# Reasoning in Description Logics with Wellington 1.0 System Description*

Ulrich Endriss

Department of Computer Science, King's College London, Strand, London WC2R 2LS, UK, Email: endriss@dcs.kcl.ac.uk

http://www.dcs.kcl.ac.uk/research/groups/logic/wellington/

## 1 Introduction

Description logics are formal knowledge representation languages with a relatively simple syntax and well-defined semantics. According to the description logic paradigm, knowledge is divided into a terminological part (TBox), where concepts like *beverages that are carbonated and have some ingredient that is alcoholic* are defined, and an assertional part (ABox), where individuals are related to each other and asserted as being instances of certain concepts. For an introduction to the field and an overview of main directions of current research we refer to [1].

In this paper we introduce a new description logics based knowledge representation and reasoning tool, the WELLINGTON system, which is currently being developed by the Group of Logic and Computation at King's College London.

## 2 System Description

Unlike a number of other description logic systems, that have been written in functional languages, WELLINGTON is being developed in Java. By choosing a mainstream object-oriented language rather than a functional one we hope to make the system more accessible to users outside the description logics community. Java in particular allows for the development of (almost) platform-independent software. For most system configurations applets can be launched from a web browser without the need to install any additional software. WELLINGTON 1.0 is available as both a Java application and an applet and may be run online over the Internet or can be downloaded for local use from our project web site (see top of page).

Currently, the system supports ABox reasoning in the standard description logic $\mathcal{ALC}$ (without global axioms). Using the ABox consistency checking algorithm it is also possible to check the consistency of a given concept formula and to check the subsumption relation between two given concept formulas.

WELLINGTON 1.0 implements a multimodal tableaux-like calculus with a number of optimisations, including lexical normalisation, semantic branching with heuristic guided search, beta simplification (disjunctions entailed by one of their subformulas on the same branch are not expanded), non-branching beta rules (also called boolean constraint propagation), and backjumping. Furthermore, in order to minimise the time required for comparing formulas the implementation assures that for each (syntactically) distinct formula not more than one object is

*This work is part of the *Data Driven Logic Algorithms* project, which is funded by the EPSRC under grant reference number GR/L91818.

created. An overview of optimisation techniques for description logic tableaux may be found in [5]. WELLINGTON seems to perform well, but to date no detailed evaluation has been carried out.

On the calculus level, one aspect where our system apparently differs from many others is, that one proof gives rise to exactly one tableau, on which each branch may hold formulas labelled by different ABox individuals. The standard *algorithmic* presentation [4], on the other hand, assumes a number of so-called nodes, each of which contains the formulas associated with one of the ABox individuals. These formulas again are (at least implicitly) structured as a tableau. Besides being semantically clearer and closer to the presentation of tableaux calculi for e.g. modal logics, we believe that our approach will simplify the integration of mechanisms for reasoning about concrete domains [2].

## 3 Future Developments

WELLINGTON 1.0 is only the beginning. In the long run we intend to develop a system for ABox and TBox reasoning in the description logic proposed in [6], which extends $\mathcal{ALC}$ by a number of features, notably arithmetical constraints over numerical aspects of sets of role-fillers, complex role terms and hierarchies, as well as various generalised quantifiers. The current prototype can already be used to manage knowledge bases encoded in that language, but the reasoning services are yet to be implemented.

The first obvious extension of the current version will be to allow for unfolding of acyclic concept definitions. Then it will be possible to check ABox consistency, concept consistency, and concept subsumption with respect to a TBox. This in turn will provide the basis for a concept classification algorithm.

Furthermore, we plan to augment WELLINGTON with the ability to reason about concrete domains [2]. In cooperation with the LIIA Strasbourg we are currently defining a general Java interface for concrete domain reasoning that will be integrated into both CICLOP [3], the description logic system developed in Strasbourg, and WELLINGTON. This will allow us to exchange implementations of particular domains without the need to alter any code in the main systems.

In the context of concrete domains we are particularly interested in domains that can be used to combine description logics with temporal reasoning mechanisms.

## References


[1] F. Baader. Logic-based knowledge representation. In M. J. Wooldridge and M. Veloso, editors, *Artificial Intelligence Today, Recent Trends and Developments.* Springer-Verlag, 1999.

[2] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence, IJCAI'91*, Sydney, 1991.

[3] F. Beuvron, F. Rousselot, M. Grathwohl, D. Rudloff, and M. Schlick. CICLOP (system description). In *Proceedings of the International Workshop on Description Logics, DL'99*, Linköping, 1999.

[4] I. Horrocks. *Optimising Tableaux Decision Procedures for Description Logics.* PhD thesis, University of Manchester, 1997.

[5] I. Horrocks and P. F. Patel-Schneider. Optimising description logic subsumption. *Journal of Logic and Computation*, 9(3):267–293, 1999.

[6] H. J. Ohlbach. A theory resolution style ABox calculus. Extended abstract. In *M4M, Methods for Modalities 1, Workshop Proceedings.* ILLC, University of Amsterdam, 1999.