

離散型生産システムの挙動モデルにおける SPIN を活用した要制御部位の検出手法

高塚 佳代子

宮崎大学工学部教育研究支援技術センター

1. 概要

現在開発中の「モデルに基づく離散型生産システム制御系の系統的設計手法」では、プロセスネットワーク型の中間モデルで表現された対象システム内に潜在する「制御を要する危険部位」の検出が重要な課題となっている。本稿では、並列分散システムの動作検証ツールである Spin[1]のシミュレーション機能を活用し、このような「危険部位」を動的に絞り込む方法について提案する。

2. 背景と目的

本研究室では、離散型生産システムの挙動を表現するための状態チャートベースのモデル「拡張時間状態チャート」を開発し、そのモデルに基づき制御系を実現するための手法を開発してきた。ここで、対象とするシステムは、例えば、医薬品の生産バッチプラントのような多品種少量で、変種変量生産のフローショップ型装置群を持つようなシステムである。生産要求はバッファリングされ、時間駆動でスケジュールされるが、通常予定通りには処理は進まない。そこで、この種のシステムでは、計画がくるっても対応できるようにするための方法として、時間駆動で得られたスケジュールリング結果を事象駆動(順序情報)に変換してシステムに与える必要がある。この時、新たな不都合{バッティング、非決定性}が生じる可能性があるため、それがどこで起こるかを事前に検出しておくための手法の開発が必要となる。この問題に関し、本研究では、プロセスネットワーク型モデルの構造的特徴に基づいて要制御部位を顕在化させるという手法を開発してきた。しかし、この方法では、要制御部位を完全に絞り込むことはできないことと、要制御部位であるにもかかわらず検出できないケースも在る、といった問題点がある。以上のことから、本研究の目的は、プロセスネットワーク型モデルの見かけ上は制御を要する箇所として検出されるが実際には競合が起こらない部分を除外する方法、及び、見かけからは検出されないような制御を要する箇所を見つける方法を提案することである。以下では、本手法全体の枠組み、及び、本研究室で既に開発されているプロセスネットワーク型モデルの構造的特徴、即ち「見た目」から「制御を要する可能性のある箇所」を検出する方法について説明した後、本研究目的について示す。

3. 本制御系実現手法の枠組み

図1が本研究室で開発してきた制御系を実現するための枠組みである。ここでは、2種類の挙動モデルを採用するが、その理由は、「プロセスネットワーク型モデル」は、物質情報の流れを自然に表現できるため、バッティングや非決定性が何処で起こり得るか?を顕在化させやすいのに対し、「拡張時間状態チャート」の方は、明確な階層性・並行性を持つため、制御系のドライバーを操作するためのプログラムである実行モジュール群への変換がしやすいためである。従って、まずプロセスネットワーク型モデルで制御を要する箇所を見つけ、その各々の箇所に制御内容を独立に埋め込むことが出来るような拡張時間状態チャートを、プロセスネットワーク型モデルからの変換によって作成し、そこから実行モジュールをテンプレートを用いた形式的な手順に従って抽出させるという流れを採用する。

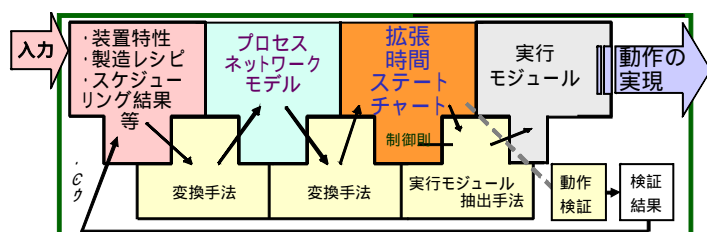


図1 挙動モデル「拡張時間状態チャート」に基づく制御系設計手法

4. 「制御を要する可能性のある箇所」をプロセスネットワーク型モデルの構造的特徴から検出可能とするための仕組み

図2で示すように、プロセスネットワークは、並行動作する各プロセスをノードで、メッセージの通信路をアークで表現し、プロセスがストリーム並列で動作するネットワークのことが、本研究では、更に、バッティングや非決定性が起こり得る箇所が常に同じ構造的特徴として現れるようにするための記法上の制限を与えた[2]。その制限を与えるために、まず、通常のプロセスネットワークの概念を具体化するための幾つかの記法や意味付けを次のように与えた。即ち、プロセスは、「内部プロセス」の集まりであり、それらは選択的に実行され、これらの内部プロセスは更に小さいプロセスである「操作」のシーケンスとして定義され、アークの根元は、「装置の要求」などのメッセージが出力されるべきタイミング（具体的には、特定の操作の直前か直後）を表わす位置に付けさせ、一方のアークの先端は、それを受領すべきノード自身に付けさせるようにし、「メッセージの受領」=「使用を許可」という意味付けをした。以上の取り決めの下で、バッティングや非決定性が起こり得る箇所が常に同じ構造的特徴として現れるようにするために、「XOR 関係にある複数のアークの根元、或いは先端は、同一部位に集める」という記述上の制限を与えることとした。

5. 見かけ上は制御を要する箇所として検出されるが実際には競合が起こらない箇所を検出する方法

検出方法を形式化するには、汎用性という意味で無理があったため[3]、本研究では、並列分散型システムのための検証ツール「SPIN」のシミュレーションツールを利用した検出方法を開発した(図3参照)。本方法では、検出のためのプログラムである「検出用プローブ」を作成するのだが、その処理内容は、システムの実行中に「見かけ上制御を要する箇所」でのメッセージの発生を監視し(図中(1))、実際に競合が起きたパターンを抽出、及び記録し(図中(3))、実行を続けさせるために便宜上与えて置いた優先順位に基づいて、競合解消を行わせ(図中(4))、以上を優先順位を変えて繰り返すことで起こり得る全ての競合パターンを検出する(図中(5)から(1)へ戻り繰り返す)、というものである。以上のような検出用プローブをシステムへ埋め込んで網羅的にシミュレーション実行し、最終的に記録されなかったパターン、つまり競合が起きなかったパターンを「除外すべき対象」と判定する。

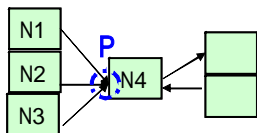


図2 プロセスネットワークの例

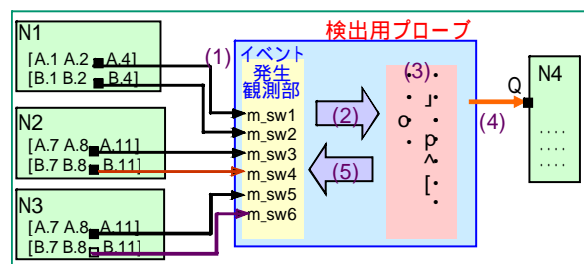


図3 「検出用プローブ」を使用した検出方法の概観

6. 制御を要するがプロセスネットワーク型モデルの見かけ上からは検出されない箇所を検出する方法

この方法に関し、SPINを活用した次のような方法を考えた。まず、モデルの構造からは検出できない潜在的な要制御部位が「あるかどうか」のチェックは、SPINでのシミュレーション実行を網羅的にを行い、システムが停止するかどうかによって行う。もし、停止した場合は、SPINが提供する「シミュレーションログ」により、停止時の状態変数の値を参照することが出来るので、「何故停止したか？」の原因を解析することができる。更に、「原因」が同定できた場合は、今回見つかった要制御部位が構造的な特徴として現れるようにするための修正を、プロセスネットワーク型モデル上へ施す。もし、原因を同定できなかった場合は、シミュレーションログに表示させる状態変数の種類や数を変更し、再度シミュレーションを行う。

7. プロセスネットワーク型モデルから PROMELA への変換

Spinを利用するためには、Promelaで記述されたモデルが必要であるため、本研究では、プロセスネットワーク型モデルからPromelaへの変換手法の開発と、自動変換システムの作成を行った。図4は、本研究で開発した変換方法の基本的な考え方を示したものである。具体的には、データ抽出用テンプレートを用いてプロセスネットワーク型モデルの情報を抽出し、Promelaソーステンプレートを用いてテキスト形式に集約するというものである。具体的には、プロセスネット

ワーク型モデルの“ノード”を、ノード名というキーで抽出し、Promela での“PROCESS”へ変換する(図中)。また、ノードの内部データの部分、即ち、操作の一手順と見なしている部分を、操作名として抽出し、“内部 PROCESS”へ変換する(図中)。更に、メッセージ送信のための“アーク”は Promela での“チャンネル”へ変換するが、アークの先端は、プロセス、或いは操作の開始条件や終了条件として抽出し(図中)、チャンネルの IN へ変換する。一方、アークの根元は、ノード或いは操作の開始条件や終了時に出力されるメッセージとして抽出し、チャンネルの OUT へ変換する(図中)。本研究では、以上のような変換を自動的に行わせる変換システムを実装した。

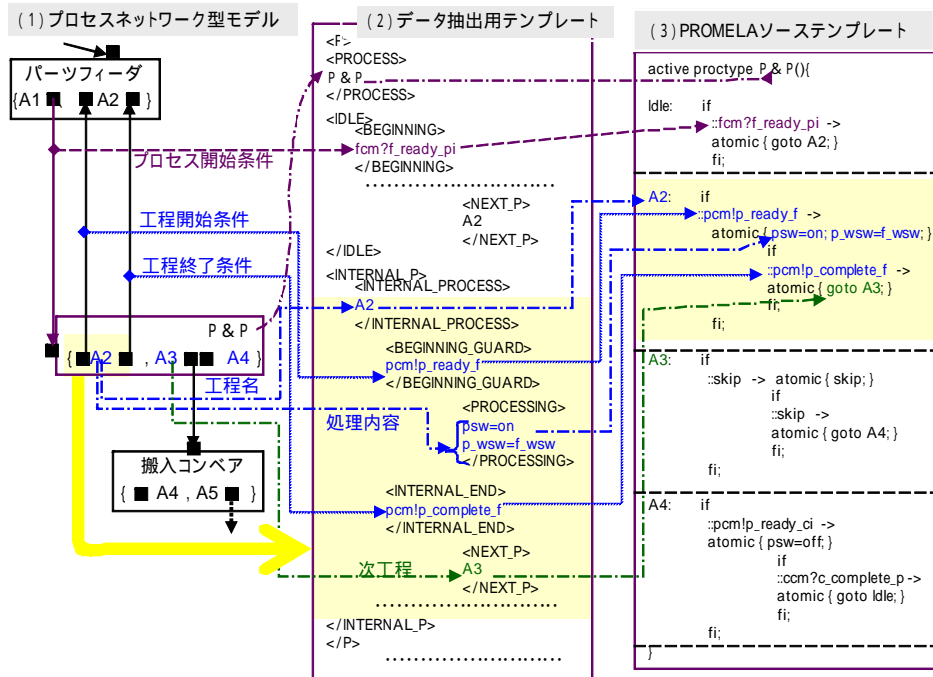


図4 プロセスネットワーク Promela 変換システムによる変換例

8. 検証実験

本研究で開発した手法の検証を行うために、図5のような2工程のフローショップ型生産システムであるFA実験装置を対象として実験を行った。

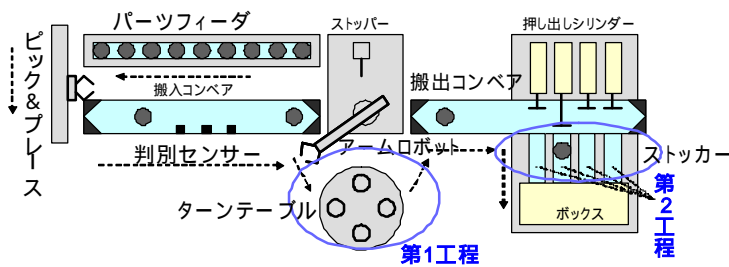


図5 対象プラント

まず、本研究で開発した変換手法と自動変換システムの妥当性の検証を行った。具体的には、FA装置のプロセスネットワーク型モデル(図6)を、本研究で開発した変換システムへ入力したところ、変換システムは、データ抽出用テンプレートを用いてプロセスネットワーク型モデルの情報を抽出し、Promelaソーステンプレートを用いてテキスト形式に集約し、正しいPROMELAプログラムを生成した。そして、非決定性が残っている部分に関してはランダムに処理を選択するSPINのランダムシミュレーション機能を用いて得られたPromelaプログラムを実行したところ、プロセスネットワーク型モデルの記述どおりに動作することが確認された。

次に、見かけ上は要制御部位としての構造的特徴を持つが、実際には競合が起こらないような箇所を検出除外する方法の妥当性について検証した。即ち、まず、対象システムの構造的な特徴上、制御を要する可能性のある箇所、具体的にはアームロボット使用要求の競合が発生する可能性のある箇所に対し、そこに埋め込むための検出用プローブを作成し、それを先に生成したPROMELAプログラム中に埋め込んだ(図7)。そして、SPINのシミュレーションモードでシステムを

実行し、イベントの発生競争を監視、検出、記録させた。その結果、例えば図中 P の部分では、競合する可能性のあるメッセージ 8 個に関する全ての組み合わせパターン 247 通り中、実際に競合するパターンとしては 39 パターンが抽出され、それ以外は本来要制御部位ではない箇所であることが判った。なお、以上の結果は、ハンドトレースによる結果と一致していたことから、本検出方法は妥当であることが確認できた。

更に、プロセスネットワーク型モデルの構造的特徴として現れない潜在的な要制御部位が提案する手法で検出可能かどうかの実験を行った。具体的には、アームロボット使用の競合に関する要制御部位の一部が顕在化されないようなモデルを使用し、それに関する制御は埋め込まれていないようなシステムの Promela プログラムを自動生成させ、それを SPIN でシミュレーション実行した。その結果、システムは停止し、SPIN のシミュレーションログから、停止時の状況として、「ターンテーブル上の加工装置がすべて加工中であるにもかかわらず、アームロボットが搬入コンベア側からのワークを搬送してしまう」といったデッドロックが発生したことが判った。この内容から、実行停止の原因はアームロボットに対する制御の不備であることが特定できるため、プロセスネットワーク型モデルを修正し、再び実行を試みたところ、今度は停止することなく実行された。以上のことから、シミュレーションログから要制御部位の同定と適切な対応の提案が実行できることが確認できた。

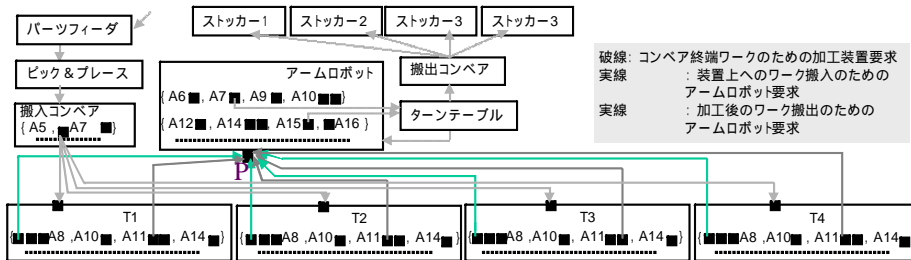


図 6 プロセスネットワーク型モデルの概観

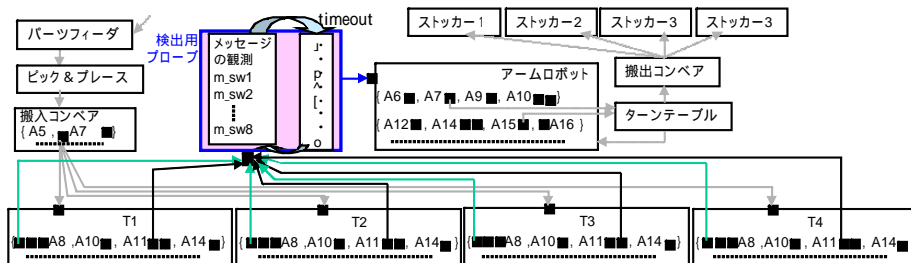


図 7 対象システムへのプローブの埋め込み

9. 考察とまとめ

以上のことから、プロセスネットワーク型モデルを使用する本制御系設計手法では、支援ツールとして SPIN を利用することにより、「どのように制御すべきか」以前に認識する必要のある「何処を制御すべきか？」を的確に指摘できる可能性があることが分かった。

10. 参考文献

- [1] Gerard J.Holzman: THE SPIN MODEL CHECKER
- [2] 高塚佳代子 富田重幸: プロセスネットワークに基づく離散事象システムのモデル化手法、第 48 回離散事象システム研究会予稿集 (2010)
- [3] 高塚佳代子 富田重幸: 離散型生産システムの挙動モデルにおける要制御部位の検出手法 (- SPIN を活用した動的検出法の提案 -) 第 45 回離散事象システム研究会予稿集予稿集 (2009)