Fowler's Email DSL

H. Conrad Cunningham

16 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.

Fowler's Email DSL

Background

The Email case study is based on the Email Message Builder DSL given in Chapter 35 (Method Chaining) 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/.

This DSL uses Fowler's Expression Builder and Method Chaining patterns [1] with what he calls a *progressive interface*.

This case study presents a simple example DSL for constructing email messages. This internal DSL uses the progressive interface to ensure that the elements of the message are only added in a particular order: first the to's, then the cc's, then the subject, and finally the body.

Here is an example usage of the DSL given by Fowler:

```
message = MessageBuilder.build
          .to("fowler@acm.org")
          .cc("editor@publisher.com")
          .subject("Error in book")
          .body("Sally Shipton should read Sally Sparrow")
```

At least one "to" field and the "body" must be given. However, the "cc" and "subject" fields are optional. There may be multiple occurrences of "to" and "cc".

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's code is written in C#. I reimplemented Fowler's DSL in Scala 2 [2,4], seeking to use the same approach as Fowler. I had the C# code for the DSL processing, but I did not have the code for the semantic model. So, I guessed at what was intended in some cases. The overall implementation in Scala can probably be redesigned to take better advantage of Scala features.

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].

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

Scala versions

- Scala internal DSL using Method Chaining and Progressive Interfaces
- test script `EmailProgressive.sh`

Acknowledgements

I developed the Email DSL case study for my prototype offering of the graduate Software Language Engineering class (eventually CSci 658) in Spring 2009. I based it on an example in Chapter 35 (Method Chaining) in 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 C#. 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/