

FOMA 端末ソフトウェア プラットフォーム “MOAP”の開発

FOMA 端末のソフトウェアを開発する際に活用可能な Linux OS および Symbian OS^{*1} 向けのソフトウェアプラットフォーム “MOAP” を開発した。このプラットフォームにより品質向上、コスト低減、開発期間の短縮が図れる。

つじ ひろゆき おおの けいすけ さいとう てつ
辻 宏幸 大野 圭祐 齊藤 哲

1. まえがき

1999年のi-modeサービスの導入以降、携帯電話サービスは音声通話主体のサービスから、Webサイトの閲覧、電子メールの送受信、着信メロディおよびJava^{*2}アプリといったコンテンツのダウンロードなど、データ通信主体のサービスに移行した。また、2001年に導入されたFOMA (Freedom Of Mobile multimedia Access) 端末では、高速データ通信を活かしたTV電話やi-motionなどの映像系サービスが開始された。

このような多様で高性能なサービスを実現するために、移動端末ソフトウェアの複雑化が進み、開発規模も爆発的に増大している。移動端末ソフトウェアの複雑化や開発規模の増大は、移動端末ベンダとドコモにとって開発期間の遵守やソフトウェア品質の確保といった面で大きな負担となっている。

これらの問題を解決するためにドコモでは、複数の移動端末ベンダやソフトウェアベンダが、移動端末ソフトウェアの開発において共通に使用できるソフトウェアプラットフォームであるMOAP (Mobilephone Oriented Application Platform) の開発を行った。

本稿では、2004年11月に導入したMOAPの開発概要について述べる。

2. 移動端末のソフトウェア開発の問題点

従来の開発では、移動端末ベンダが、それぞれ独自にア

*1 Symbian OSおよびすべてのSymbian関連の商標およびロゴはSymbian Ltd.の商標または登録商標である。

*2 Java：米Sun Microsystems社が提唱しているネットワークに特化したオブジェクト指向型開発環境。

アプリケーション、ミドルウェア、OS (Operating System)、デバイスドライバといった移動端末ソフトウェア全体を開発していた。しかし、サービスが多様化するのに伴い、限られた開発期間ですべてのソフトウェアを1社で開発するのは困難となってきたため、一部のソフトウェアについては、ソフトウェアベンダから購入して搭載するようになった。

ところが、ソフトウェアベンダ製のアプリケーションを搭載する場合には、移動端末ベンダ独自のミドルウェアとのインタフェース部分の開発が必要となる。これは、ソフトウェアベンダにとって、大きな負担となって来る。また、アプリケーションの高機能化に伴い、ミドルウェア部分においてもソフトウェアの複雑化と開発規模の増大は止まらず、保守作業の負担が無視できなくなってきた(図1)。このような状況において、今後も高度なサービスを開発し提供していくためには、移動端末ソフトウェア開発の大幅な効率化が必要となる。

そこで、図2のように、移動端末ベンダ間あるいは移動端末の機種間でOSとミドルウェアなどの基盤ソフトウェアをそれぞれ共通化することにより、移動端末ソフトウェアの開発規模が削減される。さらに、ソフトウェアベンダ製のアプリケーションと移動端末ベンダ独自のミドルウェアとのインタフェース部分の開発が不要となり、移動端末開発におけるQCD(品質:Quality, コスト:Cost, 納期:Delivery)の向上を図ることができる。

このように、移動端末ソフトウェア開発における問題点を解決するため、複数の移動端末ベンダ、ソフトウェアベンダが共通に使用できるソフトウェアプラットフォーム“MOAP”を開発した。

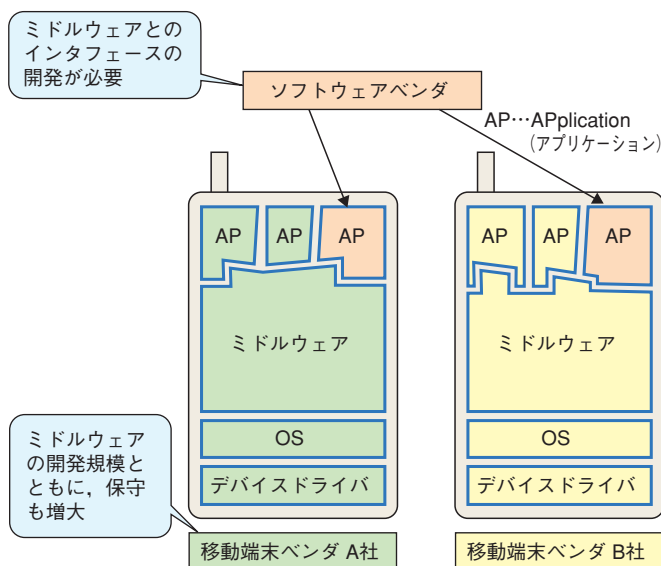


図1 従来の移動端末の開発

3. ソフトウェアプラットフォーム

3.1 高機能OS

移動端末ソフトウェアにおける機能充実を考えた場合、大きな懸念となるのが、そのベースとなるOSの機能である。これまで移動端末のOSには、 μ ITRON (Industrial TRON) などのリアルタイムOS (以下、RTOS (Real Time Operating System)) が採用されていた。これはリアルタイム性能を重視し、非力なCPU (Central Processing Unit) のパワーを効率的に引き出し、応答性の高いアプリケーションの開発が可能であったためである。

しかし、RTOSには、複数のアプリケーションを、安定して動作させる機能が乏しいことや、ツールが整備されておらず、開発に特殊な知識とスキルが要求されるという問題点があった。

そこで、開発効率の向上と、今後の高度なハードウェアへの対応をにらんで、MOAPでは、機能の拡張性も高く、複数のアプリケーションを障害なく動かすためのマルチスレッド、メモリ保護といったソフトウェア管理機能を標準でサポートしている高機能OS “Linux OS”, “Symbian OS” を採用した。この2種類の高機能OSをベースに、Linux OS向けプラットフォーム“MOAP(L)” およびSymbian OS向けプラットフォーム“MOAP(S)” を構築している。

3.2 ミドルウェア

MOAPでは、上述のLinux OSとSymbian OSをベースに、以下のようなミドルウェアを構築している。

- (1) ユーザインタフェース (UI: User Interface)

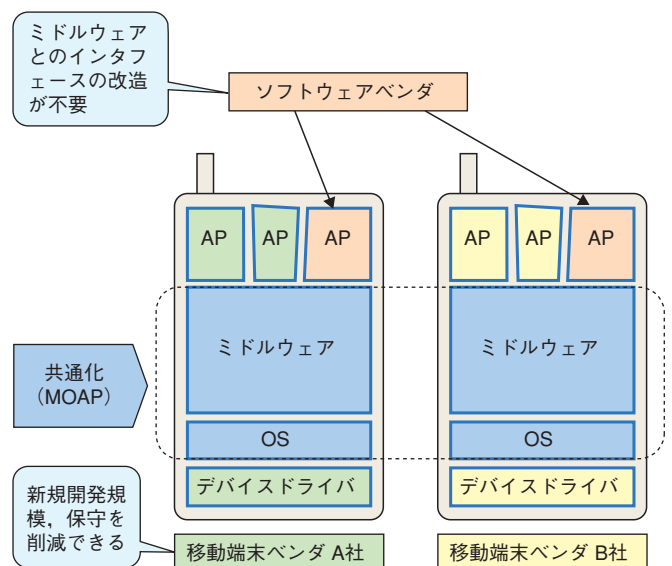


図2 MOAPによる移動端末の開発

MOAPでは、ウィンドウやボタン、リストボックスなどの表示部品とキーボードなどによる操作といった一般的なUIの機能に加えて、テンキーと十字キーだけで十分な操作性を実現するために、例えばボタン長押しなど移動端末特有のUI機能を提供している。これらを用いることにより移動端末特有の操作に対応したアプリケーションを容易に開発できるようにしている。

MOAP(L)では、Linux OSにおいて広く使われているUIであるX-WindowとGTK+(Gimp ToolKit +)をカスタマイズすることによって対応している。またMOAP(S)では、OSが提供するUIの上位に独自のオブジェクト指向ライブラリを構築することによって対応している。

(2) アプリケーション管理

移動端末では、電話帳やメールなど複数のアプリケーションを協調して動作させる必要がある。そのため、MOAPでは、キー入力や電話の着信などを契機としたアプリケーションの開始終了処理機能、アプリケーション間でデータをやりとりするための通信機能、複数のアプリケーションが液晶ディスプレイ(LCD: Liquid Crystal Display)やスピーカなどを交互に利用する際の排他管理機能、そして、移動端末の起動時や終了時のシーケンス管理機能などを有している。加えて、アプリケーションの動作状態を常時監視し、異常な状態が発生した場合における速やかな障害検出と障害情報の記録機能を有している。

(3) ストレージ管理

カメラの画素数の向上や動画、音楽データの取扱いなどによって、移動端末のアプリケーションが処理するデータのサイズも肥大化してきた。これに伴い、移動端末に装備されるストレージデバイスも内蔵フラッシュメモリだけでなく、SDメモリカード、メモリースティックなど多様化してきている。

MOAPでは、これら多彩なストレージデバイスに対して、統一されたデータの書込み/読出し/削除などの機能を提供している。

(4) 装置状態管理

移動端末のアプリケーションは、移動端末の開閉状態、電界強度や電池残量、マナーモードや端末ロックの状態などを認識し、これに対応した動作をしなければならない。

MOAPでは、これらの情報を各アプリケーションが個別に管理する必要がないように、これらを一元管理する装置状態管理の機能を提供する。また、例えば移動端末の開閉など、それぞれの状態の変化が、アプリケーションへ自動的に通知されるようになっている。

(5) 移動通信ネットワークを使用した通信機能

移動通信ネットワークを使用した通信には、音声通話やTV電話などの回線交換による通信やi-modeなどのパケット交換による通信などの種別があり、それぞれについて特定の制御や処理が必要となる。

MOAPでは、それぞれの通信種別ごとに必要な制御機能を提供する。例えば、回線交換による通信では、留守番電話やドライブモードといった付加サービスに関する設定機能などを提供している。パケット交換による通信では、パケット通信の開始や終了などの制御機能などを提供している。

(6) アイコン管理

受信電界の強度や電池残量、外部メモリの装着、メールの着信や未読、SSL(Secure Sockets Layer)や赤外線などといった通信状態などを知らせるために、LCD画面上にさまざまなアイコンを表示させている。

MOAPでは、この仕組みを統合的に実現するために、状態変化に応じて、自動的にアイコンを表示/変化させる機能を有している。

3.3 開発環境

以上のような充実したOSとミドルウェア上において、移動端末ベンダ、ソフトウェアベンダがアプリケーション開発を進めることをサポートするため、MOAPでは移動端末のソフトウェア自体に加えて、ソフトウェアの開発環境を提供する。この開発環境を用いることにより、PC上での効率的なアプリケーション開発を行うことが可能である。

MOAP(L)では、Linux OS上で利用可能なエディタやデバッガ、構成管理ツールが利用可能である。MOAP(S)では、Microsoft社の統合開発環境であるVisual C++^{®*3}および関連製品が利用可能となっている。また、それぞれのプラットフォームにおいて、PC上でアプリケーションの動作確認ができるように、移動端末エミュレータが提供され、移動端末上でのソフトウェアの動作を確認することが可能となっている。写真1にMOAP(L)の移動端末エミュレータを示す。

4. あとがき

本稿では、移動端末ソフトウェアの複雑化と開発規模の増大に対応するために開発したMOAPの概要について述べた。MOAPは移動端末ソフトウェアの開発に必要な共通的な機能を提供することにより開発効率の向上を図ってお

*3 Visual C++[®]は、米国Microsoft Corporationの、米国およびその他の国における登録商標または商標である。



写真1 MOAP(L) 移動端末エミュレータ

用語一覧

CPU : Central Processing Unit
 FOMA : Freedom Of Mobile multimedia Access
 GTK+ : Gimp ToolKit +
 ITRON : Industrial TRON
 LCD : Liquid Crystal Display (液晶ディスプレイ)
 MOAP : Mobilephone Oriented Application Platform
 OS : Operating System
 RTOS : Real Time Operating System
 SSL : Secure Sockets Layer
 UI : User Interface (ユーザインタフェース)

り、すでにFOMA 901iシリーズに適用されている。

今後は、さらに高機能化、複雑化が進むドコモの移動端末開発において、より効率化を高めていくため、MOAPのさらなる共通化機能の拡充を図るとともに、移動端末ベンダ、ソフトウェアベンダに対してタイムリーに提供していくことが重要である。

また、ソフトウェアプラットフォームの将来的な課題として、ハードウェアやOSの違いをミドルウェアで隠蔽することによるハードウェア抽象化やマルチプラットフォーム化の強化、ミドルウェアでの多言語や国外仕様への柔軟な対応による国際展開についても、検討を行っていく。