

Android

伊藤 博高, 中村 彰之

Hiroataka ITO, Akiyuki NAKAMURA

1 はじめに

近年, 携帯電話は急速に進化しており, パソコンからではなく携帯電話からインターネットにアクセスする機会が増えている. その中で, 携帯電話業界は進化していく 3G 携帯電話によって開発コストが大きくなる一方, 高性能なものを安く, 且つ早く開発しなければならない事態に直面している. そのため携帯電話会社のほとんどが自社開発の OS ではなく汎用 OS を用いることで開発コストと開発期間を縮小している. しかし, 開発環境は端末によって異なるためソフトウェア開発者への負担は変わっていない.

この背景から, モバイル分野でのビジネスに着目したのが Google である. Google はインターネット上の広告や検索サービスにより収益を上げている企業である. 世界で 37 億人が携帯電話を使っているが, 全員がインターネットにアクセスしているわけではない¹⁾. そこで, 携帯電話ユーザのインターネット利用率を高め, Google のユーザを増やせば更なる利益に繋がる. そのためには, 携帯電話からでも Google を利用し易い環境の普及が必要である. そこで, Google は OS と開発環境をセットにしたプラットフォームを開発し, 無料で提供することにした. それが Android である.

本稿では, Android の構造, 特徴について説明し, 現状および今後の展望を述べる.

2 Android

Android とは, OS やミドルウェア, ユーザインタフェース, アプリケーションなどの多くのソフトウェアを含んだ携帯電話の開発プラットフォームのことである. インターネットを通じて全てのソースコードが無償で公開されている.

Android の OS には Linux を用いているが, 必要最低限の動作を担うカーネルのみが採用されている. それは, 携帯電話ではメモリが限られており, コンパクトで高速にしなければならないことと, Linux が採用している GNU Public License をできるだけ避けているからである. このライセンスがあると, プログラムを改変した場合にそのソースコードを公開しなければならないという制約が付いてしまう. そのため, 開発者によって今後修正を加えられると予想される部分は Google が独自で開発し, Apache Software License 2.0 という製造ソースの公開を義務付けないライセンスとして採用することで, その制約を取り除いている.

2.1 構造

Android の構造を Fig.1 に示し, Android の構造を最下層の Linux カーネルから順に説明する.

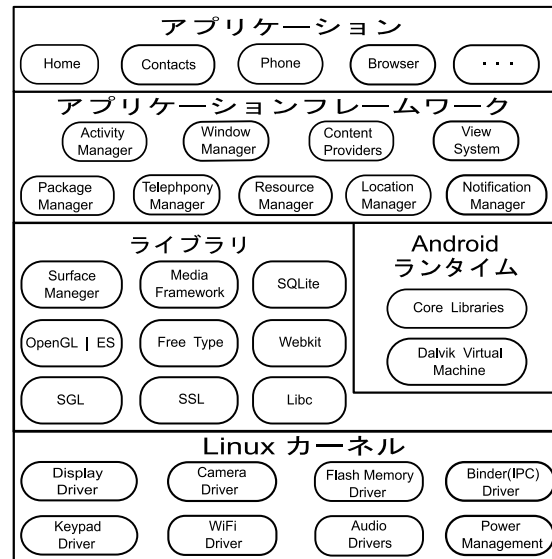


Fig.1 Android の構造 (参考文献²⁾ より参照)

Linux カーネルのドライバ群はハードウェアを駆動するためのソフトウェアである. ドライバは, デバイスと Linux の間を取り持ち, デバイスの機能を Linux から利用できるようにしている. そのため, 新しいデバイスが開発された時, Android 用のドライバを新たに用意するだけでデバイスの機能を利用可能になる.

Linux カーネルの上には複数のアプリケーションから共通利用されるプログラムをまとめたライブラリがある. また, その内部に Android ランタイムが定義されている. ランタイムはアプリケーションの実行の際に必要なソフトウェアであり, 主に Core Libraries と Dalvik Virtual Machine (以下 Dalvik VM) の 2 つに分かている. Android のアプリケーションはこの Dalvik VM 上で動作する. 開発言語には Java が用いられ, 一般的な Java プログラミングで用いるライブラリの内, 役立つ機能の多くが Core Libraries として利用できる.

アプリケーションフレームワークは Android のアプリケーション動作に関係し, 共通で利用される Application Programming Interface (以下 API) を規定しており, ユーザがアプリケーションを利用する際, Android 独自のアプリケーションを作り出す箇所でもある. API とはソフトウェアを開発する際に使用できる関数の集合のことであり, 開発者はアプリケーションフレームワークで

公開された API を用いてアプリケーションを作成することができる。

Android の中でも特徴的なアプリケーションについては次節で詳しく説明する。

2.2 アプリケーション

Android 上のアプリケーションはアーキテクチャの複数の要素を協調させて動作している。このアプリケーションが動作する仕組みは複雑であるが、開発を行う際はその複雑さを感じさせない構成となっている。それは、組み込みのソフトウェア開発であるが、PC などの開発環境や用いている Java クラスライブラリが比較的一般的なものだからである。アプリケーションには主に「Activity」、「Intent」、「Intent Receiver」、「Service」、「Content Provider」の 5 つの要素があり、いくつかを組み合わせて構成される。これらの要素について説明する。

• Activity

アプリケーションは複数の画面から構成されているが、その画面を作る要素が Activity である。アプリケーションの画面遷移は新しい Activity の起動により、新しい画面へ遷移する。画面の遷移パターンは基本的に「進む」と「戻る」の動作である。画面を先に進める度に履歴スタックへ新しく生成した画面の情報を積み重ねて保管している。古い画面は一時停止状態にして新しい画面を表示する。戻る時は履歴を一つずつ戻して一時停止の画面を表示する。このように「進む」と「戻る」を画面のスタック管理で実現している。この仕組みにより、アプリケーションを作っている際に、特別な記述をしなくても、「戻る」動作が実現できる。

• Intent

アプリケーション内部にある Activity の遷移や、アプリケーションから別のアプリケーションの遷移を行うために用いるのが Intent である。これは次の画面で行わせる動作や Activity の名前や特徴を書いた、メッセージを伝えるボールのようなものである。Intent を作成してアプリケーションフレームワークに投げると、Intent の情報により起動すべきアプリケーションを判断する。起動したアプリケーションで、Intent フィルタという Intent を受け取った際にどのような処理を行うかを示したクラスに従い、処理を行う。

• Intent Receiver

Intent を受け取って処理する役目を担うのが Intent Receiver である。Intent は Activity か Intent Receiver に対して投げられる。何かのイベントが生じた時に Intent Receiver に対して Intent が投げられた場合は、画面を持たないバックグラウンドで処理するような動作を行う。例えば、電話が着信した時に音が鳴るなど、外部からイベントが飛んできて電話内部の状態が変更され、それに伴ってアプリケーションの画面に依存しない処理を行う。

• Service

Service は画面を持たずバックグラウンドで動き続けるプログラムである。通常、アプリケーションの処理は Activity に記述すればよいが、音楽プレイヤーの場合では、画面が消えたと同時に音楽も止まってしまう。そのため「ながら再生」をするために、バックグラウンドで動かせるようにするのが Service である。また、Service の状態を変更させる場合には Activity プログラムから Service に接続して操作する。

• Content Provider

Content Provider は電話帳アプリや写真アルバムのようにアプリケーション自身が保持するデータを、他のアプリケーションに対して公開し、利用可能にする時に使う仕組みである。インタフェースをシステム上で公開しておくことで、ほかのアプリケーションは API 経由でデータの問い合わせや削除、更新、追加などができる。

2.3 利点

本節では、Android の利点を紹介する。

2.3.1 境界のないアプリケーション

Android 上のアプリケーションは、API を介して主要な携帯端末機能にアクセスできる。アプリケーションが他のアプリケーションの機能を使う際には Intent を使用する。Intent に受け渡すべきデータが入っていた場合、そのアプリケーションの Activity にデータを引き渡し、適切な処理が行われる。これにより、友人の住所と地図を連携させるといった、ウェブと携帯電話をシームレスに繋いだアプリケーションが構築できる。また、Android には Webkit がライブラリに含まれている。WebKit は Web ページの描画やアプリケーションとしてのインタフェースを形成するための土台として用いられている。その API を用いることで、HTML や JavaScript などをアプリケーションに容易に埋め込むことができる。

2.3.2 平等なアプリケーション

Android は、携帯電話に付属する基本的なアプリケーションとサードパーティ製のアプリケーションを差別しない。従って、ブラウザやメールなどの携帯電話機能の全部または一部をアプリケーションで置き換えることもできる。これは Intent の仕組みにより動作するアプリケーションを決定させることができるためである。起動アプリケーションの決定には Intent フィルタで記述されている値を参考にする。このようにアプリケーション名固定でなく、メール機能やブラウザ機能として Intent が発行し、それに該当するアプリケーションを起動するため、アプリケーションの着せ替えが実現できる。

2.3.3 並列実行可能なアプリケーション

Android では複数のアプリケーションを平行して実行できる。これはアプリケーションが自分自身のプロセスで動作するマルチプロセスシステムになっているため

である．全ての Android アプリケーションは，自身の Dalvik VM のインスタンスを用いてそれ自体のプロセス内で動作する．Dalvik はデバイスが複数の VM を効率的に動作できるように記述されている．これにより，アプリケーションのクラッシュ時に他のアプリケーションの巻き添えを防ぐことができる．また，一部例外的に root 権限で動くものを除いて，他のプロセスを見ることができないといったセキュリティを向上させることが可能である．

3 Android の現状

3.1 実用例

2008 年 10 月に初めて Android を搭載した携帯電話「G1」を米 T-Mobile 社が発売した．特徴は，携帯電話を最初に立ち上げた時に Gmail のアカウントの入力を求められることであり，全てを Gmail のアカウントで管理するようになる．また，Google Maps を利用した時に，G1 には GPS が内蔵されているため，所在地がすぐに表示でき，ストリートビューモードに切り替えることもできる．更に，Android Market というウェブサイトで，Android 携帯電話向けの新しいアプリケーションをダウンロードできる．しかし，日本語入力にはまだ対応していない³⁾．現在，G1 は半年で契約ユーザ数が 100 万件を突破してシェアを広げている．

3.2 動向

Google は，Android の発表と同時に，普及促進団体「Open Handset Alliance (以下 OHA)」を立ち上げており，現在 Google を始め，NTTDocomo や KDDI，Soft-Bank などの携帯電話会社，半導体会社，携帯電話メーカー，ソフトウェア会社，サポートパートナー企業など，併せて 48 社の企業が加盟する団体となっている．OHA では Android に対応した端末の開発やデバイスの展開，API の開発やパッチの提供を行っている⁴⁾．

更に Android を携帯電話以外の組み込み機器に利用することを目的とした一般社団法人「Open Embedded Software Foundation (以下 OESF)」が今年 3 月に発足した．OESF に Google は参加していないが情報交換は行われている．組み込み機器を開発する企業には，個別の製品の市場が小さいため，共通の開発環境がなく開発効率が良くなかった．そこで，組み込み機器の開発に Android を利用することで開発環境が共通化されれば，開発者は必要な機能を選択して製品を組み上げることができる．現在，情報家電や Set-Top Box，VoIP 関連，計測・制御機器，システム・コアなどのワーキング・グループが活動しており，携帯電話用の機能しか用意されていない Android に組み込み機器用の機能の拡張を試みている．Fig.2 に開発例を示す．企業向けの IP 電話機器に必要な機能としては，Session Initiation Protocol (以下 SIP) や Security Architecture for Internet Protocol (以下 IPsec) などの通信制御や暗号化のプロトコルが開発中である．Android にこれらの機能を追加することで IP 電話やデジタルテレビ，カーナビやオーディオ機器など

の GUI を備えた多様な機器に搭載できるようになる．

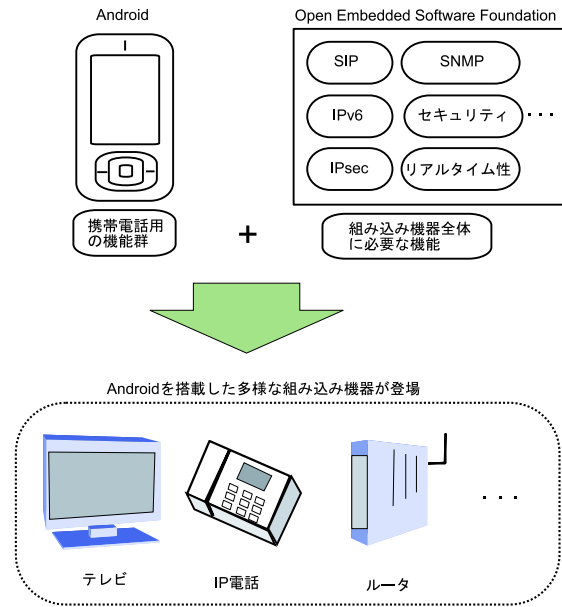


Fig.2 OESF の狙い (参考文献⁵⁾ より参照)

4 今後の展望

Android の普及に伴って，家電製品への導入も考えられる．例えば，冷蔵庫に Android を搭載すれば，冷蔵庫内を管理するソフトが開発され，賞味期限や在庫などを簡単に把握することもできる．冷蔵庫内情報から今ある材料を用いたレシピの検索も可能になるだろう．更に，現在はまだ Android の SDK に Bluetooth の API は用意されていないが，将来的には Bluetooth や無線 LAN によって家電製品同士の通信も可能であり，携帯電話からも状況を把握できると考えられる．

また，Google による Android 市場は始まったばかりである．今後 Android が生活に浸透していけば街中どこにいても公共の Android 搭載機器により，携帯電話がなくても自分のアカウントでインターネットを利用することができるようになるだろう．

参考文献

- 1) GSM World
<http://www.gsmworld.com/>
- 2) Tech-On
<http://202.214.174.10/article/NEWS/20081022/159959/?ST=mobile.PRINT>
- 3) 日経トレンドネット
<http://trendy.nikkeibp.co.jp/article/column/20090403/1025211/?P=3>
- 4) goo ビジネス EX
<http://bizex.goo.ne.jp/tool/it/17163/5906/>
- 5) ITpro
<http://itpro.nikkeibp.co.jp/article/COLUMN/20090225/325488/?SS>