

Abstractionless Programming in App Inventor

Audrey Seo
Computer Science Department
Wellesley College
Wellesley, Massachusetts, U.S.
aseo@wellesley.edu

Abstract

App Inventor is a web-based blocks programming environment that allows people of all ages and experiences to build mobile apps, introducing new programmers to fundamental programming concepts such as variables, conditionals, iteration, and procedures. Past studies show that programmers do not use procedures, App Inventor’s key means of abstraction for avoiding duplicate code, very often, which suggests that App inventor programmers, instead of using procedures, merely duplicate their code. This research focused on detecting duplicated event handlers in two datasets of App Inventor code. I found that about 13% of App Inventor event handlers are duplicates. Surprisingly, over 49% of this code cannot be extracted into a procedure without using the rarely used blocks for manipulating generic components, which could be further preventing App Inventor programmers from using procedures as abstractions. Therefore, more work needs to be done to make programmers more aware of generics and how to use them.

Keywords App Inventor, procedure extraction, computational thinking, blocks programming, abstraction

1 Introduction

App Inventor is a web-based blocks programming language that allows users with little to no coding experience to quickly create and deploy mobile applications. In the in-browser App Inventor IDE, programmers drag and drop user interface and functional components onto a simulated phone screen, and then combine programming blocks to specify the behavior of the app. As of July 2018, App Inventor has over 7 million users who have created 30+ million apps.

Two data sets of App Inventor projects were obtained for this research. The first data set, referred to as the 10K data set, consists of all of the projects created by 10,003 randomly selected App Inventor programmers. In total, it contains 30,983 projects. The second data set, dubbed the 46K data set, was chosen based on studying so-called “prolific” App Inventor programmers: those with 20 or more programs each.

The 46K dataset contains 1,546,056 projects that were created by 46,320 programmers¹.

An App Inventor project, once downloaded, actually consists of several parts: JSON and XML files for each of the screens involved in the app. In order to streamline analysis, these two parts were combined and parsed back into JSON. However, several of the projects in each of the datasets failed to parse correctly, so the actual number of projects studied in this research was

- 30,851 of the 10K projects, and
- 1,545,284 of the 46K projects.

Last summer, it was discovered that procedures, a crucial abstraction mechanism in App Inventor, only 15% of random projects and 18% of the 46K projects used procedures. [2].

2 Detecting Opportunities for Procedure Extraction

My main goal was to determine if App Inventor programmers are missing opportunities for procedure extraction, i.e. capturing nearly identical code patterns with procedures, as in Fig. 1. I wrote a program that looked for nearly duplicate event handler bodies, each of which could be replaced by a call to a single new procedure with the appropriate arguments within a single screen of an app, that had at least five blocks to be consistent with other work [1]. I found that duplication is prevalent in both data sets, where 18.7% of the 10K programmers and 87.6% of the 46K programmers had at least one missed opportunity for procedure extraction. Some other results are shown in Fig. 2 and 3.

An unexpected result was that 55% of the 10K and 49% of the 46K duplicated code could not be extracted into a procedure without using so-called generic components and blocks. These are needed in situations where a procedure must abstract over a particular component (such as a label or button) rather than a simple value like a number or string. See Fig. 1. As Table 1 indicates, generic blocks are very rare, possibly because programmers do not know about them or find them hard to use. Furthermore, very few resources that demonstrate how generics are used exist, and in the App Inventor IDE, generic component and generic method, getter, and setter blocks are kept in entirely different menus, which does

BLOCKS+, November 04, 2018, Boston, MA, USA
2018. ACM ISBN ...\$15.00
<https://doi.org/>

¹129 of these programmers overlap with the 10K programmers.

