流れ作業ライン制御へのルール型制御方式の適用

——製鉄所のビレット精整ライン制御への適用——

都 島 功*・田 代 勤*・薦 田 憲 久*
 馬 場 和 史**・高 倉 修 一***

Application of Rule Based Control to Flow Line Control ——Billet Conditioning Line Control——

Isao TSUSHIMA, Tsutomu TASHIRO, Norihisa KOMODA, Kazushi BABA
 and Syuuichi TAKAKURA

In factory automation systems, it is strongly desired to provide highly flexible control software because of frequent system modification. However, short term development and modification of control software is difficult for traditional control software which is developed by using procedural general purpose programming languages.

Recently, knowledge engineering methods which can describe control logics directly and in non-procedural manner have been applied to many fields. However, few applications have been put to practical use. Furthermore, very little is reported about these implementation.

We have applied the knowledge engineering method to real time control of billet conditioning line in steel works.

The paper presented rule based control system applied to billet conditioning line and its evaluation. First, the software configuration in which high flexibility, high response and easy interface to plant were considered, is presented. Next, the practical method for checking consistency of control logic is presented. Finally, evaluation results about the

application are presented. Based on real data, it is presented that control software is developed step by step and easily modified. As the results, control software development manpower were reduced to 1/3 compared with traditional methods. Additionally, useful know-how such as the ratio of inference time for response time (about 20%), software construction method of rule based control system with which control logics can be modified only by data modification are presented.

Key Words: rule based control, knowledge engineering, flexibility, real time control

1. ま え が き

今日、ユーザ要求は多様化しており、その内容は短期間に変化している。そのため、製品のモデルチェンジに伴う、製造ラインの頻繁な変更への迅速な対応がFactory Automation システムに強く要求されている。従来、このようなシステムの制御ソフトは、手続き型で汎用のプログラム言語 (FORTRAN 等) により開発されている。しかし、制御論理がプログラム中に分散し、埋れてしまうため、1度書かれたプログラムから制御論理を読み出すことが作成者でも困難な場合が多い。また、制御知識の利用順序を規定する命令語が知識間を強く結び付けてしまうため、知識の追加、変更、削除が困難となる。さらに、制御論理を考案する生産技術者は一般にソフトの非専門家であるため、プログラマを介さない制御論理の変更ができない。

最近、前述した問題点を解決するために、知識工学応用のアプローチが現われてきている。これは、プロダクションシステムに代表されるように、対象の論理を IF (条件), THEN (結論) 型のルールとして表現

* 第23回計測自動制御学会学術講演会で発表 (昭 59・7)

* (株)日立製作所システム開発研究所
 川崎市麻生区王禅寺 1099

** 川崎製鉄(株)水島製鉄所 倉敷市水島川崎通 1 丁目

*** (株)日立製作所大みか工場
 日立市大みか町 5-2-1

* Systems Development Laboratory, Hitachi, Ltd.,
 Kawasaki

** Mizushima Works, Kawasaki Steel Corp.,
 Kurashiki

*** Omika Works, Hitachi, Ltd., Hitachi
 (Received December 13, 1984)
 (Revised April 24, 1985)

し、その論理の利用順序はシステムが用意する推論機構が自動的に決定するもの（非手続き型）である。これにより、データ構造を考える必要がなく、論理を利用順序と独立に記述でき、日本語的な表現もそのまま扱うことができるので、生産技術者でもプログラムに組込まれた論理を直接理解できるようになる。

知識工学の応用例には、研究レベル、実験レベルでは、鉄道ダイヤの乱れを回復させるスケジュールを作る運転整理問題¹⁾、電力系統事故時の復旧系統決定の問題²⁾、原子炉等のプラント異常診断の問題^{3),4)}、下水処理プロセスの運用制御支援問題⁵⁾などがある。

一方、実用レベルのシステムは、計算機システム構成設計支援エキスパートシステム R1⁶⁾、石油鉍脈試験データ解析システム Dipmeter ADvisor⁷⁾等、極めて少ない。この理由として、実用システムでは、推論機構はシステムの 1 部にしかすぎず、まわりに多くの機能が必要と考えられる。たとえば、Dipmeter ADvisor の報告⁷⁾では、推論機構のコード量は 8%、ルールは 22% と報告されている。従って、知識工学応用システムの実用化のためには、まわりに従来ソフトによる十分なサポートを行うとともに、それらと推論機構との接続の方式を確立する必要がある。しかし、推論機構を含めた知識工学応用システムとしての実現方法については、ほとんど報告されていない。第 2 の理由としては、推論機構に入れる論理の完全性の問題である。実用システムでは常に正しい結論の出ることが要求される。ルールチェックに関しては、メタ知識を活用することで入力ルールを構文上、意味上、因果関係上で検査し、ルール間の合理性を自動的に保持することを目指す知識獲得システムの研究⁸⁾がなされているが、ルール間あるいは入力されている知識間の整合性をチェックするレベルであり、論理そのものの良し悪しを論じるものではない。

筆者らは、今回製鉄所のビレット精整ラインを対象に、知識工学手法を初めて実時間制御に実適用した⁹⁾。そこで、実システムへの知識工学適用上の方式、手順、実用面からの解析結果をまとめたので報告する。まず、変更の容易性、プラントとの簡易なインターフェースを考慮した、ルール型制御システムの構成を示す。つぎに、ルール記述された制御論理の合理性チェックの現実的な方法を提案する。つぎに、製鉄所のビレット精整ライン制御システムの開発過程の分析から、制御論理の変更、段階的 version-up が極めて容易になったこと、応答時間に占める推論機構の処理時間の割合、ソフト開発工数、提案するルールチェック方法が有効であったことなどを、実データにより

示す。

2. 制御対象のモデル

FA では、物の分岐、合流制御や設備への作業割付制御など、各種条件の組合せで制御指令が決まるタイプの制御が多くある。その 1 例として、1 段の流れ作業工程の例を Fig. 1 に示す。本工程では、分岐点に到着したパレットをコンベアで作業ステーションへ搬送し、作業完了後、払出しライン（合流点）へ搬送する。各ステーションは複数種類の作業を実施でき、その能力は、設備番号が小さいほど大とする。この工程の分岐制御をルール記述すると、Fig. 1 に示す 4 ルールになる。なお、このルールの $\langle \min. z \rangle$ は、条件を満足する z のうち、最小の値（能力大の作業ステーション番号）を選択するユーザ手続きである。

Fig. 1 に示した 1 段の工程を多段につなげた流れ作業工程、また、無人搬送車を物流手段とするジョブショップ型の作業工程¹⁰⁾の場合も、物流制御、作業割付制御等の多くの制御は条件組合せ型制御となり、それらはルール型制御向きの制御対象である。今回、ルール型制御方式を適用したビレット精整プロセス制御は、典型的な流れ作業工程の物流制御である。

Fig. 1 A Simple flow line model and rule description of its control

3. ルール型制御システム

本章では、推論機構と他ソフトとの関係、変更の容易性を考慮したルール型制御システムとしてのソフト構成およびソフトの作り方について述べる。なお、推論機構自体については、文献(11)、(12)、(13)を参照されたい。

まず、プロセスとのインターフェイスの問題である。プロセスからの信号の検出、その信号をもとにしたトラッキング、制御指令から制御信号への変換の3つのプログラムはほぼ標準化されている。トラッキングテーブルから推論機構のワークテーブルへの設備現況書き込み機能、推論機構で決定された制御指令の読み出し機能に関して、文字列、パラメータを与えるだけで前述した機能を満足するインターフェースルーチンを設けた。このルーチンを含め、Fig. 2により、ルール型制御システムのソフト構成を説明する。制御モニタタスクはトラッキングタスクから起動されると、発生したイベント内容を調べ、制御指令決定タスクへ知らせる。制御指令決定タスクはイベント情報を取り込み、トラッキング情報も考慮して、適用すべきルール群を選択する（使用した推論機構では、推論時間の短縮のため、予めルールをイベントの種類に対応してルール群に分類している）。そして、各ルール群処理では、推論機構のワークメモリをクリアするルーチン INITFA、つぎに設備現況を書き込むルーチン

WRITFA がコールされる。WRITFA には、書き込みたい設備現況を示す文字列とその現況の具体的なパラメータ値を引数として与える。高速推論プログラム FÖRWD で決定された制御指令は、最終結論読み出しルーチン READRS で読み出され、それは制御信号出力タスクで具体的な制御信号へ変換され、設備へ発信される。なお、READRS には、読み出したい制御指令を示す文字列と変数を与える。

つぎに、ルール型制御システム全体としての変更容易性の実現に関して述べる。実機デバッグでは制御論理の変更は再コンパイル、再リンケージ等のための時間がなかなかとれず、変更論理を再実行できるまでに長時間かかっている。この解決には、データの変更だけで対応可能にすることである。制御論理の修正が、設備現況のワークテーブルへの書き込みプログラムの修正に及ばないように、ルール群に関係なく、一括して設備現況の登録を行うソフトの作り方をすることが必要である。

4. ルールのチェック方法

ルールのチェック方法で、実用システムに使用できそうなものに、ルール間のつながり図を作成してチェックする静的な方法⁸⁾、条件を設定し、推論機構を動かす、所望の結論が得られるかをシミュレートする動的な方法⁹⁾がある。しかし、これらは、制御論理そのものの良し悪しをチェックできるものではない。

Fig. 2 Software configuration of rule based control system

```

F 1
IF (GRダイシヤ<3>セイジョウ)
IF (GR<3>セイジョウ)
IF (GRダイシヤ<3>サイ ナシ)
THEN (GRダイシヤ<3>シヨウ カノウ)
F 2
IF (GRダイシヤ<4>セイジョウ)
IF (GR<4>セイジョウ)
IF (GRダイシヤ<4>サイ ナシ)
IF (サイテイレザイ ナシ)
THEN (GRダイシヤ<4>シヨウ カノウ)
F 3
IF (GRダイシヤ<X>セイジョウ)
THEN (GRダイシヤ<X>シヨウ フカ)
F 4
IF (GR<X>セイジョウ)
THEN (GRダイシヤ<X>シヨウ フカ)
F 5
IF (GRダイシヤ<X>サイ ナシ)
THEN (GRダイシヤ<X>シヨウ フカ)
F 6
IF (サイテイレザイ ナシ)
THEN (GRダイシヤ<4>シヨウ フカ)

```

	Condition 1 GRダイシヤ<X>; セイジョウ/ イジョウ	Condition 2 GR<X>; セイジョウ/ イジョウ	Condition 3 GRダイシヤ<X>; ダイアリ/ ダイナシ	Condition 4 サイテイレザイ; アリ/ ナシ	
Rule 1	セイジョウ(X=3)	セイジョウ(X=3)	ダイナシ(X=3)	—	GRダイシヤ<3> シヨウ カノウ
Rule 2	" (X=4)	" (X=4)	" (X=4)	ナシ	GRダイシヤ<4> シヨウ カノウ
Rule 3	イジョウ	—	—	—	GRダイシヤ<X> シヨウ フカ
Rule 4	—	イジョウ	—	—	"
Rule 5	—	—	ダイアリ	—	"
Rule 6	—	—	—	アリ	GRダイシヤ<4> シヨウ フカ

(1) An example of rules

(2) An example of rules described by proposed format

Fig. 3 Rule checking format

筆者らの対象は、ルール記述されたプログラムを人間の介入の許されない閉ループコントロールの中で動かさなければならない実時間制御のため、実稼動前に条件そのものの抜けやルールそのものの抜け等の論理自体のチェックが確実にできる方法が必要であり、人間が介入するが、その要求を満す、現実的な方法を以下に提案する。

ルールそのままの型では、条件、結論が全て記述されているか、各条件がどの結論と関連しているかが把握しにくく、制御論理の十分なチェックは困難である。この解決には、条件と結論との相互関係の構造化が必要である。そこで、排他条件を1つの条件項目とし、Fig. 3の(2)のように、各ルール毎にその条件を該当する条件項目のもとに位置付ける。また、設備の使用法の違いから下記の原則に従いルールを分類し、変数を含む条件項目にすべきか、定数指定の条件項目にすべきかを決定する。

(タイプ1) 同種の設備に共通の使用方法を指定するルール(パラメータ部を変数で指定するルール)…変数を含んだ条件項目で扱ってよい。

(タイプ2) 同種の設備群に対し、個々に使用方法を定める(パラメータ部を定数で指定するルール)が、設備間にその使用に関し優先順序等の相互関係を指定するルール…変数付きの条件項目とすると、優先順序記述のため、1つの条件項目下に複数の条件を設定することになり、これが他の設備番号のルールとの相互関係をわかりにくくする。そのため、定数指定の条件項目とする。

(タイプ3) 同種の設備群に対し、個々に使用方法を定める(パラメータ部を定数で指定するルール)が、それが各設備固有の条件のみで決まるルール…Fig. 3のルール1, 2が対応する。ルール自体のパラメータは定数であるが、各設備の使用は、各設備固有の条件で決まるため、タイプ2での問題は発生しない。そのため、変数を含む条件項目とする。

5. ビレット精整ラインのルール型制御

5.1 ビレット精整ライン

ビレット精整ラインでは、圧延後の、10m 前後の棒

Fig. 4 Billet conditioning line

材であるビレットを検査，手入れ（研削），仕分けし，出荷する（Fig. 4）。

まず，材を探傷機1で検査する．無傷の材（無手入材と呼ぶ）はテーブル4～10を直進し，仕分けラインへ搬送される．傷の有る材（手入材と呼ぶ）はテーブル7から移載機3にて台車3あるいは4に移され，グラインダ3あるいは4で研削後，移載機4にてテーブル8に戻される．研削後の材は探傷機2にて再検査され，傷なしとなった材は仕分けラインへ，まだ傷のある材は台車4を介してグラインダ4にて再手入される．手入前の材を給材，手入後の材を集材と呼ぶと，移載機3は給材処理だけを，移載機4は集材処理だけを行う．稼動中の移載機3，4のエリアをゾーン2，ゾーン2の前に同様の設備群が将来増設されるエリアをゾーン1と呼ぶ．なお，移載機的一方が故障時，同一ゾーン内の他方が給材と集材の両処理を行う．この場合，材は双方向に流れる．仮置設備では材の円滑な流れを確保するために一時保管を行う．

5.2 制御ルール

Fig. 5 にルール群の全体構成を示す（ルール数の合計は260）．以下に，分割の考え方を示す．

（1）ゾーン間の負荷を平滑化するための材の行先ゾーン決定，デッドロック防止のための材の投入タイミング制御は投入口でしか実施できないので，投入口制御とゾーン内制御を分ける．

（2）各移載機制御にはかなり共通機能があるが，制御論理の修正時にその内容をわかりやすくすること，移載機の増設に対応できるようにすることのため，移載機別に制御を分ける．

（3）異常時制御は，正常時制御に比べ，使用頻度が非常に少ないため，両制御を分ける．

（4）正確な研削時間の予測は難しいため，早めに移動先を決め，移動すると，処理待ちが大となる．この解決のため，吊上げ位置の決定と卸し位置の決定とを分ける．

つぎに，ルール群内のルール分割の考え方を示す．

制御指令は，各種設備の使用可能状況の組合せから決まる．そのため，設備の使用可能状況を抽出する中間結論ルールを設備の種類毎に作る．制御指令となる最終結論ルールは，絞られた中間結論群を条件として作成する．これにより，一度に考慮すべき条件数が限られるため，ルールが容易に作成できる．

上述した考え方を全ルール群に適用したが，紙面の都合から，Fig. 6 に示す比較的ルール数の少ない移載機3給材卸しルール群のみを説明する．

制御方針は，グラインダ稼動率の向上から，卸せる台車があればそこに卸す（ルール15）．卸せる台車はないが，仮置設備に卸せる場合，次材（給材）が既にテーブル7に来ていれば，仮置設備に卸す（ルール16）．以上の最終結論を出すには，台車に卸せるか，仮置設

Fig. 5 Overall configuration of rule groups

チュウカン ケツロン 1		*チュウカン ケツロン 3*	
F	1	F	10
IF	(GRダイシヤ<3>セイジョウ)	IF	(タイレ<7>ザイ アリ)
IF	(GR<3>セイジョウ)	IF	(タイレ<7>イキサキ ゾーン<2>)
IF	(GRダイシヤ<3>ザイ ナシ)	THEN	(ジザイ アリ)
THEN	(GRダイシヤ<3>シヨウ カノウ)		
F	2	*チュウカン ケツロン 4*	
IF	(GRダイシヤ<4>セイジョウ)	F	11
IF	(GR<4>セイジョウ)	IF	(キュウザイ カリオキ<3>ザイ ナシ)
IF	(GRダイシヤ<4>ザイ ナシ)	IF	(キュウザイ カリオキ<3>セイジョウ)
IF	(サイテイレザイ ナシ)	THEN	(キュウザイ カリオキ<3>オロシ コウホ)
THEN	(GRダイシヤ<4>シヨウ カノウ)	F	12
F	3	IF	(キュウザイ カリオキ<3>ザイ アリ)
IF	(GRダイシヤ<X>イジョウ)	IF	(キュウザイ カリオキ<4>セイジョウ)
THEN	(GRダイシヤ<X>シヨウ フカ)	IF	(キュウザイ カリオキ<4>ザイ ナシ)
F	4	THEN	(キュウザイ カリオキ<4>オロシ コウホ)
IF	(GR<X>イジョウ)	F	13
THEN	(GRダイシヤ<X>シヨウ フカ)	IF	(キュウザイ カリオキ<3>イジョウ)
F	5	IF	(キュウザイ カリオキ<4>ザイ ナシ)
IF	(GRダイシヤ<X>ザイ アリ)	IF	(キュウザイ カリオキ<4>セイジョウ)
THEN	(GRダイシヤ<X>シヨウ フカ)	THEN	(キュウザイ カリオキ<4>オロシ コウホ)
F	6	*チュウカン ケツロン 5*	
IF	(サイテイレザイ アリ)	F	14
THEN	(GRダイシヤ<4>シヨウ フカ)	IF	(イサイキ<3>サギョウチュウ OFF)
チュウカン ケツロン 2		IF	(イサイキ<3>ザイ アリ)
F	7	IF	(イサイキ<3>セイジョウ)
IF	(GRダイシヤ<3>シヨウ カノウ)	THEN	(オロシ ザイ アリ)
THEN	(GRダイシヤ<3>オロシ コウホ)	*サイシュウ ケツロン*	
F	8	R	15
IF	(GRダイシヤ<3>シヨウ フカ)	IF	(GR ダイシヤ<X>オロシ コウホ)
IF	(GRダイシヤ<4>シヨウ カノウ)	IF	(オロシ ザイ アリ)
THEN	(GRダイシヤ<4>オロシ コウホ)	THEN	(GR ダイシヤ<X>へ オロセ)
F	9	R	16
IF	(GRダイシヤ<3>シヨウ フカ)	IF	(ジザイ ニテ ハンテイ)
IF	(GRダイシヤ<4>シヨウ フカ)	IF	(ジザイ アリ)
THEN	(ジザイ ニテ ハンテイ)	IF	(キュウザイ カリオキ<X>オロシ コウホ)
		IF	(オロシザイ アリ)
		THEN	(キュウザイ カリオキ<X>へ オロセ)

Fig. 6 Before-grinding material unloading url group

Table 1 Equipment status and control command

Equipment status (including material)	Control command
<ul style="list-style-type: none"> • <u>Equipment</u> <ul style="list-style-type: none"> • position • normal/trouble • empty/occupied by material • in operation/idle • <u>Material</u> <ul style="list-style-type: none"> • position • flawed/ flawless • destination (zonel/zone2/go straight) • before-refining/after-refining (for flawed material) 	<ul style="list-style-type: none"> • <u>for table</u> <ul style="list-style-type: none"> • destination (zonel/zone2/ go straight) • <u>for transfer machine</u> <ul style="list-style-type: none"> • destination (table i/carrier i/temporary storage i) • descent

備に卸せるか等の中間結論を出すルールが必要であり、それは、設備毎に考えればよい。

中間結論1のルール1～6は、台車、グラインダの正常/異常、台車上に材あり/なし、再手入材があり/なしの条件から、台車3、4が使用可能か不可を結論する。中間結論2のルール7～9は、中間結論1を条件に、卸し候補とする台車番号を結論する。グラインダ4の方は再手入作業を優先させるため、台車3を4より優先する(ルール7、8)。ただし、両台車が使用不可ならば、次材にて判定すると(ルール9)、中間結論3のルール10は、テーブル7に給材があるかどうかを判定する。中間結論4のルール11～13は、仮置設備3を4より優先することも考慮し、卸し候補とする仮置設備番号を結論する。中間結論5のルール

14は、移載機3が作業中でなく、材を掴んでいることを確認して、卸し材ありを結論する。以上、設備毎に作成したルールの中間結論1～5の組合せが、最終結論ルール15、16の結論となる。

5.3 制御システムの構成

計算機ハードウェア構成をFig. 7に示す。プロセス信号検出、制御信号出力は16ビットマイコンHIDIC-08Lで、それらを結びつける制御出力決定(推論機構を含む)等は制御用ミニコンHIDIC-V90/50で行っている。

ソフトウェア構成は3章で述べたFig. 2に従った。留意した点を以下に示す。

(1) 3章で述べたように、現地デバッグでの修正がデータの変更だけで済むように、Table 1に示す、き

Fig. 7 Computer hardware configuration

め細かい設備現況を、ルール群に関係なく、一括して推論機構に取込んだ。

(2) 制御指令の方は、マイコンとのインタフェースをシンプルにするため、必要最小限とした。

(3) ルールの振分けは、応答性を高めるため、各イベントに対し、多くとも2つのルール群が動けば済むように、細かい振分けとした。

なお、Fig. 2に示したソフトウェア構成において、推論プログラムとインターフェースルーチンは、FORTRAN 77により、他のモジュールはSPL (Structured Programming Language) により作られている。

6. ルール型制御方式の評価

(1) 応答性

応答時間の目標値は、グライнда処理効率に影響を与えないこと、材の流れのスピード等を考慮して、最大1secと設定していた。これに対し、Table 2に示すように、マルチタスク下で、制御指令決定タスクの最大応答時間 568 msec (平均は 128 msec) の実測値を得て、目標値の達成を確認した。

Table 2 Response time of control command decision task

	Waiting time (a)	Execution time (b)	Response time (a+b)
Maximum time (m sec)	29	539	568
Average time (m sec)	11	117	128

Table 3 Processing time of inference program

Rule group name	Number of rules	Processing time /on rule group (m sec)			Processing time /one rule (msec)
		Maximum	Minimum	Average	
Rule group (B)	48	98	54	57	1.2
Rule group (C)	19	34	21	22	1.2
Rule group (D)	9	14	11	12	1.3
Rule group (G)	19	30	21	23	1.2
Rule group (H)	8	16	9	10	1.3

推論プログラム (FORWD) だけの処理時間は、Table 3に示すように、最大のルール群である材の投入タイミング制御ルール群でも、最大 98 msec (平均 57 msec) の実測結果であった。同一ルール群に対する、処理時間のばらつきは、適用時点で設備現況および中間結果の数が異なることによる。なお、1ルール当りの平均時間は 1.2 msec であった。

応答時間に占める推論機構の処理時間の割合は、制御指令決定タスクの実行開始から完了までの時間 117 msec に対する1ルール群当りの平均処理時間 26msec の割合と考えることができ、20%強であった (1ルール群とした理由は、各イベントに対し、ほとんど、1ルール群しか動かないようにルール群の振分けを行ったため)。

(2) 拡張性、保守性

ルール型制御方式により、制御プログラムが段階的に開発できたことを示すために、マクロなルール数の時間推移を Fig. 8 を使って説明する。

まず、正常時のルール群を固め、つぎに異常時のルール群を作るという段階の開発方針をとった。正常時のルール群はA, C, D, G, H (Fig. 5参照) であり、ルールの合計は約60であった。その後、設備の故障時、代替設備による自動運転の可能化を実現するため、ルール群E, F, I, Jを追加し、ルール数は約200になった。しかし、材の投入タイミング制御をしないとデッドロックが発生することが分り、ルール群Bを加え、ルール数は251になった。以上が現地デバッグ前までのルール数の推移である。

現地デバッグでは、2回のルール修正を行い、現在順調に稼動している。その修正は運転状況を見てルールの不具合が判明したために実施した。その1つの給材吊上げルール群Cの修正を説明する。修正前のルー

Fig. 8 Transition of number of rules

ルでは、テーブル上の材に関し、テーブル7上に吊上げる材があることを条件としていた。しかし、テーブル7に材がなくとも、その上流に吊上げるべき材があれば、移載機に吊上げ指示を先に出し、移載機をテーブル7の前で待機させる方が効率的と分り、ルール修正した。この修正を含め、最終的には260ルールになった。これらの修正は、データのみの変更で済んだ。

以上は、ルール数のマクロな推移であり、各フェーズでは、ミクロに変動している。これは、4章で提案したルールチェック方法によるもので、ルール数の約1割程のルールの抜けや不合理なルールを抽出できた。そのチェック方法では、条件項目そのものの抜け、あるルールに対する条件の抜けが最も多く抽出できたが条件の組合せが全てを尽してないことの判明から、ルールの追加もかなりあった。

一方、条件を設定して、推論機構を動かし、所望の結論が得られるかをシミュレートする方法も適用したが、ルールの文字列、単語の不統一のミスをいくつか発見できたのみであった。そのミスの発見が少なかった理由は、シミュレーションする前に、エディタのファインド機能（検索したい文字列を指定すると、その文字列のある箇所を順次示す機能）で大部分をチェッ

Fig. 9 Reduction in the manpower for program development

ク、修正したことによる。

(3) ソフト開発工数

Fig. 9 に示すように、ソフト開発工数は、従来の手続き型言語を利用した場合に比べ、1/3に低減できることが分った。なお、従来のソフト開発工数の値は、過去の実績データに基づいた推定値である。この主な要因は、以下の項目と考える。

(i) システム設計段階で決定されるルールが従来のプログラムに対応する。そのため、従来、必要であった、システム設計書をもとにした、テーブル設計、処理フロー設計、プログラミングが不要となる。

(ii) プログラム（ルール）がシステム設計段階で高品質化でき、最も工数の要する現地デバッグの工数が大幅に低減できた。

7. あとがき

以上、ルール型制御方式が実時間性を満足していること、制御ソフト開発工数が1/3になることなどの有効性を示し、あわせてルール型制御システムの構成、ルールのチェックなどに関し、ルール型制御方式適用上の有用な知見を示した。

本適用は、川崎製鉄(株)・水島製鉄所・システム部の斉藤達部長、山崎順次郎課長ほか同部の方々、(株)日立製作所・大みか工場の出村吉晴部長、同・システム開発研究所の春名公一部長、松本邦顕主任研究員ほか、同社の方々の御指導の下に行われたもので、関係各位に深謝する。

参考文献

- 1) 荒屋, はか: ヒューリスティクスを利用した列車運転整理手法, 電学会論文誌C, **103**-11, 249/256 (1983)
- 2) 松本, はか: 知識ベースに基づく電力系統復旧方式の決定法, 電気学会論文誌B, **103**-B-3, 175/182 (1983)
- 3) 元田, はか: 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, **22**-9, 791/796 (1983)
- 4) 栗原, はか: プロダクションシステム応用のプラント異常診断方式, 電学会全国大会 (1983)
- 5) 前田: ルールベースによる下水処理プロセス運用制御支援法, 第22回 SICE 学術講演会予稿集, 169/170 (1983)
- 6) McDermott, J.: R1: A Rule-based Configurer of Computer Systems, Artificial Intelligence, **19**-1, 39/88 (1982)
- 7) R.G. Smith: On the Development of Commercial Expert Systems, AI Magazine, **5**-3, 61/74 (1984)
- 8) 渡辺: 知識獲得支援システム, 計測と制御, **23**-9, 767/773 (1983)
- 9) 都島, はか: ルール型制御方式の実プロセス制御への適用, 第23回 SICE 学術講演会予稿集, 273/274 (1984)
- 10) 都島: FA 搬送制御方式の提案とそのルール記述の試み, 電気学会東京支部大会予稿集, 386/387 (1983)
- 11) 都島, はか: 知識情報処理技術のFAへの応用, 情報処理, **25**-4, 378/385 (1984)
- 12) 田代, はか: 分散型システム用ルールベースコントローラ, 第2回知識工学シンポジウム予稿集, 115/120 (1984)
- 13) N. Komoda, et al.: An Autonomous, Decentralized Control System for Factory Automation, IEEE Computer, **17**-12, 73/83 (1984)