The Description Logic Reasoner CICLOP (Version 2.0)

François de Bertrand de Beuvron, Martina Kullmann,
David Rudloff, Michael Schlick, François Rousselot

LIIA, ENSAIS, 24, bd de la Victoire, F-67084 Strasbourg Cedex
beuvron@liia.u-strasbg.fr
http://www-ensais.u-strasbg.fr/liia/ciclop/ciclop.htm

1 Introduction

Description logic knowledge representation languages provide means for expressing abstract knowledge about concepts composing a terminology (TBox), as well as knowledge about concrete facts, i.e. objects instantiating the concepts, which form a world description (ABox). Since description logics are provided with a formal syntax and formal model-theoretic semantics, sound and complete reasoning algorithms can be formulated.

The description logic system CICLOP (Customizable Inference and Concept Language for Object Processing) [2] has been developed as a system for practical use. Thus, its facilities have been motivated by its applications. For Version 2 of CICLOP, the former system (Version 1.3) has been completely re-implemented and some extensions have been made.

2 System Description

The applications which have motivated the development of CICLOP are from the areas of configuration [7], semantic indexing of corpora [8], natural language querying of databases [3], and decision support for disaster response [4]. They have shown a need for facilities to manage the knowledge base, as well as for highly expressive knowledge representation languages.

To structure big knowledge bases, CICLOP allows for the definition of multiple disjoint TBoxes and corresponding ABoxes. Thus, concepts can be grouped together according to their context. Each TBox can be associated with an appropriate expressiveness. Even if the domains associated with different TBoxes are disjoint, individuals from different domains can be related by so-called connector roles.

The basic expressiveness supported by CICLOP is $\mathcal{ALC}$. Besides, it is possible to define features, primitive role hierarchies, inverse and transitive roles, as well as general concept inclusion axioms, i.e. CICLOP can deal with cyclic axioms. These language features can be combined as required by the application, as far as the resulting logic is still decidable.

Reasoning with respect to a knowledge base composed of several TBoxes and ABoxes with different expressivenesses is based on tableau algorithms which are sound and complete. Inferences are provided for the standard TBox and ABox reasoning tasks. The implementation uses optimization techniques such as lexical normalization and encoding, dependency directed backtracking, and model caching [5]. Furthermore, CICLOP uses blocking to deal with terminologies containing cycles, transitive and inverse roles. A description of the underlying algorithms can be found in [7].

Besides, CICLOP provides means for terminology closure. A closed TBox does not allow for the introduction of new concepts. Also, concepts have to be defined to be concrete or abstract, and individuals are forced to belong to concrete concepts only ([9], [7]). Dividing the knowledge base into multiple TBoxes and ABoxes allows for closing only parts of it.

Furthermore, CICLOP allows for the defini-

tion of TBoxes which represent concrete domains [1]. A numeric TBox allows for representing constraints in form of linear equations and inequalities, whereas in string TBoxes concepts can be defined by means of a set of possible and impossible strings. The implementation of a concrete domain interface for CICLOP is a joint work with the Department of Computer Science of King's College, London, whose description logic system WELLINGTON is also implemented in Java. The interface allows for associating concrete domains without important changes in the main system.

Since CICLOP has been implemented in Java (JDK 1.2), it runs on any operating system providing a Java Virtual Machine. Also, it can be executed as an applet within any web browser supporting Java 2. This makes the system easily usable. Besides, CICLOP is provided with a graphical user interface and a text interface based on the syntax specified in [6]. Furthermore, the Java API can be used to build other applications on top of the system.

3 Future Work

The user interface is planned to be extended to visualize the satisfiability checking process, i.e. to display internal events like constraint expansion, individual creation, and clash detection together with the expansion graph. Besides, the expressiveness of the system will be extended to feature chains with corresponding agreement and disagreement. Preliminary tests show that CICLOP performs well compared to other description logic systems, but standard benchmark tests have still to be done.

References

[1] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *Proc. of the* 12^{th} *International Joint Conference on Artificial Intelligence*, pages 452–457, Sidney, Australia, 1991.

[2] F. de Bertrand de Beuvron, F. Rousselot, M. Grathwohl, D. Rudloff, and M. Schlick. Ciclop. In *Proc. of the International Workshop on Description Logics '99, System Comparison*, Linköping, Sweden, 1999.

[3] F. de Bertrand de Beuvron, F. Rousselot, and D. Rudloff. Interpretation of description logics for natural language and for databases. In *Proc. of the International Workshop on Description Logics '97*, Paris, France, 1997.

[4] M. Grathwohl, F. de Bertrand de Beuvron, and F. Rousselot. A new application for description logics: Disaster management. In *Proc. of the International Workshop on Description Logics '99*, Linköping, Sweden, 1999.

[5] I. Horrocks. *Optimizing Tableaux Decision Procedures for Description Logics.* PhD thesis, University of Manchester, Manchester, England, 1997.

[6] P. F. Patel-Schneider and B. Swartout. Description-logic knowledge representation system specification. Technical report, AI Principles Research Department, AT&T Bell Laboratories, 1993.

[7] M. Schlick. *CICLOP: Les logiques de description appliques la configuration.* PhD thesis, Université de Haute Alsace de Mulhouse, France, 1999.

[8] A. Todirascu, F. de Bertrand de Beuvron, and F. Rousselot. Using description logics for indexing documents. In *Proc. of the IAR Annual Meeting '99*, Strasbourg, France, 1999.

[9] R. Weida. *Closed Terminologies and Temporal Reasoning in Descriptions for Plan Recognition.* PhD thesis, Columbia University, New York, NY, 1998.