

# REFLEXIVE METAPROGRAMMING IN RUBY

## TUTORIAL PRESENTATION

H. Conrad Cunningham  
Department of Computer and Information Science  
University of Mississippi  
University, MS 38677  
(662) 915-5358  
[cunningham@cs.olemiss.edu](mailto:cunningham@cs.olemiss.edu)

## ABSTRACT

Ruby is an interpreted, dynamically typed, object-oriented application programming language [10]. It has been in existence for more than a decade, but in the past three years interest in the Ruby language and the programming styles it enables [2] has exploded in the practitioner community [8, 9]. Much of the explosive growth in interest has been because of the advent of the Ruby on Rails Web application framework [7, 11].

Ruby on Rails is essentially a domain-specific language (DSL) [1, 6] for web applications. It is implemented as an internal (or embedded) DSL [4, 5] using Ruby's flexible syntax and extensive reflexive metaprogramming facilities [3]. Reflexive metaprogramming is the capability of a program to both inspect and change its own program structures [12]. It has long been a staple of languages such as Smalltalk and Lisp, but the recent interest in Ruby has renewed interest in metaprogramming and in techniques like internal DSLs.

This tutorial does not concern itself with Ruby on Rails, but, instead, focuses on the Ruby features that make Rails possible. It examines Ruby with an emphasis on the language's distinctive reflexive metaprogramming facilities and techniques. These include the ability to query a class to determine what its methods, instance variables, and superclasses are—facilities that exist in mainstream languages like Java. Ruby also includes more exotic seeming facilities such as the ability to intercept calls to missing methods, to define new classes and methods dynamically, and to evaluate strings as Ruby code. The tutorial uses internal DSLs to motivate the utility of these language features.

## PRESENTER

H. Conrad Cunningham is Professor and Chair of Computer and Information Science at the University of Mississippi (Ole Miss). His professional interests include software architecture, programming methodology, programming languages, and concurrent and distributed computing. He has a BS degree in mathematics from Arkansas State University and MS and DSc degrees in computer science from Washington University in St. Louis. Cunningham has taught courses on software engineering, software architecture, software components, object-oriented programming, and functional programming. In the Fall 2006 semester, he taught a graduate course that included considerable content on Ruby programming, including its functional programming and reflexive metaprogramming features, and on domain-specific languages.

## REFERENCES

- [1] Bentley, J., Programming pearls: Little languages, *Communications of the ACM*, 29 (8), 711-721, 1986.
- [2] Carlson, L., Richardson, L., *Ruby Cookbook*, O'Reilly, 2006
- [3] Freeze, J., Creating DSLs with Ruby, *Artima Developer*, [http://www.artima.com/rubycs/articles/ruby\\_as\\_dsl.html](http://www.artima.com/rubycs/articles/ruby_as_dsl.html), March 2006.
- [4] Fowler, M., Language workbenches: The killer-app for domain specific languages? <http://www.martinfowler.com/articles/languageWorkbench.html>, June 2005.
- [5] Fowler, M., Generating code for DSLs, <http://www.martinfowler.com/articles/codeGenDsl.html>, June 2005.
- [6] Mernik, M., Heering, J., Sloane, A. M., When and how to develop domain-specific languages, *ACM Computing Surveys*, 37 (4), 316-344, 2005.
- [7] Ruby on Rails Project, <http://www.rubyonrails.org>, accessed December 2, 2006.

- [8] Tate, B., *Beyond Java*, O'Reilly, 2005.
- [9] Tate, B., *From Java to Ruby: Things Every Manager Should Know*, Pragmatic Bookshelf, 2006.
- [10] Thomas, D., Fowler, C., Hunt, A., *Programming Ruby: The Pragmatic Programmers' Guide*, Second Edition, Pragmatic Bookshelf, 2005.
- [11] Thomas, D., Heinemeier Hansson, D., *Agile Web Development with Rails*, Second Edition, Pragmatic Bookshelf, 2006.
- [12] Wikipedia: The Free Encyclopedia, Metaprogramming,  
<http://en.wikipedia.org/wiki/Metaprogramming>, accessed December 2, 2006.