From Plain Prolog to Logtalk Objects: Effective Code Encapsulation and Reuse

Paulo Moura

Dep. of Computer Science, University of Beira Interior, Portugal
Center for Research in Advanced Computing Systems, INESC–Porto, Portugal
pmoura@di.ubi.pt

Abstract. Prolog affords concise, elegant, and clean solutions for many interesting problems, but is not immune to the software engineering challenges of large-scale application development. Code modularization, using modules or objects, is a key feature to keep projects manageable. Since most literature, instruction, and practice focus exclusively on object-oriented languages derived from imperative languages, objects are perceived as alien to logic programming while modules are considered a natural fit. Logtalk is an object-oriented logic programming language that can use most Prolog implementations as a back-end compiler. Logtalk objects are about code encapsulation and reuse, providing an alternative to Prolog module systems, and enabling natural solutions for a wide range of problems that would be awkward to solve using modules. This talk presents the Logtalk design goals, followed by a tutorial on Logtalk programming and some application examples. The talk ends with a discussion on the problems and benefits of developing Logtalk as a portable Prolog application.

The slides used in this talk are available at
http://logtalk.org/papers/ iclp2009/logtalk_iclp2009.pdf

About the speaker. Paulo Moura, PhD, is a Researcher at the Center for Research in Advanced Computing Systems (CRACS), INESC Porto, Portugal, and an Assistant Professor at the University of Beira Interior. His research interests include design and implementation of logic programming systems, multi-paradigm programming languages, and declarative object-oriented programming. His best-known work is the Logtalk object-oriented logic programming language. He prefers coding to writing papers, makes quixotic attempts to move Prolog standardization forward, and stalks innocent Prolog implementers with bug reports.

Acknowledgements. A word of gratitude to the Logtalk users for their suggestions, words of encouragement, bug reports, and other contributions. A special thanks to the Prolog implementers for their groundwork on developing and continually improving their compilers, in particular those working on open source projects. This work is partially supported by the FCT research project MOGGY (PTDC/EIA/70830/2006).

P.M. Hill and D.S. Warren (Eds.): ICLP 2009, LNCS 5649, p. 23, 2009.
© Springer-Verlag Berlin Heidelberg 2009