

# Fowler's Computer Configuration DSL

H. Conrad Cunningham

16 April 2022

## Contents

<b>Fowler's Computer Configuration DSLs</b>	<b>1</b>
Background . . . . .	1
Source Code . . . . .	2
Scala versions (2009) . . . . .	3
Acknowledgements . . . . .	3
References . . . . .	3

Copyright (C) 2009, 2018, 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 Computer Configuration DSLs

### Background

The Computer Configuration DSLs are internal DSLs based on examples given in Chapter 35 (Method Chaining) and Chapter 38 (Nested Closures) of Martin Fowler's book *Domain-Specific Languages* [1].

Fowler's book defines a pattern language for developing domain-specific languages. His DSL Patterns Catalog is online at <http://martinfowler.com/dslCatalog/>.

Below is an example expressed as a method-chaining DSL program. This example uses Scala's parameterless methods instead of the Java's zero-argument methods that Fowler uses in Chapter 35 [[1].

```
computer
  .processor
    .cores(2)
    .i386
  .disk
    .size(150)
  .disk
    .size(75)
    .speed(7200)
  .sata
  .end
```

Below is the same example expressed as a nested-closures DSL program. This example uses the Scala syntax instead of the Ruby syntax Fowler uses in Chapter 38 [1].

```
computer {
  processor {
    cores(2)
    i386
    processorSpeed(2.2)
  }
  disk {
    size(150)
  }
  disk {
    size(75)
    diskSpeed(7200)
    sata
  }
}
```

## Source Code

I originally developed this DSL in Spring 2009 by referring to an in-work version of Fowler's book on his website, so my examples may not precisely follow the book as published in late 2010.

Fowler used Java for the Chapter 35 method-chaining version of the DSL and Ruby for the Chapter 38 nested-closures version [1]. I reimplemented both using Scala 2 [2,4]. The overall implementation in Scala can probably be redesigned to take better advantage of Scala features.

I had the Java and Ruby code for the DSL processing, but I did not have the code for Fowler's semantic models. I guessed at what was intended in the semantic model design and built a Scala semantic model that is used for both the method-chaining and nested-closures implementations.

In addition to the Method Chaining DSL pattern, the Scala implementation derived from Chapter 35, like Fowler’s Java implementation, uses the Expression Builder and Context Variable DSL patterns [1].

In addition to the Nested Closures DSL pattern, the Scala implementation derived from Chapter 38, like Fowler’s Ruby implementation, uses the Expression Builder, Context Variable, and Object Scoping DSL patterns [1].

I updated the Scala programs in 2018 to make sure they would compile and execute under the then-current Scala release. However, I have not updated the code yet for Scala 3 [3,4].

TODO: Update the Scala code to be compatible with Scala 3 without deprecation warnings.

### Scala versions (2009)

- Scala semantic model (shared)
- Scala internal DSL using Method Chaining
- Scala internal DSL using Nested Closures and Object Scoping
- CompConfig.sh test script

### Acknowledgements

I developed the Computer Configuration DSL case study for my prototype offering of the graduate Software Language Engineering class (eventually CSci 658) in Spring 2009. I based it on examples in Chapter 35 (Method Chaining) and Chapter 38 of Martin Fowler’s book *Domain Specific Languages* [1].

I originally developed this DSL in Spring 2009 by referring to an in-work version of Fowler’s book on his website, so my examples may not precisely follow the book as published in late 2010.

Fowler’s code is written in Java and Ruby. I used Scala 2 [2,4]. In 2018, I updated the Scala programs to make sure they would compile and execute under the then-current Scala 2 release.

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 Scala code for these DSLs have not yet been updated for Scala 3 [3,4].

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] Martin Fowler and Rebecca Parsons. 2010. *Domain specific languages*. Addison-Wesley, Boston, Massachusetts, USA.
- [2] Martin Odersky, Lex Spoon, and Bill Venners. 2008. *Programming in Scala* (First ed.). Artima, Inc., Walnut Creek, California, USA.
- [3] Martin Odersky, Lex Spoon, and Bill Venners. 2021. *Programming in Scala* (Fifth ed.). Artima, Inc., Walnut Creek, California, USA.
- [4] Scala Language Organization. 2022. The Scala programming language. Retrieved from <https://www.scala-lang.org/>