

Fowler's Reader DSL

H. Conrad Cunningham

16 April 2022

Contents

Fowler's Reader DSLs	1
Background on DSLs	1
Source Code for DSLs	1
Ruby Shared Modules	2
Ruby Direct Configuration and Testing of Reader	2
Ruby Single-Pass External Text DSL	2
Ruby Two-Pass External XML DSL	2
Ruby Internal DSL	2
Acknowledgements	2
References	3

Copyright (C) 2009-2022, H. Conrad Cunningham
Professor of Computer and Information Science
University of Mississippi
214 Weir Hall
P.O. Box 1848
University, MS 38677
(662) 915-7396 (dept. office)

Browser Advisory: The HTML version of this textbook requires a browser that supports the display of MathML. A good choice as of April 2022 is a recent version of Firefox from Mozilla.

Fowler's Reader DSLs

Background on DSLs

The Reader DSLs case study is based on the running example in the following June 2005 Blog posts by Martin Fowler:

- Language Workbenches: The Killer-App for Domain Specific Languages?
[2]

- Generating Code for DSLs [3]

Source Code for DSLs

I originally developed the Reader DSLs in Fall 2006 for my offering of a graduate special topics class Ruby and Software Development by referring to the descriptions and C# code in Fowler's blog posts above. I redesigned the DSL in Ruby [5,6].

TODO: Test and revise Ruby code as needed for 2022 releases of Ruby.

Ruby Shared Modules

- DSL Reader Framework module (ReaderFramework.rb)
- DSL Reader Utilities mix-in module (ReaderUtilities.rb)
- Data input file (fowlerdata.txt)
- Text DSL description (dslinput.txt)
- XML DSL description (dslinput.xml)

Ruby Direct Configuration and Testing of Reader

- BuilderDirect.rb

Ruby Single-Pass External Text DSL

- TextSinglePass.rb

Ruby Two-Pass External XML DSL

- TwoPass.rb
- class BuilderExternal source code generated by TwoPass.rb

Ruby Internal DSL

- RubyDSL.rb

Acknowledgements

I originally developed the Reader DSLs in Fall 2006 for my offering of a graduate special topics course on Ruby and Software Development. I referred to the descriptions and C# code in two of Martin Fowler's Blog posts from June 2005:

- Language Workbenches: The Killer-App for Domain Specific Languages? [2]
- Generating Code for DSLs [3]

My class was focusing on Ruby [5] that semester, so I redesigned the Reader DSLs to use Ruby. That semester was my use or teaching of Ruby and my first practical experience with DSLs. In addition to Fowler’s Blog posts, I consulted the classic “PickAxe book” on Ruby programming [6] and Ruby-related articles by Freeze [4], Buck [1], and others.

I sometimes used the Reader DSL as an example in my Software Language Engineering (CSci 658) course offerings in the 2009-18 period, but those offerings used Scala, Lua, or Python instead of Ruby.

I retired from the full-time faculty in May 2019. As one of my post-retirement projects, I am continuing work on possible textbooks based on the course materials I had developed during my three decades as a faculty member. In January 2022, I began refining the existing content, integrating separately developed materials together, reformatting the documents, constructing a unified bibliography (e.g., using citeproc), and improving my build workflow and use of Pandoc. I adapted this index page from a portion of my Spring 2018 course notes.

The Ruby code for these DSLs have not yet been updated for modern Ruby circa 2022.

I maintain this chapter as text in Pandoc’s dialect of Markdown using embedded LaTeX markup for the mathematical formulas and then translate the document to HTML, PDF, and other forms as needed.

References

- [1] Jamis Buck. 2006. Writing domain specific languages (blog post). Retrieved from <https://weblog.jamisbuck.org/2006/4/20/writing-domain-specific-languages.html>
- [2] Martin Fowler. 2005. Language workbenches: The killer-app for domain specific languages? (Blog post). Retrieved from <http://www.martinfowler.com/articles/languageWorkbench.html>
- [3] Martin Fowler. 2005. Generating code for DSLs (blog post). Retrieved from <https://www.martinfowler.com/articles/codeGenDsl.html>
- [4] Jim Freeze. 2006. Creating DSLs with Ruby. Retrieved from <https://www.artima.com/articles/creating-dsls-with-ruby>
- [5] Ruby Community. 2022. Ruby: A programmer’s best friend. Retrieved from <https://www.ruby-lang.org>
- [6] David Thomas, Chad Fowler, and Andrew Hunt. 2004. *Programming Ruby* (Second ed.). Pragmatic Bookshelf, Raleigh, North Carolina, USA.