

解説

FA 用情報処理技術



知識情報処理技術の FA への応用†

都島 功†† 薦田 憲久†† 田代 勤††

1. はじめに

FA が過去の自動化と異なる点は、変更に対する柔軟性の追求にある。これに対し、ハード面では、ロボット、NC 機械などが開発され、実用に供されている。しかし、ソフト面では、柔軟な FA システムを計画するためのシミュレータや機器群を制御するためのプログラムは柔軟性に欠ける。

計画段階では、各種の例外処理にも柔軟に対応できるシステム構成、運用方法を決定するために、実際レベルで多ケースのシミュレーション評価が必要である。しかし、従来使用されている GPSS (General Purpose Simulation System) などは、対象記述言語がマクロすぎて詳細なモデル化が困難であること、モデル作成、変更に多大の工数を要するという問題点を持つ。

実稼働段階では、十分に事前評価されたシステムでも、製品のモデルチェンジなどにより機器制御プログラムの変更は生じる。従来、制御プログラムは FORTRAN レベルの汎用言語やラダーダイアグラムを用いて手続き的にプログラミングされる。この方法では、制御内容(知識)がプログラムの中に埋め込まれ、理解しにくいこと、ある部分の変更が他に及ぶという問題点を持つ。

以上の問題点は、制御内容を簡単かつ確実に記述できる方法および記述されたモデルに対する、柔軟な処理系がなかったことによる。この解決のために、人間が FA システムの操作、運転を行う場合の知識をそのまま表現できるようなシミュレータ、制御方式の研究が進められている。それらは、知識を蓄える知識ベースと推論・問題解決機構とを分離しており¹⁾、知

識の入れ替えだけで各種の FA 制御システムができる。すなわち、上記のアプローチにより、システムの変化に対する柔軟な対応、同じ知識のさまざまな利用などが実現される。

上述したアプローチを FA へ応用しようとする試みにはマクロに2つの方向がある。1つは、予め、動作順序が決められたもとの、同期・排他的制御を行う“順序組合せ型制御”(シーケンス制御)を対象とする。順序的知識を、Petri-Net²⁾(PN と略す)を FA 用に拡張したモデルで記述し、そのモデルに基づいてシミュレーション、制御を行うものである。他の1つは、各種条件の組合せで制御指令が決まる“条件組合せ型制御”を対象とする。条件と結論の対からなる制御規則の知識を、知識工学における Production System (PS と略す)^{3),4)}の“IF (条件), THEN (結論)”ルールで表現し、状況に合わせて制御指令を決定するシミュレータ制御方式である。

以下では、PN, PS を FA システム向きのシミュレーション、実時間制御へ応用している研究概要について述べる。

2. FA 応用の知識情報処理技術の動向

典型的な FA システムでは、ロボットと周辺機器(圧入機など)からなる作業ステーションが複数設置され、無人台車が物を搬送する。ステーション内の機器群に対しては順序組合せ型制御が実施される。各ステーションへの作業割付制御、台車搬送制御などのショップ全体の制御に対しては、動作順序を予め決めておくことが困難であり、条件組合せ型制御が実施される。

はじめに、順序組合せ型制御への PN の応用について述べる(図-1)。PN は、OS (Operating System)、コンパイラの非同期、並列処理を表現し、解析するために提案されたグラフ表現モデルである。その主な特徴は、システムの表現が図形で行われるため、システ

† Knowledge Information Processing Technique in Factory Automation by Isao TSUSHIMA, Norihisa KOMODA and Tsutomu TASHIRO (Systems Development Laboratory, Hitachi, Ltd.).

†† (株)日立製作所システム開発研究所

ム全体を容易に理解できること、グラフ理論を駆使してシステムの動作を解析できることなどである。この PN を順序組合せ型制御向けに拡張したモデルで表現し、そのモデルに基づいてシミュレーション、制御を行う研究がされている。MFG (Mark Flow Graph…東工大、長谷川、1976)⁵⁾ は PN をシーケンス制御向けの表現モデルに拡張したものである。フランスでは、国営の生産自動化推進協議会 (ADEPA) が同様に PN を拡張した GRAFCET を 1979 年に発表し、シーケンス制御内容の仕様記述法として使用をすすめている。これらの表現モデルは当初、シーケンスの記述、設計支援が主であった。その後、シミュレーションシステムへ展開された。さらに最近では、実時間制御への応用をはかるための研究、開発⁷⁾⁻¹⁰⁾が進められている。制御応用のためには、プロセス入出力処理機能が必要になるとともに、高速化、高信頼化、高保が守化図られなければならない。この方向の研究として、MFG 利用の制御システム (東工大、長谷川、1982)⁷⁾、Petri-Net-Based Sequencer (モンテリオール大、Chocron、1980)⁸⁾、SCR (Station Controller…藤田、村田、1983)^{9)、10)} が発表され、SCR は実用化段階にある¹¹⁾。

つぎに、条件組合せ型制御への PS の応用について述べる (図-1)。PS は、制御規則などの断片的な知識が “IF (条件), THEN (結論)” 型のルールで与えられると、それらを組合せ、状況に応じた結論を出す。PS では、ルールはすべて同質であり、互いに直接の干渉はなく、知識ベースを通じてのみ干渉する。その

ため、ルールのモジュラリティが高く、ルールの追加・削除・変更が容易である。この特徴と条件組合せ型制御における制御規則が IF-THEN ルールそのものであることに着目し、制御規則の記述に PS を応用したシミュレータ CASE-PLANET¹²⁾ (Computer Aided System Engineering-Plan Network…田代、1982) が発表された。PS の応用に加えて、設備属性の記述に Frame Model を応用したシミュレータに、KBS (Knowledge Based Simulation…Reddy、1982)^{13)、14)}、ROSS¹⁵⁾ (Rule Oriented Simulation System…McArther、1981) がある。さらに最近では、実時間制御への PS の応用も研究されている。実時間制御応用のためには、高速化、高信頼化などが重要な課題であるが、この方向の研究として、Rule Based Language を用いた制御システム (カーネギーメロン大、Bourne、1982)¹⁶⁾、詳細を後述する SCD (Station Coordinator…田代、1983)¹⁷⁾⁻²⁰⁾があり、SCD は実用化段階にある¹¹⁾。

3. シミュレーションへの応用

シーケンス制御に対しては、次章の実時間制御への応用で詳しく述べるので省略する。本章では、条件組合せ型制御向きシミュレータの代表として KBS を解説する。

KBS はジョブショップ型の生産管理システムを対象としたシミュレーションシステムである。シミュレートすべき対象の構成要素 (例えば機械、待ちキュー) の属性、動作規則などを個別に記述し、それら

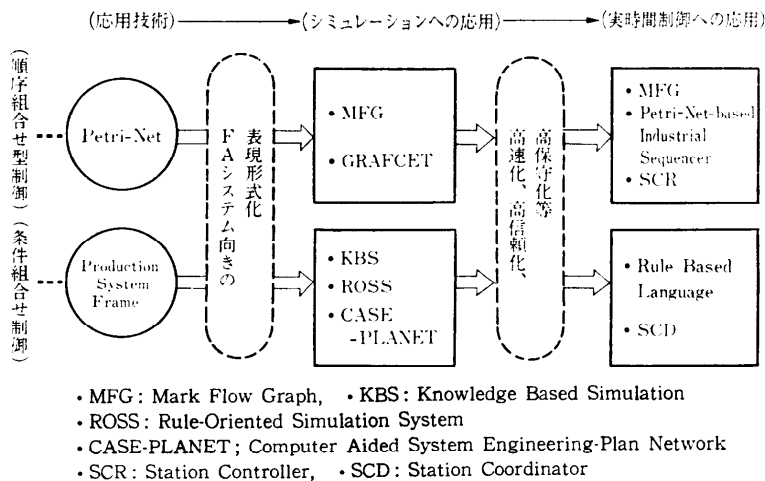


図-1 FA 応用の知識情報処理技術の動向

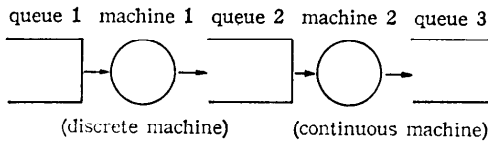


図-2 2-機械システム

を KBS で組み合わせることによりシミュレーションを実行していく。Frame model を利用しているため、属性などを階層的に定義でき、しかも上位要素の性質が下位要素に「伝承」という性質により、あらゆる要素属性の記述から詳細な記述までを一元的に管理することができる。

図-2 に示す 2-機械システムを例に KBS によるシミュレーションモデルの記述法を示す。discrete machine は 1 度に 1 つの対象物を処理する装置である。continuous machine は複数の対象物を連続的に処理する装置である。queue 1, 2 は first-in first-out タイプのキューである。machine 1, 2, queue 1 の定義を図-3(a)に示す。machine 1 には上位概念として discrete machine (図-3(b)) が定義されているので、対象物を machine 1 に load/unload するルールとして、discrete machine の定義中の discrete-load-rule と discrete-unload-rule が使用される。例えば、前者

のルールの定義を図-3(c)に示す。ルールは、“IF (条件), THEN (動作)”の型で記述される。同様に、queue 1 の定義中の arrival rule 1 も図-3(d)のように定義される。

シミュレーションの実行は、図-3(e)の prime-rule の実行で開始される。prime-rule の条件は常に真であるため、THEN 部の動作、すなわち order file からオーダーが読み出され、図-3(f)に示すような、各オーダーに対するイベントを書きならべた calendar が生成される。この例では、各オーダーに対して、queue 1 への arrival イベントを生成する。

calendar 中の event-list の先頭である event 1 が選ばれる。図-3(g)に event 1 の内容、すなわち order 1 が時刻 0 に queue 1 へ arrives することを示す。KBS は arrives を解釈するために、queue 1 (図-3(a))の arrives 属性の内容をみると、arrival-rule 1 である。その定義 (図-3(d))から判るように、もし、機械が空きならば、イベント(EVENT-TIME: 0.0; EVENT-NAME: LOAD; EVENT-FOCUS: machine 1; EVENT-PARAM; order 1) が生成され、calendar に登録される。このイベントは、machine1 に関する load 属性を評価することによって実行される。machine 1 は discrete machine タイプの 1 つで

```

{{machine 1
  {INSTANCE discrete-machine
    INPUT-Q: queue 1
    OUTPUT-Q: queue 2
    SERVICE-TIME: 0.5}}
  {{machine 2
    {INSTANCE continuous-machine
      INPUT-Q: queue 2
      OUTPUT-Q: queue 3
      SERVICE-TIME: (FUNCTION stime)
      CAPACITY: 5}}
  {{queue 1
    {INSTANCE fifo-queue
      SOURCE: nil
      DESTINATION: machine 1
      ARRIVES: arrival-rule 1}}

(a) 構成要素の記述例

{{discrete-machine
  {{IS-A machine
    LOAD: discrete-load-rule
    UNLOAD: discrete-unload-rule}}
  {{DISCRETE-LOAD-RULE
    {INSTANCE rule
      IF: state=free & input-source=not empty
      THEN: ask input-rule & update contents slot &
            change state to busy & execute statistics-rule}}
  (b) discrete machine の定義
  (c) discrete-load-rule の定義

{{arrival-rule
  {INSTANCE rule 1
    IF: (status of machine=free)
    THEN: (schedule a load for the machine)}}
  {{prime-rule
    {INSTANCE rule
      IF: (t)
      THEN: (Function read-orders order-file)}}
  (d) arrival-rule 1 の定義
  (e) prime-rule の定義

{{calendar
  {INSTANCE priority-queue
    EVENT-LIST: (event 1, event 2, event 3,...)}}
  {{event 1
    {INSTANCE event-notice
      EVENT-TIME: 0.0
      EVENT-NAME: arrives
      EVENT-FOCUS: queue 1
      EVENT-PARAMS: order 1}}
  (f) prime-rule 実行後の Calendar の内容
  (g) event 1 の内容
  
```

図-3 マニュファクチャリング・セルの KBS による記述例

あるため、前述したように、そのタイプの machine で定義された load-rule (図-3(c))を伝承する。

以上のように、KBS は断片的な知識である属性、ルールを組み合わせ、イベントを発生させながらシミュレーションを実行する。

4. 実時間制御への応用

本章では、順序組合せ型制御に対して SCR を、条件組合せ型制御に対して Rule Based Language, SCD を述べる。

4.1 順序組合せ型制御への応用

SCR は PN を PI/O の条件、繰返し作業、ファイルのアクセスなどを扱えるように拡張した C-net (Control-net) モデルをシーケンスの記述方式として採用し、C-net で記述されたプログラムに基づき機器の制御を高速に行える処理機構を実現したものである。

C-net は動作及び状態を表わすボックス、ボックスの動作が実行中あるいは状態が占有されていることを示すトークン、ボックス間のトークンの移動を制御するゲート及びボックスとゲートの接続関係を示すアークの4つの基本構成要素からなる。図-4に組立ロボットとネジ締めロボットから構成されるステーションの動作を C-net で記述した例を示す。C-net ではト

ークンは番号付の丸印で示され、後述の動作規則にしたがってボックスを移動する。ボックスにトークンが入ると、ボックスで指定された動作を行わせる。トークンのゲート通過規則は、(1) 入力側のすべてのボックスにトークンが存在する (同期)、(2) 出力側のすべてのボックスにトークンが存在しない (排他)、(3) ゲートに定義されたゲート条件が真である、の3条件が満足されるとき、ゲートは通過可能であり、入力側のすべてのボックスのトークンが消滅し、出力側のすべてのボックスに各々、1つのトークンが発生する。

SCR の全体構成を図-5 に示す。C-net で記述された動作モデルは、エディタ/モニタ端末より C-net 入出力部を通じ、ボックス情報テーブル、ゲート情報テーブルに分けて格納される。前者のテーブルには動作定義データが、後者のテーブルにはボックスの接続関係を示す C-net の構造データが格納される。トークン制御部は、ゲート情報テーブルを調べ、先に述べたゲートの通過条件を満たしているゲートを抽出し、トークンを動かす。新たにトークンが入ってきたボックスについて、プロセスインタフェース部が制御メッセージ、メッセージ送信先をボックス情報テーブルより読み出し、設備群へ制御指令を送出する。ステーションからの各種信号は、プロセスインタフェース部に

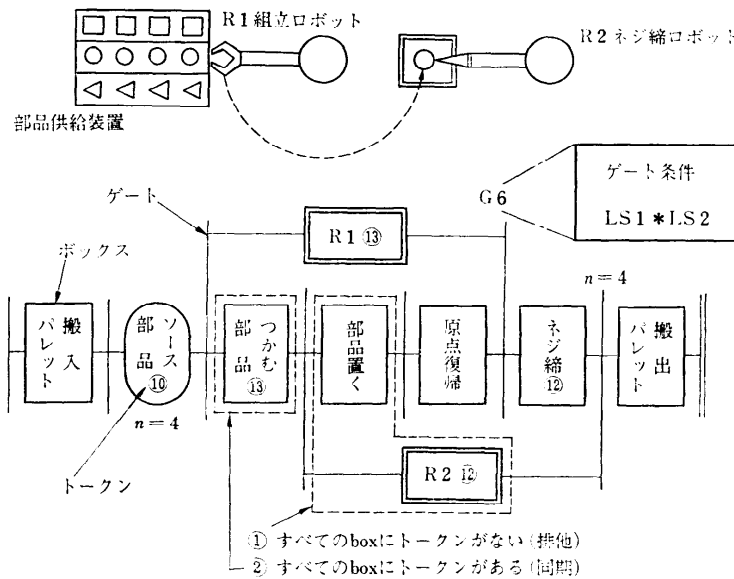


図-4 C-net による記述例

より解析され、ボックス情報テーブル、ゲート情報テーブルのステータスが変更される。以上の制御部をFAコントローラとして実現するため、16ビットマ

アイコンを、エディタ/モニタ端末には16ビットパソコンを用いている。

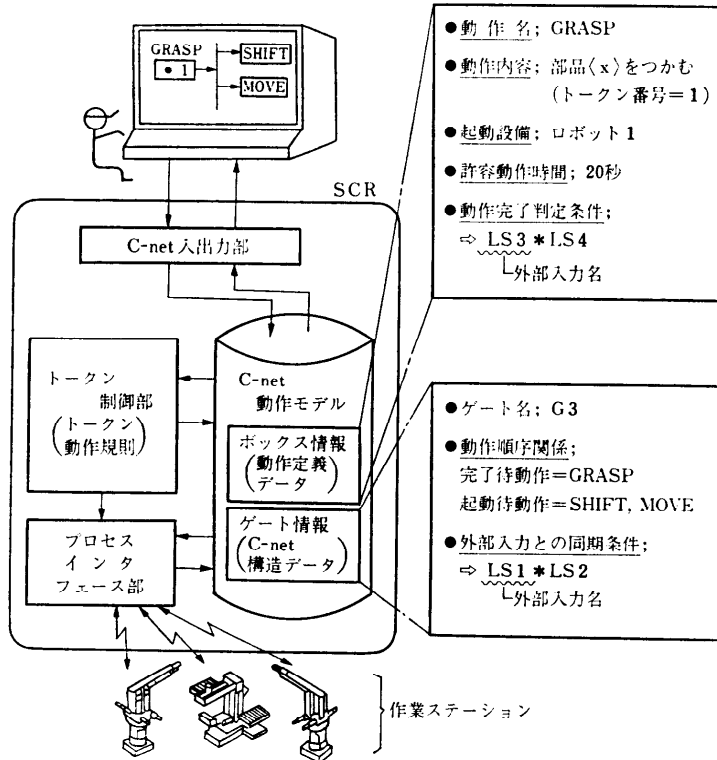


図-5 SCR の構成

- A (AND (Acquired Billet InRack) (NOT (Located Billet InRack))) → {(Locate Billet InRack)}
- B (AND (At Robot LoadStage) (Located Billet InRack) (NOT (Gripped Billet))) → {(Pass Rack BilletLocation ToRobot) (Acquire Billet InRack)}
- C (AND (At Robot LoadStage) (Gripped Billet)) → {(Move Robot ToFurnaceStage)}
- D (AND (Gripped Billet) (Cold Billet) (At Robot FurnaceStage) (NOT (Moving FurnaceHearth))) → {(Open FurnaceDoor) (Pass Furnace BilletLocation ToRobot) (Place Billet InFurnace) (Close FurnaceDoor)}
- E (AND (At Robot FurnaceStage) (NOT (Gripped Billet)) (NOT (Full Furnace))) → {(Move Robot ToLoadStage)}

図-6 Rule Based Language による記述例

4.2 条件組合せ型制御への応用

はじめに、Rule Based Language を説明する。これは制御規則を (antecedent)→{consequent} のように表現し、左辺には判定すべき条件を、右辺にはその条件が満たされた場合に出力すべき制御指令を記述する。図-6 の記述例からわかるように、条件、制御指令を文字列で与えることができる。そのため、記述がしやすい、制御の内容が理解しやすいというメリットがあり、制御プログラムの変更が容易になる。なお、Rule Based Language においては、各制御指令が1つのルールによって決定される。すなわち、デシジョンテーブルと同等の方式を採用している。適用例としては、図-7 に示すロボット、作業台、炉などからなるマニファクチュアリングセルの制御である。上述

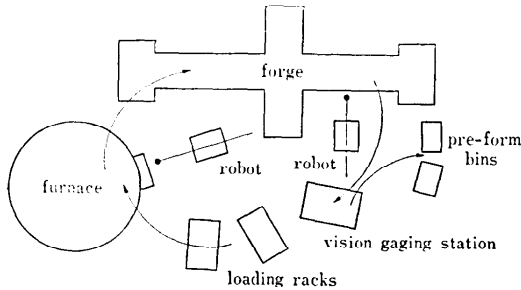


図-7 マニファクチュアリング・セル

の図-6 に示したルールは、ロボットが cold billet を loading rack から furnace まで動かすためのルールである。

つぎに、SCD について説明する。

制御規則を日本語により “IF (条件), THEN (中間結論あるいは最終結論 (制御指令に対応))” のルールで与えておけば、ルールを組み合わせ、機器状況に応じた制御指令を自動的に出力する。ルールは、図-8 に示す3種類の型式が用意されている。条件、結論を記述する文字列は内容を表わす固定部と、設備名、番号などを示すパラメータ部とからなる。IF 部は and 条件として扱われる。型式1ではパラメータ部に定数を、型式2では変数を記述する。型式3はユーザー手続きを推論過程に組み込むためのものである。

SCD の構成を図-9 に示す。端末よりルールがルールテーブルに入力される。IF 部パターンマッチング部は、IF 部の条件がワークテーブル内に取り込まれたシステム現況と一致するかを判定する。一致の場合、結論書き込み部は、該当するルールの結論が中間結論であれば、それをワークテーブルに追加する。以上の処理が繰り返され、最終結論 (制御指令に対応) に至ると、制御指令が設備に送られる。

図-10 は、制御指令が得られるまでのルールの適用過程を示す例である。

SCD は、以上の説明からわかるように PS を利用している。ただし、従来の PS はその適用対象が

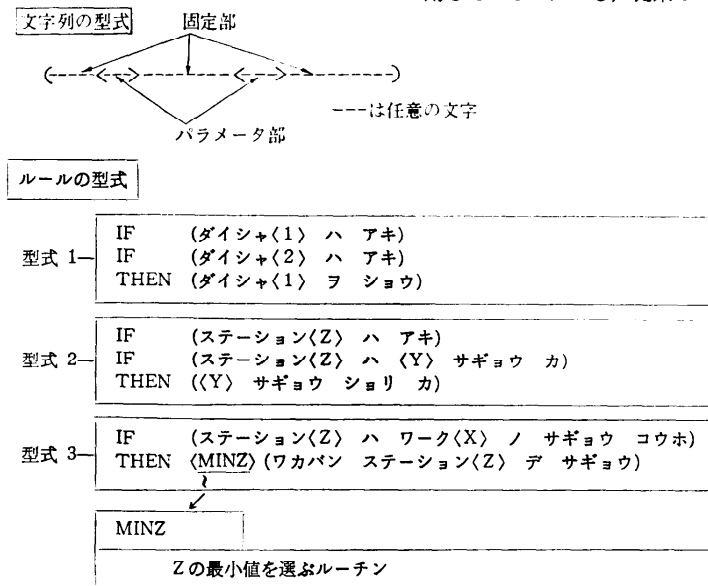


図-8 文字列、ルールの型式

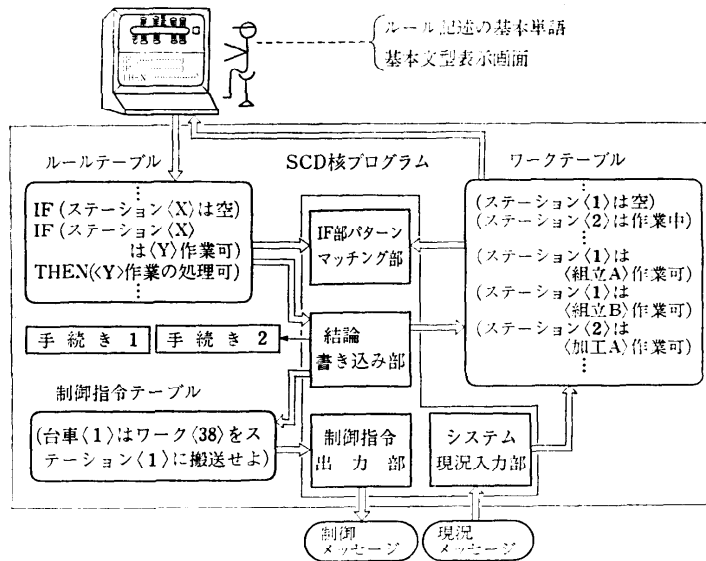


図-9 SCD の構成

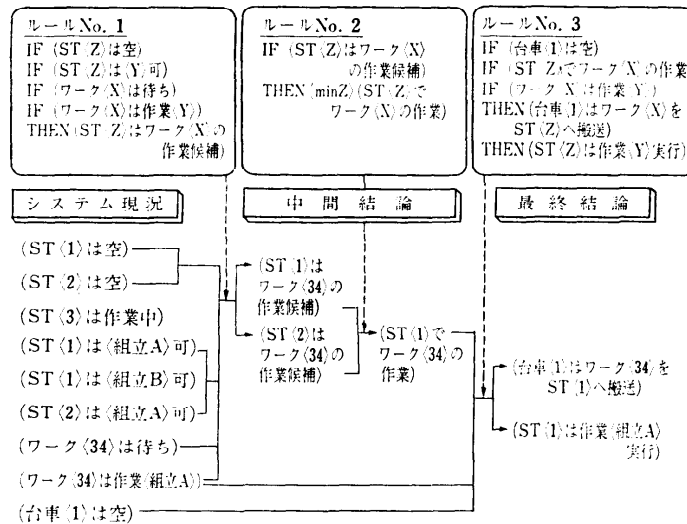


図-10 ルールの適用過程

コンサルテーションであるため、実時間制御に耐えうる処理速度を実現していない。そこで、SCD では文字列を計算機内部ではコード化し、関係あるパラメータ情報をポインタで結合することによって、パターンマッチングに要する時間を大幅に低減している。その結果、実時間制御で要求されるリアルタイム性を実現している。なお、SCD は SCR 同様 16 ビットマイコン

上で動いている。

適用実験例として、FA 組立ショップの無人台車制御への適用があり、実際レベルに近い搬送制御方式が約 80 ルールで実現されている¹⁹⁾。

5. おわりに

FA システムのシミュレーション、実時間制御への

知識情報処理技術の応用について概説した。シミュレーションに関しては実用レベルの応用がいくつかある。一方、リアルタイム性、高信頼化、高保守化が要求される実時間制御への応用については研究が活発であり、実適用も出始めている。なお、FA 以外の分野への知識情報処理技術の応用も積極的にはかられている²¹⁾⁻²⁴⁾。

今後、製品のライフサイクルがますます短期化し、かつ多種小量生産化の方向にある。そのため、新製品の立上げをいかに早くするかが重要な課題となる。これを解決する1つの手段として、本稿で述べた、ユーザ・プログラマブル、ユーザ・メンテナブルであり、かつ現場の知識を有効に利用できる、知識情報処理型制御方式がますます重要になってくると考える。

最後に、ご指導いただいている(株)日立製作所システム開発研究所、川崎淳所長、井原廣一副所長、春名公一部長、松本邦顕主任研究員に感謝いたします。

参 考 文 献

- 1) 上野：知識の表現と推論，システムと制御，Vol. 27, No. 7, pp. 432-439 (1983).
- 2) Peterson, J. L.: Petri Net Theory and Modeling of Systems, Prentice-Hall (1981).
- 3) 市川：制御からみた知識工学，計測と制御，Vol. 22, No. 9, pp. 749-754 (1983).
- 4) 小林：事象駆動形システムと知識工学，電気学会，Vol. 103, No. 3, pp. 199-203 (1983).
- 5) 長谷川他：マーク流れ線図とその性質について，第19回自動制御連合講演会予稿集，pp. 225-226 (1976).
- 6) 戸塚：シーケンスの新しい表現法—仏 ADEPA の新提案の紹介，機械設計，Vol. 25, No. 2, pp. 117-126 (1981).
- 7) 長谷川：マークフローグラフと FA への応用，計測と制御，Vol. 22, No. 11, pp. 946-951 (1983).
- 8) Chocron, D. et al.: A Petri-Net-Based Industrial Sequence, Proc. IEEE 1980 Int. Conf. and Exhibition on Industrial Control and Instrumentation, pp. 18-22 (1980).
- 9) 藤田他：FA 用ステーションコントローラ (SCR) の提案，第9回 SICE システムシンポジウム講演論文集，pp. 261-266 (1983).
- 10) 村田他：ワークステーション制御用ステーションコントローラ (SCR) のソフト構成の提案，情報処理学会第27回全国大会，4D-11 (1983).
- 11) 松本他：ファクトリーオートメーションにおける自律分散制御システム，日立評論，Vol. 65, No. 12, pp. 5-10 (1983).
- 12) 田代他：離散システム計画・設計用シミュレーションシステム CASE-PLANET におけるプロダクションシステムを利用した運用方式シミュレート方法，第25回自動制御連合講演会論文集，pp. 473-474 (1982).
- 13) Reddy, Y. V. et al.: KBS: An Artificial Intelligence Approach to Flexible Simulation, CMU-RI-TR-81-1, Robotics Institute, Carnegie Mellon Univ., Pittsburg (1982).
- 14) Reddy, Y. V. et al.: INET: A Knowledge Based Simulation Model of a Corporate Distribution System, Proc. Trend & Applications, Automating Intelligent Behavior Applications and Frontiers, pp. 109-118 (1983).
- 15) McArthur, D. et al.: An Object-Oriented Language for Constructing Simulations, Proc. of Int. Joint Conf. on Artificial Intelligence, pp. 809-814 (1981).
- 16) Bourne, D. A. and Fussell, P.: Designing Programming Languages for Manufacturing Cells, CMU-RI-TR-82-5, Carnegie-Mellon-Univ. (1982).
- 17) 田代他：ルール記述に基づくシステム制御方式，計測と制御，Vol. 22, No. 9, pp. 786-790 (1983).
- 18) 田代他：プロダクションシステムを用いた離散型システムの運用自動決定機構，第1回知識工学シンポジウム予稿集，pp. 123-126 (1983).
- 19) 都島他：FA 搬送制御方式の提案とそのルール記述の試み，昭和58年電気学会東京支部大会予稿集，pp. 386-387 (1983).
- 20) 松本他：自律分散制御システム，昭和58年電気学会東京支部予稿集，S. 7-5 (1983).
- 21) 井原：知識工学の産業への応用，電気学会，Vol. 103, No. 3, pp. 204-208 (1983).
- 22) 坂口：知識工学を応用した電力系統の制御，計測と制御，Vol. 22, No. 9, pp. 797-802 (1983).
- 23) 荒屋：人工知能の交通システムへの応用，システムと制御，Vol. 27, No. 7, pp. 440-447 (1983).
- 24) 元田：システムの機能に関する知識を用いた原子炉異常診断の試み，計測と制御，Vol. 22, No. 9 (1983).

(昭和58年12月21日受付)