Abstract

Protégé provides a complete ontology and knowledge base management tool. Along with JESS, JessTab provides one method of rule based reasoning over a Protégé ontology and knowledge base. However once JessTab rules have been created for a knowledge base, they are explicitly tied to it as they name particular classes and slots, which greatly hinders their reuse with further knowledge bases. We have developed a two phase process and a supporting tool to support the reuse of JessTab rule sets. The first phase involves changing the class and slot references in the rule set into an abstract reference; the second phase involves automatically mapping between the abstract rules and further knowledge bases. Once mappings have been defined and applied for all the classes and slots in the abstract rules, the new rule set can then be run against the new knowledge base. We have satisfactorily tested our tool with several ontologies and associated rule sets; moreover, some of these tests have identified possible future improvements to the tool.

1 Introduction

Ontologies have become one of the most widely used forms of domain knowledge capture. When used effectively they provide us with an explicit definition and a common understanding of a domain and the properties, relationships and behaviours of its components that can be communicated between people and machines.

RDF and RDFS [14], DAML+OIL and OWL [1] are representational formalisms for describing ontologies. Other languages provide mechanisms for querying these representations, for example RDQL [21]. Similarly, languages such as SWRL [9] and the various RuleML projects [25] allow one to formally specify rules for reasoning with the content of an ontology. While these formalisms provide a way to capture queries, an inference engine is still required to run them. One rule engine currently growing in popularity is JESS, the Java Expert System Shell; for examples of two recent projects involving JESS see [8] and [12].

JESS was originally developed by the Sandia National Laboratories as a Java implementation of the popular C Language Integrated Production System (CLIPS) [3], although it has since evolved into a powerful Java rule engine and scripting language in its own right [7]. The rise in JESS’s use may be in part
due to the useful JessTab [5] plugin for the widely used Protégé1 [24] ontology editor which allows developers to run JESS rules against an ontology created and populated in Protégé.

1.1 Ontology Tied Rules

As with CLIPS, JESS requires explicit definitions of the data types that will be used in the form of templates. Conveniently, there is a mapping command in JessTab which automatically produces these templates based on the classes and slots of a Protégé ontology. Along with the templates, JESS requires a set of facts (which are instantiations of the templates) to reason over. Again, helpfully the mapping command automatically creates facts from the corresponding instances which are defined as part of a Protégé project. When writing JessTab rules, the developer refers to these templates and facts as if they had been created as part of the main JessTab program. In doing this, the rules are explicitly tied to the ontology as they are required to name particular classes and slots.

Having the rules tied to a particular ontology in this way is unavoidable, but it greatly hinders reusing a set of JessTab rules developed for one ontology with additional ontologies/knowledge bases. This is because reuse of a set of rules requires one to carry out a manual mapping between the class and slot names in the JessTab rules and those in the second (and subsequent) ontologies/knowledge bases. Further, this would be a tedious and very error prone process.

For this reason, we have developed a plugin for Protégé, which supports the developer with this task. Given a set of JessTab rules (JessTab rules differ slightly from standard JESS rules as they need to link to the Protégé ontology) and a further ontology, our tool attempts to automatically map concept names featured in the rules to concept names in the "new" ontology. To achieve this, we make use of techniques used in the ontology mapping, merging and alignment sub-fields, namely partial and exact string comparisons and synonym look-up in a lexical database (WordNet [6]). We also provide facilities for the user to define mappings manually.

In section 2 we discuss some current ontology mapping and merging tools; in section 3 we outline two scenarios where our tool could be used; in section 4 we briefly outline our tool's functionality; in section 5 we describe experiments we have performed and report some results. In section 6 we discuss some modifications inspired by our experiments which should improve our tool's performance and section 7 concludes this report with a summary of our findings.

2 Related Work

As mentioned above, we make use of techniques originally developed in the ontology mapping, merging and alignment fields. There have been various approaches to these tasks including use of specially designed algebras [17], use of

1Protégé-2000 and its successors up to the latest version, Protégé 3.1