Composing Modular Ontologies with Distributed Description Logics

Luciano Serafini and Andrei Tamilin

Center for Information Technology
Fondazione Bruno Kessler
Trento, Italy
{serafini,tamilin}@fbk.eu

Summary. This chapter demonstrates the use of the Distributed Description Logics framework (DDL) and the distributed reasoner DRAGO as formal and practical tools for composing modular ontologies from purely terminological SHIQ ontology modules. According to DDL vision, a modular ontology can be formally represented by a distributed T-box, comprising a set of separate T-boxes (one for each ontological module), which are pairwise interrelated by “bridge rules” (inter-module connectives allowing to access and import knowledge contained in modules). The chapter gives the semantic explanations of knowledge import via bridge rules as well as presents the distributed tableaux reasoning technique for its computation. Practically, the implementation of the distributed tableaux in DRAGO reasoner and its use for modular ontology composition is described and experimentally evaluated.

12.1 Motivation and Approach

Recent years have witnessed the theoretical birth and practical maturation of the web ontology technology. Significant amount of academic research has been directed on the proposal and later the World Wide Web consortium standardization of representational languages for publishing ontologies on the web, such as OWL [4], proving the appropriateness of Description Logic formalism for being an underpinning theory for performing the formal analysis of ontologies [3, 2, 1, 15, 13], and finally the development of effective reasoning algorithms [12, 18, 19, 25, 26, 16] and practical implementations of automatic inference systems [14, 11, 10] facilitating the automated processing and use of ontologies. Despite the important steps ahead toward strengthening the web ontology technology, the distinguishing feature of the developed representation languages and tools is their monolithic treatment of ontologies and a little support for organizing ontologies in a modular way.

Practical need for ontology modularization and clarification of what a modular ontology is can be perceived from two complementary scenarios. A promising way of dealing with large-scale ontologies is to decompose (partition) them into a collection of smaller, more specific ontologies, which, together with the relations between
them, constitute the representation that is semantically equivalent to the original ontology. Conversely, it may be desirable to compose a set of ontologies into a coherent network that can be referred to as a single entity. In both cases, the ultimate ontology can be referred as modular – it comprises a set of autonomous ontological modules, which are interrelated by inter-module connectives. The inter-module connectives enable modules to access the knowledge contained in other connected modules and allowing, therefore, the integral use of the whole modular ontology.

In the current proposal of web ontology language (OWL) the support of ontology modularization is represented by the construct owl:imports. The import functionality of OWL, however, has several limitations moderating its modularization capabilities. Theoretically, the semantics of OWL is defined in such a way that all ontologies, imported and importing, share the single global interpretation. As a consequence, imported and importing ontologies cannot describe differently the very same portion of the world, using different perspectives and levels of granularity, without rising a logical contradiction. The practical consequence of the global semantics of OWL results in a global reasoning approach implemented in existing reasoners. The global reasoning approach consists in taking all ontologies participating in the import, combining them together in a unique reasoning space and further reasoning in it. The final shortcoming of owl:imports concerns with its ability to import ontologies only as a whole, while it is often the case in practice that only a certain part of imported ontology is of interest to the importing ontology.

Taking the above disadvantages, we see the following challenging requirements to be satisfied by the practically utilizable ontology modularization approach:

- Supporting semantic locality of ontological modules. This requirement addresses the ability of modules to have semantics different from the created modular ontology, being expressed in local languages of different expressive power so that each module can be dealt with a reasoner, specially tuned for it.
- Preserving loose coupling and autonomy of ontological modules. In other words, modularization technology should be able to keep ontological modules as separate contributors to the desired modular ontology, rather than hardly integrating them in a monolithic entity. This consequently implies that some sort of a distributed reasoning is required to operate over a set of autonomous ontological modules.
- Enabling a partial reuse of knowledge from ontological modules. In many practical cases we would like to reuse only part of a certain module, since the other part of the module may be irrelevant to the application domain of the modular ontology or, in the worst case, can even contradict some of its knowledge.
- Preserving directionality of knowledge import. This requirement means that the ontology module should not be affected by the modular ontology importing it. For example, if the importing ontology becomes inconsistent the modules it imports should not be automatically rendered as inconsistent too.
- Ensuring a scalability of modular ontology technology. Scalability criterion addresses two aspects: representation and reasoning. The scalability of the representation means the ease of large-scale ontology maintenance through its