第4回　組込みソフトウェアの転機

青山幹雄／新潟工科大学情報電子工学科

組込みソフトウェアは，これまで機器メーカ固有の技術によって開発されてきた．PC用ソフトウェアのようには一般に知られていない，いわば，ソフトウェアのロストワールドであった．しかし，Windows CEやPersonal/Embedded Javaなどの新しい技術と携帯情報機器，自動車用コンピュータなどの新たなビジネスの出現があいまって，組込みソフトウェアが見直されている．すなわち，組込みソフトウェアのオープン化である．特に，この分野は我が国のソフトウェア技術の国際的競争力が高いといわれている．だが，この変革の中で我が国の組込みソフトウェア技術が国際競争力を維持できるだろうか？　最近の動向を軸に組込みソフトウェアのあり方を考えてみよう．

ロストワールドからリアルワールドへ

組込みソフトウェアは，その名が示唆するように，ハードウェアを制御するためにROMに書き込まれて機器に組み込まれる．そのため，以前からハードウェア部門で開発することも少なくなかった．私も4ビットや8ビットプロセッサ用組込みソフトウェアをアセンブラで開発した経験がある．

しかし，家電や自動車など，多くの機器が組込みソフトウェアで制御されるようになり，組込みソフトウェアが差別化の鍵となっている．このような組込みソフトウェアには，図-1に示す2つの形態がある．

1つは，ハンドヘルドPC，パームトップ，携帯電話などデスクトップPCと同様に利用者の目に見える使い方を主とするパーソナル組込みソフトウェアである．一方，家電製品や自動車，プラントなど機器の中に組み込まれ利用者からは目に見えないが，多様なハードウェアの制御を高度化する機器組込みソフトウェアがある．たとえば，乗用車ではエンジン，変速機など機器ごとにプロセッサが組み込まれ，国内の新車では20個程度のプロセッサが搭載されているといわれる．

では，組込みソフトウェアはどのように開発されているのであろうか．図-2はTRON協会による組込みソフトウェア開発のサーベイ(http://tron.um.u-tokyo.ac.jp/TRON/ITRON/survey97)をグラフにしたものである．最近では32ビットプロセッサの使用が増え，32ビットプロセッサを使用するシステムでは規模が1Mバイトを超えるソフトウェアが40%以上を占める．手のひらに収まる携帯電話にさえ数メガバイトのメモリと数百キロステップのソフトウェアが組み込まれているのである．

一方，組込みソフトウェアは，ハードウェア機器メーカが固有の技術で開発してきた．図-2の調査では，32ビットプロセッサを使用するシステムの90%以上がOSを使用しているが，4ビットプロセッサを使用するシステムではOSを使用していない．このようなシステムでは，イベントに対応する表形式でプログラムを起動するなどの個別制御方法をとる．開発言語も，最近ではC言語などの高水準言語に移行しつつあるが，アセンブラを使用することも少なくない．

歴史は繰り返すか？

近年のデスクトップPCに象徴されるように，情報産業は，垂直統合から水平分業への転換を繰り返してきた．今，組込みソフトウェアにもその転換の波が押し寄せている1)~3)．

図-3に示すように，組込みソフトウェアは，ハードからアプリケーションに至るまで特定のベンダが開発する垂直統合の世界であった．しかも，組込みソフトウェアには安全性，信頼性，性能などの面で高度な要求がある．1円単位のコスト削減が要求され，使用メモリ量やプロセッサなどに厳しい制約が課せられる．さらに，いったんROMとして出荷した後は変更がきわめて困難であり，バグは多額の損失を招くことが多い．

だが，この世界にも高機能化の波が押し寄せてきた．携帯情報機器のようにデスクトップPCとの連携が求められたり，一から開発していては市場の変化に対応できない場合もある．もはや1社ですべての開発を担うのが困難となり，自社にないコンポーネントは外部調達する水平分業への転換の兆しが見えてきた．組込みソフトウェアのオープン化である．

図-1　組込みソフトウェアの形態

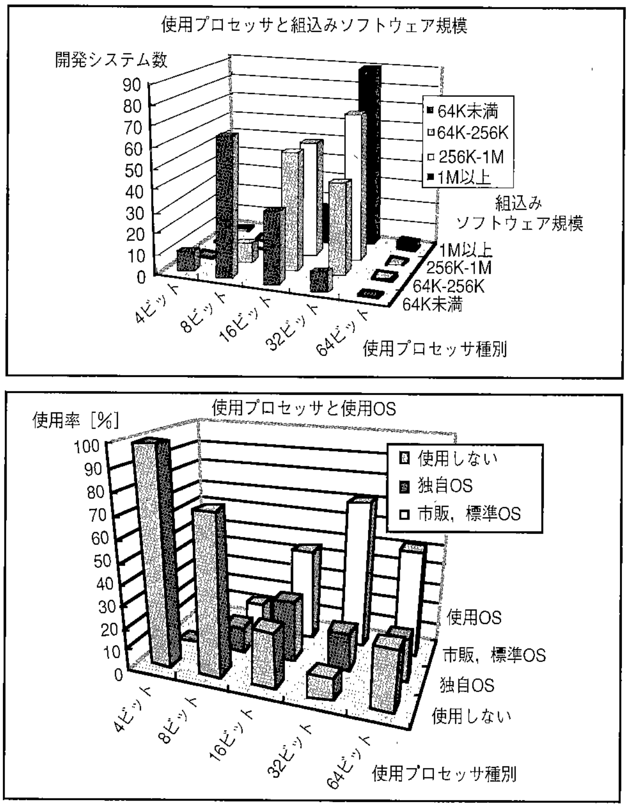

図-2　組込みソフトウェア開発状況の調査例

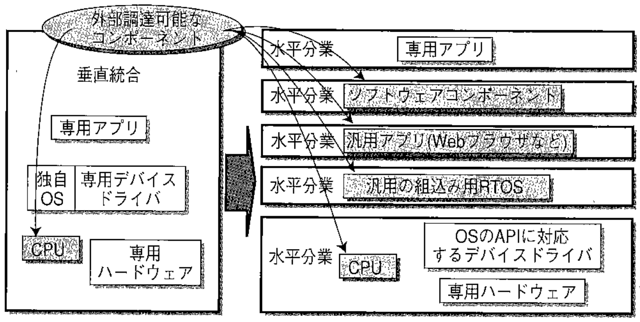

図-3　垂直統合から水平分業へ

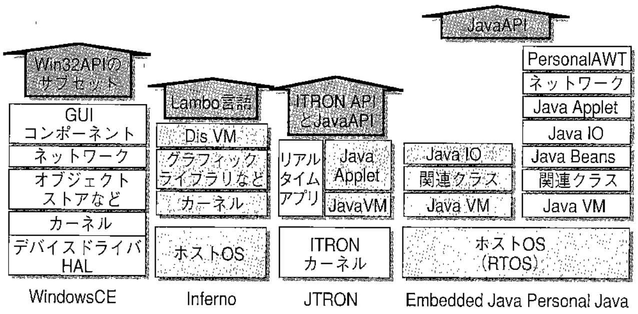

図-4　新しいRTOSアーキテクチャの例

第3のプラットフォーム戦争

組込みソフトウェアのプラットフォームが新たな技術開発競争の場としてクローズアップされている．

●組込み用RTOS

第1のプラットフォーム技術は組込みソフトウェア用RTOS(Real-Time OS)である．これまで，RTOSはできるだけコンパクトで，かつ，できるだけ処理を行わないことが求められていた．つまるところ，OSというレベルではなかった．このため，ある組込みソフトウェアでは，各プログラムでメモリセグメントの切替えをせざるをえず，切替えのリセット漏れのバグで大変苦労した．しかし，組込みソフトウェアの規模の増大，扱うデータの量の増大と種類の多様化の結果，本格的なOSが必要となっている．

一方，RTOSはこれまでニッチマーケットであった．米国には，Embedded Systems Programming誌のBuyer's Guideに掲載されているだけでも50社以上のRTOSベンダがある．さらに，多くの組込みソフトウェアベンダも独自にRTOSを開発している．一企業内でも，機器や部門ごとにRTOSを開発することもある．だが，MicrosoftのWindows CEやSun MicrosystemsのPersonal Java，Embedded Javaの出現で，RTOSは，デスクトップPC用OS，サーバ用OSに次ぐ第3のOS戦争の場となった観がある[3]．これらのOSのアーキテクチャを図-4に示す．

図-1に示したように，組込みソフトウェアには2つの流れがある．パーソナル組込みソフトウェアではデスクトップとの連携が重視される．一方，機器組込みソフトウェアでは信頼性やコンパクト性が重要である．既存のRTOSは，機器組込み用としてコンパクト性を重視しているので，カーネルは数十Kバイトに収まるものが多い．Javaでは，機器組込み用とパーソナル用をEmbedded JavaとPersonal Javaの2つのアーキテクチャに分けて提案している点で新しい試みといえる．Javaの場合は，ホストOS上にJava VMを搭載する方法に加え，JavaでOSそのものを実装する方法もある．しかし，ハードリアルタイムシステムと呼ばれるタイミングの遅れが致命的な問題を引き起こすシステムの場合や処理能力上GC(Garbage Collection)が問題となることがある．

Windows CEはWindows95/NTとの親和性が特長であるので，パーソナル用として携帯情報機器に使用されている．TRONは国内で多数使用され，Javaへ対応したJTRONの仕様も公開されている．Lucent TechnologiesのBell Laboratoriesでは，UNIXの開発者であるRitchieらがホストOS上の一プロセスとして実行するネットワーク指向のRTOSとしてInfernoを開発している．

研究ではReal-Time MachなどいくつかのRTOSが開発されている．

●組み込み用Webブラウザ

プラットフォームの第2の技術は，Webブラウザなどのネットワークソフトウェアであろう．特に，コンパクトでかつカーナビや携帯電話などの小さな画面に表示できるスケーラビリティのあるWebブラウザが開発されている．たとえば，JavaをベースとするSun MicrosystemsのPersonal WebAccess，Spyglassの

Device Mosaicがある．Device Mosaicのサイズは400KB以下で，ブラウザのGUIそのものをHTMLで記述できるので，カスタマイズ可能である．

さらに，アクセスと国内メーカはCompact HTMLと呼ぶ規格を1998年4月にW3C(World Wide Web Consortium)に提案した．アクセスは，Compact HTMLに対応したブラウザNetFrontを提供している．

●組込み用分散オブジェクト環境とコンポーネント

第3のプラットフォーム技術は，今後必要となる分散オブジェクト環境である．たとえば，Windows CE上ではWindowsと同様ActiveX/COM (Component Object Model)が利用できる．JavaではPersonal Javaで，ネットワークなどのサービスが提供されている．一方，OMG(Object Management Group)ではリアルタイムソフトウェア向け，組込みソフトウェア向けCORBA(Common Object Request Broker Architecture)の仕様として，それぞれRealtime CORBA，Minimum CORBAが検討されている．

さらに，これらの分散オブジェクト環境上で組込みソフトウェアを実現するソフトウェアコンポーネントの標準化が推進されている．ActiveXの業種向けコンポーネント，JavaのコンポーネントアーキテクチャであるJavaBeans，CORBAのドメイン固有コンポーネントがある．

組込みソフトウェアの開発技術

●組込みソフトウェア開発の現状と問題点

図-5は組込みソフトウェアの開発技術を開発プロセスに沿って示す．次の点に特徴がある[5),6)]．

(1)リアクティブ/リアルタイム・ソフトウェアの開発

組込みソフトウェアは，外部信号に対応して処理するリアクティブソフトウェアや指定時間に処理を行わなければならないリアルタイムソフトウェアの一種と考えられる．このようなソフトウェアの開発では時間の概念が入ってくるので，非リアルタイムソフトウェアに比べ設計や試験が難しい．さらに，例外処理や障害処理の開発に多くの労力を要している．

これに対し，組込みソフトウェアの開発方法論や開発管理などの整備が遅れている．チューニングなど経験と勘に頼る面もある．しかし，規模の増大や開発期間の短縮に対応できるためには，経験を体系化し，共有しなければならない．さらに，リアルタイムオブジェクト指向開発技術やスケジューリング理論などの新しい技術を取り入れることも必要となる．

(2)ソフトウェアとハードウェアのコデザイン

ソフトウェア開発とそれを組み込むハードウェアの開発とを並行して進めるコデザイン(Co-Design)が必要である．開発初期はソフトウェアとハードウェア間のインタフェースの決定や，それぞれの開発の同期をとるなど，ソフトウェアに限らず多様な技術の集積と

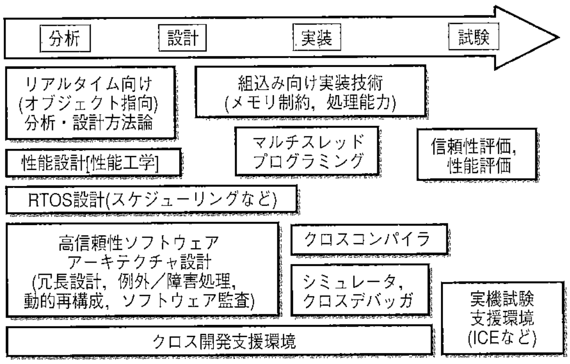

図-5　組込みソフトウェアの開発技術

複数部門が協調して開発できる仕組みが必要になる．

(3)クロス開発と開発支援環境

開発環境と実行環境が異なるクロス開発を行う．特に，試験ではハードウェアができていない場合もあるので，ICE(In-Circuit Emulator)上の疑似プロセッサにダウンロードして実行したり，シミュレータを利用することもある．一般に，クロス開発環境に比べ実行マシン(実機)上の試験環境は整備されていないことが多く，実機上のデバッグは難しい．一方，クロス開発環境上では実機のハードウェアとソフトウェア間の微妙なインタフェースが試験できない問題もある．

設計などの上流工程の開発環境も，デスクトップPCやビジネスアプリケーションに比べ未熟である．自社で独自に開発した環境を利用することも少なくない．最近は，市販の開発環境を利用する傾向にある．たとえば，MicrosoftのETK(Windows CE Embedded Toolkit for Visual C++)ではデスクトップ用開発環境を組込みソフトウェア開発へカスタマイズしている．

●オープン化がもたらす新たな問題点

(1)信頼性，性能などの品質保証の問題

RTOSなどの外部から調達したコンポーネントを組み合わせて開発したシステム全体の品質保証が問題となる．これは，すでに企業システムのダウンサイジングの過程で問題になってきた．機器組込みソフトウェアではきわめて高度な要求もある．たとえば，自動車エンジン制御ではμ秒単位の処理精度を必要とする．一方，1996年6月に起きたソフトウェアバグによるAriane 5の打ち上げ失敗のように，航空宇宙機器や大規模プラント制御では莫大な損失を引き起こす可能性がある．

(2)技術変革への対応の問題

OSなどの構成要素のオープン化は開発技術のオープン化を促す．デスクトップやサーバの世界で起きた急速な技術変革が組込みソフトウェアでも起こるだろう．特に，水平分業ではOSのように標準化が鍵となる．これは，OSに留まらず，流通向けPOS (Point Of Sales)端末やプラント制御などの分野でソフトウェアコンポーネントの標準化が進んでいる．従来の企業内に閉じた技術開発ではこの変革に取り残されるだろう．

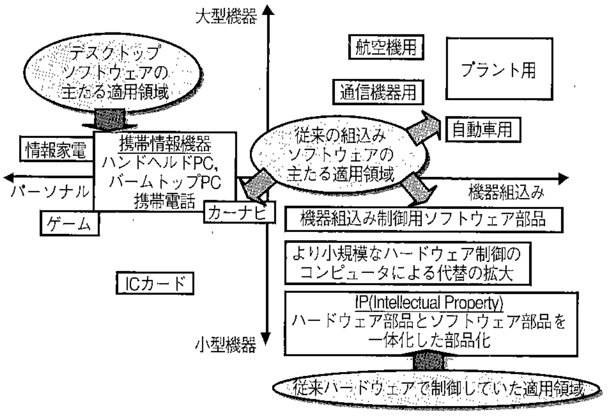

図-6　組込みソフトウェアの適用領域の拡大

広がる組込みソフトウェアの世界

図-6に示すように，組込みソフトウェアの適用領域は一層拡大するだろう．第1の方向は，携帯情報機器のようなパーソナルな領域での拡大であり，情報家電などの新しい市場の創造が期待できる．

第2の方向は機器組込みの領域である．これまでハードウェアで制御されていた機器をソフトウェアで制御する．プラント制御などのより大型の機器への適用と機器組込みなどのより小型の機器への適用が期待できる．機器組込みではハードウェアとソフトウェアの融合が進み，両者を一体化して再利用するIP(Intellectual Property)などの新たな動きが見られる．

電子機械工業会の統計(http://www.eiaj.or.jp)では，1997年度国内で生産された組込み用プロセッサは20億個を超え，世界に占める我が国のシェアは高いといわれている．今後，携帯情報機器やセットトップボックスといった情報家電のように，新たな組込みソフトウェアの開発が目白押しである．このような，我が国が得意としてきた分野で新しいニーズに対応する組込み用ソフトウェアの開発が期待される．

進化する組込みソフトウェア

新しいニーズに対応して組込みソフトウェアも進化している．

たとえば，自動車はビジネスや行楽のための情報サービスとエンジンなどの高度な機器制御の2つの面から多様な組込みソフトウェアを必要とする．

情報サービスでは，1996年4月から道路交通情報システムVICS(Vehicle Information and Communication Systems)が始まっている．今後，高度道路交通システムITS(Intelligent Transport Systems)の開発などソフトウェアへのニーズが高い．たとえば，VICSの情報に加えて駐車場の位置やレストランなどの情報を携帯電話とWebで検索できる．

一方，機器制御では，安全性，操作性，環境対策の面から，より高度な制御が要求されている．制御用コンピュータは機器ごとに組み込まれているが，LANを用いて連携する方向にある．この結果，昨年から，自動車用コンピュータシステムのアーキテクチャが次々に提案されている．Microsoftの提案によるAuto PC，IntelによるConnected Car PC，IBMなどによるThe Network Vehicleである．The Network Vehicleでは，Vehicle Control BusとMedia Busの2つのバスを介して自動車制御と情報サービスを連携する．

特に，機器組込みソフトウェアは高度な信頼性，安全性を保証する必要がある．安全性の面では，単に，ソフトウェアの信頼性だけではなく，利用者とのインタフェースが重要であることは，いわゆるハイテク飛行機の操縦性の問題が提起している．Boeing 777には新規開発2.5メガステップを含む4メガステップのソフトウェアが組み込まれているという[4]．しかし，デスクトップPCとは異なり，パイロットの眼にはこれだけの組込みソフトウェアの存在すら見えない．自動車でも，最近発表されたAuto PCでは，安全性の観点から，音声認識による操作や音声読み上げなどの音声によるユーザインタフェースが提供されている．

組込みソフトウェアの戦略転換点

組込みソフトウェアも，今や，大規模ソフトウェアとなった．今後，その需要は一層増えるであろう．しかも，安全性や信頼性の面で厳しい要求を満たす必要がある．このような大規模高信頼性組込みソフトウェアを効率よく開発するには，従来の技術の延長でよいのだろうか？開発の人材は確保できるのであろうか？

かつて，大規模組込みソフトウェアの新規開発において，アセンブラから高水準言語に転換する経験をした．あわせて，開発方法，管理方法などの開発技術も一新した．この時，開発に先立って，1年近く実際のソフトウェアのプロトタイプ開発と評価を行い，新しい開発技術の評価と導入，普及を行った．開発現場では，技術の転換は一朝一夕にはできない．しかも，開発現場は恒常的な繁忙の中にある．だが，ここで述べたように，さまざまな点で組込みソフトウェアを取り巻く環境は大きく変化している．

今，組込みソフトウェアの開発のあり方を見直し，戦略転換を図る必要があるのではないだろうか？

参考文献

1) Comerford, R.: Pocket Computers Ignite OS Battle, IEEE Spectrum, pp. 43-48 (May 1998).
2) リック クック：Javaが開く組み込みシステム&リアルタイムOS新時代, ジャバワールド, pp.102-110 (July 1998).
3) 原田 衛, 伊藤元昭：ソフトウェア危機－ディジタル家電が水平分業へ, 日経エレクトロニクス, 1998年3月23日号, pp. 139-158.
4) Pehrson, R. J.: Software Development for the Boeing 777, Cross Talk, Vol. 9, No. 1 (Jan. 1996).
5) Stankovic, J.A. et al.: Strategic Directions in Real-Time and Embedded Systems, ACM Computing Survey, Vol. 28, No. 4, pp. 751-763 (Dec. 1996).
6) 高田広章(編): 特集「組込みシステム開発の現状」, 情報処理, Vol.38, No.10, pp.870-910 (Oct. 1997).

(平成10年6月7日受付)