

CS298 Proposal

JavaFX as a Domain-specific Language in Scala/Groovy

Sadiya Hameed
sadiya@gmail.com

Advisor:

Dr. Cay Horstmann (horstmann@cs.sjsu.edu)

Committee Members:

Dr. Jon Pearce, Department of Computer Science, SJSU (pearce@cs.sjsu.edu)

Dr. David Taylor, Department of Computer Science, SJSU (taylor@cs.sjsu.edu)

Introduction

Domain-specific programming languages (DSLs) are designed for a particular problem domain and promise substantial expressiveness and ease of use in their specialized area over general-purpose programming languages [3]. JavaFX is such a domain-specific language aimed at speedy development of rich-client Java user interfaces. The core of JavaFX is JavaFX Script, a declarative scripting language with a high degree of interactivity with Java classes [7].

The goal of the project is to examine the feasibility of mimicking the functionalities provided by JavaFX as a DSL in Groovy and Scala languages and to reason about the suitability of these languages as DSL hosts. Scala claims to provide a unique combination of language mechanisms that make it easy to smoothly add new language constructs and facilitate the definition of new statements [5]. Groovy has had practical success in providing DSLs for XML builders, ORM, etc. and its creators claim that it is particularly well suited for writing a DSL [6]. The plan is to create DSLs, simulating selected JavaFX features, in both languages to facilitate evaluation and comparison of their abilities to be DSL hosts.

CS297 Overview

During this first phase, the focus was on getting familiar with JavaFX and its features, Scala and Groovy languages, the provisions in those languages that facilitate addition of dynamic behavior, and simulating a single feature of JavaFX in the two languages. The feature selected for simulation was the “automatic data-binding” capability of JavaFX. The first phase ended successfully with prototypes created in both languages. The prototypes provided syntax very similar to the original JavaFX syntax for the bind feature.

Proposed Schedule

Timeline	Task
Week 1, 2 (1/24 – 2/06)	Prepare and submit the CS298 proposal. Binding with

	expressions in Scala and Groovy
Week 3 to 5 (2/07 – 2/27)	Simulate dur operator in Scala and Groovy
Week 6, 7 (2/28 – 3/12)	Simulate do/do later keywords in Scala and Groovy
Week 8, 9 (3/13 – 3/26)	Implement list comprehension features from JavaFX in Scala and Groovy
Week 10 to 13 (3/27 – 4/23)	Write up of CS298 report
Week 14, 15 (4/24 – 5/07)	Submit report to committee and prepare oral presentation

Deliverables

1. DSLs in Scala and Groovy with the following features
 - a. Incremental dependency-based evaluation (automatic data binding)
 - b. **dur** operator
 - c. **do/do later**
 - d. list comprehensions
2. A sample application, for comparison, in JavaFX and the same application in Scala and Groovy DSL.
3. A written report on the project, with an overview of JavaFX features implemented and Scala and Groovy features used for the implementation. The report will also detail the comparison of the results achieved with each of the two languages and evaluations of the individual results.

Innovations and Challenges:

- Innovatively utilizing the dynamic features of Scala and Groovy to provide the feature set of JavaFX within the existing languages.
- Working with developing languages with highly fluid feature sets and nascent Integrated Development Environments.

References

1. Odersky, M. And Zenger, M. 2005. Scalable component abstractions. In *Proceedings of the 20th Annual ACM SIGPLAN Conference on Object Oriented Programming, Systems, Languages, and Applications* (San Diego, CA, USA, October 16 - 20, 2005). OOPSLA '05. ACM, New York, NY, 41-57.
2. Henry, K. A crash overview of groovy. *Crossroads*, 12(3), ACM Press, May 2006.
3. Mernik, M., Heering, J., and Sloane, A. M. 2005. When and how to develop domain-specific languages. *ACM Computing Surveys (CSUR)* 37, 4 (Dec. 2005), 316-344.
4. van Deursen, A., Klint, P., and Visser, J. 2000. Domain-specific languages: an annotated bibliography. *SIGPLAN Not.* 35, 6 (Jun. 2000), 26-36.
5. Odersky et al. A Tour of the Scala Programming Language. Programming methods laboratory EPFL, May 2007. Available from <http://www.scalalang.org/docu/files/ScalaTour.pdf>
6. Writing domain specific languages. (Accessed November 2007) <http://groovy.codehaus.org>.
7. JavaFX Community. (Accessed January 2008) <https://openjfx.dev.java.net/index.html>