Overview of Android
Concurrency 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

• Know the motivations for Android concurrency & concurrency frameworks
Motivating Android’s Concurrency Frameworks
Motivating Android Concurrency Frameworks

• Android’s concurrency frameworks also address design constraints

See developer.android.com/guide/components/processes-and-threads.html#Threads
Motivating Android Concurrency Frameworks

- Android’s concurrency frameworks also address design constraints, e.g.
  - “ANR” dialog is generated if the UI thread blocks too long

The UI thread can’t block for more than several seconds, so it can’t be used for long-duration operations

See developer.android.com/training/articles/perf-anr.html
Motivating Android Concurrency Frameworks

- Android’s concurrency frameworks also address design constraints, e.g.
  - “ANR” dialog is generated if the UI thread blocks too long
  - Network calls are disallowed on the UI thread by default

See blog.vogella.com/2012/02/22/android-strictmode-networkonmainthreadexception
Motivating Android Concurrency Frameworks

- Android’s concurrency frameworks also address design constraints, e.g.
  - “ANR” dialog is generated if the UI thread blocks too long
  - Network calls are disallowed on the UI thread by default
  - Non-UI threads can’t access UI toolkit components directly

UI toolkit components aren’t thread-safe

See android-developers.blogspot.com/2009/05/painless-threading.html
Motivating Android Concurrency Frameworks

- Android’s concurrency frameworks also address design constraints, e.g.
  - “ANR” dialog is generated if the UI thread blocks too long
  - Network calls are disallowed on the UI thread by default
  - Non-UI threads can’t access UI toolkit components directly

Java concurrency mechanisms alone cannot address these constraints!!

See www.dre.vanderbilt.edu/~schmidt/LiveLessons/CPj ava
Motivating Android Concurrency Frameworks

- Android’s concurrency frameworks also address design constraints, e.g.
  - “ANR” dialog is generated if the UI thread blocks too long
  - Network calls are disallowed on the UI thread by default
  - Non-UI threads can’t access UI toolkit components directly

See developer.android.com/guide/components/processes-and-threads.html#WorkerThreads
Motivating Android Concurrency Frameworks

- The “Buggy Downloader” app motivates the need for Android’s concurrency frameworks

See github.com/douglascraigschmidt/POSA/tree/master/ex/M4/BuggyDownloader
Motivating Android Concurrency Frameworks

- The “Buggy Downloader” app motivates the need for Android’s concurrency frameworks
- “Buggy1” throws an exception since the image is downloaded in the UI thread
Motivating Android Concurrency Frameworks

• The “Buggy Downloader” app motivates the need for Android’s concurrency frameworks
  • “Buggy1” throws an exception since the image is downloaded in the UI thread
  • “Buggy2” throws an exception since a UI component is accessed via a background thread
End of Overview of Android Concurrency Frameworks (Part 1)
Overview of Android
Concurrency 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

- Know the motivations for Android concurrency & concurrency frameworks
- Recognize the structure & functionality of Android’s concurrency frameworks
Learning Objectives in this Part of the Lesson

- Know the motivations for Android concurrency & concurrency frameworks
- Recognize the structure & functionality of Android’s concurrency frameworks, e.g.
  - Handler, Messages, & Runnables (HaMeR) framework

See code.tutsplus.com/tutorials/concurrency-on-android-using-hamer-framework--cms-27129
Learning Objectives in this Part of the Lesson

• Know the motivations for Android concurrency & concurrency frameworks

• Recognize the structure & functionality of Android’s concurrency frameworks, e.g.
  • Handler, Messages, & Runnables (HaMeR) framework
  • AsyncTask framework

See developer.android.com/reference/android/os/AsyncTask.html
Overview of Android Concurrency Frameworks
Overview of Android Concurrency Frameworks

- Android defines two primary concurrency frameworks:
  - Looper
  - Handler
  - Message Queue
  - Runnable
  - Background Thread
  - Executor
  - AsyncTask

UI Thread (main thread)
Overview of Android Concurrency Frameworks

- Android defines two primary concurrency frameworks
- Handlers, Messages, & Runnables (HaMeR)

Operations running in one or more background threads can post/send their results to the UI thread

See developer.android.com/training/multiple-threads/communicate-ui.html
Overview of Android Concurrency Frameworks

- Android defines two primary concurrency frameworks
  - Handlers, Messages, & Runnables (HaMeR)
  - AsyncTask

Operations run in one or more background threads & publish results to UI thread without manipulating threads, handlers, messages, or runnables

See developer.android.com/reference/android/os/AsyncTask.html
Overview of Android Concurrency Frameworks

- Both frameworks have pros & cons & are used extensively throughout Android

<table>
<thead>
<tr>
<th></th>
<th>Async Task</th>
<th>Posting Runnables</th>
<th>Sending Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Usability (Simple)</strong></td>
<td>+++</td>
<td>+++</td>
<td>++</td>
</tr>
<tr>
<td><strong>Usability (Complex)</strong></td>
<td>+++</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>+++</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td><strong>Flexibility</strong></td>
<td>++</td>
<td>+</td>
<td>+++</td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td>++</td>
<td>+++</td>
<td>+++</td>
</tr>
</tbody>
</table>

See upcoming part on “Evaluating Android’s Concurrency Frameworks”
Overview of Android Concurrency Frameworks

- Android’s concurrency frameworks are often used to decouple user interactions from computation & communication

Long-duration & (potentially) blocking operations run in background thread(s)
Overview of Android Concurrency Frameworks

- Android’s concurrency frameworks are often used to decouple user interactions from computation & communication.
Overview of Android Concurrency Frameworks

- Android’s concurrency frameworks are often used to decouple user interactions from computation & communication.

See [github.com/douglascraigschmidt/POSA/tree/master/ex/M4/SimpleImageDownloads](https://github.com/douglascraigschmidt/POSA/tree/master/ex/M4/SimpleImageDownloads)
Overview of Android Concurrency Frameworks

- Android’s concurrency frameworks are often used to decouple user interactions from computation & communication.

Background threads perform long-duration image downloads.
Android’s concurrency frameworks are often used to decouple user interactions from computation & communication.

**Overview of Android Concurrency Frameworks**

- **Message Queue**: Synchronized message queue passes results from background thread(s) to UI thread.

- **Looper**

- **Handler**

- **Runnable**

- **Background Thread A**

- **Background Thread B**
Overview of Android Concurrency Frameworks

- Android’s concurrency frameworks are often used to decouple user interactions from computation & communication.

**Diagram:**
- **Looper**
- **Message Queue**
  - **Message**
  - **Handler**
  - **Runnable**
  - **Background Thread A**
  - **Background Thread B**
- **UI Thread** (main thread)
  - **UI thread displays the image to the user**
End of Overview of the Android Concurrency Frameworks (Part 2)