows. In Section 2 we describe the clinical domain briefly. In Section 3 we summarize the modelling of our clinical knowledge base. Section 4 illustrates in some depth how objects, classes and the most common relationships in an object-oriented approach can be specified in DAML+OIL. In Section 5, we will shown how we have specified derivation rules "If...then..." in DAML+OIL. Finally, Section 6 summarizes and concludes the paper.

2 Summarizing the clinical domain

Our medical knowledge base has been modelled for a heuristic application in the domain of the *Red Eye* (RE) diagnosis [11]. The latter is a common activity carried out in primary care units. Recommendations about carrying out the diagnosis of the red eye include to take a careful history and to make a focused ophthalmology examination. Following these recommendations, the knowledge base must include, at least, knowledge about the ophthalmology anamnesis, eye examination, related clinical procedures, eye pathologies and relations among all of this knowledge.

3 Modelling the ophthalmology knowledge base

The domain knowledge base has been developed by reusing some theories in the core library described in [5], and later, by extending these theories with some descriptions taken from INTERNIST-I [9] and with more specific concepts to our clinical domain. We have also followed the International Classification of Diseases *ICD-9-CM* [13], for representing pathologies. Currently, there is a 10th Revision, but the 9th Revision is still followed in Spain. In this way, we have obtained a standard representation vocabulary, which is being revised by following UMLS (Unified Medical Language System) [8].

The library of medical ontologies of [5] specifies a set of medical conceptualizations taken from both medical literature and implemented systems, including general categories of medical knowledge, such as the following theory levels:

– Generic Patient, which models the medical activities for each patient.
– Test and Therapies, representing actions undertaken by medical agents.

– Diseases, modelling each disease as a clinical process whose evolution is described through finding and clinical abstraction values over time.
– Findings, Clinical-state-abstractions, Drugs, etc.

We have divided our knowledge base in several parts, where each part corresponds to one of these theory levels. In this paper, we will focus in the levels 'Generic-Patient' and 'Tests', in order to describe the most important aspects used in the DAML+OIL specification.

4 Specifying objects and relationships of our knowledge base in DAML+OIL

Generic-Patient models the information to record during each patient visit, such as patient identifying data, patient visit record and the anamnesis (Fig. 1). The latter includes chief complaint, history of present illness and past medical history.

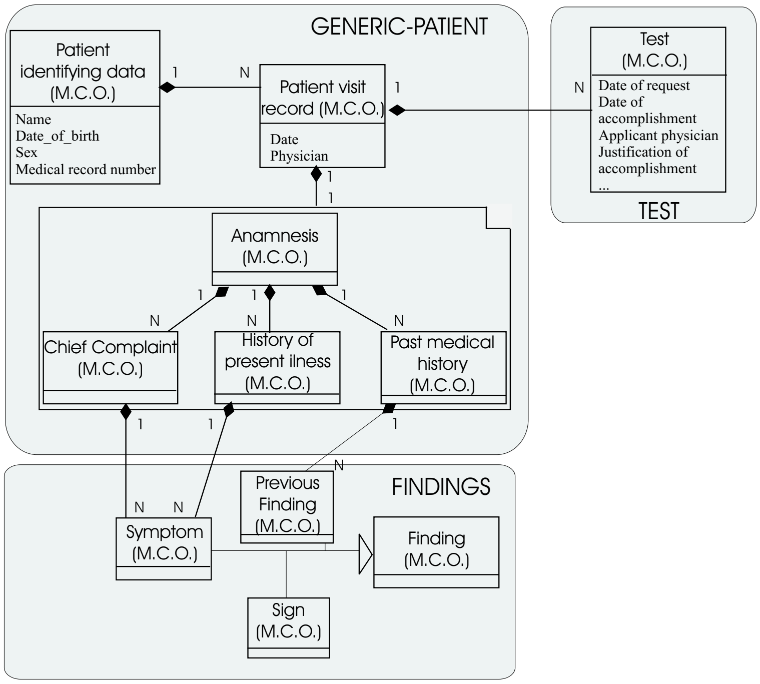

Fig. 1. Modelling the level 'Generic-Patient'

4.1 Specifying objects and attributes

In DAML+OIL, objects are described by giving a name for the object class. The definition of some classes in Generic-Patient is shown in Fig. 2.

```
<daml:Class rdf:ID="PatientIdentifyingData">
</daml:Class>

<daml:Class rdf:ID="PatientVisitRecord">
</daml:Class>

<daml:Class rdf:ID="Anamnesis">
</daml:Class>

  <daml:Class rdf:ID="ChiefComplaint">
</daml:Class>

<daml:Class rdf:ID="HistoryOfPresentIllness">
</daml:Class>

<daml:Class rdf:ID="PastMedicalHistory">
</daml:Class>
```

Fig. 2. Defining classes in DAML+OIL

Each attribute of a class of objects is defined by a kind of DAML+OIL property (named daml:DatatypeProperty), which relates objects to datatype values. For example, we have defined the attribute *Name* as an unique property, which maps strings into XML Schema (Fig. 3).

```
<daml:DatatypeProperty rdf:ID="Name">
<rdfs:comment>
Name is a DatatypeProperty whose range is xsd:string.
Name is also a UniqueProperty (can only have one name)
</rdfs:comment>

<rdf:type rdf:resource="http://www.daml.org/2001/03/daml+oil#UniqueProperty"/>
<rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#string"/>

</daml:DatatypeProperty>
```

Fig. 3. Defining an attribute for a class in DAML+OIL

A second example is the attribute *DateOfBirth*, which is a subproperty of the property *Date*. The latter maps dates into XML Schema (Fig. 4).

The internal structure of a class can be described by adding property restrictions to the class. Figure 5 shows some property restrictions added to the class *PatientIdentifyingData*.

```
<daml:DatatypeProperty rdf:ID="Date">
<rdfs:comment>
Date is a DatatypeProperty whose range is xsd:date.
</rdfs:comment>
<rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#date"/>
</daml:DatatypeProperty>


<daml:DatatypeProperty rdf:ID="DateOfBirth">
<rdfs:comment>
DateOfBirth is a subProperty Of Date and it is also a UniqueProperty
(can only have one date)
</rdfs:comment>
<rdfs:subPropertyOf rdf:resource="#Date"/>
<rdf:type rdf:resource="http://www.daml.org/2001/03/daml+oil#UniqueProperty"/>
</daml:DatatypeProperty>
```

Fig. 4. Defining 'date' attributes in DAML+OIL

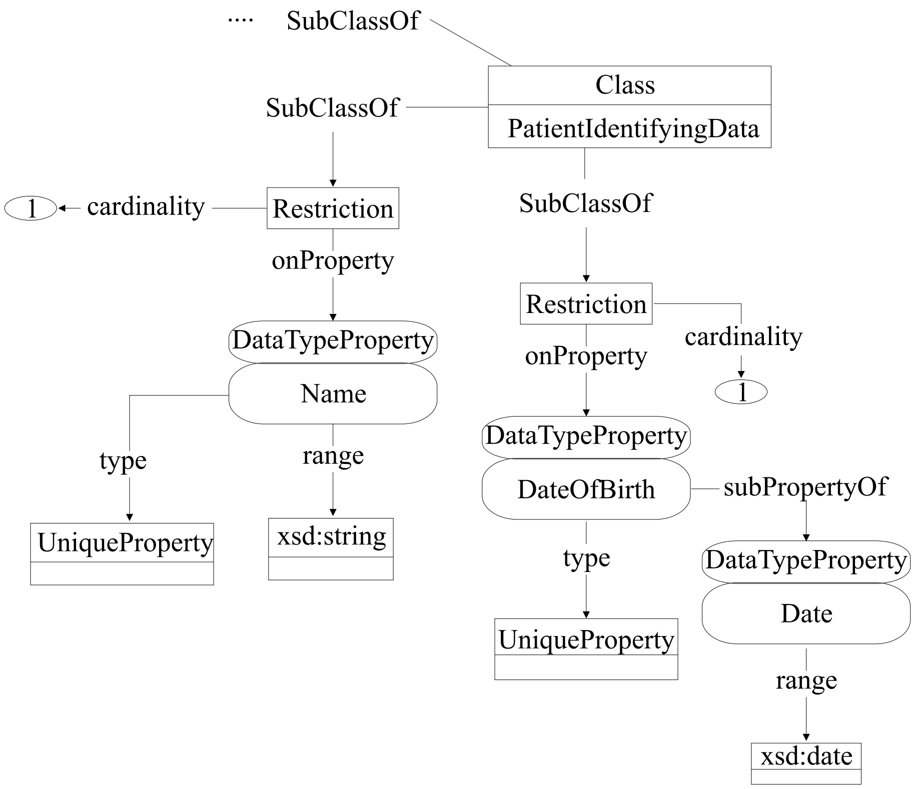

Fig. 5. Adding two property restrictions to the class PatientIdentifyingData

4.2 Specifying standard relations in an object-oriented approach

In an object-oriented approach, one standard relation is an aggregation. For example, for each patient there can be one or several visit records. This information can be modeled by an aggregation relation between one *PatientIdentifyingData* and many *PatientVisitRecords*. We have specified this kind of relation by:

1. Defining a new property *hasVisit*, which is a binary relation connecting the two classes implied in the aggregation (Fig. 6).
2. Adding a new property restriction on the class *PatientIdentifyingData*, such as it is displayed in Fig. 7, where the minimum cardinality 1 specifies that the aggregation relation is 1:N.

```
<daml:ObjectProperty rdf:ID="hasVisit">
<rdfs:domain rdf:resource="#PatientIdentifyingData"/>
<rdfs:range rdf:resource="#PatientVisitRecord"/>
</daml:ObjectProperty>
```

Fig. 6. Defining a binary relation in DAML+OIL

Binary relations between 1 object and N objects have been specified by defining a new daml:ObjectProperty and adding a new restriction on the class. For example in Fig. 8 the relation labeled as *Can be carried out by*, between *Exploration Phase* and *Primitive Phase* in the Tests theory, is an 1:N binary relation with one relation attribute, named *Choice Criteria*. Firstly, two new properties have been defined in this case: one for defining the binary relation and another for defining the relation attribute. Secondly, two restrictions have been added to the class *Exploration Phase*, as it is shown in Fig. 9.

On the other hand, a n-ary relation can be specified by defining n-1 properties and adding n-1 restrictions to one class. For example, in Fig. 8 the relation *Collects by using* is a 4-ary relation, which represents the knowledge about techniques to be used, pharmacological actions to be applied, and signs to be collected in each phase. Three properties can be defined: *Collects* between *PrimitivePhase* and *Sign*, *ByUsing* between *PrimitivePhase* and *Technique*, and *ByApplying* between *PrimitivePhase* and *PharmacologicalAction* (Fig 10).

5 Specifying rules 'If .. Then ..' in DAML+OIL

In our knowledge base we have modelled derivation rules as associations between two expressions, where each expression is a class containing four slots:

- A concept/relation.
- A slot.
- An operator (such as equal, greater, etc.).
- A value.

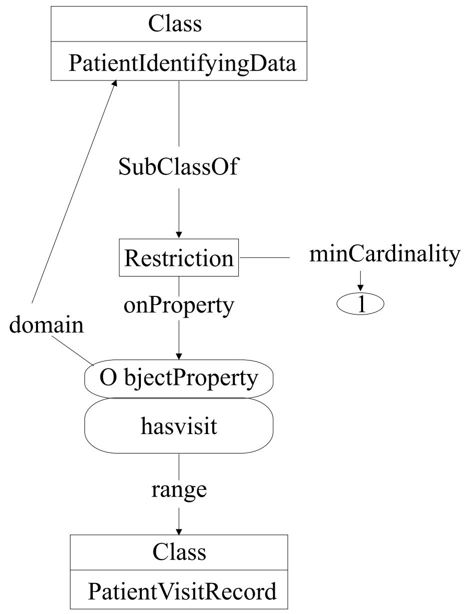

Fig. 7. Specifying a standard aggregation for the class PatientIdentifyingData

We have also considered that expressions can be nested by using logical operators (such as and, or). A simple example of a derivation rule is shown in Figure 11.

Each expression, such as *Distance-from-Snellen-Chart.decimalvalueOD<0.8* can be specified by defining a new class, for example, *DfSCOD-evaluation*, as an intersection of the class *Distance-from-Snellen-Chart* and a DAML+OIL restriction on a property (in this case, decimalvalueOD). This restriction can use user-defined datatypes (Fig 12).

A set of nested expressions can be specified by means of the boolean combination provided by DAML+OIL, such as daml:intersectionOf, daml:unionOf and daml:complementOf (Fig. 13).

Finally, a rule can be expressed by a name, an antecedent and a consequent (Fig. 14). In this example, the range of the antecedent is the class *Decreased-Visual-Acuity-Antecedent* and the range of the consequent is the class *Abstracted-Decreased-Visual-Evaluation.*

6 Conclusions

In this paper we have shown how a knowledge base for a heuristic application has been developed. We have distinguished two phases: modelling and representation. The modelling phase was carried out by applying CommonKADS. The representation phase has turned on the specification of the modelled knowledge in the Web language DAML+OIL.

As the Web is a universal medium for exchanging data and knowledge, a language for the Semantic Web, such as DAML+OIL, must have enough expressive power and provide support for both Syntactic and Semantic Inter-operability [4]. In this paper, we have revised several types of common relations in an object-oriented approach and we have shown how these types

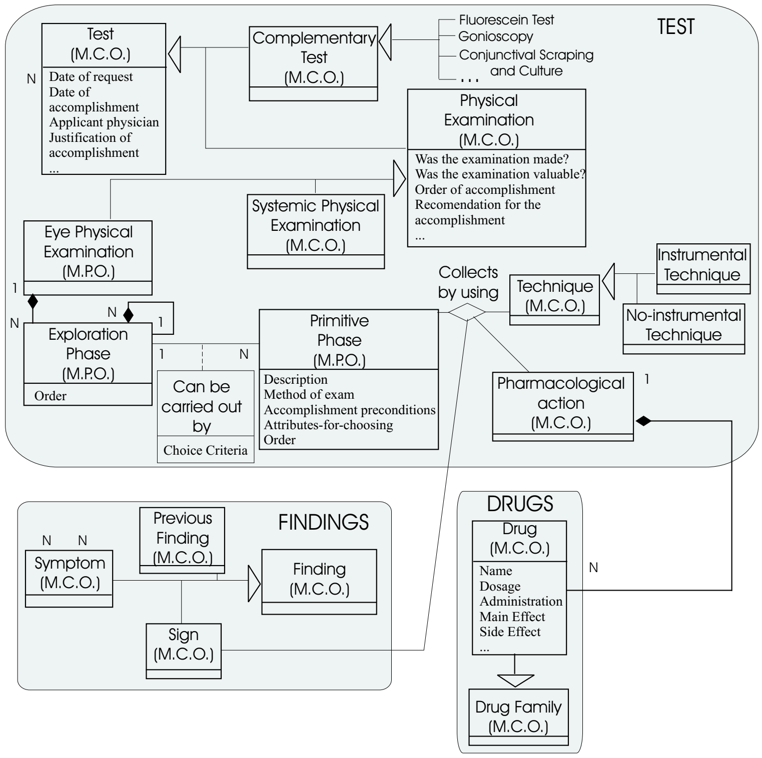

Fig. 8. Modelling the level 'Test'

```
<rdf:ID="CarriedOutBy">
<rdfs:domain rdf:resource="#ExplorationPhase"/>
<rdfs:range rdf:resource="#PrimitivePhase"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="ChoiceCriteria">
<rdfs:domain rdf:resource="#ExplorationPhase"/>
<rdfs:range rdf:resource="#OrderedListOfCriteria"/>
</daml:ObjectProperty>

<daml:Class rdf:about="ExplorationPhase">
<rdfs:comment>
 Several primitive phases can be carried out by a method of examination,
ie: One or more primitive phases can be carried out to one 'ExplorationPhase'
(through the object property 'CarriedOutBy').
</rdfs:comment>
<rdfs:subClassOf>
<daml:Restriction daml:minCardinality="1">
<daml:onProperty rdf:resource="#CarriedOutBy"/>
</daml:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<daml:Restriction daml:Cardinality="1">
<daml:onProperty rdf:resource="#ChoiceCriteria"/>
</daml:Restriction>
</rdfs:subClassOf>
</daml:Class>
```

Fig. 9. Defining a binary relation in DAML+OIL

```
<daml:ObjectProperty rdf:ID="Collects">
<rdfs:domain rdf:resource="#PrimitivePhase"/>
<rdfs:range rdf:resource="http://aiff.usc.es/elchus/medicalontology/daml+oil-f#Sign"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="ByUsing">
<rdfs:domain rdf:resource="#PrimitivePhase"/>
<rdfs:range rdf:resource="#Technique"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="ByApplying">
<rdfs:domain rdf:resource="#PrimitivePhase"/>
<rdfs:range rdf:resource="#PharmacologicalAction"/>
</daml:ObjectProperty>

<daml:Class rdf:about="PrimitivePhase">
<rdfs:subClassOf>
<daml:Restriction daml:minCardinality="1">
<daml:onProperty rdf:resource="#Collects"/>
</daml:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<daml:Restriction daml:minCardinality="1">
  <daml:onProperty rdf:resource="#ByUsing"/>
</daml:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<daml:Restriction daml:minCardinality="0">
  <daml:onProperty rdf:resource="#ByApplying"/>
</daml:Restriction>
</rdfs:subClassOf>
</daml:Class>
```

Fig. 10. Defining a 4-ary relation in DAML+OIL

```
IF
        Distance-from-Snellen-chart.decimalvalueOD <0.8
                                OR
        Distance-from-Snellen-chart.decimalvalueOI <0.8
                                OR
             Abstracted-visual-line.difference>=2

THEN
      ABSTRACT
           Visual-acuity-using-no-pinhole.abstracted='decreased'
```

Fig. 11. An example of a derivation rule

```
<daml:Class rdf:ID="DfSCOD-evaluation">
<daml:intersectionOf rdf:parseType="daml:collection">
<daml:Class rdf:about="#Distance-from-Snellen-Chart"/>
<daml:Restriction daml:cardinality="1">
 <daml:onProperty rdf:resource="#decimalvalueOD"/>
   <daml:hasClass rdf:resource= "http://aiff.usc.es/elchus/medicalontology/daml+oil-test-dt#under18"/>
</daml:Restriction>
</daml:intersectionOf>
</daml:Class>

<daml:Class rdf:ID="DfSCOI-evaluation">
<daml:intersectionOf rdf:parseType="daml:collection">
<daml:Class rdf:about="#Distance-from-Snellen-Chart"/>
<daml:Restriction daml:cardinality="1">
 <daml:onProperty rdf:resource="#decimalvalueOI"/>
   <daml:hasClass rdf:resource= "http://aiff.usc.es/elchus/medicalontology/daml+oil-test-dt#under18"/>
</daml:Restriction>
</daml:intersectionOf>
</daml:Class>

<daml:Class rdf:ID="AVLD-evaluation">
<daml:intersectionOf rdf:parseType="daml:collection">
<daml:Class rdf:about="#Abstracted-Visual-Line"/>
<daml:Restriction daml:cardinality="1">
 <daml:onProperty rdf:resource="#diference"/>
   <daml:hasClass rdf:resource= "http://aiff.usc.es/elchus/medicalontology/daml+oil-test-dt#equalorover18"/>
</daml:Restriction>
</daml:intersectionOf>
</daml:Class>
```

Fig. 12. Specifying an expression in DAML+OIL

```
<daml:Class rdf:ID="Decreased-Visual-Acuity-Evaluation">
<daml:unionOf rdf:parseType="daml:collection">
  <daml:Class rdf:about="#DfSCOD-evaluation"/>
  <daml:Class rdf:about="#DfSCOI-evaluation"/>
  <daml:Class rdf:about="#AVLD-evaluation"/>
</daml:unionOf>
</daml:Class>
```

Fig. 13. Specifying nested expressions in DAML+OIL

```
<daml:ObjectProperty rdf:ID="Decreased-Visual-Acuity-Antecedent">
<rdfs:domain rdf:resource="#Decreased-Visual-Acuity-Abstraction"/>
<rdfs:range rdf:resource="Decreased-Visual-Acuity-Evaluation"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="Decreased-Visual-Acuity-Consequent">
<rdfs:domain rdf:resource="#Decreased-Visual-Acuity-Abstraction"/>
<rdfs:range rdf:resource="Abstracted-Visual-Acuity"/>
</daml:ObjectProperty>

<daml:Class rdf:about="Decreased-Visual-Acuity=Abstraction">
<rdfs:subClassOf>
<daml:Restriction daml:Cardinality="1">
<daml:onProperty rdf:resource="Abstract-Visual-Acuity"/>
</daml:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<daml:Restriction daml:Cardinality="1">
  <daml:onProperty rdf:resource="#Decreased-Visual-Acuity-Antecedent"/>
</daml:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<daml:Restriction daml:Cardinality="1">
<daml:onProperty rdf:resource="#Decreased-Visual-Acuity-Consequent"/>
</daml:Restriction>
</rdfs:subClassOf>
</daml:Class>
```

Fig. 14. Specifying a derivation rule in DAML+OIL

of relations can be represented in DAML+OIL. For our application, we can emphasize that DAML+OIL enjoys

– Enough expressive power: Common relations in an object-oriented approach can be expressed in DAML+OIL.
– Syntactic inter-operability: Both domain structures and data can be read easily.
– Semantic inter-operability: The basic relations are represented in DAML+OIL unambiguously, so there is a direct connection from a domain model to its representation in DAML+OIL. But, the direct connection from DAML+OIL to the domain model is lost and it cannot be easily reconstructed. However, it is very easy to extend the set of DAML+OIL modelling primitives with the needed modelling primitives. These extensions facilitate the re-engineering of the domain model from just the representation in DAML+OIL.

For example, our ontology uses a simple aggregation, which could be described by a modelling primitive including a label, a class, a set of parts in the class and a cardinality. This aggregation could be mapped to DAML+OIL directly, as follows:

– Defining a new property named the label of the aggregation
– Specifying the domain of the property as the class in the aggregation
– Specifying the range of the property as the union of the classes making up the parts of the aggregation
– Adding a new restriction to the the class in the aggregation and specifying the required cardinality.

Currently, we are revising the whole ontology with the aim to integrate the definitions and the semantic network specified in UMLS (Unified Medical Language System: "http://www.nlm.nih.gov/research/umls/").

7 Acknowledgements

The authors are very grateful to Frank van Harmelen, who have changed the direction of our research in representing knowledge on Web.

References

1. T. Bray, J. Paoli, and C.M. Sperberg-McQueen. Extensible markup language XML 1.0 specification. In *http://www.w3.org/TR/REC-xml, W3C Recommendation*, 1998.
2. D. Brickley and R.V. Guha. Resource description framework RDF schema specification 1.0. In *http://www.w3.org/TR/rdf-schema, W3C Recommendation*, 2000.
3. B. Chandrasekaran, J.R. Josephson, and R. Benjamins. What are ontologies, and why do we need them? *IEEE Intelligent Systems and their applications*, 14(1):20–26, 1999.
4. S. Decker, F. van Hermelen, J. Broekstra, M. Erdmann, D. Fensel, I. Horrocks, M. Klein, and S. Melnik. The semantic web - on the respective roles of XML and RDF. *IEEE Internet Computing*, Setember/October, 2000.
5. S. Falasconi and M. Stefanelli. A library of implemented ontologies. In *Proc. of the ECAI Workshop on comparison of implemented ontologies*, Amsterdam, 1994.

6. D. Fensel, I. Horrocks, F. van Harmelen, D. Mc.Guinness, and P. Patel-Schneider. OIL: an ontology infraestructure for the semantic web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.
7. D. Fensel. *Ontologies: a silver bullet for knowledge management and electronic commerce*. Springer-Verlag, Berlin, 2001.
8. N. Library of Medicine. *Unified Medical Language System*. Bethesda, MD, 2001.
9. R.A. Miller, H.E. Pople, and J.D. Myers. INTERNIST-I, an experimental computer-based diagnosis consultant for general internal medicine. *New England Journal of Medicine*, 37(8):468–476, 1982.
10. A. Schreiber, H. Akkermans, A.A. Anjewierden, R. Hoog, N.R. Shadbolt, W. Val de Velde, and B. Wielinga. *Engineering and managing knowledge. The CommonKADS methodology*. The MIT Press, 1999.
11. M. Taboada, J. Des, J. Mira, and R. Marín. Development of diagnosis systems in medicine with reusable knowledge components. *IEEE Intelligent Systems*, 16(6):68–73, 2001.
12. F. van Harmelen and I. Horrocks. Reference description of the DAML+OIL ontology markup language. In *http://www.daml.org/2001/03/reference.html*, DARPA Agent Markup Language Program, 2001.
13. *Clasificación Internacional de Enfermedades (9a Revisión). Modificación Clínica*. Ministerio de Sanidad y Consumo, Madrid, 1988.