CS297 Proposal

JavaFX as a Domain-specific Language in Scala/Groovy

Sadiya Hameed
shameed@email.sjsu.edu

Advisor:
Dr. Cay Horstmann (email: horstmann@cs.sjsu.edu)

Description:
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. 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.

It seems unfortunate that JavaFX is yet another language. The modern trend is to provide DSLs inside a larger host language. 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 have been invented for just this purpose
"Scala provides a unique combination of language mechanisms that make it easy to smoothly add new language constructs in form of libraries:
- any method may be used as an infix or postfix operator, and
- closures are constructed automatically depending on the expected type (target typing).
A joint use of both features facilitates the definition of new statements without extending the syntax and without using macro-like meta-programming facilities." [1]

Groovy has had practical success in providing DSLs for XML builders, ORM, etc. and its builders claim that it is particularly well suited for writing a DSL. Groovy provides various features to let you easily embed DSLs in your Groovy code. e.g.
- you can also create your own control structures by passing closures as the last argument of a method call
- it is possible to add dynamic methods or properties (methods or properties which don't really exist but that can be intercepted and acted upon) by implementing GroovyObject or creating a custom MetaClass, etc. [3]

The plan is to create prototypes in both languages to facilitate evaluation and comparison of their abilities to be DSL hosts.

Proposed Schedule:

Timeline	Task
Week 1 (8/23 – 8/29)	Prepare and submit the proposal. Basic environment setup. Read on Java Swing and 2D.
Week 2 (8/30 – 9/05)	Research features of JavaFX (e.g. bind/trigger etc.)
Week 3 (9/06 – 9/12)	Explore how other languages implement bind/trigger features
Week 4 (9/13 – 9/19)	Evaluate tools available in Scala that might enable the language to be a domain-specific language's host
Week 5 (9/20 – 9/26)	Evaluate how successful Groovy's metaprogramming features would be in implementing a domain-specific language
Week 6 to 8 (9/27 – 10/17)	Develop a prototype in Scala
Week 9 to 11 (10/18 – 11/07)	Develop a prototype in Groovy
Week 12, 13 (11/08 – 11/21)	Comparison and evaluation of prototypes and initial work on the report draft
Week 14, 15 (11/22 – 12/05)	Finalize report and presentation to committee

Deliverables:

1. Prototypes (Scala and Groovy)
2. A written report giving an overview of the semester work. The report will detail the comparison of the results achieved with each of the two languages and evaluations of the individual results.

References:

1. Martin Odersky et al. A Tour of the Scala Programming Language. Programming methods laboratory EPFL, May 2007. Available online http://www.scala-lang.org/docu/files/ScalaTour.pdf
2. Kevin Henry. A crash overview of groovy. Crossroads, Volume 12 Issue 3, ACM Press, May 2006.
3. Writing domain specific languages. http://groovy.codehaus.org, 2006
4. Marjan Mernik, Jan Heering, Anthony M. Sloane. When and How to Develop Domain-Specific Languages. ACM Computing Surveys (CSUR), Volume 37 Issue 4, ACM Press, December 2005.
5. Arie van Deursen, Paul Klint, Joost Visser. Domain-Specific Languages: An Annotated Bibliography. ACM SIGPLAN Notices, June 2000.
6. JavaFX. Sun Microsystems. http://www.sun.com/software/javafx/index.jsp, 2007.