

# Android マニフェストのエラ一定義、分類及び一部自動検出に関する研究

木村 浩一郎<sup>†1</sup> 劉 少英<sup>†2</sup>

Android 開発者は年々増加している。しかし、Android の開発環境に不慣れなユーザは、Manifest に関するエラーが発生した際にバグの特定を行うことが非常に難しい。そこで、開発者の Manifest に関するエラーを分類し、Manifest のエラーの一部を発見できるように簡易のチェッカーを用意した。このチェッカーにより、ユーザは Manifest の Activity, Service, uses-permission に関するエラーの一部を自動的に発見できるようになった。

## Error definition, classification and partially automatic detection of Android manifest

KOICHIRO KIMURA<sup>†1</sup> Shaoying Liu<sup>†2</sup>

Android developers are increasing every year. When errors in relation to Android manifest occurs, it is very difficult for a user unfamiliar to the development environment of Android to detect the errors. This paper attempts to systematically define errors and classify errors concerned with Android manifest, and describes a simple checker that helps the user to automatically detect some specific types of errors in relation to the activity of the manifest, service, and uses-permission.

### 1. はじめに

AndroidOS を搭載したスマートフォンは急速に普及が進んでおり 2012 年には 4 億台のスマートフォンが全世界で販売された[1]。それに伴い 2012 年 6 月時点で AndroidOS 向けのアプリケーションは 60 万個を超えるほどのものが、google マーケットに登録されている[2]。しかし、Android 開発に慣れていない開発者は Android アプリを開発するにあたり、AndroidOS 特有のバグや AndroidOS の開発環境に対して簡単なミスが発生させてしまうことがある。

本論文では、Android のマニフェスト中に記載する Activity, Service, uses-permission が適切に設定されているかをチェックする方法を作成した。その結果を示す。

### 2. Android マニフェスト

Android マニフェストはそのアプリケーションがどのようなリソースを用いて動作をするかというものを事前に規定しておくものである。そのため、登録されていない挙動を行った場合エラーが発生し、アプリは異常終了する。これを適切に用いることでユーザはアプリからの不正な情報流出などを避ける事が可能となる。Android アプリケーションの実装としてはプロジェクトの直下に Android.manifest として記述される。記述は xml 形式で行われ記述される項目と記述したコードに不一致がある場合、アプリがクラッシュしてしまう問題が有る。またその問題を解決するためにデバッグをする時、Android アプリの開

発に不慣れな場合コードのミスなのか Manifest のエラーなのかの特定に時間がかかる。今回は Manifest の不一致によるエラーのうち Activity, Service, uses-permission に関するエラーを発見する手法を実装した。

### 3. 実装結果

今回実装にあたって Ruby の 1.9.3 を用いて実装を行った。解析を始めるにあたって Android.manifest の解析を XML の解析を事前に行い、Activity, Service, uses-permission の情報を取得している。

#### 3.1. Activity と Service に対する問題の発見手法

Activity と Service に関するエラーとして考えられるものとしては、Activity 若しくは Service として登録されていないクラスを Activity や Service として使用した際に発生する。登録されているかどうかは、Manifest を解析することで理解することが出来る。Activity と Service の使用方法は以下の図 1 のような形で用いる。

```
Intent i = new Intent(TestTestActivity.this,
    MyClassActivity.class);
startActivity(i);
Intent j = new Intent(TestTestActivity.this,
    MyService.class);
startService(j);
```

図 1 Activity と Service の使用方法

Figure 1 The directions for Activity and Service

Intent というクラスを使用することで次に呼ばれる Activity や Service は規定される。Activity を起動するのであれば初期化時に適切なクラスを与え、startActivity メソッド

<sup>†1</sup> 法政大学大学院情報科学研究科修士課程  
Hosei University

<sup>†2</sup> 法政大学大学院情報科学研究科  
Hosei University

ドを呼び、Service を起動する場合は初期化時に startService メソッドを呼ぶことで実行することが出来る。

ここでは例として Activity の Manifest への登録忘れを見つける方法を述べる。

- startActivity メソッドにより Activity は起動されるため、まず初めのコードの解析で startActivity メソッドが存在するかを確認する。
- startActivity の直前で初期化されている Intent クラスの変数を確認し、その初期化の値を確認する。
- 初期化の形式は上記の図 1 のような形式になっているので、そこから Activity として起動されるクラス名を取得する。
- Manifest を解析した結果とクラス名を比較し、Manifest 内に存在していない場合その startActivity が実行された場合エラーが発生することがわかる。

という 5 段階を踏むことで確認が可能である。Service に関しても同様の手段で確認することが可能であった。図 1 のコードに MyClassActivity と MyService をそれぞれ Android マニフェストに適切に登録を行わずに提案手法を実行した結果は図 2 の様になった。

```
これらのActivityに対してAndroidManifestに記述がありません :  
com.testtest.MyClassActivity  
これらのServiceに対してAndroidManifestに記述がありません :  
com.testtest.MyService
```

図 2 Activity と Service の実行結果

Figure 2 The directions for Activity and Service

図 2 から適切に設定されていないクラスを発見することが出来るのが解る。

### 3.2. uses-permission に対する簡易的な問題の発見手法

uses-permission は Android アプリが本来のアプリに不必要な機能にアクセスを行わないことを保証してくれるという形になっている。そのため、作成したアプリから uses-permission の設定を行わずに、uses-permission によって保護されている機能や領域にアクセスするとエラーを発生してアプリがクラッシュしてしまう原因となる。これらは外部のライブラリを導入した際に求められる uses-permission の記述の見落とし、自分で作成したコードを確認せず使用した場合などが考えられる。今回実装した方法は、Java プログラム中で import されたクラス名から推測を行い、警告を出力するという方法である。今回は作成したアプリから HTTP の通信を行う場合を例としてあげる。Android アプリから HTTP プロトコルを用いた通信を行うためには Android.manifest に android.permission.INTERNET の登録が必要である。Android アプリで HTTP を行う際には HttpResponse インターフェースを用いる事が多い。このクラスを用いるときは import 文で、org.apache.http.HttpResponse が宣言される。この import 文が宣言されている場合は http プロトコルによるインターネット

トアクセスを行っていると考えられる、android.permission.INTERNET に関する許可が必要であると推測ができる。その場合に Android.manifest 内に適切な uses-permission が存在しない場合に、警告を出すことで確認することが出来る。ここで警告としている理由としては内部の解析を行っていないため、本当にそのクラスが使用されているかを確認できていないためである。

## 2. おわりに

今回の実装の結果により Android SDK を利用した開発の場合、Manifest に関する Activity と Service のバグは提案手法で発見することが可能となった。Manifest に関する uses-permission のエラーは import 文によって警告を出すに留まっているが、デバッグを行う際の指標になるものと考えているこれにより、Android の開発で適切に画面遷移がされない場合にバグの特定を比較的簡単に行うことが可能となったと考えられる。しかし、本手法では解決できていない問題がいくつか存在している。クラスの親子関係を作成していないため、Intent クラスを継承したクラスで startActivity もしくは startService を起動した場合特定が出来ないという問題が発生している。uses-permission の使用しているデータは手動で入力しているため、ユーザが作成した permission には適応することが出来ないため自動化が望ましい。また、外部ライブラリを用いた開発を行った際に内部をデコンパイルするなどの方法を取らないと uses-permission の問題を解決することが難しいと考えられる。今後はこの結果を利用して Eclipse のプラグインに移植を行なって行きたいと考えている。

## 謝辞

### 参考文献

- 1) 世界のスマートフォン出荷台数、2012 年は前年比 45% 増の 7 億 1,700 万台  
<http://www.computerworld.jp/topics/573/205754>
- 2) Android アプリの登録数は 60 万本で 200 億本ダウンロード  
<http://www.itmedia.co.jp/news/articles/1206/28/news016.html>
- 3) The AndroidManifest.xml File | Android Developers  
<http://developer.android.com/guide/topics/manifest/manifest-intro.html>