6N－2

プラント監視制御ミドルウェアの開発
— リアルタイムプログラミング支援環境 —

†水沼 一郎　†島川 博光　†竹垣 盛一　‡草川 英之
三菱電機(株)　†産業システム研究所　‡制御製作所

1 まえがき

プラント監視制御システムは、その処理が時間制約を満たして行われること（リアルタイム性）、および、連続して運転可能なこと（高可用性）が要求される。このような要求を満たすために、プラント監視制御システムは従来専用ハードウェアや専用OSの上に構築されることが多かった。一方近年、リアルタイムOSの標準としてIEEEによるPOSIX 1003.4, .4aが定められ、これに準拠した商用OSも出回ってきた。また、オープンシステムを用いて高可用性システムを構築した例も多く見られるようになった。プラント監視制御の分野でも、このような世の中の流れに沿って、オープンな環境の上でのシステム構築を可能とすることが要求されてきている。

我々は、POSIX 1003.4, .4aに準拠したリアルタイムOS上で、レートモノトニック解析（RMA）に基づくリアルタイムシステムのプログラミングを支援するライブラリを開発した。このライブラリは二つの大きな特徴を持つ。

一つは、高可用性のために待機冗長などの構成をとるシステムが、故障の発生によってその構成を変化（構成制御）させる場合の、タスクのスケジューリング機能を持つ点である。この特徴により、連続運転可能なリアルタイムシステムを容易に実現することができる。

もう一つは、複数の開発グループによって開発される大規模システムの構築において、組合せ試験に先だった各ブロックの時間的正当性の検証を可能とするプログラミングモデル（リアルタイムサブシステムモデル）を提供する点である。この特徴によって、検証に要する時間を短縮するだけでなく、ソフトウェアの信頼性を高めることができる。

以下、2章でRMAに基づくプログラミングの支援について、3章で高可用性システムの構築支援について、4章でリアルタイムサブシステムモデルについて、そして、5章で結論を述べる。

Development of Middleware for Plant Monitoring and Control Systems
— Programming environment for real-time systems —
Ichiro Mizunuma, Hiromitsu Shimakawa, Morikazu Takegaki, and Hiroaki Kusakawa
Mitsubishi Electric Corp.

2 レートモノトニック解析

固定優先度制御の下でのスケジュール可能性判定の手法として、レートモノトニック解析（RMA）[1]が知られており、その拡張も数多く提案されている[2][3][4]。文献[3]によると、優先度継承プロトコル（PIP）に基づいて共有資源に対する相互排除を行うn個の周期タスク$\tau_i(i=1,2,\cdots,n)$は、

$$\sum_{i=1}^{n}\frac{C_i}{T_i}+\max_{i=1}^{n}\frac{B_i}{T_i}\leq n(2^{1/n}-1)$$

が成立するればスケジュール可能である。ただし、C_i, T_i, B_iはそれぞれタスクτ_iの実行時間、周期、ブロック時間である。

POSIX 1003.4, .4aには、リアルタイムシステムのプログラミングに不可欠なプリミティブが規定されている。しかし、RMAに基づいた予測的なプログラミングを行うためには、これらのプリミティブだけでは十分ではなく、以下のようなことが必要となる。

レートモノトニック優先度割当て　RMAは、より周期（あるいは最小起動間隔）の短いタスクに、より高い優先度を割り付けることを前提としている。本ライブラリでは、各タスクに対してこのような優先度割付を行う。

実行時間C_iの見積り　各タスクの実行時間は、アプリケーションから与えられる必要がある。しかし、一般にこの実行時間を正確に見積もることは困難である。あるタスクの実行時間の超過による他のタスクへの影響を予測的にするために、本ライブラリではデッドラインの検出機能を提供している。この機能を実現するために、各タスクの起動時間やタイムアウトを管理するタイマスレッド[5]が用意される。

PIPと最悪ブロック時間B_iの保証　POSIX 1003.4, .4aではPIPに基づくセマフォの機能を定めていないが、現在利用可能な商用リアルタイムOSの多くがこの機能を提供している。また、ブロック時間を保証する方法としては、共有資源へのアクセス手段を限定することが考えられる。しかし、共有資源のデータ構造はアプリケーションに依存するので、ライブラリのみでブロック時間の保証を行うことは困難である。

3 構成制御

産業用システムは、同等の機能を持った構成要素を複数用いることによって高信頼性、高可用性を得ている

[6]。特に最近では、安価に、柔軟性を持って高信頼性、高可用性を得るために、待機冗長システム[7]として構成されることが多い。待機冗長システムは、二つのサブシステムから構成され、一方のサブシステムがシステム全体に対して要求される処理を行い、他方は待機している。前者のサブシステムにハードウェアやソフトウェアの故障や誤りが発生した場合や、メンテナンスやソフトウェアの更新のため停止する必要が生じた場合には、このサブシステムと待機していたもう一方のサブシステムの入れ換えが行なわれ、システム全体として要求される処理を継続して行なうことができる。このようなサブシステムの入れ換えは、システムの再構成（reconfiguration）[7]と呼ばれる。再構成時には、各タスクの処理切替え（状態遷移）と再構成のための例外処理を、システム全体として論理的な矛盾を起こさないような順序で行なう必要があり、また同時に、複数のタスクに関わる時間制約を満たす必要がある。再構成時のスケジューリングに際しては、これらの点を考慮しなければならない。

我々は文献[8]で、それぞれRMAによって時間制約が保証されている2台のサブシステムを用いた待機冗長リアルタイムシステムにおいて、1） 時間制約を持つある再構成時の処理に対して、その処理に関わるタスクの状態遷移数を最小にするように再構成時のタスクの状態遷移の順序を決定する、2） 1）で求められた順序にしたがった再構成がスケジュール可能か、RMAを応用した手法を用いて判定する、3） 2）の結果、スケジュール可能と判定された場合について、再構成時の各タスクへの優先度の割り付け手順を示す、という手法を提案した。この手法を用いると、実時間制約を保証する待機冗長システムの設計を柔軟性を持って行なうことができる。本ライブラリでは、この手法に基づいた再構成を行うシステムの構築を支援する。

4 リアルタイムサブシステムモデル

多くの人材を投じて開発される大規模な産業用リアルタイムシステムの構築においては、機能毎に分割されたブロック単位での個別の開発を行った後、組合わせ試験か行われるのが一般である。しかし、個々のブロック単位で論理的な正当性が検証されても、ブロック毎の時間上の振舞いが互いに大きく依存しあっており、組合わせ試験を行うまで個々のブロックの時間上の正当性を検証できなかった。さらに、あるブロックの修正が他のブロックの時間上の振舞いに影響を与えてしまうこと（副作用）が多かった。これらの理由により、検証フェーズに要する時間が大きくなっていた。

このような問題点を解決し、大規模で、かつ、高い品質を要求される産業用リアルタイムシステムの生産性を向上させるために、我々はリアルタイムサブシステムモデルを提案する。このモデルでは、各機能ブロックをプロセスに対応付け、各プロセス内では互いに密な同期をとるスレッドが並行に動作する。各プロセス内では、全てのスレッドの時間上の振舞いが予測的になるような設計がなされる。プロセス間での通信は、極力共有メモリを用いた非同期なやり取りに限定される。また、共有メモリへのアクセスにおける相互排除によるブロック時間は厳しく管理される。こうすることより、各機能ブロック間の時間上の独立性が高くなる。

このモデルに基づいてS/Wを構築することにより、個々のブロック単位で、時間上の正当性をRMAを用いてあらかじめ単体で検証することができるだけでなく、組合せ試験において、あるブロックの修正による他のブロックの時間的/論理的正当性への副作用を減らすことができる。

5 むすび

本ライブラリを用いることにより、リアルタイム性と高可用性を、柔軟性を持って確保することができる産業用システムをオープンな環境の上に構築することができるようになった。

ブロック時間の保証については、ライブラリのようなボトムアップ的な支援より、ビルダといったトップダウン的な支援の方が、共有資源へのアクセス手段の限定には向いていると我々は考えており、現在、分散リアルタイムシステムの視覚的設計の支援環境（ビルダ）を開発している。このビルダでは、ブロック時間の保証のみならず、分散環境全体でのスケジュール可能性判定の実現などもめざしている。

参考文献

[1] C. L. Liu and J. W. Layland. Scheduling algorithm for multiprogramming in a hard-real-time environment. *JACM*, Vol. 20, No. 1, pp. 46–61, January 1973.

[2] J. P. Lehoczky, L. Sha, and Y. Ding. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. In *Proceedings of the IEEE Real-time Systems Symposium 1989*, pp. 166—171, December 1989.

[3] R. Rajkumar. *Synchronization In Real-Time Systems: A Priority Inheritance Approach*. Kluwer Academic Publishers, 1991.

[4] B. Sprunt, L. Sha, and J. Lehoczky. Aperiodic task scheduling for hard-real-time systems. *The Journal of Real-Time Systems*, Vol. 1, No. 1, pp. 27–60, 1989.

[5] I. Mizunuma, H. Shimakawa, and M. Takegaki. Real-Time Middleware Based on Rate-Monotonic Theory. In *Proceedings of the IEEE RTAW'94*, pp. 58–62, July 1994.

[6] 制御システムの高信頼化手法調査専門委員会. 制御システムの高信頼化手法. 電気学会技術報告, No. 496, 1994.

[7] 南谷崇. フォールトトレラントコンピュータ. オーム社, 1991.

[8] 水沼, 神余, 島川, 竹垣. 実時間制約を保証する待機冗長システムの一設計手法. 信学会論文誌, Vol. J78-D-I, No. 8, 1995 (掲載予定).