

Javascript as an Intermediate Language for FRP

Subtitle Text, if any

Name1

Affiliation1

Email1

Categories and Subject Descriptors CR-number [subcategory]:
third-level

General Terms term1, term2

Keywords keyword1, keyword2

1. Introduction

Functional Reactive Programming (FRP) is an exciting and popular way to make interactive applications in a functional style. However, creating production level applications with FRP remains prohibitively difficult, especially for mobile platforms. While many functional programmers have the programming experience to write FRP applications, technical issues have kept this circle to only the most dedicated FRP researchers. Using Javascript as an intermediate language for FRP code eliminates most of these technical hurdles and makes publishing applications written with FRP an option for more users.

FRP exists as libraries in Haskell, the dedicated FRP language Elm, and Sodium is a project to port FRP to more language like Java and C. However, FRP often feels most natural in functional language like Haskell and Elm. Building a system that easily works very well in high-level functional languages, but Android is in Java. We want to write FRP programs for Android. Many FRP libraries exist in Haskell, and Elm is a dedicated FRP language that is very similar to Haskell.

Apps can be developed and maintained much faster using FRP. prototypes can be generated very quickly.

The development of FRP and functional languages in general relies on adoption by industry. Simplifying the work flow to bring FRP code to market will encourage future work in FRP.

Ivan Perez from Keera Studios has cross-compiled Haskell to Android and published a game. His approach is complicated and closed-source at the moment. Java has an FRP library, but Java is not well suited for FRP.

Compile FRP program to an intermediate language, then compile to Android. The intermediate language By using GHCJS, we can compile our Haskell FRP program to JS. Then using PhoneGap, we can compile that generated JS to an Android application. This is unique because it removes all the cross platform difficulties.

There are two kinds of APIs for mobile development, software and hardware. Software APIs (e.g. Google Maps) would require a .jar file loaded to the Android project - now we just use the JavaScript API for that library. Hardware APIs (e.g. acceleration data) use device specific APIs. Some JavaScript APIs exist for functionality most devices have, like accelerometers.

If native code is necessary, PhoneGap provides *plugins* as a foreign function interface (FFI) to JavaScript. Plugin code can be called via the FFI in JavaScript, which is in turn called through the source language FFI. Both GHCJS and Elm provide a JavaScript FFI.

Development may also be split between any combination of the FRP language, JavaScript, and the target language.

We developed a proof of concept app using Elm and have provided the source code and app files. This app uses FRP to read the touch interface native to the mobile hardware and control a game. The app also uses a plugin to show ads, mixing JavaScript code with Elm. It can run at ?? frames per second on an Nexus 5 running Android version X.XX. pass in (fps 200) as the time signal to a display of "toString (1000/t)" to measure actual FPS - capped at 200 on laptop.

A. Appendix Title

This is the text of the appendix, if you need one.

Acknowledgments

Acknowledgments, if needed.

References

[1] P. Q. Smith, and X. Y. Jones. ...reference text...

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CONF 'yy, Month d-d, 20yy, City, ST, Country.

Copyright © 20yy ACM 978-1-xxxx-xxxx-n/yy/mm...\$15.00.

<http://dx.doi.org/10.1145/nnnnnnn.nnnnnnn>