The Journal of Systems and Software 000 (2016) 1–14

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

New Trends and Ideas

An exploratory study on the usage of common interface elements in android applications

Seyyed Ehsan Salamati Taba[a], Iman Keivanloo[b], Ying Zou[b], Shaohua Wang[a,*]

[a] School of Computing, Queen's University, Kingston, Ontario, Canada
[b] Department of Electrical and Computer Engineering, Queen's University, Kingston, Ontario, Canada

ARTICLE INFO

Article history:
Received 15 May 2015
Revised 26 March 2016
Accepted 7 July 2016
Available online xxx

Keywords:
Mobile applications
Common UI elements
User-perceived quality

ABSTRACT

The number of mobile applications has increased drastically in the past few years. A recent study has shown that reusing source code is a common practice for Android application development. However, reuse in mobile applications is not necessarily limited to the source code (*i.e.,* program logic). User interface (UI) design plays a vital role in constructing the user-perceived quality of a mobile application. The user-perceived quality reflects the users' opinions of a product. For mobile applications, it can be quantified by the number of downloads and raters. In this study, we extract commonly used UI elements, denoted as Common Element Sets (CESs), from user interfaces of applications. Moreover, we highlight the characteristics of CESs that can result in a high user-perceived quality by proposing various metrics. Through an empirical study on 1292 mobile applications, we observe that (i) CESs of mobile applications widely occur among and across different categories; (ii) certain characteristics of CESs can provide a high user-perceived quality; and (iii) through a manual analysis, we recommend UI templates that are extracted and summarized from CESs for developers. Developers and quality assurance personnel can use our guidelines to improve the quality of mobile applications.

1. Introduction

Recently, mobile applications have become increasingly popular. The number of downloads of mobile applications on various platforms by the end of 2016 is predicted to be 211 billions (Number of Apps Downloaded, 2015), increased from 7 billion in 2009 (Number of downloads for android apps, 2014b). This large number of downloads implies that a new software industry is emerging. There are various potential reasons for the rapid growth in the number of downloads of available mobile applications. One of them could be the ease of building new applications for mobile platforms (Google's do-it-yourself app creation software, 2014). A more fundamental reason that can explain this rapid growth might be the use of proven software engineering practices, such as code reuse (Basili and Rombach, 1991). To develop a simple mobile application, there are two things that should be developed: i) User Interface (UI), and ii) business logic (i.e. source code) of the application. Ruiz et al. (2012) have shown that Java classes in mobile applications on the Android Market are reused significantly, which implies that many applications in the Android Market use very similar *logic*. As aforementioned, another important aspect of a mobile application is the user interface (*UI*).

Due to the limitations of mobile applications (*e.g.,* small screen size, network problems, and computational power (Chittaro, 2010)), developers should be more careful in designing their applications on mobile platforms than PCs (Karlson et al., 2010). Developers' negligence in the importance of UI designs is one of the major reasons for users to abandon a task on mobile applications and switch to PCs (Karlson et al., 2010). To design a UI for a mobile application, developers can adopt commonly used practices (i.e. commonly used sets of UI elements) or standard templates that are used by other developers. However, this question still remains whether the use of commonly used practices (or standard templates) can impact the user-perceived quality of a mobile application. A user-perceived quality can be defined as a user's opinion of a mobile application. The perceived quality can be quantified by the number of downloads and raters in mobile stores (Mojica Ruiz, 2013). According to the statistics, the user-perceived quality can be influenced by UI designs, stability and performance of applications (Android apps quality, 2014a).

In this paper, we are interested to know to what extent the use of commonly used practices (or standard templates) occurs in UI designs of mobile applications. To quantify the extent, we define a Common Element Set (CES), as a set of commonly used UI

* Corresponding author.
E-mail addresses: taba@cs.queensu.ca (S.E.S. Taba), iman.keivanloo@queensu.ca (I. Keivanloo), ying.zou@queensu.ca (Y. Zou), shaohua@cs.queensu.ca (S. Wang).

http://dx.doi.org/10.1016/j.jss.2016.07.010
0164-1212/

elements that appear in at least one other UI page. Moreover, we investigate whether there is a relation between using CESs and user-perceived quality of mobile applications. More specifically, we investigate whether using CESs within a category or across categories of the Android applications can impact the user-perceived quality. Furthermore, we go into more details, and investigate the characteristics of CESs related to the high user-perceived quality of the Android applications in each category. Through a manual analysis, we extract templates from the CESs, with the high user-perceived quality, extracted from the Shopping category for a number of functionalities. The UI templates are recurring solutions that solve common design problems. Recommending UI templates associated with the high user-perceive quality helps developers boost UI development for mobile applications. To the best of our knowledge, no one has focused on studying the usage of CESs and extracting UI templates with high user-perceived quality in mobile applications.

Using the user-perceived quality calculated from the statistics in the Android Market and the extracted CESs from 1292 free Android applications in 8 different categories of Google Play Android Market (i.e., Shopping, Health, Transportation, Travel, News, Weather, Finance, and Social), we address the following research questions:

RQ1) ***To what extent commonly used UI element sets occur?***

We show that CESs widely occur within each category (i.e. 60% of a UI XML file can be constructed by similar CESs that are repeated in other UI XML files). Moreover, on average, the CESs occur in 23% of mobile applications across different categories.

RQ2) ***Do the usage of commonly used UI element sets impact the user-perceived quality within and across categories?***

We observe that CESs in UI designs of mobile applications can make a significant difference in the user-perceived quality of mobile applications. More specifically, we observe that in specific application categories (e.g., Health), the perceived quality of mobile applications with CESs in their UI designs is significantly higher than the ones without CESs. Therefore, developers should take into consideration the application domain during the process of a UI design.

RQ3) ***Do the usage of commonly UI element sets have an impact on the user-perceived quality of functionalities in mobile applications?***

We observe that the usage of CESs has an impact on the user-perceived quality of functionalities in mobile applications. In almost every functionality, to achieve the high user-perceived quality, it is better to use the CESs that are used in a few number of applications, or used in more UI XML files. Additionally, for certain functionalities, it is desirable to use CES having developer-customized UI elements to achieve the high user-perceived quality.

RQ4) ***Can we extract UI templates from commonly used UI element sets with a high user-perceived quality?***

Through a manual analysis, we extracted a set of UI templates associated with the high user-perceived quality (according to the findings in our empirical study in RQ1 to RQ3). Our extracted UI templates are meaningful and helpful.

The remainder of this paper is organized as follows. First, we explain the architecture of Android applications in Section 2. We describe the experimental setup of our study in Section 3, and report our findings in Section 4. In Section 5, we discuss threats to the validity. We summarize the related literature in Section 6. Section 7 concludes our work and outlines avenues for future works.

2. Background

In this section, we briefly talk about the architecture of Android applications. Typically, Android applications are written in Java programming language using Android Software Development Kit (SDK). The Android SDK compiles the code into an Android PaKage (APK) file which is an archive file with a ".apk" extension. One APK file contains all the content of an Android application, and is the file that Android devices use to install the Android application.

There are four types of application components, including activities, services, content providers and broadcast receivers, that are the essential building blocks of an Android application. Activities are used to implement user interface screens. Services implement background processes. Content providers and broadcast receivers handle shared data and messages. Among the four listed Android application components, users only interact with activities. An Android application can consist of several activities. The guidelines (Android guidelines, 2014b) for Android developers recommend that an activity is a single, focused task that a user can do. Each activity represents a single-screen user interface (UI). As a result, only one activity can be in the foreground for users to interact with.

There are two ways to declare a UI layout for an activity: 1) Declaring UI layout elements in an XML file (aka., UI XML layout), or 2) Instantiating UI layout elements programmatically. Our premise in this work is towards the former approach since it is the recommended way by the Android design guidelines (Android guidelines, 2014b). An XML layout defines a human-readable visual structure for a user interface. Applications using the latter way are excluded from our study, as our analysis and data gathering approach cannot handle them.

Every Android application has an AndroidManifest.xml (manifest) file in its root directory. The AndroidManifest.xml contains the meta-data information of an application(e.g. the path to the source code of activities and permissions). In addition to the activities and compiled code, the manifest file plays an important role as a source of information.

3. Case study design

In this study, we investigate to what extent common element sets (CESs) occur in user interface designs of the Android applications. We examine the characteristics of CESs that may lead to the high user-perceived quality.

3.1. Data collection

The Android Market (Google Play) started with 2300 applications in March 2009. Currently there are more than 1 million applications in the market (Number of android apps, 2014a). The Android operating system has the highest market share among other competitors (MarketLead, 2014). As a result, we decided to analyze Android applications from the Android market. Moreover, we only study the free applications due to cost issues.

In the Android Market, there are 34 different kinds of categories from which we analyze 8 different categories: Shopping, Health, Transportation, News, Weather, Travel, Finance and Social. The intuition behind choosing these categories is that they encompass different functionalities of daily use of mobile applications. Table 1 shows the descriptive statistics of our study. In total, we study 1292 free android applications crawled in the first quarter of 2013.

The Android Market allows users rate applications with stars from 1 to 5 (i.e., Low to High), and write reviews for applications. The ratings of an application can show the user-perceived quality of applications, and inform potential users about the experience of the earlier users. However, the rating of an application is not solely a reliable quality measurement, as 86% of the five-star applications throughout the Android Market in 2011 are applications with very few raters (less than 10 raters) (Mojica Ruiz, 2013). Moreover, Harman et al. (2012) show that the raters have a high

Please cite this article as: S.E.S. Taba et al., An exploratory study on the usage of common interface elements in android applications, The Journal of Systems and Software (2016), http://dx.doi.org/10.1016/j.jss.2016.07.010