Static analysis for quality mobile applications

Julia Perdigueiro
MOTODEV Studio for Android Project Manager
Instituto de Pesquisas Eldorado

Eric Cloninger
Product Line Manager
Motorola Mobility
Agenda

- Motivation & Requirements
- MOTODEV App Validator
- Static Analysis for Android
- The SDK: Creating Your Own Validations
- Wrapping Up
Motivation & Requirements
In the beginning

- Internal teams from Motorola were requesting a pre-flighting tool so application quality could be increased.
- There should be a way of checking whether applications might be compatible with Motorola devices.
Scenario

- There were no tools available specific to Android back then
  - Conversations started in May 2010
- The team that developed MOTODEV Studio for Android had the knowledge to work on this new tool
We wanted the new tool to…

- A standalone, command line environment
- Consistent output mechanism
  - Possible to have IDEs like Eclipse or MOTODEV Studio parse and utilize messages effectively
- Possible to add device specifications without the necessity of rebuilding the entire application
- Possible to integrate with application build processes (IDE or command line)
... and also...

- Indicate how serious a problem is
  - Provide as much information as possible to help the user understand how (or if) they should fix a problem
- Be able to find and install updates
- Suitable for both APKs and ADT Projects contents
- Available for Windows, Mac and Linux
Choices, choices

VS.

Eclipse Rich Client Platform
Second Edition

© 2012 Motorola Mobility, Inc.
What kind of tool to develop?

- Eclipse-based products structure was already well known by the development team
- We were open to analyzing the advantages of using other structures/technologies
- Cons of a ‘pure’ Java tool were too significant
  - No free support to update mechanism (p2-like)
  - Impossible to use Eclipse plug-ins for support (Java/XML processing, etc.)
And the winner was...
MOTODEV App Validator
Today…

- Provides developers with over 30 tests to increase application quality and stability. Addresses issues such as:
  - Invalid entries in the Android manifest
  - Missing strings, image files, and layouts
  - Too few or too many permissions declared in the manifest
  - Conditions that cause apps to be filtered by the Android Market
Today…

- Checks for compatibility issues by comparing app settings against specifications for every Motorola device
  - Specifications for non-Motorola devices could also be added easily

- Available as: Command-line tool, Eclipse plug-ins and on the Web

- Runs on Windows, Mac and Linux
The architecture

- Eclipse-based
- Different front-ends provide different ways of use
- It’s alive!
  - A work in progress
  - Not only for providing new validations, but also for improving the structure and provided features
App Validator architecture
Workflows

- All front-ends have similar workflows
- Command line is the most complex one
  - User input parsing and validation
- Even with differences, the plug-in structure allows the same workflow to take place at some point
App Validator workflow

User input

Command-line Tool / RCP
- Global parameter validation

App Validator Core / API
- Checker parameter parsing
  - Model preparation
  - Checker running
  - Data checking
  - Validation

Checkers / Conditions
- Checker parameter validation

Define outputter
- Output collection

Validation output
Static Analysis for Android
Why not plain Java static analysis?

- Specific types of classes
  - Activity, Broadcast Receiver, Content Provider, Service
- Permissions
- Android Manifest file details
- Resources problems
  - Layout, Locale, etc.
- Compatibility across different devices
  - Today: smartphones, tablets
  - Soon: TV, GPS
  - Upcoming: refrigerators, cars… who knows? :}
Validation example

- Asking for more permissions than necessary
  - Can prevent your application from being shown in the application store for some devices

- Be sure of the permissions you ask for!

- Permissions may imply Features

- Requiring Features imply that the devices have the necessary support (software and hardware)
Permissions in action

Permissions

THIS APPLICATION HAS ACCESS TO THE FOLLOWING:

YOUR LOCATION
COARSE (NETWORK-BASED) LOCATION
Access coarse location sources such as the cellular network database to determine an approximate device location, where available. Malicious applications can use this to determine approximately where you are.

NETWORK COMMUNICATION
FULL INTERNET ACCESS
Allows an application to create network sockets.

PHONE CALLS
READ PHONE STATE AND IDENTITY
Allows the application to access the phone features of the device. An application with this permission can determine the phone number and serial number of this phone, whether a call is active, the number that call is connected to and the like.

STORAGE
MODIFY/DELETE USB STORAGE CONTENTS MODIFY/DELETE SD CARD CONTENTS
Allows an application to write to the USB storage. Allows an application to write to the SD card.
Validation example

```
<uses-permission
    android:name="android.permission.SEND_SMS"
/>
```

```
<uses-feature
    android:name="android.hardware.telephony"
/>
```

© 2012 Motorola Mobility, Inc.
Validation example

<uses-permission
android:name="android.permission.SEND_SMS"
/>

<uses-feature
android:name="android.hardware.telephony"
android:required="false"
/>
Validation example

- The tool looks into the Java code for API calls and makes best guesses
  - Extra permissions
  - Missing permissions
- Permissions validation is a good example of how particular static analysis for Android is
- Models for Java and different XML files have to be processed and created to make validations available
  - Uses JDT for building models for Java code
The SDK: Creating Your Own Validations
Sooo simple...

As we’re on top of Eclipse... Let’s use extension points!
- Checkers
  - Conditions
- Outputters (already included: CSV, XML, plain text)

Many basic classes already implemented within the framework
- Concentrate on implementing only the necessary code: the validation itself

Reference implementation available in the distributed package
Why so simple?

- We’re dealing with mobile developers, i.e., Android developers
- They may not be familiar with the Eclipse reality
- They hold the knowledge (and the necessity) for Android application validation
- Make it as painless as possible
- The development of validation code is not their main goal
SDK Structure

App Validator Framework

Checker

Condition

Condition

Outputter
The SDK: Creating Your Own Validations (Quick Demo)
Wrapping Up
Areas of Improvement

- Improve command-line tool startup speed
  - OSGI usage, loading time, drop unnecessary bundles, etc.

- Usability improvements
  - Specially on how the output is shown
    - Grouping
    - Having a separate view on the IDE
    - Graphical results options

- Improvements on parsing information from Android resources and Java code

- And last but not least… new validations!
Where do we get our ideas from?

- Our own development team
- Other Motorola teams
- Discussion Boards, Stack Overflow the MOTODEV Community
- Studying new features made available by each of the Android releases
- Try not to overlap with other similar tools
  - We want to add value, not replicate effort :)

© 2012 Motorola Mobility, Inc.
Thank you

Q&A

Get App Validator here: moto.ly/mds

Validate your apps on the Web: moto.ly/webvalidator

Take the MOTODEV survey & get an extra chance to win! Just stop by our booth today :)

© 2012 Motorola Mobility, Inc.
Give Feedback on the Sessions

1. Sign In: www.eclipsecon.org

2. Select Session Evaluate

3. Vote
   -1  0  +1
LICENSE NOTICES

Except where noted, sample source code written by Motorola Mobility Inc. and provided to you is licensed as described below.

Copyright © 2010-2012, Motorola, Inc. All rights reserved except as otherwise explicitly indicated.

- Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
  - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Motorola, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other source code displayed in this presentation may be offered under other licenses.

Apache 2.0

Copyright © 2010, Android Open Source Project. All rights reserved unless otherwise explicitly indicated.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Creative Commons 3.0 Attribution License

Portions of this presentation are reproduced from work created and shared by Google (http://code.google.com/policies.html) and used according to terms described in the Creative Commons 3.0 Attribution License (http://creativecommons.org/licenses/by/3.0/).