

ITRON プロジェクトと CASE ツール

豊橋技術科学大学／ITRON 専門委員会 高田 広章

ITRON プロジェクトと CASE ツールについて書くように依頼され気安く引き受けたが、なかなか難しいテーマである。そこで、まずは ITRON 仕様 OS の現状と ITRON プロジェクトの活動について宣伝させていただき、最後に CASE ツールとの関連について述べたいと思う。

■ ITRON 仕様 OS の現状

ITRON プロジェクトは、組込みシステム用のリアルタイムオペレーティングシステム(以下、RTOS と略記する)とそれに関連する仕様の標準化を行うプロジェクトである。1984年にプロジェクトを開始し、今年でちょうど 15 年になるが、その間に 3 世代に渡る ITRON の RTOS 仕様を策定してきた。具体的には、1987 年には最初の ITRON 仕様である ITRON1 仕様、1989 年には第 2 世代にあたる μ ITRON 仕様 (Ver 2.0) と ITRON2 仕様、1993 年には第 3 世代の μ ITRON3.0 仕様を公開した。

それらの仕様の中で、小規模な組込みシステム向けに設計した μ ITRON 仕様は、限られた計算能力とメモリ容量しか持たないシステムでも実用的な性能を発揮することができたために、多くの種類の組込みシステム向けのプロセッサ用に実装され、また極めて多くの組込みシステムに利用されており、この分野における業界標準の RTOS 仕様となっている。(社)トロン協会では毎年実施しているアンケート調査では、国内で開発されている組込みシステム応用機器(組込み機器)の約 30%に ITRON 仕様 OS が用いられているという結果が得られている。アプリケーション分野別では、AV 機器・家電製品・個人用情報機器・通信端末など、1 人ないしは 1 家が 1 台持つ機器において、ITRON 仕様 OS が多く使われている。

■ ITRON プロジェクト 第 2 フェーズの活動

ITRON プロジェクトでは約 3 年前から、それまでのリアルタイムカーネル仕様に焦点を置いた標準化活動から、周

辺仕様にも標準化範囲を拡大した第2フェーズの活動を展開してきた。具体的には、ソフトウェア部品のための標準化、ソフトウェア開発環境関連の標準化、アプリケーション分野毎の要求に対応した標準化の3つの方向へと範囲を拡大しつつある。

その中でも特に重点を置いているのが、ソフトウェア部品のための標準化である。近年、組込みシステムの大規模化・複雑化が急速に進んでおり、RTOS上で動作するソフトウェアのすべてを組込み機器メーカーだけで構築することは難しくなっている。そのため、組込みシステム用のソフトウェア部品(ミドルウェア)の利用が重要性を増しており、そのための標準化が求められている。ソフトウェア部品のための標準化としては、ソフトウェア部品そのもののインタフェース仕様の標準化と、ソフトウェア部品が流通する前提条件を整えるための標準化の両方が必要と考えている。

ソフトウェア部品そのもののインタフェース仕様の標準化として、ITRONプロジェクトでは、これまでにTCP/IPプロトコルスタックとJava実行環境の2つの分野に取り組み、その結果、それぞれITRON TCP/IP API仕様と

JTRON2.0仕様を策定し、いずれも1998年に公開した。これらの仕様においては、概念や用語を既存のITRON仕様と整合させ、ITRON仕様OSを使っている人には容易に学習できるように配慮している。それと同時に、他のRTOS上でも利用可能なインタフェースとなるよう、できる限りITRON仕様のRTOSの機能に依存しないような仕様としている。

またITRON仕様のRTOS上で動作するソフトウェア部品が流通する前提としては、ソフトウェアの移植性が高いことが求められている。従来のITRON仕様においては、「弱い標準化」と呼んでいる設計方針により、同じ仕様に準拠していても製品により仕様が異なる場合が多く、ソフトウェア移植性を低下させる原因となっている。一方で、弱い標準化により広い範囲のプロセッサへ適用できるスケールビリティが実現されており、弱い標準化の利点を維持しつつソフトウェアの移植性を向上させることが求められている。そのために、弱い標準である μ ITRON仕様の中に標準的な機能集合(これをスタンダードプロファイルと呼ぶ)を厳密に定める方針が考えられ、この方針に従って現在標準化検

討が進められているのが後述の μ ITRON4.0 仕様である。

第 2 フェーズにおけるその他の活動としては、RTOS の利用率が低かった自動車制御分野における独自の要求事項を調査するために、その分野の技術者を集めて行った RTOS 自動車応用技術委員会が挙げられる。この委員会の活動成果も、 μ ITRON4.0 仕様に反映されている。また、難しいと言われるデバイスドライバの設計を容易にするためのデバイスドライバ設計ガイドラインや、リアルタイム制約を持ったアプリケーションとソフトウェア部品の共存を図るためのアプリケーション設計ガイドラインの検討作業も行っている。

最近開始した活動としては、ITRON 仕様の RTOS とデバッグ環境との間のインタフェースの標準化が挙げられる。ITRON 仕様 OS は開発環境面が弱いといわれているが、これは、ITRON 仕様 OS と開発環境(特にデバッグ環境)との間のインタフェースが標準化されていないために、開発環境メーカーが各社の ITRON 仕様 OS に個別に対応しなければならないのが問題の原因の 1 つと考えられる。

そこで、ITRON 仕様 OS とデバッグ環境との間のインタフェース(ITRON デバッグインタフェース)を標準化することで、デバッグ環境を各 ITRON 仕様 OS に容易に対応できるようにすることを目指している。

これらの標準化活動を通じて、ITRON 仕様の短所を克服するとともに、常に変化する要求に対応していくことを目指している。

■ μ ITRON4.0 仕様

μ ITRON4.0 仕様は、ITRON リアルタイムカーネル仕様としては第 4 世代にあたるもので、その策定が必要な理由としては、上述のソフトウェア移植性の向上に加えて以下の 3 点が挙げられる。

○ ソフトウェア部品向けの機能の追加

従来の μ ITRON 仕様では、外販することを前提にソフトウェア部品を構築するには機能的に不足している部分があったため、その面を強化することが必要である。

○ 新しい要求・検討成果の反映

ITRON プロジェクトでは、ハードリ

アルタイムシステムの構築を容易にするために RTOS が持つべき機能の検討や自動車制御分野における RTOS に対する要求事項の調査・整理を行ってきたが、これらの新しい要求や検討成果を μ ITRON 仕様で反映する必要がある。

○ 半導体技術の進歩への対応

μ ITRON3.0 仕様を公開してからすでに5年以上が経過しており、その間に半導体技術は大きく進歩し、プロセッサの性能向上やメモリ容量の増加が著しい。

そのため、 μ ITRON3.0 仕様の策定時点では有用ではあるがオーバーヘッドが大きいため標準化を見送った機能で、現時点では許容できるオーバーヘッドで実現できると考えられるものもある。

このような理由から μ ITRON4.0 仕様では、前述のスタンダードプロファイルの規定によりソフトウェア移植性の向上を図ることに加えて、スタンダードプロファイルを越える要求に対応するために豊富な拡張機能を用意するとともに、最小の機能セットを従来の μ ITRON 仕様よりもさらにコンパクトに実現可能なものとし、従来の仕様

以上に広範なスケラビリティの実現を狙っている。また、スタンダードプロファイルに加えて、自動車制御用プロファイルを定義している。

現在、 μ ITRON4.0 仕様の標準化検討は最終局面まできており、1999年6月末に開催予定のITRONオープンセミナーにおいて、仕様を公開する予定である。

■ CASE ツールとの関連

現在のITRON仕様は、アプリケーションプログラマがRTOSを直接使用することを想定して策定しているが、組込みシステムの適用分野やソフトウェア規模の拡大に伴い、(コンピュータ技術者ではなく)制御技術者がプログラムを作らなければならない状況が増えている。また、コンピュータ技術者ではあっても、汎用機やパソコンの技術者が組込みシステム開発に従事する場合にも、RTOSを使いこなすのは難しいと言われている。実際、前述のトロン協会によるアンケート調査でも、「技術者の不足」をRTOSの問題点に挙げたケースが例年トップを占めている。

CASE ツールは、この問題を解決するための最有力手段の 1 つである。実際、アプリケーションの特性に合致した CASE ツールを使えば、ソフトウェアの開発工数削減や品質向上が実現できることは、ZIPC の実績が証明していると思う。

CASE ツールを使うという前提で ITRON 仕様 OS(ないしは RTOS)を見た場合、ITRON 仕様 OS は少なくともプロセッサの違いを隠蔽するための役割を果たすことができる。筆者は、ITRON 仕様 OS はプロセッサというデバイスのためのデバイスドライバであるという言い方をする時があるが、まさにそのような位置付けである(ただし、プロセッサを隠蔽することだけに徹するのであれば、ITRON 仕様はもっと違った姿になるだろう)。この位置付けでは、RTOS の各機能が強く標準化されていることは重要であり、 μ ITRON4.0 仕様のスタンダードプロファイル規定により、CASE ツールの構築が容易になるものと思う。

CASE ツールにとっての RTOS のもう 1 つの意義は、CASE ツールで生成したプログラムと他のプログラムの混在をサポートすることであろう。分かり

やすい例としては、汎用のデバイスドライバや通信プロトコルスタックなどは直接 RTOS 上に実装し(ないしは、実装されたものを購入し)、CASE ツールで生成したプログラムからそれらを利用する方法である。また、ZIPC の場合は状態遷移ベースのシステムに向いているというように CASE ツールには得意・不得意があるが、システムが複雑化してくると、ZIPC では状態遷移ベースの部分を構築し、他の部分は RTOS 上に直接実装する(ないしは、他の CASE ツールを使う)のが妥当であろう。

さらに、CASE ツールが並列(ないしは非同期)な振舞いや実時間処理をサポートする場合には、それを実現するためにも RTOS は有用であろう。

最後の話題として、筆者の研究室で開発している教育・研究用の ITRON 仕様 OS である ItIs について触れる。ItIs は、ターゲットシステム上で動作することに加えて、BSD UNIX 上で動作するシミュレーション環境を持っているが、同様のシミュレーション環境を Windows 上にも実現する開発を現在進めている。このシミュレーション環境を ZIPC に取り込めば、

ZIPC で生成した RTOS 機能を使った C ソースコードを、Windows 上でより完全に実行することができるように思うが、どうだろうか？ なお、ItIs はフリーソフトウェアであり、以下の web ページから入手可能である。

<http://www.ertl.ics.tut.ac.jp/ItIs/>

■ おわりに

ZIPC は、ITRON 仕様 OS と同様に、組込みシステム分野における我が国の独自技術である。組込みシステム技術は、家電機器・自動車・工作機械など、我が国が競争力を持つといわれるの産業分野になくってはならない基盤技術となっており、この分野に

おいて世界をリードする技術開発を行ってと同時に、その技術を世界に向けて発信していくことが極めて重要である。

ZIPC のより一層の発展をお祈りするとともに、ITRON プロジェクトも、ITRON 仕様がこの分野における国際標準と呼べるものとなるよう、尽力していきたいと考えている。

■ 参考

ITRON ホームページ

<http://www.itron.gr.jp/>

(たかだ ひろあき)