

Accepted Manuscript

JPC: A library for categorising and applying inter-language conversions between Java and Prolog

Sergio Castro, Kim Mens, Paulo Moura

PII: S0167-6423(15)00404-9
DOI: <http://dx.doi.org/10.1016/j.scico.2015.11.008>
Reference: SCICO 1965

To appear in: *Science of Computer Programming*

Received date: 12 March 2014
Revised date: 16 November 2015
Accepted date: 17 November 2015

Please cite this article in press as: S. Castro et al., JPC: A library for categorising and applying inter-language conversions between Java and Prolog, *Sci. Comput. Program.* (2015), <http://dx.doi.org/10.1016/j.scico.2015.11.008>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



JPC: A Library for Categorising and Applying Inter-Language Conversions Between Java and Prolog

Sergio Castro^a, Kim Mens^a, Paulo Moura^b

^a*Université catholique de Louvain, ICTEAM, Belgium*

^b*Center for Research in Advanced Computing Systems, INESC-TEC, Portugal*

Abstract

The number of approaches existing to enable a smooth interaction between Java and Prolog programs testifies the growing interest in solutions that combine the strengths of both languages. Most of these approaches provide limited support to allow programmers to customise how Prolog artefacts should be reified in the Java world, or how to reason about Java objects on the Prolog side. This is an error-prone task since often a considerable amount of mappings must be developed and organised. Furthermore, appropriate mappings may depend on the particular context in which a conversion is accomplished. Although some libraries alleviate this problem by providing higher-level abstractions to deal with the complexity of custom conversions between artefacts of the two languages, such libraries are difficult to implement and evolve, because of a lack of appropriate underlying building blocks for encapsulating, categorising and applying Java-Prolog conversion routines. We therefore introduce a new library, *JPC*, serving as a development tool for both programmers willing to categorise context-dependent conversion constructs in their Java-Prolog systems, and for architects implementing frameworks providing higher-level abstractions for better interoperability between these two languages.

Keywords: Object-Oriented Programming, Logic Programming, Multi-Paradigm Programming, Programming Language Interoperability, Separation of Concerns, Java, Prolog

1. Introduction

Languages with a large market share benefit from healthy ecosystems featuring high quality libraries and development tools. A good example is JAVA, with the added benefit of a small number of language implementations. The absence of fragmentation simplifies development efforts and helps ensuring tool and library portability. Languages with a much smaller market share often lack the resources and number of users to develop, or justify developing, a comprehensive set of libraries for common tasks. Instead, the scarce resources

Email address: kim.mens@uclouvain.be (Kim Mens)

Download English Version:

<https://daneshyari.com/en/article/4951888>

Download Persian Version:

<https://daneshyari.com/article/4951888>

[Daneshyari.com](https://daneshyari.com)