

Android開発へモデル検査の提案

株式会社クレスコ

細川 友武

t-hosokawa@cresco.co.jp

開発における問題点

Androidアプリ開発において、テストは入力値の変更により行われる。しかし、複数プロセスが存在する場合など、実行時の環境に依存して実行順序が変わることがある。すべての実行順序をテストしきれないため、検証されない状態が残ることになる。これにより、不具合が発見されないまま残ってしてしまう。

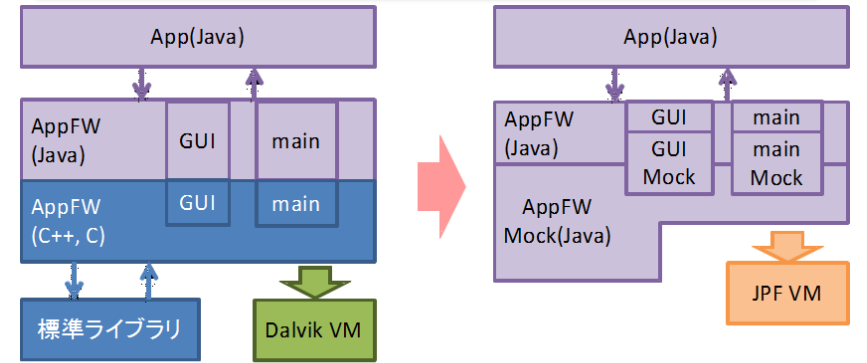
モデル検査の適用による解決

全ての状態を網羅するためにはモデル検査が適している。中でもAndroidアプリ(Java言語)のコードをそのままモデルとして利用できることから、Java Pathfinder(JPF)を適用することを提案した。

古典的モデル検査器とJPF

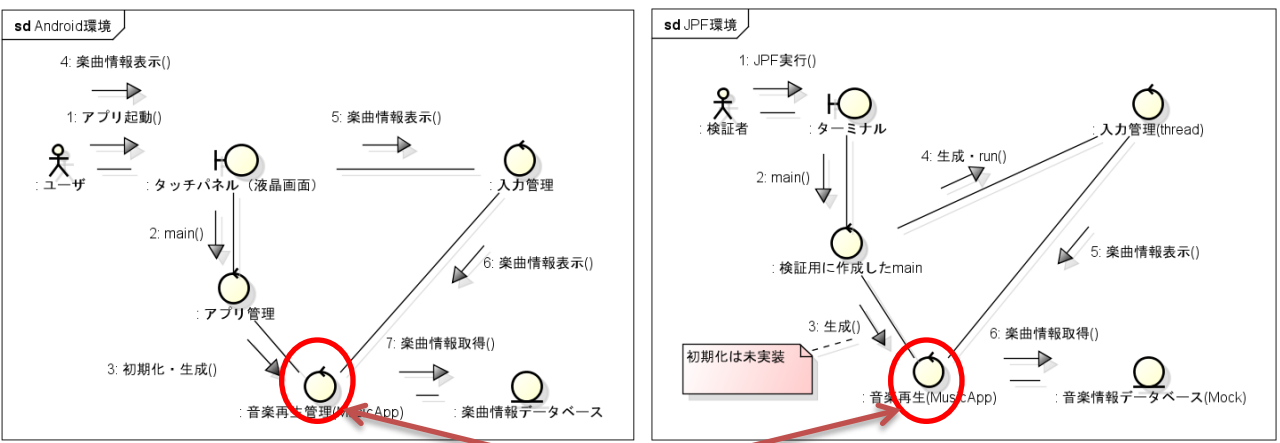
	古典的モデル検査器	Java Pathfinder
対象	仕様書、プロトコル	Javaソースコード
適用工程	上流:設計等	下流:テスト等
モデル作成	仕様書から手動で構築	ソースコードから自動抽出
問題点	モデル検査器ごとのモデル記述言語を用いるため、Java言語特有の問題を見落とす可能性がある	Javaのソースコードから作成するために状態空間が大きく、時間がかかったり使用メモリ削減が必要になることが多い

JPF適用環境構築



JPFではnativeメソッドなどのJava以外の記述が実行経路に存在すると検証できないため、JPF適用のための環境構築が必要。また、GUIなどはMock化することで状態空間を削減する。

既存アプリへの適用



検査対象にとって、Android環境と同じメッセージフローとなる環境を構築して検査を行う

まとめ

- 適用結果
 - ・Mockの動作を非決定的にすることで、単一のメソッドの入力値の違いによる問題を見つけることも可能。
 - ・複数プロセス時の実行順序の違いによる検証を行うためには、アプリの起動処理を作りこむ必要がある。
 - ・アプリ自体の問題は見つけれられるが、アプリケーションフレームワーク(AppFW)以下の問題は見つけれられない。
- 課題
 - ・Mock作成量の多さ
 - ・バージョン更新時のJPF適用環境構築
 - ・AppFW以下の検査