Overview of Android Frameworks

(Part 1)

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

• Understand how software frameworks are used in Android & Java
Overview of Frameworks in Android
Overview of Frameworks in Android & Java

- A framework is an integrated set of components that provide a reusable architecture for a family of related apps

See [www.dre.vanderbilt.edu/~schmidt/frameworks.html](http://www.dre.vanderbilt.edu/~schmidt/frameworks.html)
Overview of Frameworks in Android & Java

- Android & Java provide many frameworks
Overview of Frameworks in Android & Java

- Android & Java provide many frameworks, e.g.
  - **Android**
    - Android’s Activity framework manages lifecycle hook methods dispatched in the UI thread.
Overview of Frameworks in Android & Java

- Android & Java provide many frameworks, e.g.
  - **Android**
    - Android’s Activity framework manages lifecycle hook methods dispatched in the UI thread
      - e.g., `onCreate()`, `onStart()`, `onStop()`, `onDestroy()`, etc.

See developer.android.com/training/basics/activity-lifecycle
Overview of Frameworks in Android & Java

- Android & Java provide many frameworks, e.g.

  - **Android**
    - Android’s Activity framework manages lifecycle hook methods dispatched in the UI thread
    - A listener for button clicks is called back by Android’s GUI framework

```java
public void onClick(View v)
{
    ...
}
```

A GUI component sending an event to its registered listener
Overview of Frameworks in Android & Java

- Android & Java provide many frameworks, e.g.
  - **Android**
  - **Java**
    - A thread calls the run() hook method of a runnable

See [docs.oracle.com/javase/tutorial/essential/concurrency/runthread.html](docs.oracle.com/javase/tutorial/essential/concurrency/runthread.html)
Overview of Frameworks in Android & Java

- Android & Java provide many frameworks, e.g.
  - **Android**
  - **Java**
    - A thread calls the run() hook method of a runnable
    - The ExecutorService calls the call() hook method of a callable

See [docs.oracle.com/javase/tutorial/essential/concurrency/executors.html](https://docs.oracle.com/javase/tutorial/essential/concurrency/executors.html)
Overview of Frameworks in Android & Java

- All Android apps run inside one or more software frameworks

Your mobile apps must use multiple Android frameworks
Overview of Frameworks in Android & Java

- All Android apps run inside one or more software frameworks
- The motivation for using so many frameworks is to enhance systematic reuse

See en.wikipedia.org/wiki/Code_reuse#Systematic_software_reuse
Overview of Frameworks in Android & Java

• All Android apps run inside one or more software frameworks
  • The motivation for using so many frameworks is to enhance systematic reuse
  • Apps (& app developers) thus don’t need to “reinvent the wheel”

See en.wikipedia.org/wiki/Reinventing_the_wheel
Overview of Frameworks in Android & Java

- Android frameworks use an *event-driven programming model* to integrate app classes into them.

See en.wikipedia.org/wiki/Event-driven_programming
Overview of Frameworks in Android & Java

- Android frameworks use an *event-driven programming model* to integrate app classes into them.
  - In this programming paradigm, the flow of the program is determined by events, e.g.
    - User actions (button presses)
    - Sensor inputs/outputs
    - Messages from other threads

Overview of Frameworks in Android & Java

- The control flow in framework-driven Android apps traverses between the framework(s) & the app classes

  e.g., create an activity, service, or broadcast receiver

  ![Diagram](image)

  - Register for event
    - Event occurs
    - Event occurs
    - Event occurs
Overview of Frameworks in Android & Java

• The control flow in framework-driven Android apps traverses between the framework(s) & the app classes

e.g., a message occurs that triggers a lifecycle event
Overview of Frameworks in Android & Java

• The control flow in framework-driven Android apps traverses between the framework(s) & the app classes

• A framework calls to app code when an event of interest occurs

  e.g., *dispatch the hook methods onCreate(), onStart(), onReceive(), etc.*
Overview of Frameworks in Android & Java

- The control flow in framework-driven Android apps traverses between the framework(s) & the app classes

- A framework calls to app code when an event of interest occurs

  e.g., the app performs its processing in the context of framework thread(s)
Overview of Frameworks in Android & Java

• The control flow in framework-driven Android apps traverses between the framework(s) & the app classes
  • A framework calls to app code when an event of interest occurs
  • Control returns to the framework after an app callback is done
Overview of Frameworks in Android & Java

• The control flow in framework-driven Android apps traverses between the framework(s) & the app classes
  • A framework calls to app code when an event of interest occurs
  • Control returns to the framework after an app callback is done
  • Lather, rinse, repeat until app is done …

See en.wikipedia.org/wiki/Lather,_rinse,_repeat
Overview of Frameworks in Android & Java

• The control flow in framework-driven Android apps traverses between the framework(s) & the app classes
  • A framework calls to app code when an event of interest occurs
  • Control returns to the framework when an app callback is done
  • Lather, rinse, repeat until app is done …

Separating concerns this way helps enhance systematic software reuse & improve various quality attributes

See www.sei.cmu.edu/reports/95tr021.pdf
End of Overview of Android Frameworks (Part 1)
Overview of Android Frameworks
(Part 2)

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

• Understand how software frameworks are used in Android & Java

• Identify key characteristics of Android frameworks

Domain-specific functionality for concurrent Android programs
Key Characteristics of Android Frameworks
Key Characteristics of Android Frameworks

• Android frameworks (like all frameworks) have three key characteristics

See www.dre.vanderbilt.edu/~schmidt/PDF/Queue-04.pdf
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics
  - Exhibit “inversion of control” (IoC) via callbacks

See en.wikipedia.org/wiki/Inversion_of_control
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics
  - Exhibit “inversion of control” (IoC) via callbacks, e.g.
  - Framework controls the main execution thread
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics
  - Exhibit “inversion of control” (IoC) via callbacks, e.g.
    - Framework controls the main execution thread
  - Decides how & when to run app code
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics
  - Exhibit “inversion of control” (IoC) via callbacks, e.g.
    - Framework controls the main execution thread
    - Decides how & when to run app code
  - IoC is often called the “Hollywood Principle”

See [www.dre.vanderbilt.edu/~schmidt/hollywood-principle.txt](http://www.dre.vanderbilt.edu/~schmidt/hollywood-principle.txt)
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics
  - Exhibit “inversion of control” (IoC) via callbacks

  e.g., an Android looper dispatches a handler, which then dispatches a runnable
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics
  - Exhibit “inversion of control” (IoC) via callbacks

  e.g., an Android looper dispatches a handler, which then dispatches a runnable
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics
  - Exhibit “inversion of control” (IoC) via callbacks

  e.g., an Android looper dispatches a handler, which then dispatches a runnable

  The runnable dispatched via IoC doesn’t know/care how/why it was called back
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics
  - Exhibit “inversion of control” (IoC) via callbacks
  - Integrated domain-specific structure & functionality

Application-specific functionality

Domain-specific functionality for concurrent Android programs

Diagram:
- Looper
- Handler
- Executor
- Runnable
- Message
- Message Queue
- FutureTask
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics:
  - Exhibit “inversion of control” (IoC) via callbacks
  - Integrated domain-specific structure & functionality
    - e.g., provide capabilities that can be reused in one or more domain(s)
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics
  - Exhibit “inversion of control” (IoC) via callbacks
  - Integrated domain-specific structure & functionality
    - e.g., provide capabilities that can be reused in one or more domain(s)

Android’s frameworks focus on domains associated with mobile apps & services
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics:
  - Exhibit “inversion of control” (IoC) via callbacks
  - Integrated domain-specific structure & functionality
    - e.g., provide capabilities that can be reused in one or more domain(s)

Application-specific functionality can systematically reuse framework components.
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics
  - Exhibit “inversion of control” (IoC) via callbacks
  - Integrated domain-specific structure & functionality
  - Provide semi-complete (portions of) apps

Domain-specific functionality for concurrent Android programs

Application-specific functionality

• Looper
• Handler
• Message
• Message Queue
• FutureTask
• Runnable
• Executor
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics:
  - Exhibit “inversion of control” (IoC) via callbacks
  - Integrated domain-specific structure & functionality
  - Provide semi-complete (portions of) apps, e.g.
    - *Hook methods* plug app logic into the framework

Hook methods implement patterns that customize reusable framework classes to run app-specific logic.

Domain-specific functionality for concurrent Android programs.

Key Characteristics of Android Frameworks

• Android frameworks (like all frameworks) have three key characteristics
  • Exhibit “inversion of control” (IoC) via callbacks
  • Integrated domain-specific structure & functionality
  • Provide semi-complete (portions of) apps, e.g.
    • Hook methods plug app logic into the framework
  • Mediate interactions among common abstract & variant concrete classes/interfaces

Application-specific functionality

Runnable
Looper
Handler
Executor
FutureTask
Message
Message Queue

Domain-specific functionality for concurrent Android programs
Key Characteristics of Android Frameworks

- Android frameworks (like all frameworks) have three key characteristics
  - Exhibit “inversion of control” (IoC) via callbacks
  - Integrated domain-specific structure & functionality
  - Provide semi-complete (portions of) apps, e.g.
    - Hook methods plug app logic into the framework
  - Mediate interactions among common abstract & variant concrete classes/interfaces

  *e.g.*, Runnable is a common abstract interface whose run() hook method supports many concrete variants

See developer.android.com/reference/java/lang/Runnable.html
End of Overview of Android Frameworks (Part 2)