

# 機械図面自動認識システム"ARCADIA-M" 4N-7 — 知識表現と制御手法 —

松坂 基弘 日根 俊治  
(松下電器産業(株) 技術本部 システム研究所)

## 1. はじめに

機械図面自動認識システム"ARCADIA-M"における知識表現とその制御手法について概説する。

機械図面においては、例えば、加工具を意識して設計するため、認識すべき外形線が全く同じ図面でありながら、認識のための鍵となる寸法線の記入方法が異なる等、問題領域に固有のノウハウが数多く存在する。本システムでは、機械図面に係るこのような個々の知識を、KS (Knowledge Source) と呼ぶ独立した知識単位で記述し、KSの起動制御に関するメタ的な知識にもとづいて、柔軟な制御を実現している。

## 2. 概要

図1に、本システムにおける制御構造を示す。本システムでは、個々の知識を最小の機能単位よりなる関数に分割し、これらを組み合わせることによりKSを記述しており、後述のコントローラ、スケジューラ、及びインタプリタによって、上記KSに示された各処理用の関数が選択、起動され、共通データ領域を介して、認識処理が進められる。

## 3. 知識表現

### 3.1. 関数の説明

以下に各処理用関数群の簡単な説明を記す。

- 1) 画像処理用関数群(画像処理部<sup>[1]</sup>)  
細線化、近似符号化等、2値のラスターデータをベクタ情報に変換するまでの画像処理に必要な関数の集合。
- 2) パターン認識用関数群(パターン認識部<sup>[2]</sup>)  
端末記号、中心線、文字認識等、パターン認識に必要な関数の集合。
- 3) 図面解釈用関数群(図面解釈部<sup>[3]</sup>)  
文字情報、寸法線、寸法補助線、寸法規定点間の対応づけ等、機械図面を認識、解釈し外形線の位置や形状を決定するために必要な関数の集合。
- 4) モニタ用関数群  
画像処理用、パターン認識用、及び図面解釈用の各関数群により書き込まれる共通データ領域内の状態を監視するために必要な関数の集合。

### 3.2. KSの説明

本システムでは、機械図面に関する個々のKSを統一的に、"IF (条件部) THEN (実行部)"のルール形式で記述することにより、各KS間のモジュール性を高めるとともに可読性、保守性に優れた知識表現形式を採用している。一方、機械図面に関する知識は、例えば細線化処理のように必ず一度だけ実行される基本的なもの、認識過程において矛盾が生じた場合にのみ実行される例外的なもの、及びそれらの知識をいつ評価すれば良いかという制御に係るメタ的なものに大別することができる。本システムでは、処理の効率を考慮して、事象

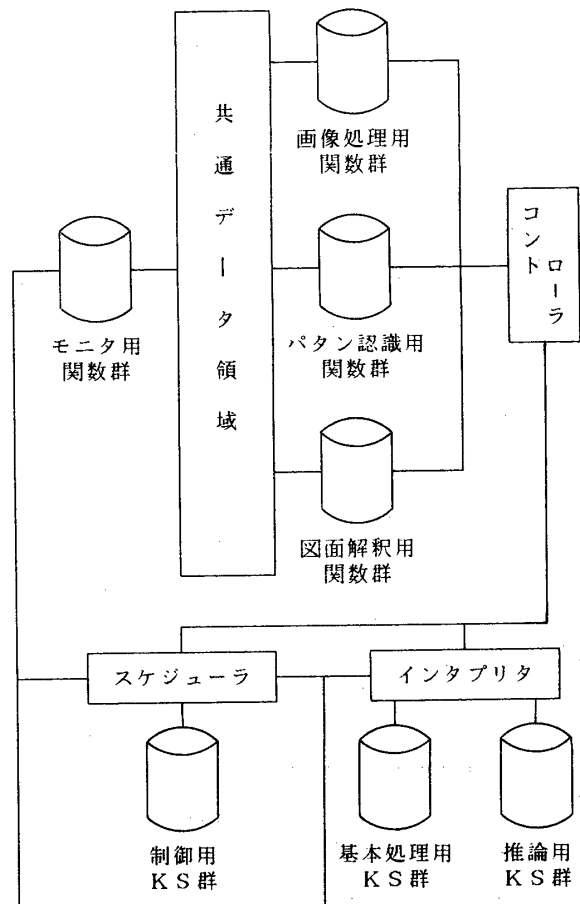


図1. 制御構造

AUTOMATIC RECOGNITION SYSTEM FOR MECHANICAL DRAWINGS "ARCADIA-M"  
- KNOWLEDGE REPRESENTATION AND CONTROL METHOD -

MOTOHIRO MATSUSAKA SHUNJI HINE

SYSTEMS RESEARCH LABORATORY MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD

駆動型の基本処理用KSと目標駆動型の推論用KSを併用し、これらの上位にメタ的な制御用KSを設けることにより、階層的な知識表現の体系をとっている。

#### (1) 制御用KS

共通データ領域内の状態により、知識表現の階層性を利用して、多数のKSの中から評価すべきタイミングにあるものを効率的に選択することができる。本KSは、共通データ領域内の状態が、トリガとなる条件部を満たす場合に起動され、基本処理用KS、又は推論用KSのKSグループ名を与える。(図2、3)

#### (2) 基本処理用KS(事象駆動型)

本システムの入力対象となる機械図面に描かれる外形線は、直線・円・円弧を組み合わせた任意形状が許されるため、特に、認識過程の初期段階においては、入力さ

IF

(条件部:モニタ用関数)

THEN

(実行部:<<KSグループ名>>)

図2. 制御用KS

<<KSグループ名>> ::

<KS名>

<KS名>

.....

図3. KSの階層的表現

<KS名> :

IF

(条件部:モニタ用関数)

THEN

(実行部:画像処理・パターン認識・図面解釈用関数)

図4. 推論用KS

IF

is\_corresponding\_character\_node\_over()

THEN

<<KS\_reform\_outline>>

<<KS\_reform\_outline>> ::

<ks\_reform\_chamffer>

<ks\_reform\_fillet>

.....

<ks\_reform\_chamffer> :

for all\_dimension\_line\_list( nodimen );

IF

is\_character\_attribution((\$1)nodimen.C)

get\_indicated\_element((\$1)nodimen.

(\$0)&noindce.omitelem)

check\_number((\$1)omitelem.GE.2)

THEN

get\_fixed\_node((\$1)noindce.(\$0)termnode)

get\_middle\_node((\$1)omitelem.termnode.

(\$0)midlnode)

create\_element((\$1)LINE.(\$0)&newelem)

replace\_element((\$1)omitelem.newelem.termnode.

(\$0)omitnode)

copy\_element\_attribution((\$1)omitelem.newelem

repair\_indicated\_element((\$1)noindce.newelem)

delete\_element((\$1)omitelem)

delete\_node((\$1)omitnode)

図5. 面取り部分の外形線データ修正例

れた画像データに依存して処理が進められる。本KSは、例えば細線化処理のように、必ず一度だけ行われる処理を記述したものであり、この様な例の場合、条件部には"not\_care"と記述する。なお、実行部関数は、画像処理用・パターン認識用・図面解釈用関数からなる。

#### (3) 推論用KS(目標駆動型)

本KSは、ボトムアップ的な基本処理用KSの起動後、共通データ領域内に矛盾が生じた場合に起動される。共通データ領域内の状態により仮説を立て、それを検証する形で条件部が評価され、仮説が正しい場合には、実行部に示された各処理用の関数群が順次実行され、共通データ領域内の矛盾箇所を自動的に修正する。(図4)

なお、図5に、面取り部分の外形線データを修正するKSの一例を示す。機能及び機能の対象である引数を陽に記述することができる。また、日本語でコメントを加えることも可能である。

## 4. 制御手法

本システムにおいては、常に共通データ領域内の状態を判断しながら制御が進められる。まず、スケジューラが制御用KSの条件部を評価し、次に起動すべきKSグループ名をインタプリタに引き渡す。インタプリタは、そのKSグループ名が基本処理用のものか、推論用のものかを判断して、KSグループに階層的に登録されているKSの条件部を評価し、総ての条件が満たされる場合に、実行部に記述されている各処理用の関数名をコントローラに引き渡す。コントローラは、これら関数を順次起動し処理を進める。

従って、共通データ領域内の状態により、基本処理用KSの実行部関数が起動されている場合は、制御の流れはボトムアップ的であるが、推論用KSの実行部関数が起動されている場合、制御はトップダウン的な流れに変更される。本システムでは、図面解釈上矛盾が生じた場合にのみ、画像処理用・パターン認識用・図面解釈用の各関数群を再起動することにより、機能的にバックトラック制御を実現している。

## 5. おわりに

本システムでは、機械図面に係る個々の知識をKSと呼ぶ独立した単位で階層的に記述し、KSの起動制御に関するメタ的な知識にもとづいて制御を行うことにより、知識と処理手順の分離を図った。これにより、機能の拡張やメンテナンスを容易に行うことが可能となる。

### 〔参考文献〕

- [1] 本講演論文集 4N-8
- [2] 本講演論文集 4N-9
- [3] 本講演論文集 4N-10