An Universal GUI for Theorem Provers

Boško Stanković[⋆†]

[⋆]Military Academy, University of Belgrade, SERBIA
zekave@yahoo.com

Nenad Krdžavac, Vladan Devedžić[†]

[†]FON - School of Business Administration, Department of Information Technology, University of Belgrade
POB 52 Jove Ilića 154 11000 Belgrade, SERBIA
nenadkr@galeb.etf.bg.ac.yu, devedzic@etf.bg.ac.yu

Abstract

This paper describes architecture of an universal graphical user interface (GUI) for any theorem prover. The implemented solution is tested on a GUI for description logics. The solution is implemented using Eclipse development tool. It contains a set of plug-ins for tracking the process of reasoning based on tableau.

1 Introduction

Main disadvantage of GUI for DLs reasoners (1), (2), (3) is a lack of graphical representation of reasoning process . We came on idea to make GUI for reasoning process that is extensible and flexible. It would allow other developers to make their own representations of that reasoning process. That GUI should make learning of reasoning based on description logics easer. We found that Eclipse is tool that meets all the requirements.

2 The GUI Architecture

Eclipse (www.eclipse.org) is an extensible platform based on plug-in architecture (4). The basic mechanism of extensibility is adding new plug-ins. Process of extending Eclipse is very simple. All that should be done is to copy the plug-in jar file into the folder *plugins*. Plug-ins can be related to each other through dependencies or extensions. Plug-in defines extension points so other plug-ins can extend or modify its behavior (4). Plug-in can define new views for representation or manipulation of information. Views usually depend on or interact with other views (5).

Eclipse 3.0 introduced Rich Client Platform (RCP) (8). RCP contains the minimal set of plug-ins needed to build a

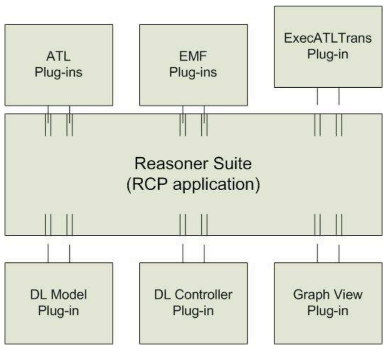

Figure 1: Architecture of a GUI for Theorem Provers

platform application with a user interface (6). Our solution is an RCP application based on three separate plug-ins implemented using MVC design pattern (7). The idea is that users can change any of plug-ins or add their own to extend application's functionality. Because of this advantages the GUI may be used with any theorem prover.

Fig. 1 describes the architecture of the Reasoner Suite application. Additional plug-in sets ATL (Atlas Transformation Language) (9) and EMF (Eclipse Modeling Framework) (10) are used by this solution.

DL Model plug-in (Fig. 1) loads tableau model into EMF repository using EMF framework. It is intended to keep model

and information about achieved on screen representation of reasoning process.
Graph View plug-in (Fig. 1) is a graph presentation of some reasoning process. A view has methods for adding and removing nodes and relations. Since view is intended to be reusable, the logic for graph modification is placed in the separate class.
DL Controller plug-in (Fig. 1) contains listeners that are answering on users commands. When user want to go to the next or previous step, registered listeners perform model modification without affecting the graph. Model modification automatically notifies all registered views so they can update themselves. The views are responsible for updating themselves. Depending on process direction Graph View plug-in needs to add or delete node from the graph. Each node represents one concept instance. Node has name of the instance, connections to other nodes (optionally) and a set of related concepts.
ExecATLTrans plug-in's (Fig. 1) function is executing ATL transformation that generates tableau model from model based on DL metamodel during reasoning process [Krdzavac07]. This function can execute other transformations by changing the source and target metamodel and ATL transformation file path.

3 Implementation Details

DL Model plug-in (Fig. 1) is implemented using Eclipse modeling framework (10). During initialization the model is read from an Ecore file into EMF repository. EMF has built-in support for model change notification and reflective API for manipulating EMF objects. The base interface of every EMF class is EObject. Every EObject can maintain a list of observers. Model and each part of it are of type EObject and observers can be attached separately. In DL Model there is a reasoning process analyzes indicator. It contains reference on last displayed concept instance. Graph View is registered as observer on analyzes indicator. Any other views can register themselves on it affecting that analyzing control (start, next, previous, end) can be done from more views.

4 Conclusion and Future Work

The paper described architecture of a GUI for any theorem prover (reasoner) based on Reach Client Platform (RCP). Future research will include implementation a plug-in for comparing performance of a few representative DLs reasoner.

References

I. Horrocks, *The FACT System*, In. H. de Swart, editor, Automated Reasoning with Analytic Tableaux and Related Methods, In Proceedings of the International Conference on Tableaux98, No. 1397, in Lecture Notes in Artificial Intelligence, Springer-Verlag, 307–312, 1998.

V. Haarslev, R. Mller, *Description of the RACER System and its Applications*, In Proceedings of the International Workshop on Description Logics (DL-2001), pp.131-141, Stanford, USA, 2001.

E. Sirin, B. Parsia, *Pellet: An OWL DL Reasoner*, In Proceedings of the International Workshop on Description Logics(DL2004) ,Whistler, British Columbia, Canada, June 6-8, 2004.

A. Bolour, *Notes on the Eclipse Plug-in Architecture*,July 3, 2003.
Online.[Avaliable]: http://www.eclipse.org/articles/Article-Plug-in-architecture/plugin_architecture.html

C. Pandit, *Make your Eclipse applications richer with view linking*, November 15, 2005.
Online.[Avaliable]: http://www-128.ibm.com/developerworks/opensource/library/os-ecllink/

Rich Client Platform, Retrieved March, 2007, from:
http://wiki.eclipse.org/index.php/Rich_Client_Platform

C. Moock, *Essential ActionScript 2.0*, chapter 18, O'Reilly, June 2004.

J. McAffer ,J.M. Lemieux, *Eclipse Rich Client Platform: Designing, Coding, and Packaging Java Applications* , Addison-Wesley Professional, October 2005.

F. Jouault, I. Kurtev, Transforming Models with ATL, *In Proceedings of the Model Transformations in Practice Workshop at MoDELS*, Montego Bay, Jamaica, 2005. [Online].Available:
http://sosym.dcs.kcl.ac.uk/events/mtip/submissions/jouault_kurtev_transforming_models_ with_atl.pdf

Eclipse Modeling Framework, Retrieved March, 2007, from: http://www.eclipse.org/modeling/emf/?project=emf

N. Krdžavac, D. Djurić, V. Devedžić, *MDA-Based Architecture of a Description Logics Reasoner*, In Proceedings of 3rd IFIP Conference on Atrificial Inteligence and Innovations (AIAI-2006), June 7-9, Greece, 98-105, Athens, 2006.