

The Moby Scheme Compiler for Smartphones

or, Is That a Parenthesis in Your Pocket?

Shriram Krishnamurthi

Computer Science Department
Brown University
sk@cs.brown.edu

For the past decade, PLT Scheme has been investigating reactive programming languages and libraries. This has resulted in innovative programming tools for synchronous and asynchronous Web applications, GUIs, and more.

Our most recent foray is the World library, which makes the traditional event loop functional. Whereas event loops usually invoke callbacks that return no interesting value, each World callback returns the changing state of the computation, which the event loop in turn passes to the next callback. As a result, programmers can perform seemingly-imperative reactive computations with pure functional expressions.

Now we are exploring World on mobile phones. Many mobile applications listen to event sources ranging from network traffic and location sensors to haptic sensors; process the resulting events; and effect the world through actuators. It is tempting to ask whether these applications, too, can be written as pure functional expressions in the World style.

Moby is an experimental compiler to evaluate this question. It consumes Scheme programs that use World, augmented with phone-specific libraries. It compiles them to J2ME, a standard for mobile platforms. It already seamlessly translates numerous functional animations. Compared to their J2ME equivalents, some of these programs exhibit a significant reduction in size and conceptual complexity.

Acknowledgments This is joint work with Danny Yoo, Matthias Felleisen, and Kathi Fisler. Thanks to Emmanuel Schanzer and Larry Finkelstein for suggestions and encouragement, and to Zhe Zhang for his programming contributions. This work is partially supported by NSF grants.

