Ontologies for the Semantic Web in Casl

Klaus Lüttich, Till Mossakowski, and Bernd Krieg-Brückner

BISS, FB3 – Dept. of Computer Science, Universität Bremen
{luettich, till, bkb}@tzi.de

Abstract. This paper describes a sublanguage of Casl, called Casl-DL, that corresponds to the Web Ontology Language (OWL) being used for the semantic web. OWL can thus benefit from Casl's strong typing discipline and powerful structuring concepts. Vice versa, the automatic decision procedures available for OWL DL (or more precisely, the underlying description logic $\mathcal{SHOIN}(\mathbf{D})$) become available for a sublanguage of Casl. This is achieved via translations between Casl-DL and $\mathcal{SHOIN}(\mathbf{D})$, formalized as so-called institution comorphisms.

1 Introduction

The internationally standardized Web Ontology Language (OWL) [10] is a major contribution to the upcoming Semantic Web [11, 5] that proposes a new form of web content meaningful to computers. One problem of the documents on the web is the restricted ability to search for certain topics without any knowledge how an author or organisation names the concept. Another problem results form multimedia files like audio or movie files which cannot be indexed by techniques available today; the meaning must be given by meta data. However, just giving a text describing a piece of multimedia yields only a very limited aid for searching.

Therefore, the W3C (World Wide Web Consortium) and Tim Berners Lee proposed the Semantic Web, where the meaning is given by shared and extended ontologies that provide organised knowledge about certain domains; thus the contents of the web is accessible by computers. Hence, it becomes possible e.g. to search for the least cost of a phone call from Singapore to Germany. The visitor from Europe does not need any knowledge of the foreign language, because the query is given in a semantic-based language that is also provided by the Singapore telephone company. Indeed, with Swoogle [1], a first search engine for OWL and RDF documents is available.

In this work, we interface OWL with the specification language Casl [6, 9]. Casl provides a strong typing discipline, which allows to find conceptual errors at an early phase. Moreover, powerful structuring constructs allow the modularization of large theories into manageable pieces. Both features are present in OWL in a very limited form only. We hence propose a sublanguage of Casl, called Casl-DL, which corresponds to OWL DL in expressive power, but which retains the above mentioned advantages. Casl-DL can also be used to interface Casl with efficient decision procedures that are available for description logics.

J.L. Fiadeiro, P. Mosses, and F. Orejas (Eds.): WADT 2004, LNCS 3423, pp. 106–125, 2005.

The paper is organised as follows: Section 2 recalls the underlying description logic $\mathcal{SHOIN}(\mathbf{D})$ of OWL DL. Section 3 describes the Web Ontology Language OWL DL. Section 4 introduces CASL and the sublanguage CASL-DL. Section 5 continues with translations between OWL DL and CASL-DL. Section 6 concludes the paper. Last but not least an appendix collects some tables showing the concrete translations between $\mathcal{SHOIN}(\mathbf{D})$ and CASL-DL constructs including semantics for the $\mathcal{SHOIN}(\mathbf{D})$ constructs.

2 $\mathcal{SHOIN}(\mathbf{D})$

$\mathcal{SHOIN}(\mathbf{D})$ is an expressive description logic [3, 19, 18]. Its main purpose is the definition of hierarchies of concepts and roles. In terms of logic, concepts are unary and roles are binary predicates. The general properties of concepts and roles are collected in a so-called TBox. By contrast, the ABox represents a particular database, i.e. defines individuals to belong to concepts and roles. It also defines concepts and roles involving predefined datatypes. See Fig. 1 for an example of a TBox describing the class definitions of a family.

$$
\begin{aligned}
Woman &\equiv Person \sqcap Female \\
Man &\equiv Person \sqcap \neg Woman \\
Mother &\equiv Woman \sqcap \exists hasChild.Person \\
Father &\equiv Man \sqcap \exists hasChild.Person \\
Parent &\equiv Mother \sqcup Father \\
Grandmother &\equiv Mother \ \sqcap \exists hasChild.Parent \\
MotherWithManyChildren &\equiv Mother \sqcap \geqslant 3\ hasChild \\
Wife &\equiv Woman \ \sqcap \exists hasHusband.Man
\end{aligned}
$$

Fig. 1. Example TBox: Family

The standard description logic that is the base of all description logics is called $\mathcal{ALC}$. $\mathcal{ALC}$ has a notation for the universal concept $\top$ and the bottom concept $\bot$ (representing the always true and the empty predicate). Moreover, new concepts can be built with unions, intersections and complements of concepts. Finally, concepts can be universally or existentially projected along roles (e.g. $\exists hasChild.Person$ means the concept that consists of all individuals having some person as their child).

The logic $\mathcal{ALC}_{R^+}$ adds the possibility to specify roles to be transitive. This logic is also abbreviated by $\mathcal{S}$, which is the first letter of the name $\mathcal{SHOIN}(\mathbf{D})$. Likewise, the other letters are used for various features of description logics [3, pp.494-495]. The letter $\mathcal{H}$ adds role hierarchies (i.e., the possibility to specify inclusions between roles) and the letter $\mathcal{I}$ adds inverse roles (i.e. the possibility to generate a new role by just swapping the arguments of a given role). Unqualified number restrictions and nominals are added by $\mathcal{N}$ and $\mathcal{O}$. The former allow