# Notes on Domain-Specific Languages: Chapter Index

**H. Conrad Cunningham**

**27 April 2022**

## Contents

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.

# Notes on Domain-Specific Languages

## Introduction to DSLs

The chapter *Introduction to DSLs* [3] motivates and defines the general domain-specific language concepts and techniques.

- as HTML
- as PDF

## Reader DSLs

Fowler's Reader DSLs include both external and internal Ruby-based DSLS adapted from Fowler [4,5].

- as HTML
- as PDF

## State Machine DSLs

The State Machine (Secret Panel) DSLs are Scala-based external DSLs adapted from Fowler [7].

- as HTML
- as PDF

## Computer Configuration DSLs

The Computer Configuration DSLs are Scala-based internal DSLs adapted from Fowler [7].

- as HTML
- as PDF

## Email Message-Building DSL

The Email Message Building DSL is a Scala-based internal DSL adapted from Fowler [7].

- as HTML
- as PDF

## Lair DSLs

The Lair Configuration DSLs are mostly internal DSLS adapted from Fowler [6]. One is an external DSL. All are in Lua; some are also in Python.

- as HTML
- as PDF

## Survey DSL

The Survey DSL is a Ruby-based internal DSL that I developed in my research [2]. Thee idea was partly motivated by a sidebar in Bentley [1].

- as HTML
- as PDF

## Sandwich DSL Projects

### Haskell

I developed the Haskell-based Sandwich DSL as scaffolding for a programming project in a Haskell-based course. It is now included as a section in Chapter 21 of *Exploring Languages with Interpreters and Functional Programming.*

- as HTML
- as PDF

### Scala

I subsequently developed the similar Scala-based Sandwich DSL for the same purpose in a Scala-based class.

- as HTML
- as PDF

## Exam DSL Project

I developed the Haskell-based Exam DSL as scaffolding for a programming project in a Haskell-based course. It is now included as a section in Chapter 21 of *Exploring Languages with Interpreters and Functional Programming.*

- as HTML
- as PDF

## Expression Tree Calculator

- as HTML
- as PDF

## Exploring Languages with Interpreters

During the 2013-18 period, I partially developed three different sets of interpreters. I developed the first in Lua in 2013 (KILT), but I was not satsisfied with the modularization and some of the structures I used. So I rewrote it in Lua in 2016 for the CSci 450 class. Actual use of the Lua interpreter for class was difficult, partly because of the dyanmic typing of Lua. So I began a third set using the statically typed Haskell language, which also had good algebraic data types and pattern matching.

- Fall 2017-18 ELI Calculator language interpreter includes the following source code modules:
  - *REPL* modules
    - Prefix REPL `PrefixCalcREPL`
    - Infix REPL `InfixCalcREPL`
  - *Parser* modules
    - Prefix parser `ParsePrefixCalc`
    - Infix parser `ParseInfixCalc`
  - *Lexical Analyzer* module `LexCalc`
  - *Abstract Synax* module `AbSynCalc`
  - *Evaluator* module `EvalCalc`
  - *Environments* module `Environments`
  - *Values* module `Values`
  - Skeleton simplify and derivative module `Process AST`
- Fall 2017-18 ELI ImpCore interpreter modules (prefix syntax) code mostly works but needs a bit of update to match recent changes to ELI Calculator:
  - REPL module
  - Recursive descent parser module
  - Lexical analyzer module
  - Abstract Syntax module
  - Evaluator module
  - Environments module
  - Values module
  - Test Imp Core(in work, not current)
- Fall 2016 Lua Expression Language 1 interpreter folder
- Fall 2016 Lua Imperative Core interpreter folder
- Fall 2013 Kamin Interpreter in Lua Toolset (KILT) folder
- Kamin-Budd Interpreters original source code folder

## Acknowledgements

See the Acknowledgements section for the "Introduction to DSLs" chapter for information about the overall development of this set of notes. See the corresponding sections of the DSL chapters for information about each DSL case study.

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 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] Jon Bentley. 1986. Programming pearls: Little languages. *Communications of the ACM* 29, 8 (1986), 711–721.

[2] H. Conrad Cunningham. 2008. A little language for surveys: Constructing an internal DSL in Ruby. In *Proceedings of the ACM SouthEast conference*, Auburn, Alabama, USA, 282–287.

[3] H. Conrad Cunningham. 2022. *Notes on domain-specific languages.* University of Mississippi, Department of Computer and Information Science, University, Mississippi, USA. Retrieved from https://john.cs.olemiss.edu/~hcc/docs/DSLs/NotesDSLs.html

[4] Martin Fowler. 2005. Language workbenches: The killer-app for domain specific languages? (Blog post). Retrieved from http://www.martinfowler.com/articles/languageWorkbench.html

[5] Martin Fowler. 2005. Generating code for DSLs (blog post). Retrieved from https://www.martinfowler.com/articles/codeGenDsl.html

[6] Martin Fowler. 2008. One lair and twenty Ruby DSLs. In *The ThoughtWorks anthology: Essays on software technology and innovation*, ThoughtWorks, Inc. (ed.). Pragmatic Bookshelf, Raleigh, North Carolina, USA. Retrieved from https://media.pragprog.com/titles/twa/martin_fowler.pdf

[7] Martin Fowler and Rebecca Parsons. 2010. *Domain specific languages.* Addison-Wesley, Boston, Massachusetts, USA.