

Agile and Efficient Domain-Specific Languages using Multi-stage Programming in Java Mint

GPCE'10 Tutorial Proposal

Mathias Ricken

Rice University
mgricken@cs.rice.edu

Edwin Westbrook

Rice University
emw4@cs.rice.edu

Walid Taha

Halmstad University
walid.taha@hh.se

1. Title

Agile and Efficient Domain-Specific Languages using Multi-stage Programming in Java Mint

2. Abstract

Domain-specific languages (DSLs) are a powerful productivity tool because they allow domain experts, who are not necessarily programming experts, to quickly develop programs. DSL implementations have unique constraints for programming languages because they must be efficient, in order to ensure high productivity, but they must also be agile, in order to meet the rapidly changing demands of their domains. In this tutorial we show how multi-stage programming (MSP) can be used to build staged interpreters, which combine the agility of interpreters with the efficiency of compilers. The tutorial is conducted in Java Mint, a multi-stage Java based on recent work incorporating MSP into imperative object-oriented languages. In the first half of the tutorial, we introduce MSP by demonstrating how to write a staged interpreter for a number of basic language constructs, such as recursive functions, conditionals, and let expressions. In the second half, we extend our staged interpreter to take advantage of several well-known compiler optimizations, including type inference, constant folding, and static parallel loop scheduling. We highlight the opportunities afforded by using MSP with object-oriented design to quickly create efficient DSL implementations.

3. Names of Presenters

Mathias Ricken and Edwin Westbrook

4. Short Biographies of the Presenters

Mathias Ricken is a doctoral candidate in the Programming Languages Team at Rice University and one of the principal developers of the DrJava integrated development environment. His research interests include concurrent programming, extending the Java language, and computer science education. He is the developer of the Concutest concurrent unit testing framework and has created various experimen-

tal extensions of Java to address, for instance, programming with meta-data. Currently, Mathias is contributing to Java Mint, a multi-stage extension of Java that allows safe and expressive statically typed program generation and specialization in an imperative language setting.

Edwin Westbrook is a post-doctoral researcher at Rice University. His primary interests are in developing techniques for implementing and verifying properties of domain-specific languages (DSLs). He has worked on a number of projects in this area, including: Cinic, a type theory for building machine-checked proofs of properties of DSLs using a new approach to higher-order abstract syntax; Java Mint, a multi-stage version of Java used for efficient implementations of DSLs; and Acumen, a DSL for designing cyber-physical systems.

5. Brief Outline of Tutorial

This tutorial uses Java Mint to introduce MSP in imperative, object-oriented languages. We will progress as follows:

1. Introduction to MSP, the three language constructs called “bracket”, “escape” and “run”, and their use in Java Mint.
2. Brief overview of MSP applications other than interpreters.
3. Development of a staged interpreter for a small DSL. We incrementally add:
 - (a) Arithmetic expressions
 - (b) Conditionals
 - (c) Recursive functions
 - (d) Let expressions
4. Application of compiler optimizations in the interpreter for our DSL. We discuss:
 - (a) Extending the interpreter with multiple data types
 - (b) Type inference
 - (c) Constant folding
 - (d) Static parallel loop scheduling

6. Relevance for GPCE/SLE Audience

We feel this tutorial is relevant for GPCE/SLE for two reasons. First, the tutorial discusses implementation techniques for DSLs, which traditionally have been a topic of interest to both GPCE and SLE. DSLs provide a powerful software engineering/component engineering technique because they equip domain experts with the ability to write programs quickly and effectively. This in turn allows a greater modularity of software systems by distributing code production between domain experts and DSL implementors.

As a second reason, this tutorial is especially relevant to GPCE because it discusses a new MSP language, Java Mint. MSP is a form of generative programming that has been an important topic at many previous GPCE conferences. This tutorial will serve to disseminate cutting-edge research in combining MSP with imperative languages [1, 2]. We will also make MSP accessible to a wider audience than in the past by showing how it can be used in a mainstream object-oriented language such as Java.

References

- [1] Rice PLT. Mint Multi-stage Java Compiler. Available at <http://www.javamint.org>.
- [2] Edwin Westbrook, Mathias Ricken, Jun Inoue, Yilong Yao, Tamer Abdelatif, and Walid Taha. Mint: Java multi-stage programming using weak separability. In *PLDI '10: Proceedings of the 2010 ACM SIGPLAN conference on Programming language design and implementation*, pages 400–411, New York, NY, USA, 2010. ACM.