

Android

田辺 竜也, 鍵谷 武宏

Tatsuya TANABE, Takehiro KAGITANI

1 はじめに

近年携帯電話メーカーは、これまでの独自開発による OS よりも、専業開発による汎用 OS を採用する流れが大きくなってきている。これは携帯電話の多機能・高機能化により大きくなった開発費の削減、および開発期間の短縮を狙ったことである。

しかし現在、携帯端末に汎用 OS を用いてはいるものの、その上で動くアプリケーションの開発環境は携帯電話会社により異なっている。これがアプリケーション開発者への負担となり、高速サイクルが求められる携帯電話において 1 つの障害となっていた。そこで、その問題を解決するものとして OS と開発環境が 1 つとなった Android が登場した。以下に Android の OS として、また開発環境としての特徴について示す。

2 現在の携帯電話について

現在の携帯電話は多機能・高機能化により、1 台当たり 100 億円もの開発費が必要となり、その大部分をソフトウェアが占めるという状態である。また、ソフトウェアが複雑なものになるため、事故が多発した。このため、各携帯電話メーカーは自社独自の OS から、品質が安定して汎用性の高い専業開発の OS を採用することが主流となりつつある。また、携帯電話を独自 OS から汎用 OS に切り替えることで、それぞれの携帯電話にあった独自性が失われないようにするため、アプリケーションの開発環境は携帯電話会社が独自のものを構築している。

以下に携帯電話向け汎用 OS と、アプリケーション開発環境について示す。

2.1 現在の携帯電話用 OS

- Symbian OS

英 Symbian 社が提供している携帯電話向け OS。スマートフォンにおいては現在 70% と圧倒的シェアを誇っている。メモリの節約、省エネなど、携帯電話向けに特化した OS であるが、そのためかなり複雑な C++ コードとなっている。

- Windows Mobile

Microsoft 社の携帯向け OS ブランド。他のアプリケーションを起動したまま、他のアプリケーションを起動して作業できるマルチタスクが大きな特徴。しかし、アプリケーションを起動し続けていると、実行に必要なメモリを圧迫するため、端末の動作速度が遅くなることがある。

2.2 現在の開発環境

- S60

Nokia が開発した Symbian OS 用のアプリケーションプラットフォーム。個人が自由にネイティブアプリケーションを作成、実行できるので、海外でのシェアが圧倒的に高い。

- MOAP

NTT ドコモが中心となって推進している FOMA 向けの共通プラットフォーム。具体的には、OS とミドルウェア、ブラウザなどが含まれている。MOAP は Symbian OS に対応する MOAP(S)、Linux に対応する MOAP(L) という 2 つのバージョンがある。

- BREW

CDMA 携帯電話向けアプリケーションのプラットフォーム。携帯電話間の仕様の違いを吸収し、単一のプログラムで多数の携帯電話に対応できるように設計されている。

2.3 アプリケーション開発の問題点

多くのユーザー確保のために、携帯会社が幅広いニーズに答えようとしているので、そのぶん様々な開発環境が生まれてきている。なぜなら、どの会社も同一の開発環境でアプリケーションの開発を行えば、携帯電話の機種に関係なく利用できるアプリケーションやサービスになるので、携帯電話会社は自身の差別化ができなくなるからだ。しかし、これはアプリケーション開発者にとって大きな負担となり、結果的に新たなアプリケーションやサービスの開発が遅れることになる。

3 Android

3.1 Android とは

Android は 2007 年 11 月 5 日に Google から発表された携帯電話向けのオープンソースの汎用プラットフォームである¹⁾。この Android には OS、ミドルウェア、ユーザーインタフェース、コアアプリケーションや SDK などをすべて含む、包括的な携帯電話開発ツールである。この Android をベースにした最初の携帯電話は、2008 年下半期にも商品化が見込まれている。

3.2 Android の構成

Android のアーキテクチャは Fig.1 に示すように、全部で 5 つの層で構成されたモジュラー型²⁾である。これは、モジュール構造にすることで、ハイエンドな端末にもローエンドな端末にも幅広く対応することができるからである。またこの構造は、コスト削減と開発期間短縮に

も貢献している。Android の構成内容¹⁾ を以下に示す。

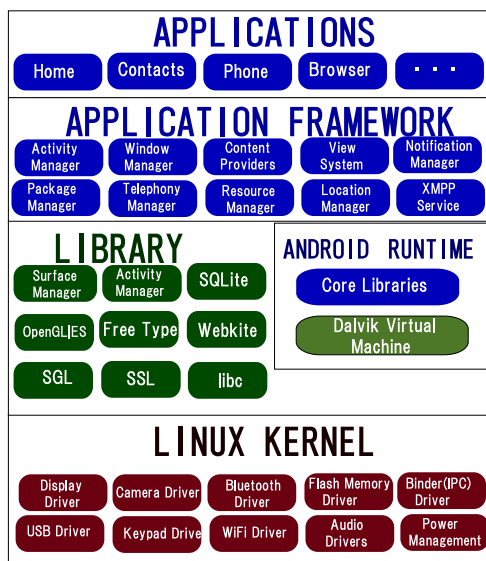


Fig.1 Android の構成 (出典:自作)

● Applications

e-mail クライアント, SMS プログラム, カレンダー, 地図, ブラウザ, アドレス帳などの携帯電話で共通に使われるコア・アプリケーションがあらかじめインストールされている。なお, 全てのアプリケーションが, Java 言語で実装されている。

● Application Framework

アプリケーションから使用可能なフレームワーク。この中にはコア・アプリケーションが使用している部品 (コンポーネント) を多数まとめており, 開発者もアプリケーションと同様に使用できる。

● Libraries

Android システムのあらゆるコンポーネントによって使用されている C/C++ のライブラリ集。このライブラリは Application Framework を通じて開発者へ公開される。

● Android Runtime

Android の全てのアプリケーションは, Runtime 上で起動する。Runtime 環境は独自で開発された Dalvik VirtualMachine が採用され, 開発言語は Java で行う。

● Linux Kernel

セキュリティ, メモリ管理, プロセス管理, ネットワークスタック, ドライバモデル等のコアシステムを Linux version 2.6 の機能に頼っている。

3.3 開発者にとってのメリット

3.3.1 ライセンスの一本化

現在, 携帯電話用のソフトウェアの多くはサードパーティーからライセンスを取得する必要がある。³⁾ 例えば, Web ブラウザに Opera を搭載するにしてもライセンスを受ける必要があるし, Midi の着信音を再生するなら

Midi シーケンサーのライセンスを受けなくてはならない。OS や各アプリケーションをあわせると, 1 つの携帯電話に必要なライセンスの数が増え, それだけ膨大な法的手続きが必要になる。これは携帯電話の改良および開発を目指す人たちにとって, 大きな障害となる。Android には様々なアプリケーションを包括し, かつそれを 1 つのライセンスで提供している。これによってライセンス取得にかかる面倒な手間が簡略化され, また開発期間の短縮につながる。

3.3.2 新しいサービスの実現

Android を利用すれば携帯電話のネット上でもマッシュアップが容易になる。⁴⁾ マッシュアップとは, ウェブ上に公開している API の機能に独自の機能を組み合わせて新しい Web 上のサービスを作る手法のことである。PC を使っている場合, Google の提供している Web サービスをマッシュアップして, 新しいサービスを生み出すことが容易にできる。しかし携帯電話ではサービスそのものに制限もあり, そうしたマッシュアップができないことが多かった。オープンソースプラットフォームである Android が普及すれば, これまで垂直統合されていた携帯電話市場が水平分業に向かい, サードパーティやベンダーが提供するモバイル向けアプリケーションを交互に利用するサービスが生まれ, 携帯電話市場でもマッシュアップを図れる。

なお Android の API はいくつかのコードに分割され, ライブラリとして実装されている。以下に Android でマッシュアップに利用できる API をいくつか紹介する。

- Surface Manager : サブシステムの表示へのアクセス管理や複数のアプリケーションの 2D, 3D のグラフィックレイヤーを違和感なく統合できる。
- WebKit : ウェブエンジンにはアップルが中心となって開発されているオープンソースの WebKit を採用している。SVG・RSS・タブブラウズといった最新のブラウザが備える機能に対応する。
- OpenGL | ES : 3D グラフィックスのためのプログラムインタフェースである OpenGL の 3D ライブラリ。様々な機械で 2D/3D グラフィックを画面に表示させることができる。

3.4 携帯電話会社にとってのメリット

開発者にとって, 異なる携帯端末でも同一の環境でアプリケーション開発を行えるのは大きなメリットである。しかしそうすると, 携帯電話会社にとって「他社との差別化を図ることができるか」というのが大きな問題点となる。この点において, Android はライセンスに Apache License 2.0 を採用している。このライセンスが採用されたのは, Android 採用企業が開発したソフトウェアを Open Handset Alliance に提供する必要がなく, 商業化しやすいライセンスだからである。たとえば他のフリーソフトウェアライセンスには, Free Software Foundation (FSF) の GNU General Public License (GPL) があ

る。このライセンスの下で配布されているソースコードを自社製品に利用した場合、その製品のソースコードは開示しなくてはならない。Apache License はこれを要求していない。ソースコードを公開することは Apache License でも可能であるが、その判断は個人や組織に任せており、必須ではない。このため、Android をベースに、一部の機能を取り除いたり、独自の機能を付け加えたりしたものを自社の商用製品として売り出すことが出来る。

4 今後の展望

Google はオープンなプラットフォームを開発・推進していくための企業連合 OHA (Open Handset Alliance) を設立したり、賞金総額 1000 万ドルにものぼるアプリケーション開発コンテストを行うなど、Android により新しいサービスが生まれるチャンスを数多く作っている。これにより Android のキラーアプリケーションといえるものが生み出されれば、今後 Android が携帯電話の主流になる可能性も十分ありえる。

また Google は携帯電話以外の携帯用端末にも Android を採用することも視野に入れている。携帯電話以外のあらゆる端末から Google は情報を収集できるようになれば、Google は膨大な情報を手に入れることができ、それらを整理することで、更に新たなサービスの提供ができるかもしれない。

参考文献

- 1) フリー百科事典 Wikipedia
<http://ja.wikipedia.org/wiki/Android>
- 2) IT 用語辞典 e-Words
<http://e-words.jp/w/E383A2E382B8E383A5E383A9.html>
- 3) ソフトウェアライセンスを一本化する Google の Android - ITmedia エンタープライズ
<http://www.itmedia.co.jp/enterprise/articles/0711/08/news068.html>
- 4) 「携帯でもマッシュアップを」—Google の「Android」
<http://www.itmedia.co.jp/news/articles/0711/06/news071.html>
- 5) Google-OHA, Android の携帯革命
<http://alexa.livedoor.biz/archives/51060397.html>
- 6) YOMIURI ONLINE(読売新聞)
<http://www.yomiuri.co.jp/atmoney/news/20080322-OYT1T00820.htm>
- 7) 「Android」アプリコンテストを開催
<http://www.itmedia.co.jp/news/articles/0711/13/news015.html>
- 8) Android とプラットフォーム
<http://blog.novsix.com/2007/12/google200712.html>
- 9) 携帯電話-フリー百科事典 Wikipedia
<http://ja.wikipedia.org/wiki/>