Carnegie Mellon University
Computer Science Department

THESIS PROPOSAL

Michael Coblenz

User-Centered Design of Principled Programming Languages

Friday, September 14, 2018 – 1:30 p.m. – GHC 6501

Programming languages are designed to facilitate software development by people. They are subject to two very different sets of requirements: first, the need to provide strong safety guarantees to protect against bugs; and second, the need for users to effectively and efficiently write software that meets their functional and quality requirements. This thesis argues that fusing formal methods for reasoning about programming languages with user-centered design methods is a practical approach to designing languages that make programmers more effective. The thesis is supported by two language design projects: Glacier is an extension to Java that enforces transitive class immutability; Obsidian is a new programming language for writing blockchain programs.

From two observations about typical blockchain applications, I derived two features that motivated the design of Obsidian. First, blockchain applications typically implement state machines, which support different operations in different states. Obsidian uses typestate, which lifts dynamic state into static types, to provide static guarantees regarding object state. Second, blockchain applications frequently manipulate resources, such as virtual currency. Obsidian provides a notion of ownership, which distinguishes one reference from all others. Obsidian supports resources via linear types, which ensure that owning references to resources are not accidentally lost.

The combination of resources and typestate results in a novel set of design problems for the type system; although each idea has been explored individually, combining them requires unifying different perspectives on linearity. Furthermore, no language with either one of these features has been designed in a user-centered way or evaluated with users. Typical typestate systems have a complex set of permissions that provides safety properties, but these systems focused on expressiveness rather than on usability. Obsidian integrates typestate and resources in a novel way, resulting in a new type system design with a focus on simplicity and usability while retaining the desired safety properties. I propose to integrate formal methods with user studies to show both that the language has provable safety properties and that users can use Obsidian more effectively than existing languages for common blockchain programming tasks.

Thesis Committee:
Jonathan Aldrich, Co-Chair
Brad A. Meyers, Co-Chair
Frank Pfenning
Joshua Sunshine
Gail Murphy, University of British Columbia

Thesis Summary: http://www.cs.cmu.edu/~mcoblenz/proposal.pdf