

生産システムにおける

6S-4

並行プロセス制御プログラムの構成法

丹羽博幸* 山縣敬一** 牧之内三郎***

(*日航情報開発(株) **大阪大学工学部 ***大阪国際大学)

1. はじめに

生産システムにおけるコンピュータコントローラの応用には、極めて多様な機能が要求されるようになってきている。例えば、1台のNC工作機械を制御する場合であっても、単にNC指令の転送と送りの制御だけにとどまらず、いろいろなセンサーを用いた加工状態の監視を考えると、多くの機能モジュールの協調動作が必要になる¹⁾。さらに最近では、マルチマイクロプロセッサによるコントローラが容易に構成できることと相俟って、搬送機械や工作機械やロボットなどが協同作業をする生産セルの制御が重要になってきている。本研究では、このような状況のもとで、並行プロセス間のメッセージ・パッシングに重点を置いたシステムの記述から、実時間制御プログラムを構成する方法を考察している。

2. 応用面からの並行プロセスの記述

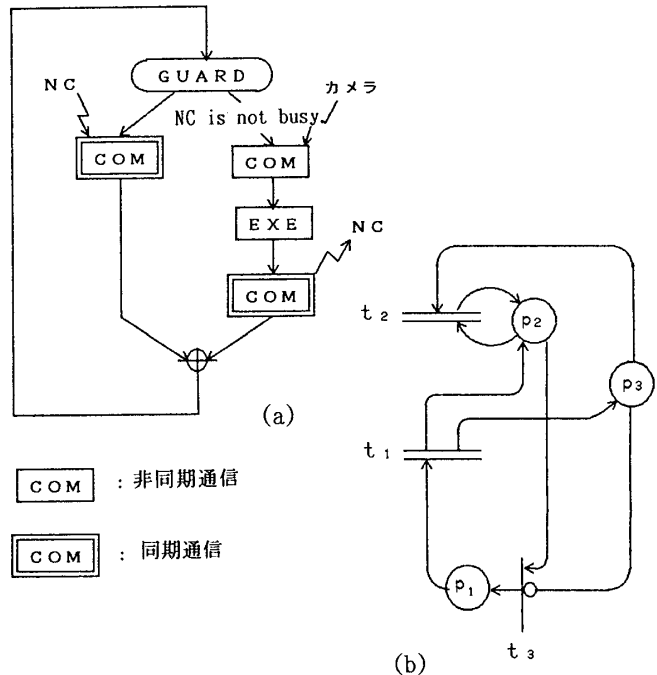
従来からの制御分野での実時間処理では、マルチタスクの状態遷移に重点を置き、これにいろいろな同期機構と通信機構を設け、全体をリアルタイムモニタが統括するという構成法を取っている。一方応用面に視点を置いてみると、一般の計算分野で、並列性の度合いが強くなるとデータフローモデルが重要になってくるのと同様に、実時間処理に於いてもイベント駆動あるいはメッセージ転送に視点を置いたシステムの記述が必要になってくるのである²⁾。大阪大学精密工学科では、NCフライス盤や組立ロボットの基礎的制御実験を試みているが、応用システムの記述と実行時のプログラムの処理形態の間にはかなりの開きがあって、この間の間隙を埋める工夫が色々行なわれている。これからの制御プログラム構成法としては、表1のようなステップを考えておかなければならない^{3), 4)}。

3. システム記述の構成要素

表1には、それぞれ特長のあるモデルが使われている。以下に、簡単な例を用いて、具体的なシステムの構築にそれらがどのように使われるかを示す。図1はマテリアル・ハンドリング・ロボットの例で、(a)がコントロールフロー・グラフ、(b)が部分ペトリネット

表1 制御プログラム開発のステップ

1	それぞれの制御機能(あるいは制御対象)に応じて逐次プロセスとしてのタスクプログラムを書き下し、モジュールとして登録する。
2	タスク間のコントロールフローグラフを定義することによって、並行に動作するタスクを指定し、計算負荷の分配を調整する。
3	並行に実行されるタスクプログラム間に通信チャンネルを設定し、通信による同期機構を付与する。
4	得られたシステムについてペトリネットによる解析を行ない、病的な状況が起こる可能性のないことを確かめる。
5	コントロールフローグラフとその中のメッセージ・パッシングを管理するリアルタイムモニタのもとで実行モードに入る。



t₁: NCへ工作物セット完了を伝える同期通信
 t₂: NCからの加工終了のメッセージを待つ同期通信
 t₃: カメラから部品の情報を受け取る非同期通信

図1 ロボットのシステム記述

トを示している。基本的な動作は、NC工作機械が動作中でなければ、カメラからの情報により工作物を受け取って工作機械にセットし、その旨をNC装置に通知する。また、選択的実行によって、NC装置から加工完了の通知を受け取る。ここでは、非同期通信、同期通信、ガードと云った概念（これらはAdaやoccamのもつ手法に対応する）が使われていることが分かる。同図(b)のネットは、トランジションの所で他の装置からのメッセージを受け取るようになっており、まだ並行プロセスとしては完成していない。

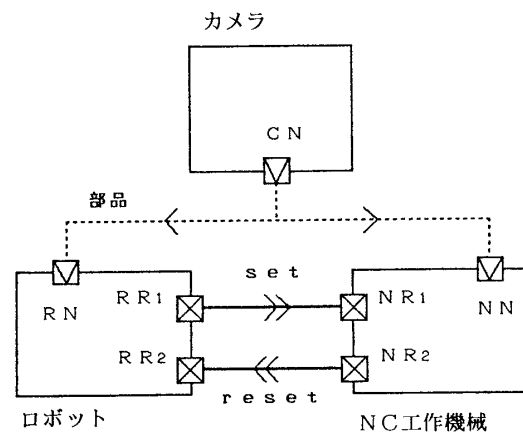
4. システムの統合化

モジュール間を通信チャネル（ソフトウェアの意味での）で結合するという考え方に従って³⁾、システム全体を記述する。このことは、複雑なシステムを容易に構成できるという意味と、実時間処理に固有の問題として、モジュール間の結合が実行時に動的に変わることを考慮することの二つの目的をもっている。結合の仕方をいろいろ変えたときのシステムの検証が、しばしば必要になるからである。

図2に示すように、図1のロボットに対して並行動作をするカメラとNC工作機械を通信機構を介して接続することを考える。同期を取る部分だけを抽出してペトリネットを作ると、図3のようになる。右上部分がロボット（図1に対応）、左上部分がNC工作機械で、下部がカメラである。トランジション1, 2, 3の意味は図1と同様である。トランジション4は、工作機械がカメラから工作物の種類を知らされる部分であり、トランジション5の発火でカメラからの通知動作が開始される。このようにして、システム全体のペトリネットが構成できると、可達木による解析が可能となる。本研究の最終目標は、言語処理系とシステム解析ツールの併用によって効率の良い制御プログラムの実現を達成することである。

5. まとめ

コンピュータ・コントローラに関する基礎的研究のため、現在、マルチボード・コントローラの試作を行っており⁵⁾、ソフトウェア作成段階に必要な考え方を述べた。具体的な設計ツールは、まだ開発途中であるが、制御プログラムの作成に於いても、単なる手続きの記述にとどまらず、メッセージ・パッシングを考慮したシステム記述がますます重要になってくると思われる。



☑ : 非同期通信用ポート

☒ : 同期通信用ポート

→→→ : 同期通信

--->--- : 非同期通信

図2 通信機構を用いた結合

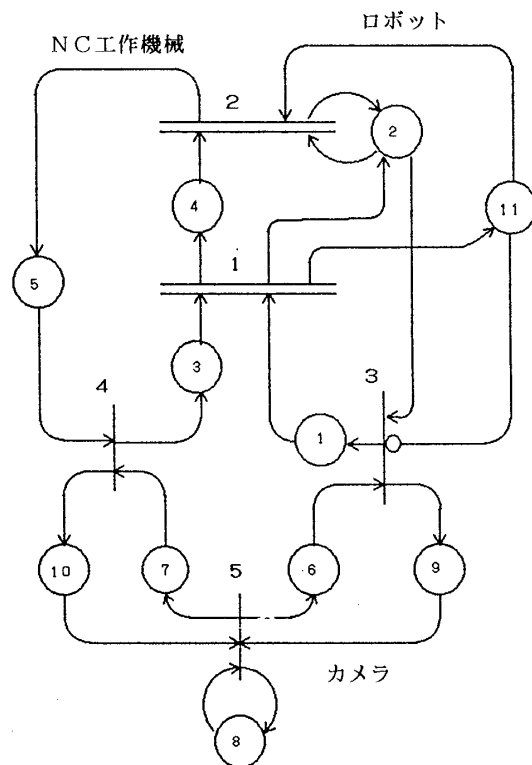


図3 ペトリネットの統合

参考文献

- 1) Makinouchi, S. et al., 1985, "Computer Numerical Control for Assuring Machining Accuracy," *Robotics and CIM*, 2, 1.
- 2) 山縣敬一, 牧之内三郎: ミニコンピュータにおけるイベント処理指向型のリアルタイム・モニター, *情報処理*, 18, 6(1977).
- 3) Sloman, M. et al., 1985, "The CONIC Toolkit for Building Distributed Systems," *IFAC Proc. Distributed Computer Control Systems*.
- 4) Peng, D. et al., 1987, "Modeling Concurrent Task Execution in a Distributed System," *IEEE Trans. C-36*, 4.
- 5) Yamagata, K. et al., 1988, "Designs of Hardware Architecture and Control Kernel for Multi-Microcomputer Systems in Flexible Manufacturing," *Proc. USA-JAPAN Symposium on Flexible Automation*.