

携帯電話機のソフトウェアプラットフォーム

太田 賢 (株) NTT ドコモ 総合研究所
ohtak@nttdocomo.co.jp

鈴木 敬 (株) NTT ドコモ 総合研究所
t_suzuki@nttdocomo.co.jp

照沼和明 (株) NTT ドコモ 移動機開発部
terunuma@nttdocomo.co.jp

携帯電話機には、iモード等のモバイルインターネット機能の搭載以降、従来の音声通話機能に加えて、データ通信を利用した多彩なサービスを提供するために多くの機能が搭載されている。さらに、データ通信の高速化、ハードウェア性能の向上に合わせて高度化するサービスに呼応して、携帯電話機に搭載される機能は増加を続けている。このような背景のもと、ソフトウェア開発規模は急激な拡大傾向にある(日経ビジネス 2005.3.28)。

急激なソフトウェア開発規模増大は、コスト増加、不具合発生リスクの増加をもたらす。よって開発規模を抑えながら、タイムリーに新規サービスの導入を図るため、高機能な汎用 OS、共通的な機能やインターフェースを備えるミドルウェア、PC エミュレータによる開発環境等を含むソフトウェアプラットフォームの重要性が高まっている。本稿は、携帯電話機の全体構成とソフトウェア開発の背景について述べた後、ソフトウェアプラットフォームの構成要素として、1. 基盤となる汎用 OS、2. アプリケーション管理やマルチメディア機能、システム制御などの共通機能を提供するミドルウェア、3. 開発環境を事例とともに解説した後、今後のソフトウェアプラットフォーム技術の展望を述べる。



携帯電話機とソフトウェア

◎携帯電話機の構成

携帯電話機は、移動体通信の基本サービスである電話やデータ通信等を提供する際、電波状態の変動や高速移動に対応しつつ基地局と連携して動作するため、厳格なリアルタイム処理が必要とされる。従来、携帯電話機は1つのCPU上でリアルタイムOS (RTOS: Real Time Operating System) を動作させ、通信処理やその他のアプリケーション処理を実行する1CPU構成をとっていた。しかし、第3世代携帯電話機では高速のデータ通信を実現するため、通信処理の処理負担が重くなり、1つのCPUでアプリケーション処理も同時に実行させることが難しくなった。そこで、従来のCPUをもっぱら通信処理用とし、アプリケーション処理用のCPUを新たに導入する2CPU構成の機種が増えてきた(NIKKEI MICRODEVICES, April 2004)。通信処理用のCPUはC-CPU (Communication Central Processing Unit) と呼ばれ、RTOSを動作させるのに対し、アプリケーション処理用のCPUはA-CPU (Application Central Processing Unit) と呼ばれ、SymbianやLinuxなどの汎用OS (OS: Operating System) を動作させる。2CPU構成は、処理性能の向上、A-CPU上の汎用OSの搭載によるソフトウェ

ア開発効率の向上、RTOS側の通信処理用のソフトウェア資産の継承という利点があるのに対し、1CPU構成は低コスト、少面積、省電力において有利である。主に低価格帯の機種は1CPU構成、高価格帯の機種が2CPU構成をとっているが、1CPU構成においてはC-CPUの高性能化、2CPU構成においてはC-CPUとA-CPUの1チップ化による低コスト化、省電力化の取り組みがなされている。

図-1に、文献1)に示されている1チップ化された2CPU構成の携帯電話機アーキテクチャの一例を示す。ここでは、ベースバンド処理部は、C-CPUとW-CDMA (Wideband Code Division Multiple Access) 回路、GSM (Global System for Mobile Communications) / GPRS (General Packet Radio System) 回路、および、これらを接続する内部バスから構成され、W-CDMA方式とGSM/GPRS方式のデュアルモード通信のベースバンド処理を行う。C-CPUには、RTOSであるμITRON (Micro Industrial TRON (The Real-time Operating system Nucleus)) が搭載され、通信制御プロトコルスタックが実装される。通信制御プロトコルスタックでは主に、携帯電話機と移動網間の無線回線制御、移動管理・呼制御、UIM (User Identity Module) 情報の読み出し・書き込み処理が行われる。

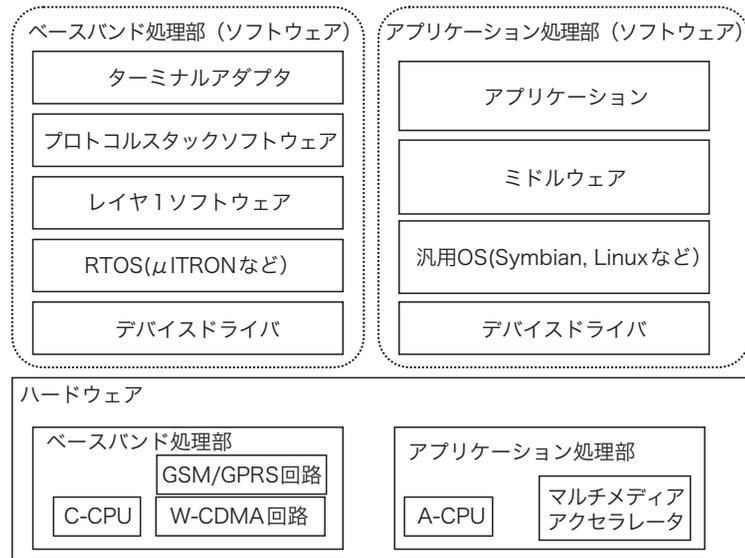


図-1 2CPU構成の携帯電話機アーキテクチャの一例

一方、アプリケーション処理部は、A-CPUとマルチメディアアクセラレータから構成され、OS、ミドルウェア、動画やオーディオ等のマルチメディアアプリケーションを動作させる。マルチメディアアクセラレータは、演算量の大きいマルチメディア処理に特化したハードウェアやDSP(Digital Signal Processor)であり、A-CPUの負荷を軽減することによる低消費電力化とマルチメディアアプリケーションの高速処理を実現する。

本稿はこの2つの処理部のうち、アプリケーション処理部に焦点を当てる。アプリケーション処理部は、携帯電話サービスの多様化、高機能化に伴って開発規模が急激に増加している部分であり、次節で述べるように、開発コストの増加、ソフトウェア品質の低下などの解決すべき課題を抱えている。

◎携帯電話機ソフトウェア開発の背景

携帯電話機は、電子メールやWWWブラウザなどのモバイルインターネットへの対応、デジタルカメラや音楽再生、テレビ、非接触ICカードなどの多様な機能の取り込み、Java等のアプリケーション実行環境の提供など、パソコン化と多機能化が進んでいる。その進化とともに、携帯電話機のソフトウェア規模は急激に拡大しており、2005年には500万行に達していると言われていた(日経ビジネス2005.3.28)。その一方で、利用者は“電話”端末としての信頼性を要求し、どのようなアプリケーションや機能を実行中でも電話やメールの着信を正しく受け付けること、フリーズや異常終了することなく、長時間動作することを期待する。

携帯電話機の開発には、次々に追加される新しい機能やサービスに対応しつつ、高い信頼性と品質、タイムリーな納期、開発コストの抑制などの要求を満足することが求められている。これに対し、従来の携帯電話機のアプリケーション処理部のソフトウェア開発においては、以下の3つの課題があった。

第1の課題として、携帯電話機の高機能化に伴い、リッチなユーザインタフェースを提供するためのウィンドウシステムや各種プロトコルスタック、ファイルシステムなど、OSおよびミドルウェアの機能性の向上が求められている。一方、従来の携帯電話機のアプリケーション処理部は1CPU構成において、CPUやメモリの制約の厳しい環境でも動作可能な μ ITRONなどのRTOS上に構築されていたが、多くのRTOSが標準的に提供する機能は、カーネルとシステムコール、一部の通信プロトコル等に限定される場合が多く、各携帯電話機ベンダ(端末ベンダと呼ぶ)が独自にウィンドウシステム、ファイルシステム、通信プロトコルスタックなどの機能を整備する必要があった。また、携帯電話機のサービスや機能の実現において、マルチタスクやアプリケーション間の連携を必要とする際、RTOSでは複数のアプリケーションを安定して動作させるための機能が乏しく、その制御に複雑な開発が必要であった。

第2の課題として、サードパーティ製のソフトウェア部品の取り込みのため、共通的なミドルウェアやAPIの整備が求められている。端末ベンダは当初、アプリケーション、ミドルウェア、OS、デバイスドライバを含む携帯電話機ソフトウェア全体を開発していたが、サー

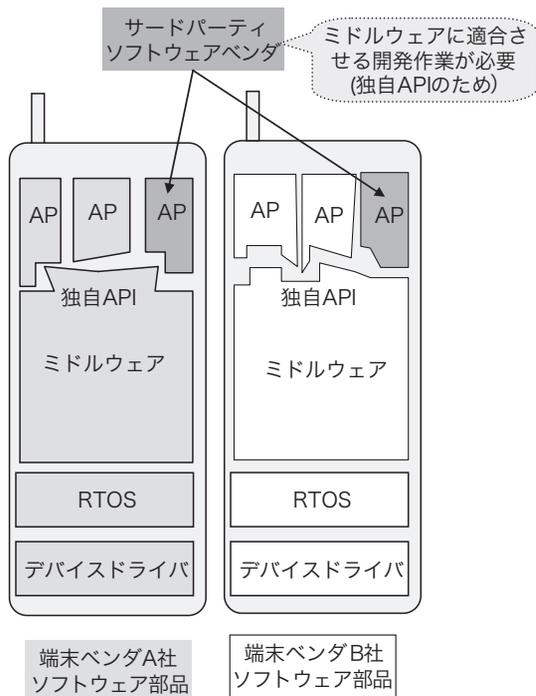


図-2 従来の携帯電話機開発

ビスの多様化に伴って開発するソフトウェア部品も増加し、1社ですべてをまかなうことが困難となった。そのため、サードパーティ製のソフトウェア部品を搭載するようになったが、図-2に示すように、端末ベンダが独自に整備したミドルウェアとAPIへ適合させるための開発作業が必要であり、そのための開発コストが発生していた。

そして第3の課題としては、携帯電話機の振る舞いを再現するPCエミュレータや試験やデバッグのためのツールなどの開発環境についても、端末ベンダが個別に整備する必要があるが、ソフトウェアベンダが利用できる標準的な開発キットがなかった。実際に携帯電話機のソフトウェアが動作する環境だけでなく、開発環境の整備も、ソフトウェア開発コストの増大や品質低下の問題を解決する上で重要な課題である。



ソフトウェアプラットフォーム

携帯電話機のソフトウェアプラットフォームは、前記課題を解決するために提供されるソフトウェア群として捉えることができ、(1) 複数のアプリケーションを安定して動作可能で、機能の拡張性も備える高機能・汎用OS、(2) 共通のミドルウェアおよびAPI、(3) 開発環境、の組として定義できる。代表例としては、Symbian OS向けのUIQ (User Interface Quartz) や Series 60、OS

とミドルウェアを統合的に提供している Windows Mobile、REX (Realtime Executive) OS 向けの BREW (Binary Runtime Environment for Wireless)、NTTドコモが組み込み Linux と Symbian OS 用に提供している MOAP (Mobilephone Oriented Application Platform)^{2), 3)}がある。MOAPは組み込み Linux 用が MOAP (L)、Symbian OS 用は MOAP (S) と呼ばれているが、本稿では以降で MOAP (L) について述べる。その他、Java 仮想マシン上で動作する J2ME (Java2 Micro Edition) も、標準的な API (クラスライブラリ) の規定や開発環境の提供などソフトウェアプラットフォームとしての性質を備えている。

本章では以下、高機能・汎用 OS として、組み込み Linux、Symbian OS、Windows Mobile の3つを概観した後、ミドルウェアおよび開発環境について述べる。

◎高機能・汎用 OS

携帯電話機用の高機能・汎用 OS として、組み込み Linux や Symbian OS、Windows Mobile が挙げられる。これらは複数のアプリケーションを安定して動かすためのマルチタスク/マルチスレッド、メモリ保護機能を持ち、各種ネットワークに対応するプロトコルスタック、ファイルシステムなど豊富な機能を備える。マルチメディアアプリケーションやゲームアプリケーションなどはリアルタイム性を要求する。リアルタイム性を満足するには、OSはプリエンティブ(高優先度タスクを低優先度タスクの前に割り込ませるなど、OSがタスクの切替えと優先度を管理すること)、かつ、ディターミニスティック(タスクの切替え・割り込みが事前に決められた時間内に完了すること、ハードリアルタイムを意味する)であることが要求される。

組み込み Linux はプリエンティブなマルチタスク/マルチスレッド OS であり、GNU GPL (The GNU General Public License) に基づくオープンソースであることが特徴である。しかし、従来の RTOS と比較して、動作に CPU やメモリなどのリソースを大量に要求すること、非ディターミニスティックであることの2つの課題があった。これに対し、組み込み Linux ベンダによって、Linux が備える機能を取捨選択可能とすることによるリソース消費の削減、スケジューラの改良によるリアルタイム性の向上、起動時間の短縮などの取り組みがなされている。組み込み Linux として、たとえば MontaVista 社の MontaVista Linux がある。

次に、英 Symbian 社の Symbian OS は、プリエンティブかつディターミニスティックなマルチタスク/マルチスレッド OS であり、前身の Psion 社による EPOC 32 オペレーティングシステムから派生した OS である⁶⁾。前述の Linux がモノリシックカーネルであるのに対して、

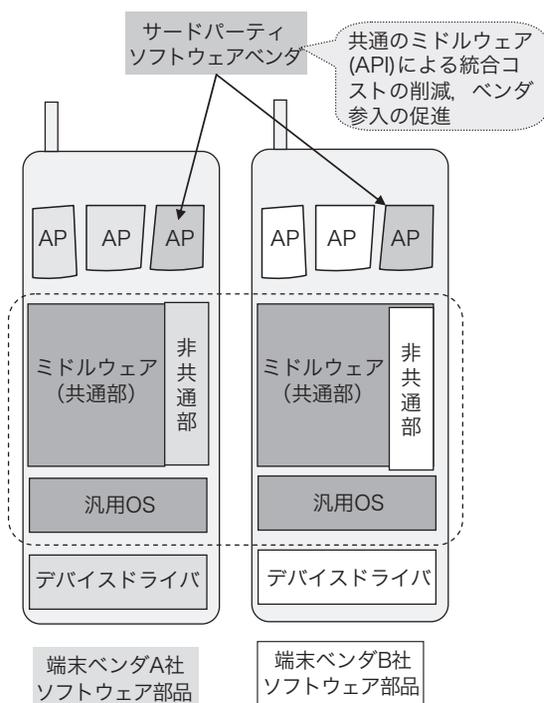


図-3 共通のミドルウェアを利用した携帯電話機端末開発

Symbian OSはマイクロカーネルのアーキテクチャを採用しているのが特徴である。マイクロカーネルは、OSが担う各種機能のうち、メモリ管理、プロセス制御、プロセス間通信などの必要最小限の基本機能をカーネル空間に残し、それ以外を上位のユーザ空間にサービスとして実装する形のアーキテクチャである。一方で、カーネルにファイルシステムやデバイスドライバ、ネットワークなど、さまざまな機能を実装するアーキテクチャはモノリシックカーネルと呼ばれる。Symbian OSでは割り込み、スケジューリング、スレッド同期などの最小限の機能のみをカーネルに組み込み、通信やグラフィクスなどその他の機能はカーネル上で動作するリアルタイム性が保証されたタスクとして実装する。

最後に、マイクロソフト社のWindows Mobileは、携帯電話機/PDA向けのOS、ミドルウェア、アプリケーションを含む統合パッケージであり、そのOSはWindows CEをベースとしている。Windows CEはプリエンティプかつディターミニスティックなマルチタスク/マルチスレッドOSである。

◎ミドルウェア

共通のミドルウェアの整備は、図-3に示すように、1. ソフトウェア部品の移植・統合コストを削減し、携帯電話機開発のコスト増加を抑制すること、2. サード

パーティのソフトウェアベンダの参入を促し、アプリケーションやミドルウェアなどの開発・流通の促進をはかること、などの効果があると考えられる。多くのミドルウェアは、SDK (Software Development Kit) やPCエミュレータなどの開発環境と一緒に提供される。

ミドルウェアが備える機能やAPIを、NTTドコモが提供するMOAP(L)を例として説明する。図-4に示すように、MOAP(L)は主に1. ユーザインタフェース、2. アプリケーション管理、3. ストレージ管理、4. 装置状態管理、5. 電話/通信機能を実装している。また、PCエミュレータと試験ツールを含む開発環境も開発されている^{3), 4)}。

- (1) ユーザインタフェース (UI: User Interface) : 携帯電話機のキー配置に基づくキーパッド入力処理やフレームバッファ制御、着信ランプやパイプレータ制御などの低レベルの入出力から、ラベルやボタン、リストボックスなどのGUI部品(ウィジェット)、アイコン表示制御(メールの着信表示や電池残量の点滅表示など)、携帯電話機特有の文字入力制御など上位のUIまでのAPIを提供している。MOAP(L)では、Linuxシステムで広く使われているX-WindowとGTK+(The GIMP(GNU Image Manipulation Program) Toolkit+)をベースにUI機能を実装している。
- (2) アプリケーション管理 : 携帯電話機では、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、アプリケーションの切替えや起動、終了、動作状態の取得のためのウィンドウマネージャの制御、アプリケーション間の通信制御、複数のアプリケーションからのリソース利用要求を統合的に交通整理するための競合管理のためのAPIを提供している。
- (3) ストレージ管理 : 内蔵フラッシュメモリや、SDカードやメモリスティックなど外部ストレージへのアクセスのためのAPIを提供している。
- (4) 装置状態管理 : 携帯電話機の開閉状態、電池残量、外部電源供給、ヘッドホン接続などの状態の取得および、その状態の監視のAPIを提供している。たとえば、携帯電話機の開閉などの状態変化をウィンドウマネージャに通知させることができる。
- (5) 電話/通信機能 : 音声通信やパケット通信、SMS、着信音、録音再生などの制御のためのAPIを提供している。たとえば、C-CPU側のベースバンド処理部を通してTCP/IPベースのパケット通信における通信開始/終了制御や、電話サービスにおける留守番電話やドライブモードといった付加サービスの設定を行うことができる。

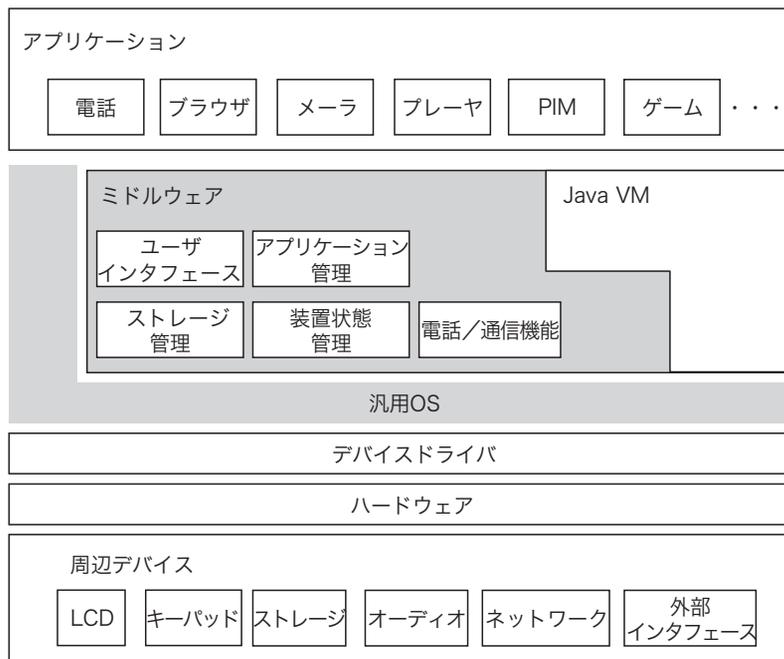


図-4 ミドルウェアの構成例

これら以外にも、IrDA や USB などの外部インタフェース制御やカメラや動画処理などのマルチメディア制御のための API も提供されている。

この MOAP (L) の特徴的な機能は、アプリケーション管理のウィンドウマネージャ制御と競合管理である。携帯電話機は、どのようなアプリケーションや機能を実行中でも任意のタイミングで電話の着信や、電波状態や電池状態の変化などさまざまな割込みイベントが起きる可能性がある。また、メーラからブラウザや電話機能、画像管理機能を呼び出すなど、複数のアプリケーションを起動し、連携して利用することもある。その際、通信やファイル I/O や音声等のリソースの競合もあり得る。その一方で、メモリ、CPU、バッテリーについて厳しいリソース制約のもと、ユーザを混乱させないため、正常動作の確保とエラーの発生の抑止が求められている。そのため、ミドルウェアとアプリケーションの両方の作り込みによって、割込みイベントの種別や、アプリケーション間の優先度、連携の組合せを事前に定義し、どのような状態でも着信した電話アプリを最前面に表示して、通信や音声等のリソースを優先的に割り当てたり、メモリ不足の際に低優先度のアプリを終了させるなどの処理を行う。

一方、開発環境については、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。また、PC 上でアプリケーションの動作が確認できるように、携帯電話機エミュレータが提供

され、事前にソフトウェアの動作を確認できる。さらに、試験の効率化のため、アプリケーションを一定のシナリオに沿って自動的に動作させる連続試験ツールも開発されている(図-5)。キー操作をまとめた試験シナリオとターゲット機種種の属性を設定する実行環境ファイルを入力すると、シナリオに沿ったキーエミュレーションを自動的に実行し、実行結果とログを出力できる。

その他の携帯電話機用ミドルウェアとして、Windows Mobile、BREW についても説明する。Windows Mobile は OS とミドルウェア、オフィスアプリやブラウザ等のアプリケーションまでが統合されたパッケージとして提供され、無線 LAN や Bluetooth などの無線通信の標準サポートや、.NET フレームワークへの対応による Web サービスやデータベースなどとの連携、豊富なマルチメディア機能の特徴とする。Windows Mobile のベースである Windows CE はベンダごとのカスタマイズが可能であったため、Windows CE 端末同士で構成が異なることがあり、アプリケーションの互換性は提供されず、ルックアンドフィールもまちまちだった。Windows Mobile は組み込みを必須とするライブラリを定義することによるアプリケーションの互換性の維持と、ユーザインタフェースの規定によるルックアンドフィールの統一をはかっている。開発環境としては、SDK と PC エミュレータが提供され、マイクロソフト社の Visual Studio により開発を行う。

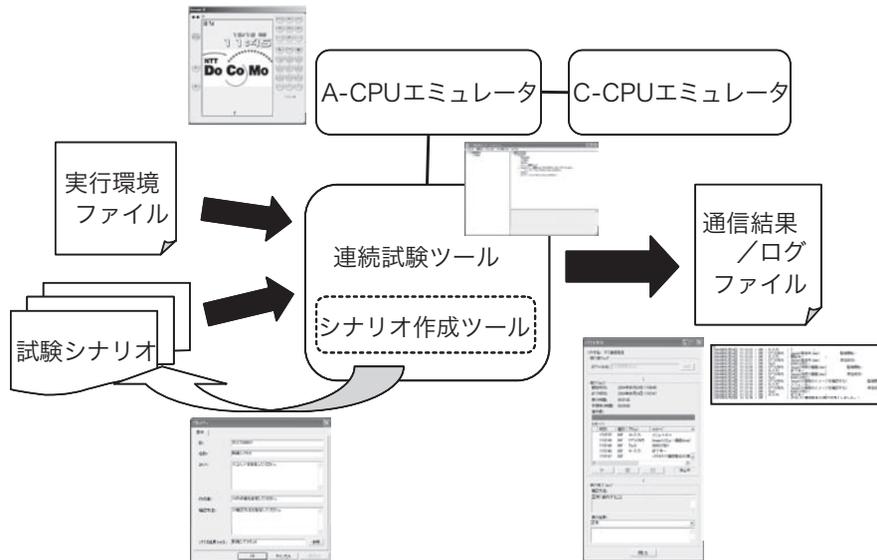


図-5 連続試験ツール

一方、BREW (<http://brew.qualcomm.com/brew/ja>) は、米 Qualcomm 社が提供するミドルウェアであり、RTOS である REX 上で動作する。C や C++ 言語で記述される BREW ネイティブアプリケーションを、サーバや PC から携帯電話機にダウンロードしてインストール可能であること、サードパーティが BREW の API を拡張可能であることが特徴である。開発環境として、SDK と PC エミュレータが提供され、マイクロソフト社の Visual C++ により開発を行う。開発キットには、アプリケーションに電子署名を付与するツールなども含まれている。

今後の技術展望

以上、携帯電話機のソフトウェア開発の背景と、現状のソフトウェアプラットフォームについて、OS とミドルウェア、開発環境の観点から解説した。最後に、ソフトウェアプラットフォームの今後の方向性として、セキュリティ、デバイス管理、マルチコア対応の3つの技術を述べる。

◎セキュリティ

ソフトウェアプラットフォームにおける汎用 OS の採用、API の共通化はソフトウェア開発における多くのメリットを提供するが、デメリットがないとはいえない。PC の世界でウィルスやスパイウェアによる情報漏洩、パソコンのクラッシュ、情報の消失、サービス拒否などの被害が増大しているのに対し、携帯電話機をターゲットにしたウィルスも徐々に増えており、2006 年 4 月の時点で 200 種を超えたという報告もある (F-Secure

社発表, <http://www.f-secure.com/weblog/archives/archive-042006.html#00000864>)。この背景には、近年の携帯電話機の汎用 OS の採用、API の共通化の流れがあると考えられる。

携帯電話機は電話帳や発着信履歴、写真などの個人情報を含むとともに、企業情報システムに組み込まれるなどビジネスユースも増加しており、PC と同様の情報漏洩やデータ破壊などのリスクが今後、懸念される。現状、携帯電話機用に小型化されたエンジンと無線によるパターンデータの更新機能を含むウィルススキャンシステムなども開発されているが、今後、ファイアウォールや IDS (Intrusion Detection System)、アンチスパム、実行環境(ドメイン)の分離による安全な実行環境の提供、ハードウェアベースのセキュリティ機能の連携など、ソフトウェアプラットフォームの拡張による、さらなるセキュリティ対策が望まれる。文献4)では、携帯電話機のハードウェア/ソフトウェアが正常に動作していることの確認機能など、携帯電話機の“信頼”を確保するための、ハードウェア/ソフトウェアアーキテクチャとプロトコルを規定し、Trusted Mobile Platform 仕様として公開している。

◎デバイス管理

移動通信サービスにおけるデバイス管理とは、販売後の携帯電話機のサポートや保守、管理、カスタマイズを提供するサービス機能群であり、携帯電話機のライフサイクルにおける管理コスト低減とサービス品質向上を目的としたものである。無線経由で携帯電話機のファームウェアの機能拡張や不具合修正を行うファームウェア更

新 (FOTA : Firmware update Over The Air) はデバイス管理の代表例であり、販売後の携帯電話機を回収しなくとも不具合修正や機能拡張が可能になる。無線経由でのファームウェアの更新によるトラフィックの増加に対処するため、ファームウェア更新技術では、更新データの差分抽出や圧縮処理により無線トラフィックを削減し、ソフトウェアの大容量化に対処している。また、数千万の加入者を収容する携帯電話事業者では非常に多数の携帯電話機に対するファームウェア更新を想定する必要があるため、トラフィック予測に基づく更新データ配信のスケジューリング等の工夫により、スケーラビリティを確保する必要がある。

デバイス管理におけるサーバクライアント間のプロトコルや管理データモデルは、OMA DM (Open Mobile Alliance Device Management) Working Group において標準化作業が進められており⁵⁾、FOTA 以外に、サービス品質を監視する機能や、品質劣化やソフトウェア障害をリアルタイムで検出し、診断・修復する機能、携帯電話機のデータや設定情報を遠隔制御する機能の標準化作業が進行中である。今後、上記の機能を搭載したソフトウェアプラットフォームが製品化されていくものと考えられる。これらにより、企業ユーザの携帯電話機の設定を管理者のサーバから一括して設定を行ったり、携帯電話機が異常動作や故障した際にも、携帯電話機を販売店などに持ち込まなくとも、遠隔から携帯電話機の状態を診断して修復することができるようになる。

◎マルチコア対応

高機能な携帯電話機の多くが通信処理とアプリケーション処理用に2つのCPUを搭載していることを述べた。一方で、携帯電話機向けのマルチコアCPUもすでに出

荷されており、今後、OSを含むソフトウェアプラットフォームがマルチコア対応していくものと考えられる⁶⁾。たとえば、マルチコア対応OSは、タスクを空いているCPUコアに動的に割り当てたり、タスクの種別に応じてCPUコアを選択する。マルチコアCPU導入の目的は、性能向上と省電力性の両立と、分散処理によるリアルタイム性の向上である。携帯電話機にはマルチメディア処理などさらなる高性能化とリアルタイム性の確保が要求されているが、CPUの動作周波数を上げると消費電力の増大を招き、物理的なCPUを増やすと部品点数が増える。マルチコアCPUにより、動作周波数を抑えながら、性能向上とリアルタイム性の確保が可能となる。

このほかにも、無線LANや3.5G、WiMAXなどの無線インタフェースの多様化への対応、家電や放送との連携など、ソフトウェアプラットフォームはさまざまな機能を取り込み、発展していくものと思われる。ソフトウェアの大規模化、高度化に対して、さらなるスケーラブルなソフトウェアプラットフォーム構築への取り組みが、今後、必要とされるものと考えられる。

参考文献

- 1) 土橋, 荒井, 秋山: 携帯電話端末プロセッサの1チップ化開発—ベースバンドプロセッサとアプリケーションプロセッサの統合—, NTT DoCoMo R&D Technical Journal, Vol.14, No.1 (2006).
- 2) 吉澤, 市川, 古月: 移動端末用ソフトウェアプラットフォーム“MOAP”の拡充, NTT DoCoMo R&D Technical Journal, Vol.14, No.1 (2006).
- 3) 辻, 大野, 斉藤: FOMA 端末ソフトウェアプラットフォーム MOAP の開発, NTT DoCoMo R&D Technical Journal, Vol.13, No.1 (2005).
- 4) Trusted Mobile Platform Specification Download Site, <http://www.trusted-mobile.org/>
- 5) OMA Enabler Releases and Specifications - OMA Device Management, http://www.openmobilealliance.org/release_program/dm_v1_2.html
- 6) 次世代組込 OS のトレンドはハイブリッド化とマルチコア対応, 日経バイト 2005 年 12 月号 (2005).

(平成 18 年 7 月 19 日受付)

