

## 解説



## ネット指向パラダイムを求めて

## 7. ペトリネットの FA 制御への応用†

長尾陽一†† 熊谷貞俊†††

## 1. ま え が き

最近, FA (Factory Automation) の進展にとともに, 生産現場においてロボット, NC 機械, 搬送装置, 立体倉庫など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. 従来, この制御ソフトウェアの開発において, PC (Programmable Controller) ではラダー図と呼ばれる低レベルの図的言語を, 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きプログラミング作業を行っている. これらの方法では,

- プログラミング時に誤りが発生しやすい,
  - 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. ペトリネットはシステムの状態をプレースで, 状態遷移をトランジションで, その因果関係を有向枝で表現する二部グラフであり, 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いくつか開発されている. 本稿では, FA 制御へのネット指向アプローチの現状を紹介する

とともに, 特に筆者らが開発した, 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 FA 制御ソフトウェア開発支援システム K-NET<sup>1)</sup> とその適用例について述べる.

## 2. FA 制御へのネット指向アプローチ

将来の FA 制御システムがいわゆる知的生産システム (Intelligent Manufacturing System) の核として実現されるためには, 設計方法論の確立が不可欠なことは明らかである. ペトリネットが一般に自律分散システムの形式的表現や数学的解析に優れた能力を発揮するモデルであることは広く認識されており, 上記の方法論の基礎を与えるものと期待されている.

ネット指向アプローチの最大の利点は, 仕様記述, 動作解析, 性能評価という設計の流れの中にネット理論を基礎とする一貫した形式的方法が導入できる点にある. 具体的な実時間システムのネットモデル表現や制御入出力信号とのインタフェースについては, 次章の応用例にみられるような問題ごとの細かな対応が必要となるが, 一度ネットモデルが確定すれば, さまざまな解析ツールの利用が可能となる. ネットモデル構築のためのグラフィックエディタやシミュレータを含むこれらのネット解析のためのツール (ネットツール) については文献 2) に詳しいが, 表-1 にその機能と特徴を示す.

EWS の高機能化と低廉化にとともに, 特にウィンドウ機能を活用したグラフィックスの進歩が著しいが, 各ネットツールもそれに応じてグラフィック/テキストエディタ, ドキュメント生成, コード生成などが時間差なく柔軟に行えるようになってきた. また, システム仕様に応じてトップダウン/ボトムアップ入力やライブラリに格納さ

† Application of Petri Nets to Control for Factory Automation by Yoichi NAGAO (System Technology Development Center, Kawasaki Heavy Industries, Ltd.) and Sadatoshi KUMAGAI (Computation Center, Osaka University).

†† 川崎重工業(株)システム技術開発センター

††† 大阪大学大型計算機センター

表-1 ペトリネットツールの機能

搭載計算機	Sun, Macintosh, MS-DOS-PC, Apollo, News ほか
メインメモリ	8MB 程度
グラフィック機能	X-window アイコン/メニュードライブ方式
グラフィックエディタ	追加/削除, 移動, 拡大/縮小
シミュレーション	アニメーション, タイムチャート出力 確率ペトリネット, 高レベルネットも可能
解析ツール	可達木生成とデッドロック検証, インバリアント計算
応用機能	プログラム自動生成, デバッグ, 並列 処理の統計解析, 性能評価, 階層化, ドキュメント生成, シーケンス制御 コード, 高レベル意思決定機構など

れたモジュールの再使用が容易に行えるツールが開発されている。実用上の利便を重視して、グラフィック入力中のテキスト記入/参照などのインタフェースが完備されたものが多い。使用計算機はいろいろであるが、主記憶 8MB 程度の容量をもったメニュー/アイコンドライブ方式のもので、ウィンドウ機能の利用が可能な機種に搭載されている。また、グラフィック仕様記述言語 (C 言語が多い) と別に発火制御記述のために PASCAL や ML (高レベル動作記述用関数型言語) を用いているものが多い。特に注目されるのは、Meta Soft 社の Design/CPN に代表される高レベルネット (色付きペトリネット) を用いた大規模分散システム設計のためのツールである。これはペトリネットの特徴である階層モデル化機能のほか、システム各部の同一動作を畳み込んだ (folding) 簡略可達グラフの作成、シミュレーション、インバリアント計算などの解析ツールが搭載されている。今後、モデル化のための簡便なインタフェース、構造的解析も含めた解析ツールの充実とシミュレーションの効率化、ほかのプログラム言語への自動変換機能などの機能拡充が進展すれば、並行システムへのネット指向アプローチの有効性が増大するものと考えられる。これらのツールを用いたネット指向アプローチは、データ通信プロトコル、プロセス制御、FMS (Flexible Manufacturing System), OA やサービス組織の構成、電話サービスの品質制御などにその実用例があり、ほかにも航空管制システム、並列デジタルシグナルプロセッサのバスコントローラ的设计に

用いられ、従来の高レベル設計手法にない威力を発揮した実績が報告されている。

さて、PC やマイクロコンピュータで単独制御される多数の NC 機械やロボット群、無人搬送台車群、ならびに立体自動倉庫よりなる大規模自動化 FMS における同期・排他などの協調制御システムや通信システムを構築し運用する際に、ペトリネット記述を設計・検証・実時間制御・通信に一貫して使用する制御ソフトウェア開発システムが 1980 年初めより欧州、日本で提案され始め、次章で紹介する K-NET 以外にも、(1)S.E.CO.I.A.<sup>3)</sup>, (2)C-NET<sup>4)</sup>, (3)FOR-ME-TOO<sup>5)</sup> などがよく知られている。(1)はハンドシェイク通信プロトコルのネットモデルを含む大規模 FMS 制御システムを実現するためにフランスの LAAS で行われたネット指向アプローチによる最初のプロジェクトである。制御フローとデータフローを含む計算機資源を二つのネットモデルで表現し、それらの同期関係を検証するための方法が提案されている。(2)は日立システム開発研究所において開発されたネット指向アプローチによる世界初の商用 FA コントローラである。実時間制御を可能とするために、トランジション発火の高速化に工夫がなされている。FMS 構築のためのネット指向方法論として現在も進行中のプロジェクトでもある(3)が興味深いので、以下に簡単に紹介する。

FOR-ME-TOO (FORMalism, MMethods and TOOLS for specification and modular programming of large real time systems) は、EC の共同プロジェクト ESPRIT の一環としてフランスの SYSECA 社が開発中の大規模実時間システムの仕様設計・検証のためのソフトウェアシステムで、その概念は図-1 で表せる。FOR-ME-TOO

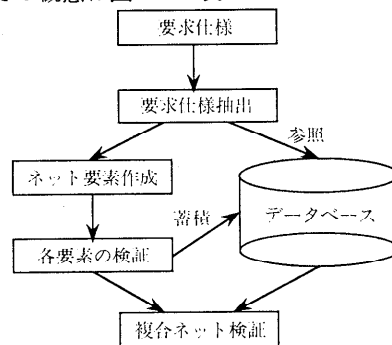


図-1 FOR-ME-TOO の構成

は、ペトリネット記述による逐次詳細化を用いたトップダウン設計手法が核となっており、Andre のネット等価性<sup>6)</sup>に関する理論的成果が取り入れられている。たとえば、B-Condition (逐次詳細化における Well-Behavedness に相当する) に基づくネット代置とネット複合化手法およびインバリエント合成による検証手法に対する自動化ツールの整備がシステム開発の主眼である。検証済の要素ネットは、その入出力等価性を考慮して以後の設計に再使用可能な形でデータベースに蓄積され、参照される。また、要素ネットの記述には色付きペトリネットが用いられている。各要素ネットに対する解析事項としては、発火系列チェック、活性解析、インバリエント、有界性などのシステム構造に関する検証があり、その解析プログラムがツールとして開発されつつある。大規模システムの要素ネット複合、代置による形式的設計手法の確立を目指したソフトウェアシステムである。

以上の例から窺えるように、ネット指向アプローチの有効性は FA システムに限らず、並行プロセスを含む複雑な非同期システムのモデル化能力と形式的な解析能力にある。また、ペトリネットは自然な階層表現に優れており、オブジェクト指向に特徴的なモジュールの部品化や、ほかの記述言語との階層的な融合が容易に行えることも利点のひとつにあげられる。将来、ほかの知識情報処理、たとえば AI 技術との融合を目指して多方面での研究が期待されているのが現状である。

次章以降では、特に FA 制御向けに開発されたネットツールである K-NET の詳細な説明と応用について紹介し、ネットツールの有効性を示す。

### 3. コンカレント制御プログラミングシステム： K-NET

#### 3.1 システムの構成

K-NET は、制御仕様をペトリネットで記述することにより制御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-NET 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制御装置で実行できる形式、すなわち C 言語記述のソースプログラム

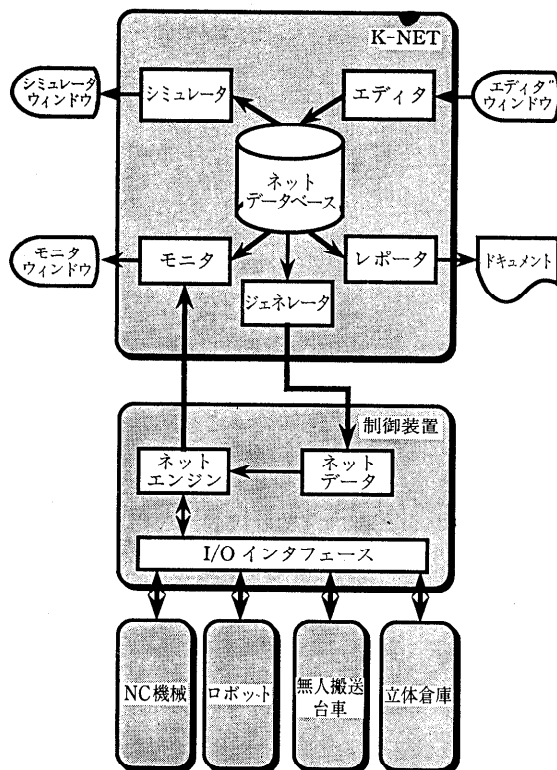


図-2 K-NET のシステム構成

に変換するためのジェネレータ、及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、制御装置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-NET のシステム構成を図-2 に示す。

#### 3.2 制御仕様の記述モデル

制御仕様を表現するために K-NET において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(プレース)、処理の流れを制御するゲート(トランジション)、両者間の関係を表す有向枝、および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 (図-3)。

##### 3.2.1 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物体の型、形状などの属性情報をもつことができ、“クラス”と呼ばれる属性の構造体によって分類される。

##### 3.2.2 ボックス

(1) ボックスの状態

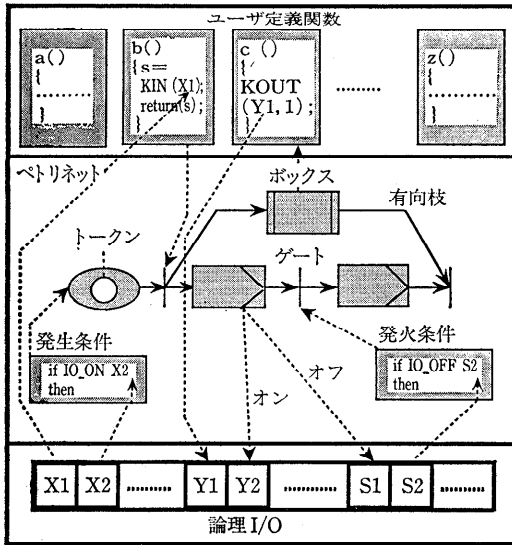


図-3 K-NET におけるネット構成要素

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”と呼ぶ。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、すなわち容量と同じ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、ゲートが発火すればトークンの移動が可能な状態である。

(2) ボックスの種類

ボックスはその機能によって、表-2 のような種類を用意している。

(a) 状態ボックス

容量1の単純なボックスである。機器の操作・運転に関する許可、禁止などの制約条件を表現する際に、機器の稼働状態あるいは遊休状態に状態ボックスを対応させる。

(b) 容量ボックス

1以上の容量をもつボックスであり、バッファあるいはキューを表現する。

(c) 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移る。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。

(d) 処理ボックス

ユーザが定義した処理関数の呼出しを含む

表-2 ボックスの種類

名称	記号	容量	指定項目
状態ボックス		1	なし
容量ボックス		n	容量 n, トークン取出し順序
時間ボックス		1	時間 t
計数ボックス		m	計数值 m
演算ボックス		1	演算子, 論理 I/O
信号ボックス		1	論理 I/O, 出力値
処理ボックス		1	処理関数, 完了条件, 後処理関数, 監視時間
発生ボックス		n	クラス, 容量 n, 発生条件, 処理関数
マクロボックス		—	なし
入口ボックス		1	なし
出口ボックス		1	なし

一連の動作を行う。すなわち、図-4 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。その後、ゲート B の処理完了条件の成立を待ち、同時にタイムアウト監視のためのタイマ C を起動する。完了条件が成立するか、あるいはタイムアウト発生時には、後処理関数 D が呼び出されるが、引き数によってその起動要因を知ることができる。後処理では、完了条件をチェックし、タイマ C をキャンセルしてボックスを励起状態に移らせるか、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

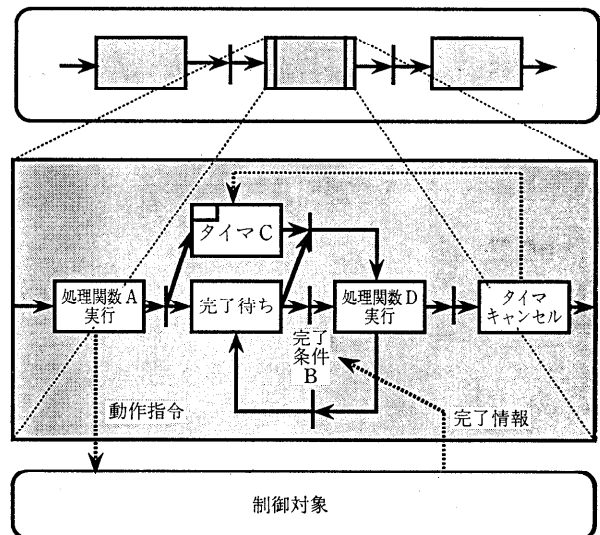


図-4 処理ボックスの機能

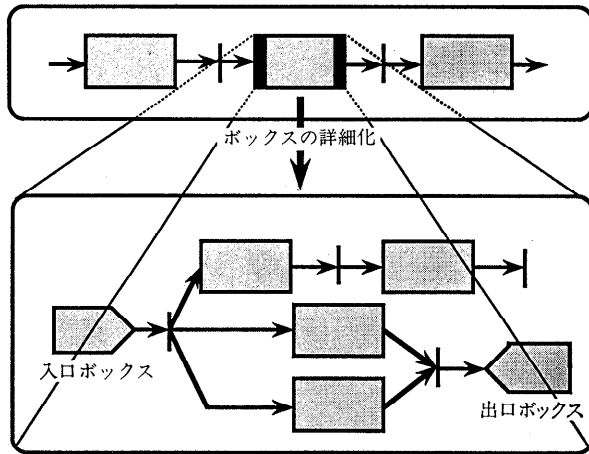


図-5 マクロボックスとその詳細化ネット

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。

(e) 発生ボックス

発生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。発生したトークンの属性値はクラスで指定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属性値の変更を行うことができる。

(f) 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。図-5に示すように、マクロボックスにトークンが入ると、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良くする。

(g) その他のボックス

上述のボックスのほかに、シーケンス制御におけるカウンタを表現する計数ボックス、論理 I/O の値に関する算術演算を実行する演算ボックス、論理 I/O に信号を出力する信号ボックスがある。

(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、その前置ボックスの

すべてが励起状態にあり、後置ボックスのすべてが受容状態にある場合に発火し、すべての前置ボックスからトークンを一つ取り出し、すべての後置ボックスにトークンを一つ置く。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え、図-6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。

(a) 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 の値などの論理式で表現された発火条件を 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。

(b) トークンのフロー制御

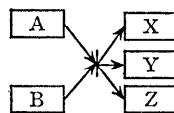
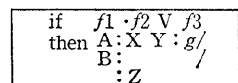
上記発火条件を満足する IF 部に対応した THEN 部の内容に従って、トークンの移動および属性継承を含むフロー制御が実行される。このとき、前置ボックスごとにユーザ定義関数の呼出しを指定することができる。

(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、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 やトークン属性値を変更するためのシステム関数を用いることができる。

(5) 論理 I/O

I/O 設計は、論理設計と物理設計の二段階で行われる。論理 I/O は、時間ボックスの時間、発生ボックスの発生条件、処理ボックスの完了条件、ゲートの発火条件などに使用されるが、外部プロセスへの入出力以外に、ネット内部における I/O としても使用できる。論理 I/O を物理 I/O に割



(a) ペトリネット表現

(b) 記述形式  
 発火条件  $(f1 \cdot f2) \vee f3$   
 トークンフロー条件  
 ボックスA→ボックスX, Y  
 関数gの実行  
 ボックスB→消滅  
 発生 →ボックスZ

図-6 ゲートの仕様記述

り当てることにより、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 の適用例

無人搬送システム制御ソフトウェアの開発に K-NET を適用した例<sup>7)</sup>を紹介する。

##### (1) 無人搬送制御システムの概要

対象とする無人搬送システムは、図-7 に示す走行経路と、立体倉庫ならびに 48 カ所のステーションを有し、最大 5 台の AGV (Automated Guided Vehicle) 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無人搬送システム制御装置 (以下、AGVS 制御装置と呼ぶ) と情報の授受を行う。AGVS 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合せが行われ、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”及び“ステーション停止”などの動作指示の系列を

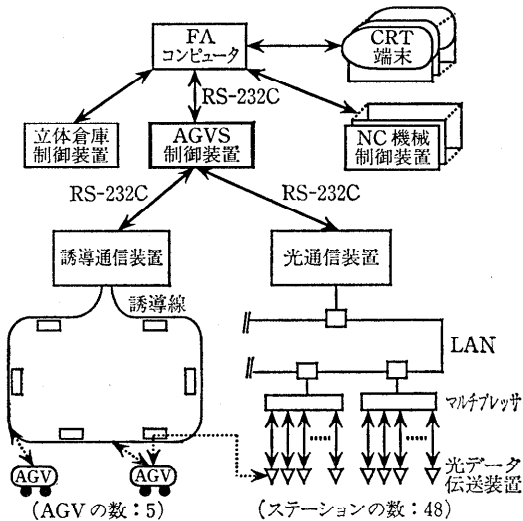
通知され、走行経路の分岐点ならびにステーションにおいて順次この指示を実行しながら自律的に移動する。なお、各 AGV の状態情報はステーション間を走行中にも誘導通信装置を経由して一定時間間隔で AGVS 制御装置へ通知される。

AGVS 制御装置は生産システム全体を管理する 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、AGVS 制御装置には、

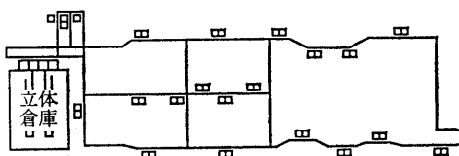
- ① 搬送要求に対し、最適な AGV を割り当てる台車割当機能
- ② 割り当てられた搬送要求に従って、AGV に対して荷積みステーションへの走行、荷積み、荷卸しステーションへの走行、荷卸しを順次実行させる荷物搬送機能
- ③ 搬送の完了した時点で、実行すべき搬送指令が存在しない場合、AGV を待機ステーションへ移動させる待機走行機能
- ④ バッテリ低電圧となった AGV を搬送終了後に充電ステーションへ移動させる充電走行機能が設けられている。このほか、異常時の対応、マンマシン、および周辺機器との通信機能がある。

##### (2) K-NET による開発

上記 AGVS 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ペ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 SUN 4 (図-8) を、現場では東芝のラップトップパソコン J 3100 を用い、また実行マシンとしては NEC の FA パソコン FC 9801 を用いた。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を含めた本無人搬送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費用の約 1/3 で達成できた。これは、開発段階に制御仕様をネットにより容易に表現できたこと、その妥当性をシミュレータを用いて十分に検証できたこと、ジェネレータを用いてネットから制御プログラムを自動生成できたこと、さらに現場でのテスト段階ではモニタ



(a) システム構成図



(b) 路線図およびステーション配置

図-7 AGV 制御システム

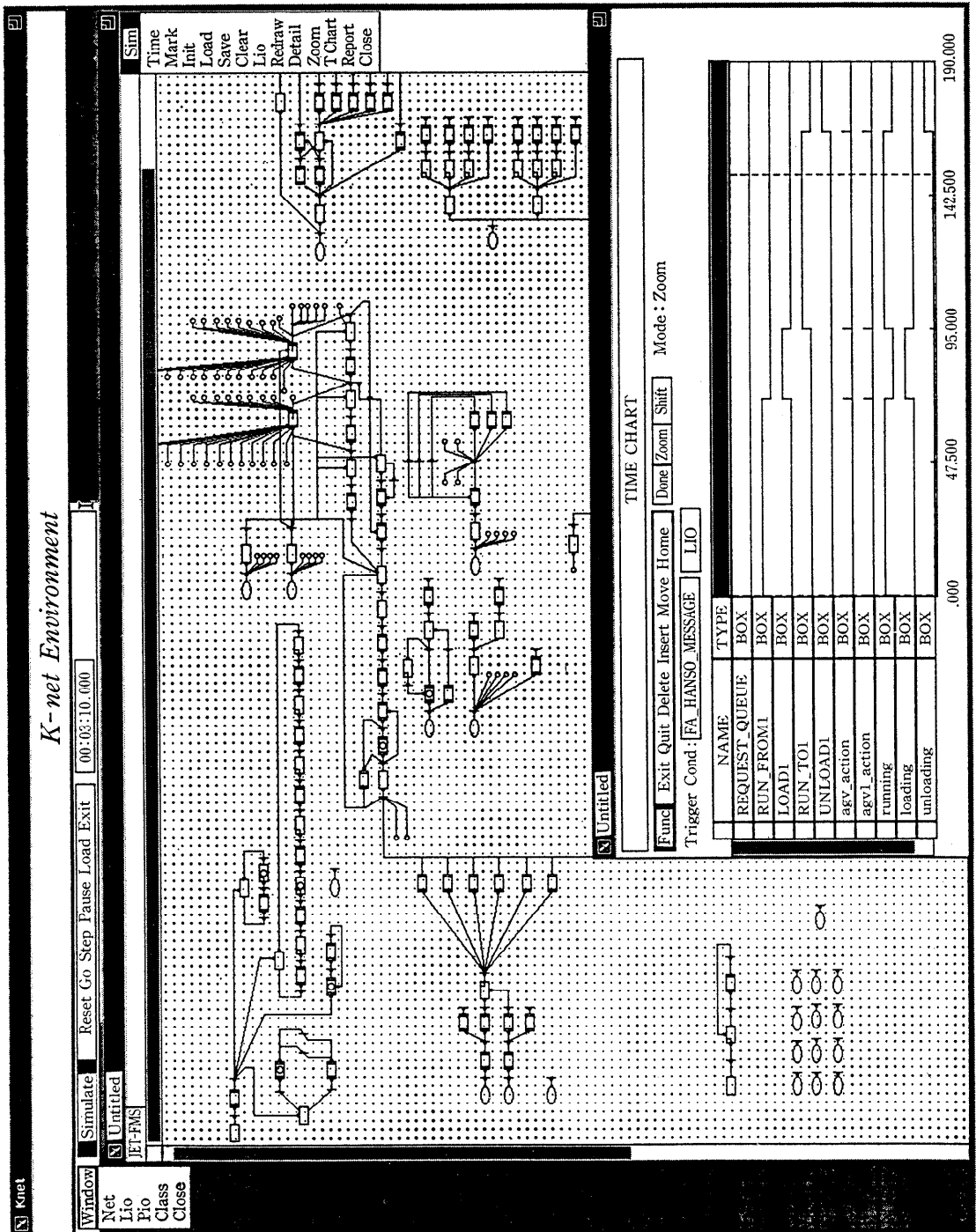


図-8 シミュレーション画面例

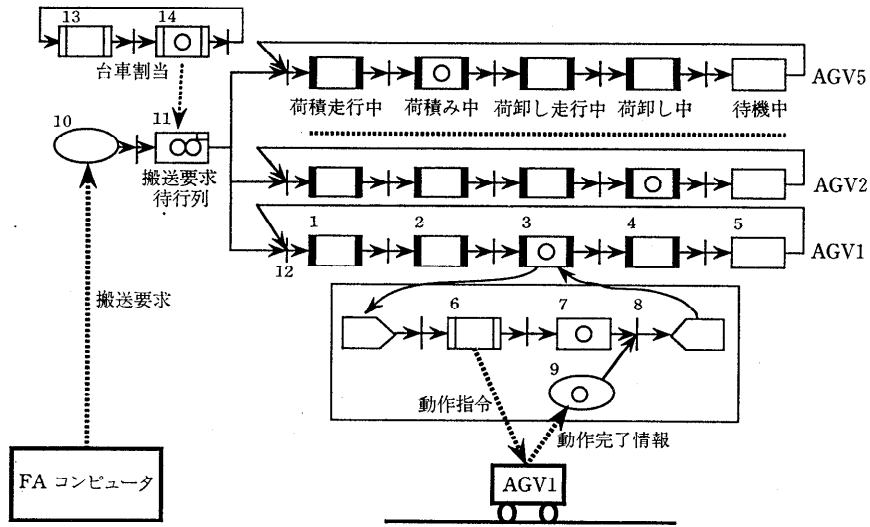


図-9 AGV 制御のネット構造

を用いて制御プログラムのリアルタイム動作を視覚的に確認できた結果である。

(3) ペトリネットモデル

AGV の運行制御に対するネットの概念図を図-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 ごとにボックス 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 AGV 1 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/O 63、トークンクラス 5、ユーザ定義関数は 82 であった。

5. あとがき

FA 制御へのネット指向アプローチの最近の動向を紹介した。特に、FA 制御ソフトウェアの開発支援ツールである K-NET について詳述した。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、モニタなどのデバッグ環境も整えやすい。K-NET では、ペトリネットの動作フローの理解が容易であり、また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る。さらに、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、ネットを明瞭に表現できる。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、高信頼の制御ソフトウェアの開発が可能である。

参考文献

- 1) Nagao, Y. et al.: Petri Net Based Programming System for FMS, IEICE Trans. Fundamentals, Vol. E75-A, No. 10, pp. 1326-1334 (Oct. 1992).
- 2) Feldbrugge, F. and Jensen, K.: Petri Net Tool Overview 1986, Lecture Notes in Computer Sciences 255, Springer-Verlag, pp. 20-61 (1986).
- 3) Valette, R. et al.: Putting Petri Nets to Work for Controlling Flexible Manufacturing Sys-



tems, Proc. IEEE Int'l. Symp. on Circuits and Systems, Kyoto, Japan, pp. 929-932 (1985).

- 4) Komoda, N. et al.: An Autonomous, Decentralized Control System for Factory Automation, IEEE Computer, pp. 73-83 (1984).
- 5) Sibille, E. et al.: Nets as Formalisms, Methods and Tools for Specification of Large Real Time Systems, Proc. of 7th European Workshop on Application and Theory of Petri Nets, Oxford (1986).
- 6) Andre, C.: Use of the Behaviour Equivalence in Place-Transition Net Analysis, Informatik-Fachberichte 52, Springer-Verlag, pp. 241-250 (1982).
- 7) 長尾他: 無人搬送制御システムへのペトリネットの応用, システム制御情報学会論文誌, Vol. 5, No. 11, pp. 31-41 (1992).

(平成5年3月1日受付)



長尾 陽一

1949年生。1972年大阪大学工学部電子工学科卒業。1974年同大学院修士課程修了。同年、川崎重工業(株)入社。ロボットシステム、無人搬送システムなどのFAシステムの研究開発に従事。著書「ペトリネットとその応用」(計測自動制御学会, 共著)。現在、同社技術総括本部システム技術開発センター応用技術部FAシステム課長。電子情報通信学会, システム制御情報学会各会員。



熊谷 貞俊

昭和43年東大工学部電気工学科卒。昭和50年大阪大学大学院修了。工学博士。同年同大学工学部電子工学科助手。昭和54年~56年カリフォルニア大学バークレー校客員研究員。昭和63年大阪大学助教授を経て、現在大型計算機センター教授。非線形回路, ネット理論の研究に従事。電子情報通信学会, IEEEなどの会員ならびに編集委員。

