# A Description Logic for Use as the ODM Core

Lewis Hart, Patrick Emery
AT&T Government Solutions, Inc.
LewisHart@att.com  PatEmery@att.com

***Abstract* - Knowledge representation has become an important part of modern information systems that require more than tacit understanding of the data they manipulate. The OMG has lead the way in software system development with its Model Driven Architecture. UML and MOF are central elements in the MDA, therefore the development of a UML/MOF based ontology standard is key to using the MDA for future information aware systems. The Ontology Definition Metamodel is that OMG standard, based on a description logic core that facilitates mappings of existing knowledge representation, such as RDFS, OWL, Topic Maps and Entity Relation models into a collection of MOF models and UML notations for knowledge systems development.**

## I. INTRODUCTION

"Ontologies are often captured in knowledge representation (KR) languages that have come out of the AI community. These languages are often structured after logic formalisms, such as predicate logic, and have a grounding in these formalisms, which supports machine interpretation (reasoning). The ontology languages considered in [this context] are those that are fragments of predicate logic." [3]

The ODM RFP seeks at a minimum the following:

- A MOF metamodel for ontology definition
- A UML profile for ontology definition
- A mapping between the UML profile and the MOF metamodel
- Support for W3C Web Ontology Language OWL DL

But in a broader sense, there are other knowledge representations, specifically Entity-Relation (ER), Topic Maps(TM) and Simple Common Logic (SCL) that should be considered, but fell outside these narrow minimum requirements. In defining the response to the ODM RFP the project team realized it could take a narrow path and meet the minimum requirements of the RFP but leave behind some of the teams broader goals or take a wider path that results in a more flexible and robust ODM solution. The team chose to take the wider path.

## II. RATIONALE

One of the issues of a broader approach is the many-to-many scaling problem of mappings between the multiple languages. It was conjectured that a core metamodel could reduce this problem. In early discussions of the core, a minimal set of classes was discussed. This set was the cross section of elements that were obviously common among most of the metamodels the team was thinking of including OWL, RDF/S, Topic Maps and SCL. The other metamodels were specializations of this core. As discussions and contemplation continued the questions arising from what additional classes to include in the core and whether the core was a pure intersection of all the metamodels or a subset of a union of all the metamodels. About the same time, discussion occurred about the UML Profile and including an ER metamodel. The group realized that a slightly different approach to the core might be better. Instead of having generalization-specialization relationship between the core and the other metamodels maybe what was needed were mappings to and from the individual metamodels and the core. Shortly after that realization, a DL meta-model as the core began to take hold.

"A Model-Driven Semantic Web Reinforcing Complementary Strengths" [4], describes briefly the teams approach:

> "In defining [the ODM metamodel mappings], the project team wishes to avoid simply defining point-to-point mappings that lead to classic $N^2$ mapping explosion. Thus, the team is exploring the feasibility of defining a Core metamodel that contains elements common to all of the metamodels. Most likely, this Core will be based on *description logic*, a constrained form of knowledge

representation that is particularly amenable to supporting automated reasoning. "

A Description Logic is appropriate to use as a common core. Description logics represent a subset of first-order predicate logic aimed at being tractable while maintaining a richness of semantic expressiveness.

In addition to defining a set of five meta-models, the team will define a series of bi-directional mappings, where each mapping is between the Core DL metamodel and one of the other four metamodels, either OWL, UML, ER, or Topic Maps.

## III. A BRIEF OVERVIEW OF DESCRIPTION LOGICS

Logic is a system or model of reasoning. The simplest kind of logic called Propositional Logic deals with simple expressions that can be assigned a value of true or false. First-Order Predicate Logic extends Propositional Logic by allowing separate symbols for predicates, subjects and quantifiers. Description Logics are a subset of First-Order predicate Logic. Daconta, Orbst and Smith describe Description Logic as, "A knowledge representation formalism (sometimes called a terminological logic, classification logic, concept logic, or term subsumption logic) based on a subset of first-order predicate logic that is declarative formalism for the representation and expression of knowledge and sound, tractable reasoning methods founded on a firm theoretical (logical) basis."[1]

Description Logic[1],[2] can be thought of as defining the core knowledge base in a knowledge representation system. A DL knowledge base is traditionally divided into three principal parts, as illustrated in Figure-1:

- Terminology or schema, the vocabulary of application domain, called the 'TBox',
- Assertions, which are named individuals expressed in terms of the vocabulary, called the 'ABox' and
- Description Language that define terms and operators for building expressions.

Note that the TBox and ABox elements represent two separate meta-levels in the application domain, that is, the TBox elements are the definitions of elements instantiated in the ABox.

Most Description Logics define, as a minimum, atomic concepts, atomic roles and two special concepts, Universal (or top) and Empty (or bottom), as their fundamental terminology. The expressiveness of a Description Logic is defined by the expressiveness of its description language. A widely used description language, given the nomenclature $\mathcal{AL}$, defines expressions constructed using this small set of operators:

- Atomic negation or complement,
- Concept intersection,
- Role value restrictions and
- Limited existential quantification.

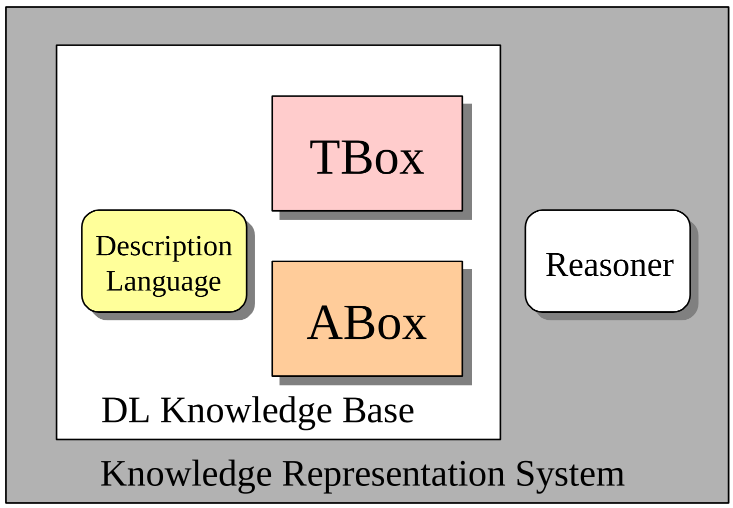


Figure 1. The primary components of a typical knowledge representation system based upon description logic.

Adding or constraining its description language can modify the expressiveness of a DL. Common modifications, followed by their nomenclature include:

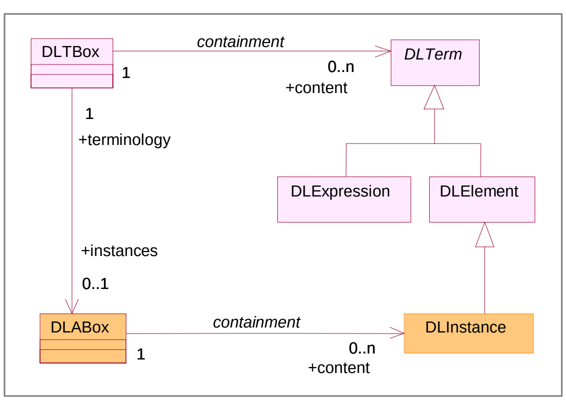
- Full expression negation ($\mathcal{C}$)
- Transitive Roles ($\mathcal{R}^+$)
- Full Existential Quantification ($\mathcal{E}$ )
- Role Hierarchies ($\mathcal{H}$)
- Inverse Roles ($\mathcal{I}$)
- Unqualified Number Restrictions e.g. < 5 ($\mathcal{N}$ )
- Functional Number Restrictions e.g. 0 or 1($\mathcal{F}$)
- Enumerated Classes ($\mathcal{O}$)
- Union constructor ($\mathcal{U}$ )
- Datatypes( $(\mathcal{D})$ )

TABLE – 1
A COMPARISON OF ODM META-MODELS BASED UPON THEIR DL'S DESCRIPTION LANGUAGE FEATURES.

| **Nomen-clature** | **Meaning** | **DL-Core** *SHIN(D)* | **UML** *AL$^{-}$HOIN(D)* | **OWL/ DL** *SHOIN(D)* | **Entity-Rel.** *AL$^{-}$N(D)* | **Topic Maps** *AL$^{--}$(D)* |
|---|---|---|---|---|---|---|
| *AL* | Atomic Concepts, Universal Concept, Bottom Concept, Atomic Negation, Intersection, Value Restrictions and Limited Existential Quantification | ✦ | ⊖ Atomic Concepts, Value Restrictions and Limited Existential Qualification. (***AL$^{-}$***). | ✦ | ⊖ Atomic Concepts, Value Restrictions and Limited Existential Qualification. (***AL$^{-}$***) | ⊖ Atomic Concepts and Value Restrictions. (***AL$^{--}$***) |
| *C* | Full Negation or Complement* | ✦ | | ✦ | | |
| *E* | Full Existential Quantification | ✦ | | ✦ | | |
| *H* | Role Hierarchies | ✦ | ✦ | ✦ | | |
| *I* | Inverse Roles | | ✦ | ✦ | | |
| *N* | Unqualified Number Restrictions e.g. < 5 | ✦ | ✦ | ✦ | ✦ | |
| *O* | Enumerated Classes | | ✦ | ✦ | | |
| *R$^{+}$* | Transitive Roles | ✦ | | ✦ | | |
| *U* | Union constructor | ✦ | | ✦ | | |
| *(D)* | Datatypes | ✦ | ✦ | ✦ | ✦ | ✦ |

✦ Fully has feature, ⊖ partial has feature.
***UE** is semantically equivalent to **C,** so, **ALUE** is often written **ALC**.

The nomenclature short hand can be used to build "names" for various families of Description Logics. For example, a common DL is *ALCR$^{+}$*, adding full negation and transitive roles to the basic *AL*. This DL is often given the nomenclature *S*. (*S* can be shorthand for either *ALC* or *ALCR$^{+}$* depending on source. Although many of these nomenclature letters are widely and commonly used, there is no universal agreement on usage.)



Figure–2. The TBox and ABox representation in the DL meta-model.

As a concrete example, OWL DL is approximately *SHOIN(D)* or equivalently *ALC R$^{+}$HOIN(D)*, expanding the *S*. OWL Lite is the less expressive *SHIF(D)*, allowing only functional, 0 or 1, number restrictions. Table-1 compares the DL meta-model with the other meta-models currently being considered for the ODM.

IV. DL CORE META-MODEL OVERVIEW

In order to support desired set of knowledge representation languages, a description language was carefully designed for the core meta-model. This DL is

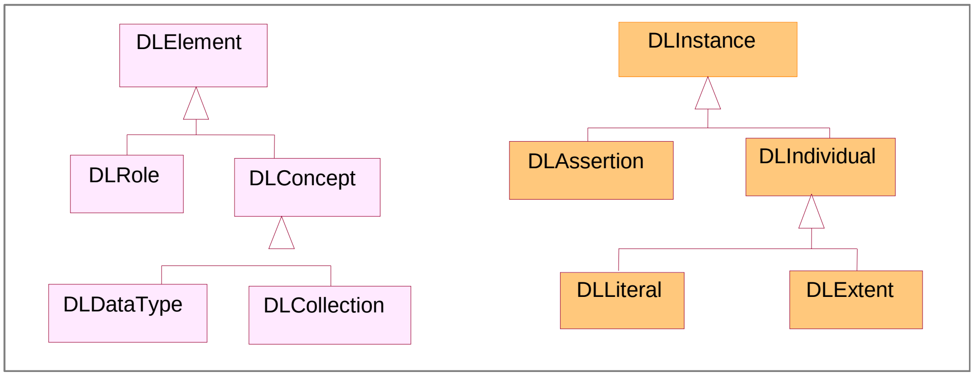


Figure –3. The specializations of TBox Elements, on left, and ABox Instances define the basic structural elements of the DL meta-model.

slightly less expressive than OWL DL, with a nomenclature of *SHIN(D)*. The DL meta-model is designed to be minimally constrained and easily extensible. While the DL meta-model is very similar to OWL DL, it was felt that a totally separate model was needed to provide a separation of concerns and to allow for general DL terminology.

The DL Meta-model implemented a Meta-Object Facility Version 2.0 (MOF) model, using UML notation. The MOF model that was developed is a straightforward implementation of the required elements.

At the top level, the DL meta- model defines meta-classes DLTBox and DLABox that correspond directly with the TBox and ABox concepts of a DL knowledgebase, shown in Figure-1. The DLTBox meta-class contains a collection of DLTerm meta-classes, and its specializations, while the DLABox contains a collection of DLInstances.

DLTerm is an abstract class that has specializations DLExpression, corresponding to the DL meta-model's description language and DLElement corresponding to concepts and roles.

In the ABox, there are specializations of DLInstance that correspond with each TBox class. DLAssertions are tuples of instances that define the members of a DLRole, right handside of Figure-3. DLIndividuals are instances that define the members of a DLConcept. Two specializations of DLConcept are defined DLDatatype with DLLiterals as instances and DLCollection with DLExtent as instances, left handside of Figure-3. The collections construct does not add any additional expressiveness but is included as a representational convenience.

DLExpressions are composed of DLTerms using DLConstructors, as shown in Figure-4. Both monadic and dyadic constructors are supported. It is the types and semantics of these constructor specializations that define the description language.

### A *Translations and Mappings*

The key to using a core meta-model is the mappings to the other ODM meta-models and to their concrete syntax. Since these meta-models are all expressed in MOF, the natural choice is a MOF specific mechanism. The Query-View-Translation, or QVT, in process OMG standard [6] is the obvious choice for defining these mappings and translations.

The relations and transformaitons defined in the mappings are rsponsible for ensuring that the semantics of the source meta-model are accurately and completely translated into the target model.

## V. CONCLUSION

With an ODM that includes more than one metamodel a core of some sort is required to avoid the N2 mapping problem and to help scope the metamodels that could be included.

Logics in general and description logics in particular are central to much of the ontology work and languages used to date. Use of a DL metamodel seems appropriate in an ontology definition

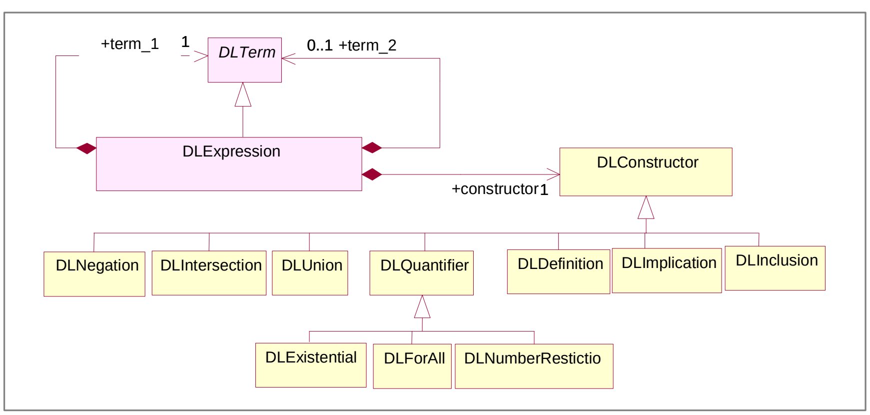


Figure-4. The DL meta-model supports both monadic and dyadic constructors to build description language expressions.

standardization effort. Strong ties exist between the ODM requirements and the resulting product under development and the W3C's recently released OWL and RDF recommendations. As these two languages have strong ties of their own to Description Logic so too will the ODM.

## VI. REFERENCES


[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider Editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, 2003, 574 pages; ISBN 0521781760

[2] I. Horrocks, U. Sattler and S. Tobies; Practical Reasoning for Very Expressive Description Logics; Logic Journal of the IGPL, Volume 8, Issue 3: May 2000;

[3]Object Management Group**,** *Ontology Definition Metamodel Request For Proposal*, OMG Document: ad/2003-03-40, http://www.omg.org/cgi-bin/doc?ad/2003-03-40

[4 ] David Frankel, Pat Hayes, Elisa Kendall, Deborah McGuinness , "A Model-Driven Semantic Web Reinforcing Complementary Strengths", To be pucblished in MDA Journal, July 2004, v00-07

[5] Michael C. Daconta, Leo J. Orbst, and Kevin T. Smith, *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*, Wiley Publishing Inc., 2003, ISBN 0471432571

[6] Object Management Group**,** *Ontology Definition Metamodel Request For Proposal*, OMG Document: ad/02-04-10, http://www.omg.org/cgi-bin/apps/do_doc?ad/02-04-10;
See also
http://neptune.irit.fr/Biblio/qvt_specification.shtml