

# Android アプリケーションのスムーズなインストールを可能とするスマートダウンローダの開発

Android™\*1 端末において、各種アプリケーションの本体ではなく、それらをダウンロードするためのダウンローダをプリインストールすることがある。しかしながら、従来のダウンローダの場合、Android の特徴であるアプリケーション間連携ができないなど、利便性が低いという課題があった。そこで、アプリケーション間連携を可能とする新たな仕組みをダウンローダに組み込むことで、本体アプリケーションのスムーズなダウンロードと利用を促進するソリューションを開発した。

サービス&ソリューション開発部 原 未来 加藤 剛志

## 1. まえがき

Android 端末ではキャリア向けサービスを含め、数多くのアプリケーションをプリインストールすることが求められており、端末の空き記憶容量確保や試験コスト削減を目的として、アプリケーションの本体ではなく、上記本体をダウンロードするダウンローダをプリインストールすることがある。

しかしながら、ダウンローダのプリインストールでは、Android OS の特徴である、他アプリケーションからインテント\*2（起動要求）を受けて本体アプリケーションが起動するアプリケーション間連携機能を利用できず、アプリケーション利用の機会が損なわれてしまう。

そこで、上記課題を解決するため

に、ダウンローダでも本体アプリケーションをプリインストールした場合と同等の動作や操作性を提供するスマートダウンローダを開発した。

スマートダウンローダは本体アプリケーションが未インストールの状態でも、本体アプリケーションの代わりに他アプリケーションからのインテントを受信することができる。さらに、受信したインテントを一時的に保存しておき、本体アプリケーションのインストールを検知した際に、上記インテントを本体アプリケーションへ送信することで、アプリケーション間連携を実現している。

## 2. Android 端末へのアプリケーションのプリインストール方法

Android 端末へのアプリケーショ

ンのプリインストール方法は大きくわけて、(a) 本体アプリケーション、(b) ショートカット、(c) 従来型のダウンローダ、(d) 今回開発したスマートダウンローダの4種類がある。各プリインストール方法の比較を表1に示す。

(a) は、前記4種類の中で唯一本体アプリケーションがプリインストールされるため、該当アプリケーションを使い始める際に、ダウンロードおよびインストールが不要であり、アプリケーションへの導線が最も短い。しかしながら、本体アプリケーションのプリインストールは、本体の記憶容量を大きく消費する点、また、Android マーケット™\*3 からの更新通知を受け取れない点が大きな課題となっている。また、本体アプリケーションが提供するすべて

表1 プリインストール方法の比較

	(a) 本体アプリケーション	(b) ショートカット	(c) 従来のダウンローダ	(d) スマートダウンローダ
端末の使用容量削減	×	○	○	○
マーケットからの更新通知	△ 初回の手動アップデートが必要	○	○	○
アプリケーション間連携 (インテントへの応答)	○	×	×	○
アプリケーション利用 までのシームレスな導線	◎ ダウンロード不要	×	×	○ インテント引継ぎによる 自動起動
その他の制約	—	・本体アプリインストール後もショートカットが残る ・ホームスクリーンに配置する必要がある	—	—

の機能を事前に試験するための試験コストがかかること、端末発売の一定期間前までに本体アプリケーションが完成している必要があることなど、端末メーカ、アプリケーション開発ベンダーにとってコストと時間が課題となるケースもある。本体アプリケーションのプリインストールにかかるこれらの課題を解決する代替案として、従来は、(b) ショートカットのプリインストール、(c) 従来型のダウンローダのプリインストールが行われてきた。(b) (c) 共に、本体アプリケーションよりも必要な容量が小さく、かつ、Android マーケットから本体アプリケーションをダウンロードすることになるため、該当アプリケーションの更新通知を受け取ることが可能になる。また、端末発売直前までに本体アプリケーションが完成していればよいため、本体アプリケーションの開発期間に余裕ができる。

(b) ショートカットのプリインストールは、該当アプリケーションをダウンロードする Android マーケット内ページへのリンク、または、web サイトへのリンクをホームアプリケーション上に配置する方法である。ショートカットの場合、本体アプリケーションのインストール後もショートカットが残るため手動削除する必要がある点、また、ショートカットをホーム画面上に配置する必要がある点が課題となる。

(c) 従来型のダウンローダは、ショートカット同様に Android マーケットや web サイトに遷移する機能をもつが、実態は通常のアプリケーションである。また、ダウンローダは、本体アプリケーションと同一の識別子（証明書とパッケージ名）をもち、本体アプリケーションより必ず小さいバージョン番号をもつため、Android システムからは、本体アプリケーションの古いバージョン

と認識され、本体アプリケーションのインストール時に上書き消失する。さらに、通常のアプリケーションであるため、ショートカットのようにホーム画面上に配置しておく必要はない。

(b) (c) のプリインストール方法により、本体アプリケーションをプリインストールする際の課題であった消費記憶容量の削減などは解決できるものの、本体アプリケーションではないゆえに、他アプリケーションからのインテントを受け取れないという新たな課題が出てくる。インテントとは、アプリケーションを起動するための情報であり、例えば、**図1**に示すように「ギャラリー」アプリケーションで画像を見ている際、気になった画像を共有するため「共有」ボタンを押すと、画像を扱える Twitter<sup>\*4</sup> アプリや Facebook<sup>\*5</sup> アプリが起動する場面などで利用されている。インテントを受け取れな

\*2 インテント：アプリケーション内の機能どうしや異なるアプリケーション間でやりとりするメッセージ、および、その仕組み、アプリケーションを起動する際にも利用される。  
\*3 Android マーケット™：Google, Inc.の商

標または登録商標。Android マーケットからアプリケーションの更新通知を受け取るには、そのアプリケーションを Android マーケットからダウンロードしている必要がある。

\*4 Twitter：アメリカ合衆国または他国々における Twitter, Inc.の登録商標。  
\*5 Facebook：Facebook, Inc.の商標または登録商標。

なくなってしまうことで他アプリケーションからの起動が行えなくなってしまう。また、インテントを受け取れないことに加え、ダウンロード操作を挟むことによりアプリケーション利用までの導線が切れてしまうという課題もある。

そこで、これらの課題を解決する方法として、新たに (d) スマートダウンローダを開発した。

### 3. スマートダウンローダの開発

#### 3.1 スマートダウンローダの概要

スマートダウンローダは、従来型のダウンローダ同様に、Androidマーケットやwebサイトに遷移するアプリケーションであるが、(1)他アプリケーションからのインテントに回答する機能、および、(2)本体アプリケーションのインストール後に、前記インテントを本体アプリケーションに引き継いで本体アプリケーションを自動起動する機能をもつ。

##### (1)インテントへの応答

すべてのAndroidアプリケーションはマニフェストファイルと呼ばれるアプリケーションの定義ファイルをもっており、この中に、アプリケーションの識別子<sup>\*6</sup>となるパッケージ名や、どのようなインテントに回答するかを宣言するインテントフィルターなどの情報が記載されている。スマートダウンローダは、基本的に、本体アプリケーションと同じ識別子とインテントフィルターをマニフェストファイルに記載してお



図1 インテントの一例

り、本体アプリケーションの代わりに、他アプリケーションからのインテントに回答することが可能である。また、応答して受け取ったインテントは、本体アプリケーションを起動するために利用するため、一時的に保存される。

##### (2)インテントの引継ぎによる自動起動

Android OSは、何らかのアプリケーションがインストールされた際に、インストール完了イベントを発行する。スマートダウンローダは、本体アプリケーションがインストールされた際の上記インストール完了イベントを検知して、本体アプリケーションにインテントを送信し、これを起動する。この際、(1)で保存しておいたインテントを読み出して利

用しているため、スマートダウンローダに画像などのオプション情報が渡されていた場合、これらの情報を本体アプリケーションにそのまま引き継ぐことが可能である。

スマートダウンローダの機能をユーザ視点の画面遷移として図2に示す。例えば「ギャラリー」アプリケーションで共有ボタンを押した際、図1で一覧表示されるアプリケーションに加え、サンプルアプリケーション「Shakes!」のスマートダウンローダも、共有先の一覧として表示されている(図2①②)。「Shakes!」を共有先として選択すると、「Shakes!」本体アプリケーションのダウンロードを促す画面が表示され、Androidマーケットなどに遷移する(図2③④)。本体アプリケーシ

\*6 アプリケーションの識別子：同一のアプリケーションとして認識されるためには、パッケージ名の他、アプリケーションの証明書も同じである必要がある。



ョンのインストール後は、「ギャラリー」アプリケーションからの画像とともに本体アプリケーションが自動的に起動する (図2⑤)。

### 3.2 アプリケーションの構成

上述のようにスマートダウンローダは、従来型のダウンローダと同様に本体アプリケーションと同一の識別子を保持しているため、本体アプリケーションのインストール完了と同時に上書き消失することになる。すなわち、ダウンローダ単体では、本体アプリケーションのインストール完了の検知や、検知後に、保存しておいたintentを本体アプリケーションに引き継ぐ動作を行えない。そこで、スマートダウンローダは、本体アプリケーションごとに用

意する連携ダウンローダアプリと、複数の連携ダウンローダから共通して利用されるブリッジランチャーアプリから構成される。ブリッジランチャーが、前述のintentの保存、本体アプリケーションのインストールの検知、および、検知後の本体アプリケーションへのintentの引渡しを行うことで、intentの転送を実現している。図3に、連携ダウンローダとブリッジランチャーの構成例を示す。

### 3.3 動作シーケンス

図4に示すスマートダウンローダの動作シーケンスフローを使って、スマートダウンローダの一連の内部的な動作を解説する。

「ギャラリー」アプリケーションなどの各種アプリケーションは、他

のアプリケーションと連携するために、画像などのオプション情報とともにintent (起動要求) を発行する (図4①)。連携ダウンローダは、まだインストールされていない本体アプリケーションの代わりに前記intentに応答する。

intentに応答した連携ダウンローダは、ブリッジランチャーにintentの保存を要求し (図4②)、ブリッジランチャーは、このintentを保存する (図4③)。

ブリッジランチャーによるintentの保存後、連携ダウンローダはユーザ操作を介してAndroidマーケットなどから本体アプリケーションをダウンロード、および、インストールする (図4④⑤)。本体アプリケーションのインストール完了とともに、連携ダウンローダは上書き消失



図3 連携ダウンロードとブリッジランチャーの構成例

する。残ったブリッジランチャーは、本体アプリケーションのインストール完了イベントを検知して(図4⑥)、保存されたインテントを読み出し(図4⑦)、このインテントを本体アプリケーションに転送する(図4⑧)。前記インテントによって、図4①で連携ダウンロードに渡された画像などのオプション情報とともに、本体アプリケーションが起動されることになる。以上が、連携ダウンロードとブリッジランチャーの連携によるスマートダウンロードの基本的な動作である。

また、図4①において、インテントは、ユーザがアプリケーションのアイコンをタップした際にも発行されるが、この場合、画像などのオプ

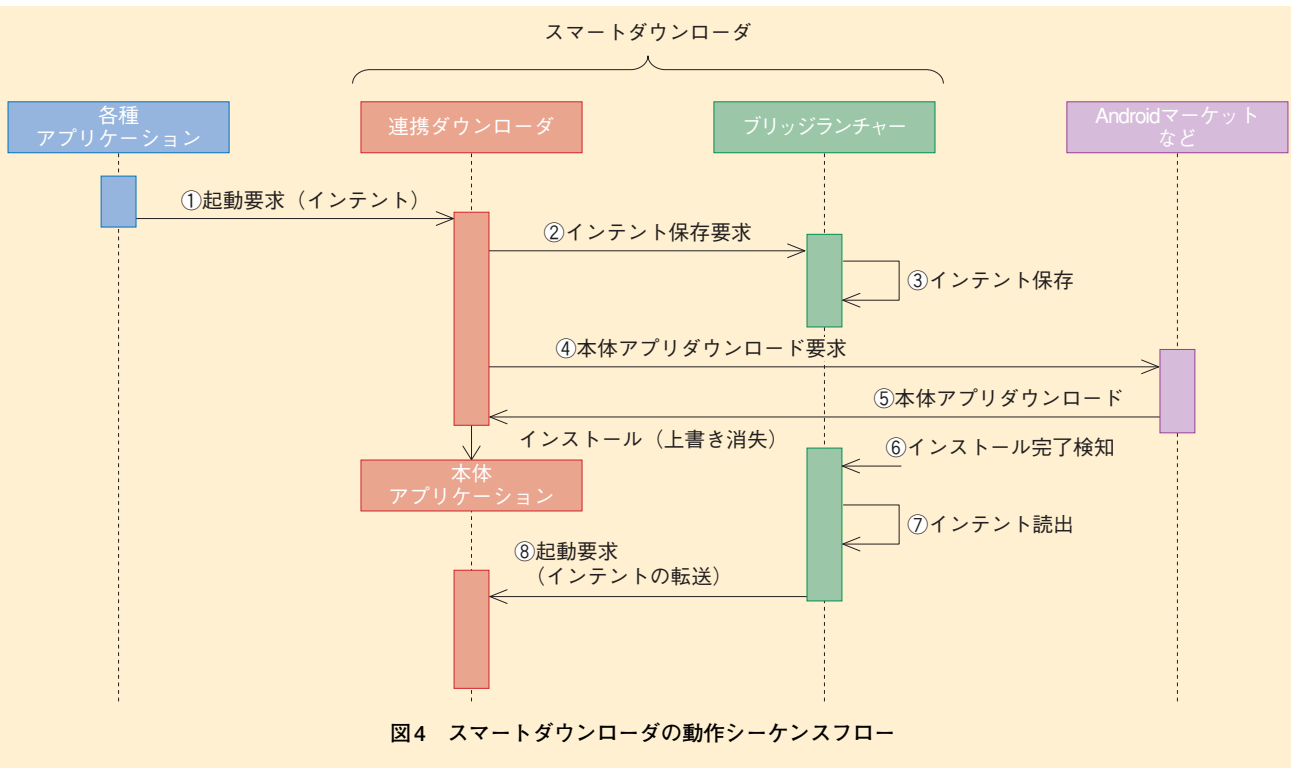


図4 スマートダウンロードの動作シーケンスフロー

ション情報が含まれずに自動起動が行われるだけで、基本的なシーケンスは前述の内容と同様である。

#### 4. あとがき

本稿では、Android 端末における新たなプリインストールソリューションとしてスマートダウンローダを解説した。スマートダウンローダは、連携ダウンローダとブリッジラ

ンチャーという2つのアプリケーションが連携してIntentの転送処理を行うことで、従来のダウンローダでは不可能だったアプリケーション間連携を実現し、ダウンロード型でプリインストールされた各サービスへ、ユーザをより多く集客することを可能にした。また、ブリッジランチャーは本稿執筆時点で、2011年秋冬モデルへのプリインストールが

決定しており、13個のアプリケーションがスマートダウンローダを採用している。

今後は、各サービス提供者がさらにスマートダウンローダを採用しやすくなるように、開発用パッケージの提供などを進めて行く予定である。