

Android OS における複数アプリ同時表示手法の提案と評価 Design and Evaluation of Displaying Multiple Applications on Android OS

森田 清隆[†] 桐村 昌行[†]

1. はじめに

Android OS (以下、Android)は、従来のスマートフォンやタブレット適用だけでなく、近年様々な組込み機器への適用が始まっている。

組込み機器に Android を適用する利点として、サードパーティ製のアプリが実行可能な環境を提供し、アプリ間で連携が可能となる点がある。しかし、従来の Android は画面に対して 1 アプリしか表示できない制約があり、アプリ間の連携時に表示するアプリの切り替えが起こる。そのため、地図を表示しながら周辺の店情報を表示するなど、複数アプリを同画面上で連携することができない。

これに対して、複数アプリを同時表示・並列実行する製品が存在するが^[1]、特定のアプリしか同時表示できない課題がある。また、複数アプリを並列実行する場合、どのアプリを優先動作させるかも課題となる。

そこで本論文では、任意の複数アプリの同時表示・並列実行、およびその状況下でのアプリの処理優先度制御を可能とする、Android 拡張手法について提案する。

2. 提案手法

Android のアプリの画面は Activity と呼ばれる要素によって構成される。この Activity がユーザと Android とのインタフェースとなる。従来の Android において Activity はダイアログなどを除くと基本的に全画面表示であり、一度に 1 アプリの Activity しか表示できない。

そこで本論文では、Android のアプリケーションフレームワーク^[2] (以下、フレームワーク)を改修することにより、複数アプリの Activity を同時表示及び操作を可能にし、それぞれのアプリに対する処理優先度制御を可能とする手法について提案する。

2.1 複数アプリの同時表示の実現手法

ひとつの Activity が画面を占有する制約は、複数の Activity を表示できない要因となる。そこで、その制約を取り除くため、Activity の表示サイズ・表示位置を管理するフレームワーク WindowManager^[3]に改修を加え、Activity の表示サイズ、表示位置を任意に変更できるように機能拡張した。これにより Activity を重ならないように配置することが可能となる。

ただし、複数の Activity を表示するには、Activity の状態の管理も必要になる。Android では、図 1 に示すように Activity を前面に表示されるべきものから順にスタックで管理している。この情報を用いて Activity の状態を管理するフレームワーク ActivityManager^[4]が、表示する Activity を決定する。従来の Android では、最前面の Activity が表示され、それ以外の背面の Activity は非表示となる。そのため、複数の Activity を表示できない。

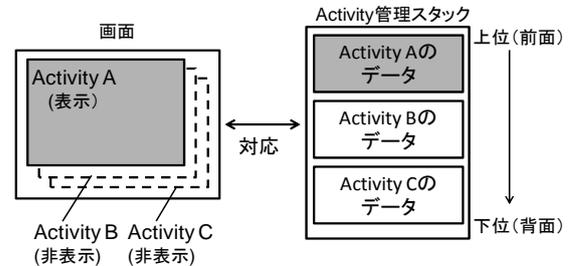


図1 スタックによる Activity の管理

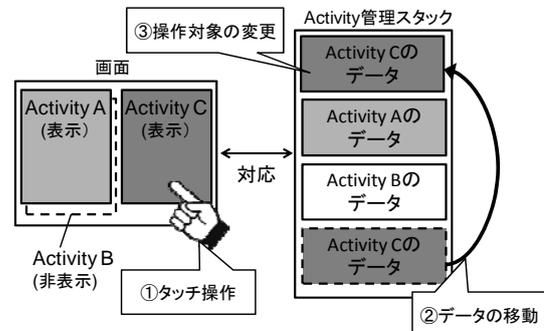


図2 操作対象となる Activity の変更

そこで、ActivityManager において Activity の表示位置の情報を管理し、前面の Activity に重ならない Activity は表示状態にするように改修を加えた。以上により、複数アプリの Activity の同時表示を可能にした。

2.2 複数アプリの並列実行の実現手法

次に 2.1 節で同時表示を可能にしたそれぞれの Activity に対して、ユーザの操作を可能にする手法を示す。Android では、最前面の Activity が操作の対象となるので、ひとつの Activity しか操作できない。これに対して、図 2 に示すように表示されている複数の Activity の内、タッチ操作の発生位置に対応する Activity をスタックの最上位に移動し、最前面の Activity となるように改修を加えた。具体的には、入力イベントを管理するフレームワーク InputDispatcher^[5]を改修し、タッチされた Activity を検知して、ActivityManager でその Activity をスタックの最上位に移動させる。

これにより、タッチ操作により操作可能な Activity を切り替えることができるので、同時表示しているアプリに対してユーザの操作が可能となる。

[†] 三菱電機株式会社 情報技術総合研究所

2.3 複数アプリの優先度制御の実現手法

従来の Android では各アプリは同一優先度で処理される。そのため、複数アプリを同時表示・並列実行した場合、各アプリの実行速度は一律に遅くなる。この場合、例えば地図アプリの動作を同時表示する店情報表示アプリの動作より優先的にやりたいなど、並列実行したアプリの中で特定のアプリを優先動作させたい場合に対応できない。

これに対して、文献[6]を参考にして、提案手法に優先度制御機構を組み込んだ。具体的には Android における各機能に対するアクセス権限を示す Permission に、新たに優先的に処理を行う権限を示す優先処理 Permission を付加し、処理優先度を向上させる機構を用意した。これにより特定のアプリの処理優先度を高くすることを可能とした。

3. 評価

Android(バージョン 4.1.1)に対し、本提案手法を実装し、評価を行った。ただし、今回の実装では同時表示できるアプリを2つに限定した。

表 1に示す評価環境において、仮想マシン上で Android SDK エミュレータを実行し、複数アプリの同時表示・並列実行の動作確認および性能評価を行った。

3.1 動作確認

ブラウザや連絡帳のような Android にプレインストールされているアプリ 15種類と自作アプリ 2種類に対して、2つのアプリを同時表示し、操作することが可能かを確認した。結果、プレインストールされた 1種類のアプリを除き、2つのアプリを同時表示、操作可能であることを確認できた。同時表示できなかったアプリは、画面の表示方向が固定のアプリであった。本実装では、このようなアプリへの対応を行っていないため、今後対応が課題となる。

3.2 性能評価

複数アプリを同時表示・並列実行した場合における優先度制御について評価した。提案手法を適用した Android に対して、従来通り優先度制御を行わない場合と、片方のアプリの優先度を高くした場合において、同じ内容の2つのアプリを左右に同時表示して並列実行した。対象とするアプリはアニメーションの描画を行うアプリであり、このアプリにおいてアニメーションの 1フレームの実行時間を測定し、フレームレート(fps)を算出した。結果を図 3に示す。

処理優先度制御を行わずに2つのアプリを同時に実行した場合、各アプリのフレームレートは単独実行時に対して約 50%になることが分かる。これに対して、優先度制御機構により片方の処理優先度を高く設定すると、優先度を高くしたアプリは単独実行時のフレームレートの約 88%を確保でき、もう片方のアプリのフレームレートは約 8%に下がることが分かる。よって、同時表示・並列実行しているアプリの内、特定のアプリの処理を優先的に処理できることを確認できた。

表1 評価環境

項目	内容・性能
プロセッサ	Intel Core i5 (4cores, 3.1GHz)
主記憶	20.0GB
ホスト OS	Window7 Professional
ゲスト OS	Ubuntu 12.04
仮想化 S/W	VMWare Player

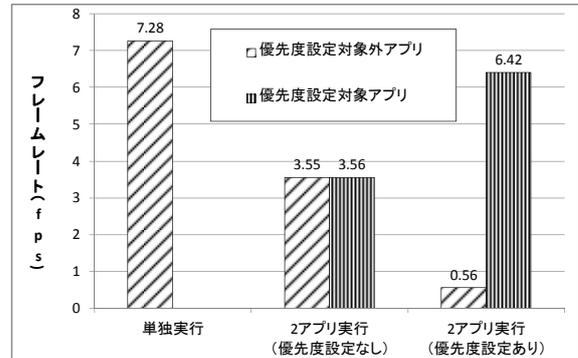


図3 2アプリ同時表示・並列実行における性能

4. まとめ

本論文では、Android のアプリケーションフレームワークの機能拡張を行い、複数アプリの同時表示可能な手法を提案し、実装を行った。実装した機能について Android SDK エミュレータ上において動作確認を行った結果、一部のアプリを除いた、大部分のアプリにおいて2つのアプリの同時表示・並列実行可能であることを確認できた。また、優先度制御機構を実装し、2つのアプリを同時表示・並列実行した場合において、特定のアプリを優先的に処理できることを確認した。

今後は、本研究成果の実機への適用及び評価を行っていく。また、本実装で同時表示できなかったアプリへの対応や同時表示するアプリ数を3つ以上にする実装も行う。

参考文献

- [1]samsung : GALAXY Note II | マルチウィンドウ, samsung, 入手先 (http://www.samsung.com/jp/galaxynote2/home/multitask.html) (参照 2013-06-05).
- [2]himmele : Android Graphics Architecture, himmele, 入手先 (http://himmele.googlecode.com/svn/trunk/Google%20Android/Android%20Graphics%20Architecture.pdf) (参照 2013-06-05).
- [3]博客频道:android 窗口管理剖析, 博客频道, 入手先 (http://blog.csdn.net/huanxido/article/details/7879529) (参照 2013-06-05)
- [4]博客频道:android Application Component 研究之 Activity(二), 博客频道, 入手先 (http://blog.csdn.net/martingang/article/details/8068430) (参照 2013-06-05).
- [5]博客频道:Android4.0 input 事件输入流程详解(中间层到应用层), 博客频道, 入手先 (http://blog.csdn.net/wlw10071986/article/details/8247138) (参照 2013-06-05).
- [6]東山知彦, 増田大樹, 落合真一, “Android の応答性評価”, 情報処理学会 第 75 回全国大会予稿, (2013).