

ペトリネットのFA制御への応用

Application of Petri Nets to Control for Factory Automation

長尾 陽一
Yoichi Nagao

川崎重工業株式会社 技術開発本部
Technology Development Group, Kawasaki Heavy Industries, Ltd.

1. まえがき

最近、FA (Factory Automation) の進展に伴い、ロボット、NC機械、搬送装置、立体倉庫等を組み合わせた生産システムを構築し、製品の多様化、製品サイクルの短期化に対応する傾向が強くなっている。このようなシステムでは、非同期・並行的に動作する複数の機器を互いに同期、排他させながら統括的に制御する必要がある。このために設置されるシステムコントローラには、上記制御機能に加え、各機器との間の情報交換や上位コンピュータへの管理情報の提供、あるいはオペレータとの対話処理機能が必要とされる。従来、このソフトウェアの開発においては、PC (Programmable Controller) ではラダーダイアグラムを、制御用計算機では Basic や C などの手続き型言語を用いて、人間の設計した制御仕様に基づいてプログラミング作業を行っている。この方法では、

- ・プログラミング時に誤りが発生し易い、
- ・作成したプログラムが設計した仕様を満足するかどうかの検証が困難である、
- ・制御ソフトウェアの保守が容易でない、

といった問題があった。

これに対して、最近、ペトリネット¹⁾を用いたシーケンス制御システムの記述モデルが注目を集めている²⁾。ペトリネットは、人間が理解し易い図的表現であり、FAシステムにおいて要求される同期、排他、並行動作といったシーケンス制御上の機能の表現が容易である。そのため、ペトリネットに基づいたシーケンス制御のためのプログラミングツールがこれまでにいくつか開発されている³⁾⁻⁷⁾。しかし、ペトリネットの利用だけでは高度なFA制御ソフトウェアを構築するには不十分であった。

そこで、筆者らは、上記問題を解決するために、従来の手続き型言語とペトリネットという図的言語を融合して複雑な制御仕様を簡単に記述できるモデルを提案し、概略設計、詳細設計、プログラ

ミング、テストの各開発段階を一貫して支援するFA制御ソフトウェア開発支援システムK-NE Tを開発した⁸⁾。これを実際のAGV (Automated Guided Vehicle) 制御システムの開発およびロボットを用いた形鋼自動加工制御システムの開発に適用し、有効な結果を得たので以下に紹介する。

2. K-NE Tの概要

K-NE Tは、制御仕様をペトリネットで記述することによりそのソフトウェアを開発するためのプログラミング環境である。K-NE Tには、ペトリネットで表現した制御仕様を入力・編集するためのエディタ、その内容を検証するためのシミュレータ、入力されたネットをコントローラで動作できる形式、すなわちC言語記述のプログラムに変換するためのジェネレータ、およびペトリネットによる制御仕様を整理した形式で印刷するためのレポータが設けられている。また、コントローラに接続してその制御状況をリアルタイムに表示するモニタ機能がある。K-NE Tの機能構成をFig. 1に示す。

K-NE Tの特長は以下のとおりである。

(1) ハイレベル・ペトリネットによる複雑な制御仕様の表現

K-NE Tで用いるペトリネットでは、トークンが属性を持つことができ、トークンの種類に応じてネット内の流れを変えることができる。さらに、事象の発生および処理時間の確率的扱いを可能としている。

(2) IF-THENルールによる動作フロー条件の表現

K-NE Tでは、状態遷移の条件としてネット上の枝(許可枝、抑止枝)⁴⁾を用いて表現するのではなく、これをIF-THENルールにより表現することにより、動作フローのみがネット表現され、制御の流れが理解し易くなっている。

(3) ペトリネットと手続き型言語の融合

K-NE Tでは、シーケンス制御における動作フローをペトリネットで、煩雑な情報処理を従来の手続き型言語Cで記述することにより、両者の特長が組み合わされて、制御の流れを理解し易い形で表現できるとともに、複雑な処理の記述も可能となっている。

*〒673 兵庫県明石市川崎町1-1
TEL 078-921-1645 FAX 078-921-1603

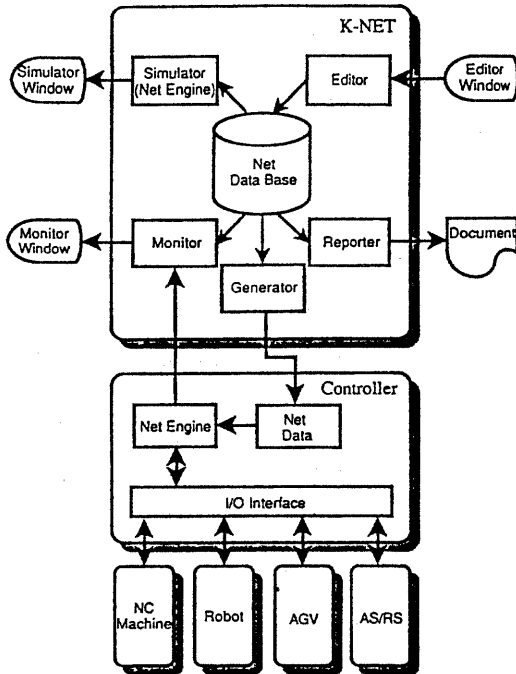


Fig.1 System configuration of K-NET

3. K-NETにおける制御仕様の記述モデル

K-NETにおいて制御仕様を表現するために用いるペトリネットは、特定の機能を有する数種類のボックス、処理の流れを制御するゲート、両者間の関係を表す有向枝、およびその有向枝に沿ってネット内を流れるトークンから構成され、さらに手続き型言語Cを用いて処理内容を記述できるユーザ定義関数、入出力を定義する論理I/Oが用いられる (Fig. 2)。各要素の詳細について以下に説明する。

3.1 トークン

トークンはシステム内を流れる物や情報に対応する。トークンはたとえばワークのタイプ、形状などの属性情報を持つことができ、“クラス”と呼ばれる属性の構造体によって分類される。

3.2 ボックス

3.2.1 ボックスの状態

ボックスはトークンの存在の状況により、システムおよび各機器の状態を表現する。ボックスに収容可能なトークンの数をそのボックスの“容量”と呼び、通常、ボックスの容量は1である。ボックスの状態としては“受容状態”および“励起状態”が定義される。受容状態とは、トークンを受け入れることができる状態、すなわち容量と同一数のトークンが存在しない状態である。また、励起状態とは、出力ゲートが発火してトークンの移動が可能な状態である。通常、ボックスはトークンが存在する場合に励起状態にあるが、後述の時間ボックス、計数ボックス、処理ボックス、マ

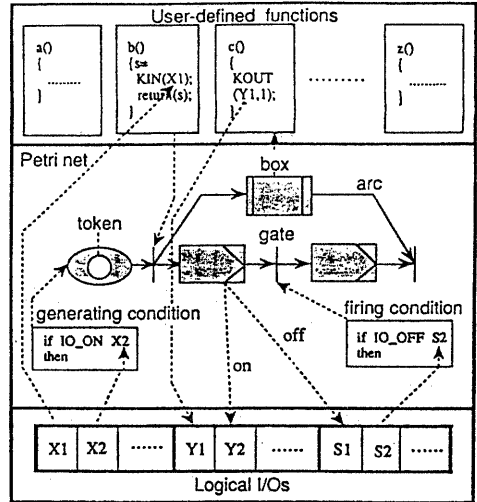


Fig.2 Net elements in K-NET

Table 1 Kinds of box

name	symbol	capacity	specifying items
state box		1	nothing
buffer box		n	capacity, token exiting order
time box		1	time t
count box		m	count value m
calculation box		1	operator, logical I/Os
signal box		1	logical I/Os, output values
process box		1	process, condition, post-process, time
source box		n	class, capacity, condition, process
macro box		-	nothing
entrance box		1	nothing
exit box		1	nothing

クロボックスではトークンが存在するからといって、必ずしも励起状態にはなく、システムの状態によって変化する。

3.2.2 ボックスの種類

ボックスはその機能によって、以下のように分類される (Table 1)。

(1) 状態ボックス

容量1の単純なボックスである。これは、機能的に以下のボックスの完全な部分集合であり、それらを用いて表現することができるが、入力の手軽さおよびネットの見易さから導入した。機器間のインターロックを表現する際に、機器の稼働状態あるいは遊休状態に本ボックスを対応させる。

(2) 容量ボックス

1以上の容量を持つボックスであり、バッファあるいは待行列を表現する。トークンの取り出し順序として、先入先出、先入後出、あるいはトークンの特定の属性値の昇順/降順を指定すること

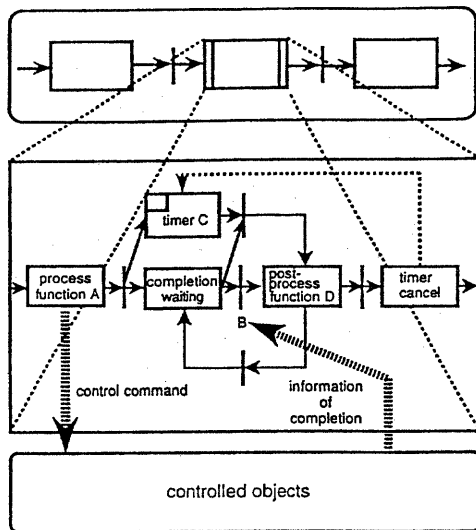


Fig. 3 Function of process box

ができる。

(3) 時間ボックス

時間の概念を導入したボックスである。トークンがこのボックスに入ってから指定時間経過後、ボックスは励起状態に遷移する。時間は数値、論理 I/O の値、確率分布のいずれかで指定される。時間ボックスは、シーケンス制御におけるタイマを表現する。また、制御対象のモデルに使用することにより、シミュレーションによるサイクルタイム計算に利用される。

(4) 計数ボックス

1 以上の容量を持つボックスであり、容量に等しい数のトークンが入って初めて励起状態になる。ボックスからトークンが出るときには、最初のトークンのみが移動し、他はここで消滅する。計数ボックスはシーケンス制御におけるカウンタを表現する。

(5) 演算ボックス

トークンがこのボックスに入ると、論理 I/O の値に関する算術演算（加算、減算、乗算、除算、代入）を実行する。

(6) 信号ボックス

トークンがこのボックスに入ると、指定された論理 I/O に信号を出力する。

(7) 処理ボックス

ユーザが定義した処理関数の呼び出しを含む一連の動作を行う。すなわち、Fig. 3 に示すように、処理ボックスにトークンが入ると、まず、A の処理関数が呼び出される。処理関数 A では、通常、外部プロセスに対する信号出力、タスク起動、その他の情報処理が実行される。その後、処理完了条件 B の成立を待ち、同時にタイムアウト監視のためのタイマ C を起動する。論理 I/O に関する論理式の積和形式で記述された完了条件 B が成立するか、あるいはタイムアウト発生時には、後処

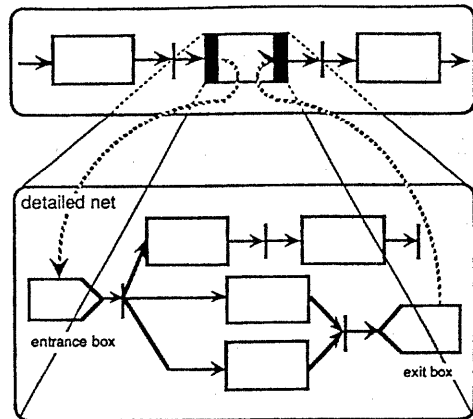


Fig. 4 Macro box and detailed net

理関数 D が呼び出されるが、引き数によってその起動原因を知ることができる。後処理では、完了条件を調べることにより、タイマ C をキャンセルしてボックスを励起状態に遷移させるか、あるいは再び完了待ちの状態に戻るかを制御することができる。このような複雑な動作に対して、情報処理を従来の手続き型言語で記述し、制御で典型的な“実行-監視-チェック”を一つのボックスで記述することにより、ネットの作成を容易にし、ネット構造を見やすくしている。

(8) 発生ボックス

論理 I/O に関する論理式の積和形式で記述された条件が成立した時点で、指定したクラスのトークンを発生させる。トークンは 1 度だけ、あるいは条件の成立している間、決定的/確率的な時間間隔で発生させることができる。発生したトークンの属性値はクラスに記述された初期値を設定されるが、引き続きユーザ定義関数が呼び出されてこれを変更することもできる。

(9) マクロボックス

より詳細なネットでも置き換えられるボックスである。Fig. 4 に示すように、マクロボックスにトークンが入ると、対応する詳細ネットの入口ボックスにトークンが入り、以降、詳細ネット内をそのネット構造に従って移動する。そして、トークンが出口ボックスに入った瞬間、上位のマクロボックスは励起状態となる。マクロボックスは、システム全体の概略設計から始まって詳細化へ至るトップダウン設計を支援し、かつシステム全体の構造の見通しをよくする。

3.3 ゲート

ゲートはトークンの流れを制御するものである。ゲートは、基本的には、その入力ボックスのすべてが励起状態にあり、出力ボックスのすべてが受容状態にある場合に発火し、すべての入力ボックスからトークンを 1 つ取り出し、すべての出力ボックスにトークンを 1 つ置く。複数のゲートの発火が競合する場合、固定的な優先順位によって発火するゲートが決定される。以上の基本機能に加

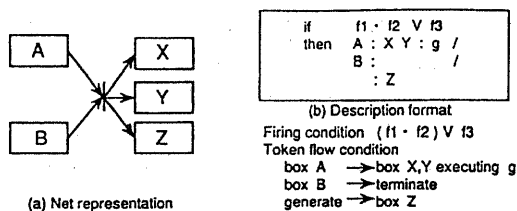


Fig. 5 Gate specification

え、Fig. 5 に示すような IF-THENルールの記述により、以下に示す発火制御が可能である。

(1) 発火条件

ボックス内のトークンの有無、その属性値、論理 I/O の値等の論理式の積和形式による発火条件を IF 部に記述する。発火条件は複数個並記することができ、最初に成立した条件に対応する制御が実行される。

(2) トークンのフロー制御

上記発火条件を満足する IF 部に対応した THEN 部の内容に従って、トークンの移動および属性継承を含むフロー制御が実行される。この時、入力ボックス毎にユーザ定義関数の呼び出しを指定することができ、トークンの属性値変更の他、トークン移動に伴う任意の処理（例えば、共有メモリに対する書き込みなど）を行うことができる。

3.4 ユーザ定義関数

手続き型言語 C で記述されたユーザ定義関数は、前述のように処理ボックス、発生ボックスあるいはゲートから呼び出される。ユーザ定義関数内では、論理 I/O やトークン属性値にアクセスするためのシステム関数を用いることができる。

3.5 論理 I/O

I/O 設計は、論理設計と物理設計の 2 段階で行われる。論理 I/O は、前述のように時間ボックスの時間、演算ボックスの被演算子、信号ボックスの信号名、発生ボックスの発生条件、処理ボックスの完了条件、およびゲートの発火条件に使用される。外部プロセスへの信号入出力以外に、ネット間での信号の授受にも使用可能である。論理 I/O を物理 I/O に割当てることにより、実際の信号が外部プロセスに出力、あるいは外部プロセスから入力される。

4. K-NET の機能

4.1 エディタ

K-NET には、ベトリネットでの制御仕様を記述するためのグラフィックエディタを設けている。編集機能として、ネットの挿入、修正、削除、移動、複写機能がある。さらに、論理 I/O の編集機能、トークンの属性構造体を規定するクラスの編集機能が設けられている。

4.2 シミュレータ

前編集機能で入力されたネットが設計通りになっているか否かを検証するためにシミュレーシ

ョン機能を設けている。シミュレーションを実行する前には、ネットのボックス上にトークンの初期配置を行う。実行には、現時点で移動可能なトークンを 1 回だけ移動させる方法と、連続的にトークンを移動させる方法が用意されている。デバッグのために、シミュレーション途中の任意の時点で、各ボックスにおけるトークンの配置/除去、各論理 I/O の値の変更を行うことができる。また、ネット上のトークンの各属性の現在値、およびトークンのボックス滞在、ゲート通過、さらに論理 I/O のオン/オフに関するタイムチャートを得ることができる。

以上の他に、確率モデルを利用したシミュレーションにより、FA システムの性能評価を行うこともできる。その結果として、トークンのボックス滞在時間、ゲート通過間隔に関する統計情報が得られる。

4.3 レポータ

本機能は、ネット情報およびシミュレーション結果を整理した形で印刷・出力する。本機能により、出力される項目は次のとおりである。

- ① ネット階層図および構成図
- ② ボックス、ゲート、論理 I/O、トークンクラスの仕様および名前一覧表
- ③ 各論理 I/O を参照するボックスとゲートの名前一覧表
- ④ タイムチャート図
- ⑤ ボックス、ゲート、および論理 I/O に関する統計情報

4.4 ジェネレータ

本機能は、入力・検証されたネットを現場で使用するコントローラに搭載するために、ネットを C 言語記述のプログラムに変換する。

4.5 モニタ

本機能は、コントローラを K-NET に接続した状態で動作し、実機試験を支援する。コントローラから送られて来る制御状況に関する情報を基にして、シミュレーション時と同じネット図におけるトークン移動として、コントローラが制御を実施する状況をリアルタイムに表示する。

5. K-NET による制御ソフトウェアの開発手順

K-NET を用いて制御ソフトウェアを開発する手順は一般に以下のようなになる (Fig. 6)。

- ① エディタを用いて、制御ネットおよび制御対象ネットを作成する。ここで、各ネット間の関係は論理 I/O によって表現される。
- ② シミュレータを用いて、制御ネットの妥当性を検証する。
- ③ 制御ネットおよび制御対象ネットを分離した後、ジェネレータを用いて、C 言語プログラムに変換し、それぞれをコントローラ用、シミュレータ用計算機上に実現する。この時、ネット間の論理 I/O は各計算機間を接続す

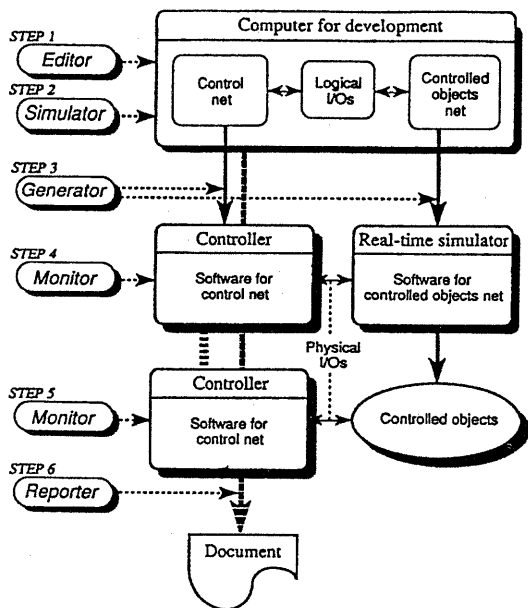


Fig. 6 Procedure for development of control software

る物理 I/O に割り当てられる。ここで、シミュレータ用計算機は制御対象動作を模擬するリアルタイムシミュレータとして機能する。

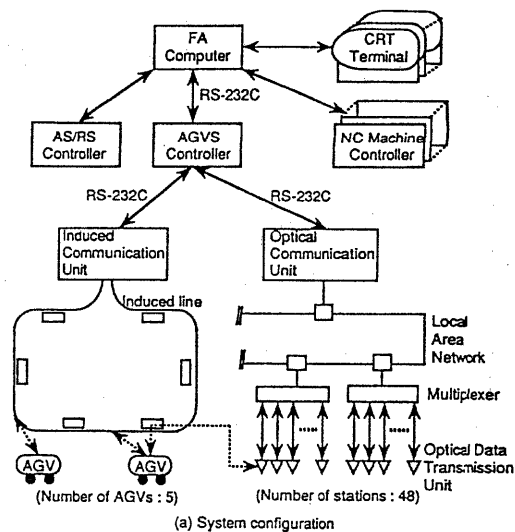
- ④ リアルタイムシミュレータを用いて、コントローラのデバッグを行う。この時、コントローラの動作はモニタを用いて確認できる。
- ⑤ リアルタイムシミュレータを実際の制御対象に置き換え、実機を用いたコントローラのデバッグを行う。この時にも、コントローラの動作はモニタを用いて確認できる。
- ⑥ 以上のテストを終了した後、レポートを用いて制御ネットの仕様書を作成する。

6. K-NETの適用例

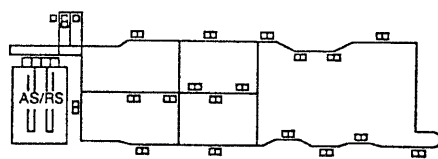
6.1 AGVシステム制御装置の開発⁹⁾

(1) AGV制御システムの概要

本システムは、Fig. 7 に示す走行経路と、自動倉庫ならびに 48ヶ所のステーションを有し、最大 5 台の AGV を投入できるように設計されている。各 AGV は、ステーション停止中に、光通信装置を経由して AGV システム制御装置（以下、AGV S 制御装置と呼ぶ）と情報の授受を行う。AGV S 制御装置からは、走行、荷積み、荷卸しの指令ならびに状態の問い合わせが行われ、各 AGV からは指令データ受取応答、動作完了情報、および状態情報が送信される。AGV は走行指令として、“直進”、“左折”、“右折”、“ステーション通過” および “ステーション停止” 等の動作指示の系列を通知され、走行経路の分岐点ならびにステーションにおいて順次この指示を実行しながら自律的に移動する。なお、各 AGV の状態情報はステーション間を走行中にも誘導通信装置を経



(a) System configuration



(b) Route map and station layout

Fig. 7 AGV control system

由して一定時間間隔で AGV S 制御装置へ通知される。

AGV S 制御装置は工場全体を管理する FA コンピュータから通知される搬送要求（荷積みステーションと荷卸しステーションの対）をもとに、光通信装置および誘導通信装置を経由して、AGV の運行制御と運行監視を行う。そのため、AGV S 制御装置には、

- ① 各搬送要求に対し、最適な AGV を割り当てる台車割当機能
- ② 割り当てられた搬送要求に従って、AGV に対して荷積みステーションへの走行、荷積み、荷卸しステーションへの走行、荷卸しを順次実行させる荷物搬送機能
- ③ 搬送の完了した時点で、実行すべき搬送指令が存在しない場合、AGV を待機ステーションへ移動させる待機走行機能
- ④ バッテリ低電圧となった AGV を搬送終了後に充電ステーションへ移動させる充電走行機能

が設けられている。この他、異常時の対応、マンマシン、および周辺機器との通信機能がある。

(2) K-NETによる開発

上記 AGV S 制御装置のソフトウェア開発に K-NET を適用した。本装置の機能の中で、荷物搬送などシーケンス的な動作はベトリネットで記述し、複雑な情報処理を含む台車割当機能、高速な処理が必要な周辺機器との通信機能、および画面表示を行うマンマシン機能は従来の C 言語で実

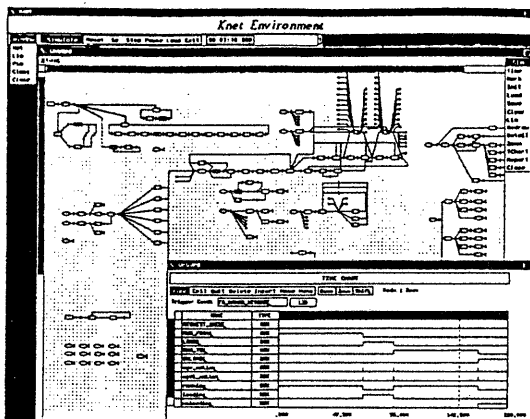


Fig. 8 Simulation window for AGV system

現した。なお、開発マシンとしては、事務所ではワークステーションSUN3を、現場では東芝のラップトップパソコンJ3100を用い、また実行マシンとしてはNECのFAパソコンFC9801を用いた。シミュレーション時のネット画面をFig. 8に示す。K-NETの利用によって、制御プログラムの設計、作成から現地試験まで含めた本AGVシステムの制御ソフトウェア開発は、従来の開発方式で要する約1/3の費用で達成できた。

(3) ベトリネットモデル

AGV群の運行制御に対するネットの概念図をFig. 9に示す。AGVの荷物搬送および待機の基本シーケンスは各AGV毎にボックス1~5で表現される。FAコンピュータから搬送要求が通知されると、それに対応するトークンが発生ボックス10に発生し、搬送要求バッファを表現する容量ボックス11に移る。AGV待機中に相当する状態ボックス5にAGVに対応するトークンが存在して、処理ボックス13あるいは14の台車割当処理により、ボックス11内の搬送要求トークンにAGV1が割り当てられると、ゲート12が発火し、ボ

ックス1にトークンが移動して荷物搬送が始まる。荷積み完了後に、荷卸しステーションへの走行に対応するボックス3にトークンが入ると、トークンはボックス6に移動し、光通信装置を経由してAGVに動作指令を送信した後、走行の完了をボックス7で待つ。AGVから走行完了が通知されると、発生ボックス9にトークンが発生し、ゲート8が発火して、次の荷卸し処理を行うためにトークンはボックス4に移動する。

本AGVS制御装置を実現するために作成したネットの規模は、ボックス284、ゲート281、論理I/O63、トークンクラス5、ユーザ定義関数は82であった。

6.2 AGVシステムシミュレータの開発¹⁸⁾

(1) シミュレータの構成

前述のAGVシステムのリアルタイムシミュレータをK-NETを用いて開発した。シミュレータは実機系のモデルとマンマシン部により構成される。マンマシン部は路線図やAGVの位置・状態の表示機能と、AGVの投入・削除、ならびにバッテリー低電圧や異常状態を強制的に発生させるためのキー入力を受け付ける機能を有している。AGVの動作を模擬する実機系のモデルは、AGVS制御装置もしくはマンマシン部からの指令に基づいて動作し、その結果をAGVS制御装置とマンマシン部に通知している。

(2) ベトリネットによるモデル

実機系のモデルとして、走行経路の構成やステーションの配置等の情報は実経路に近い2次元的な図で表現できるベトリネットで、複雑な情報処理を伴うマンマシン部およびAGVS制御装置とのインターフェースはC言語によるプログラムで実現した。

ステーション停止中のAGVはAGVS制御装置と情報の授受を行い、動作開始処理を必要とするので、ステーションはC言語によるユーザ定義関数を記述可能な処理ボックスで、ステーションや経路の分岐点および合流点の間の部分的な走行

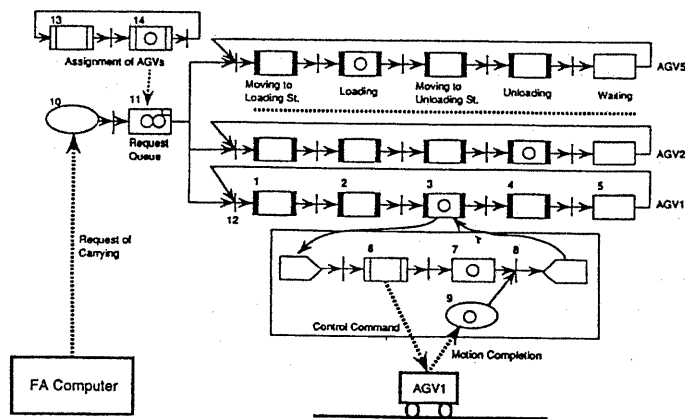


Fig. 9 Net model for AGV system control

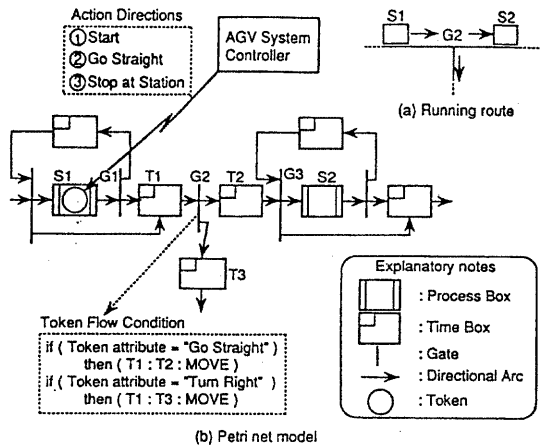


Fig. 10 Net model of AGV and route

経路は走行時間を表現するために時間ボックスで表現し、ステーションと部分的走行経路間の接続関係は、ユーザ定義関数が記述可能なゲートおよび有向枝を用いて表現している。たとえば、Fig. 10(a)に示す走行経路のペトリネットは同図(b)のようになる。また、AGVは動作指示データを属性とするトークンによって表現される。

AGV S 制御装置から走行指令が通知されると、動作指示系列が既定のデータ領域に設定され、AGV トークンの存在する処理ボックスにおいて、動作指示系列の先頭データがトークンの属性に設定される。以後、トークンの移動は、ボックスの出力側ゲートに記述されているトークンフロー条件によって制御される。そして、ゲートが発火する毎に、そこに記述されたユーザ定義関数により、トークンの位置が変更されたことをマンマシン部に通知すると同時に、トークンは次の動作指示データを設定され、ペトリネット上で移動を続ける。トークンがステーションに対応する処理ボックス

に到達すると、AGV S 制御装置に動作完了を通知して次の指令を待つ。たとえば、Fig. 10(a)におけるステーション S 1 から S 2 への走行指令では、動作指示系列(前進、直進、ステーション停止)が S 1 停止中の AGV に与えられる。最初の動作指示データ「前進」を実行した後、その属性値が「直進」に変更された AGV トークンがボックス T 1 に入る。次に、ボックス T 1 の出力側ゲート G 2 のトークンフロー条件は Fig. 10(b)のように記述されており、最初の条件が選択されて、その THEN 部に記述されているユーザ定義関数 MOVE によりトークンの属性値が「ステーション停止」に変更され、ボックス T 2 にトークンが移動する。

6.3 形鋼自動加工システム制御装置の開発¹¹⁾

(1) 形鋼自動加工システムの概要

本システムは、船舶の補強用部材を製作するために、長尺の形鋼素材に対する切断作業と、部材名称あるいは基準線等のマーキング作業を行う。システムの構成は Fig. 11 に示されるように、素材搬入装置と 2 つの加工ラインから成る。各加工ラインにはそれぞれ加工作業を行うロボットが 3 台、素材を移動させる搬送車が 2 台づつ装備されている。素材は素材搬入装置により遊休状態にある加工ラインに搬入された後、以下の作業が繰り返される。すなわち、第一搬送車により、あらかじめ定められた順序に従って、マーキングロボットあるいはプラズマ切断ロボットの作業位置に素材の加工箇所が位置決めされ、それぞれの加工作業が実行される。次に、前記工程を終了した素材は、先行素材の加工完了後に搬送車により本工程へ移動され、先の工程と同様に以下の作業が繰り返される。すなわち、第二搬送車により、ガス切断ロボットの作業位置に素材の切断箇所が位置決めされ、ガス切断作業が実行された後、切断された各部材は、搬出装置により搬送先別に仕分けられ、

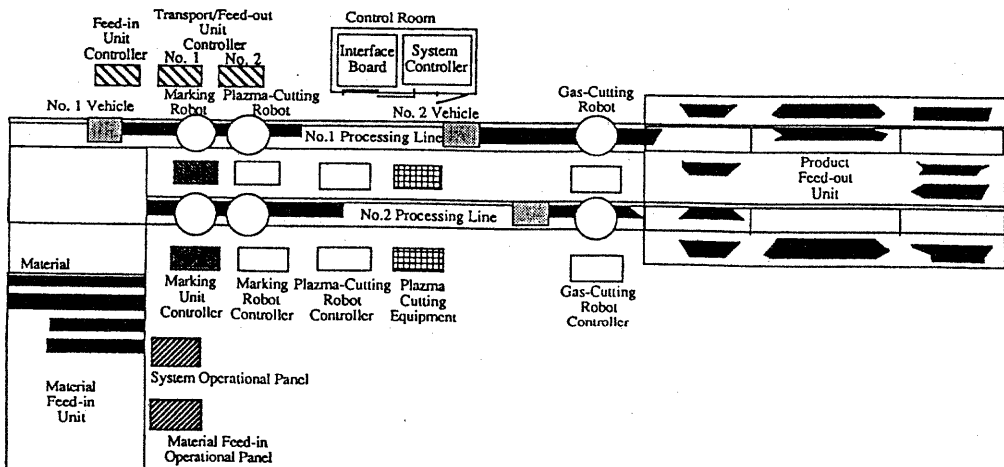


Fig. 11 Layout of automatic cutting system of shaped-steel

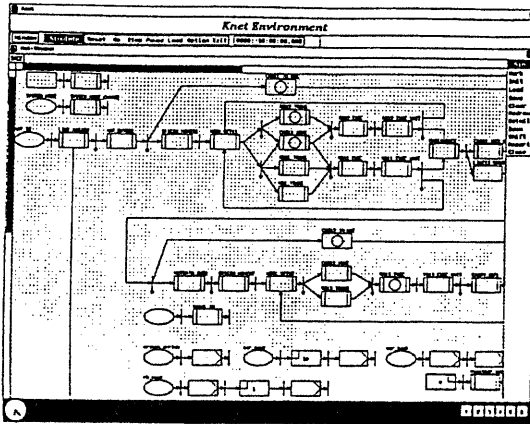


Fig.12 Simulation window for automatic cutting system of shaped-steel

切断の過程で発生したスクラップは一括して回収される。

ロボット、マーキング装置のNCデータはあらかじめオフラインの計算機(加工情報作成計算機とNCデータ作成計算機)で作成され、フロッピディスクによりシステム制御計算機に渡される。システム制御計算機はシステム全体の制御を行い、各ロボット制御装置とマーキング制御装置に対し、通信ラインを介してNCデータを伝送する。素材の搬入、搬送、および搬出の各機器の制御装置にはPCが用いられ、これらは光通信ネットワークにより別のPCを経由してシステム制御計算機に結合されている。

(2) K-NE Tによる開発

上記システム制御計算機のソフトウェア開発にK-NE Tを適用した。本機能の中で、ワーク搬送・加工などシーケンス的な制御はペトリネットにて記述し、複雑な情報処理を含む素材のライン割当機能、高速な処理が必要な周辺機器との通信機能、および画面表示を伴うマンマシン機能は従来のC言語で実現した。シミュレーション時のネット画面をFig.12に示す。本システム制御装置を実現するために作成したネットの規模は、ボックス135、ゲート120、論理I/O159、トークンクラス3、ユーザ定義関数は40であった。この場合は、制御対象機器が多いため、AGVシステムに比較して、ネット規模の割に、論理I/Oの点数が多いのが特徴である。K-NE Tの利用によって、制御プログラムの設計、作成から現地試験まで含めた本システムの制御ソフトウェア開発は、従来の開発方式で要する約1/2の費用で達成できた。

7. あとがき

ペトリネットにて制御仕様を記述する方式を用いたFA制御ソフトウェア開発支援システムK-NE Tを紹介した。ペトリネットは人間の思考とよく一致し、分かりやすい図的プログラミング言語

である。そのため、シミュレータ、モニタ等のデバッグ環境も整え易い。K-NE Tの特長は、ペトリネットの動作フローの理解の容易さと、複雑な情報処理を記述する場合には従来の手続き型言語Cで記述したユーザプログラムを用いることができ、両者の長所を融合した点にある。また、ゲートの発火条件をIF-THENルールで記述することにより、トークンの流れとその制御を分離して、ネットの明瞭さに留意している。さらに、制御仕様の記述から、プログラム作成、デバッグまで一貫して支援することにより、誤りの少ない制御ソフトウェア開発を可能としている。

本稿で述べたごとく、すでにFA制御装置ソフトウェアの開発にK-NE Tを適用し、有効な結果を得ているが、今後より幅広いFA分野へK-NE Tを適用し、より使いやすいFA制御ソフトウェア開発支援システムとしていく予定である。

参考文献

- 1) J.L.Peterson (斎藤訳)：“ペトリネット”，Vol.10, No.16, pp.1937-1965, bit
- 2) 村田，薦田：“ネット指向形シーケンスコンローラの動向”，Vol.27, No.8, pp.680-687, 計測と制御(1988)
- 3) 戸塚：“グラフセ(GRAFSET)の概要と応用”，Vol.29, No.5, pp.43-50, オートメーション(1984)
- 4) 長谷川：“マークフロウグラフ(MFG)によるシーケンス制御の表現”，Vol.29, No.5, pp.32-41, オートメーション(1984)
- 5) 田口，川合，香川：“汎用物流コントロールシステム：ΦNET”，Vol.32, No.1, pp.50-55, オートメーション(1987)
- 6) 村田，薦田，松本：“ペトリネットに基づく高フレキシブルFA制御システム”，Vol.20, No.9, pp.72-79, 計測自動制御学会論文集(1984)
- 7) 村田，薦田，解良：“色付きペトリネットに基づくリアルタイム情報処理用ソフトウェア”，Vol.29, No.12, pp.1129-1140, 情報処理学会論文誌(1988)
- 8) Y.Nagao et al.：“Petri-Net Based Programming System for FMS”，Proc. 16th Annual Conf. of IEEE Industrial Electronics Society, pp.456-461(1990)
- 9) 長尾，山内，占部，海老根，熊谷，児玉：“無人搬送車制御システムへのペトリネットの応用”，pp.183-184, 第34回システム制御情報学会研究発表講演会(1990)
- 10) 長尾ほか：“無人搬送システムシミュレータへのペトリネットの適用”，pp.101-102, 第29回計測自動制御学会学術講演会(1990)
- 11) 長尾ほか：“形鋼自動加工制御システムへのペトリネットの応用”，pp.121-122, 第36回システム制御情報学会研究発表講演会(1992)