

연삭가공 트러블슈팅을 위한 룰베이스 룰의 구성

이재경*, 김건희**, 송지복***

Production Rules Based on the Rule-Based Model for Grinding Trouble-shooting

Jaekyung Lee*, Gunhoi Kim**, Jibok Song***

ABSTRACT

Cognition and control of grinding trouble occurring during the grinding process are classified into a quantitative knowledge which depends on experimental data and qualitative knowledge which relies on skilful engineers. Grinding operations include a large number of functional parameters, since there are several ways of coping with grinding trouble. One is the qualitative method which depends on empirical knowledge utilizing the skilful experts from the workshop, the other is the quantitative method which utilizes the experimental data obtained by sensor. But, they are all difficult to accomplish from the grinding trouble-shooting system. The reason is that grinding troubles are not easily controlled in the quantitative method, and therefore, trouble-shooting has mainly relied on the knowledge of skilful engineers. Thus, there is an important issue of how a grinding trouble-shooting system can be designed and what knowledge is utilized among the large amount of grinding trouble information.

In this paper, basic strategy to develop the grinding database by taking rule-based model, which is strongly depended upon experience and intuition, is described.

Key Words : Grinding operations(연삭가공), Database(데이터 베이스), Rule-based model(룰베이스 모델), Trouble-shooting systems(트러블슈팅 시스템), Automatic manufacturing process(자동화 생산)

1. 서 론

생산가공에 있어서 데이터베이스의 활용기술은 생산자동화의 요구가 근년에 와서 급격히 대두됨에 따라 효율적인 생산을 위한 데이터베이스 구축이 연구되었다. 즉, 생산자동화 분야에 있어 작업조건 설정, 공구 선택, 가공액 선정 및 트러블 검

지·처리를 포함한 CIMS의 요구가 가속화 되고 있다. 연삭가공용 자동화 생산시스템 지원을 위한 데이터베이스에 축적되는 지식이 경험이나 기능등의 정성적 지식에 크게 의존하는 경우가 많으므로 이를 유효하게 이용할 수 있도록 데이터 베이스의 설

* 한국기계연구원 자동화연구부
** 전주대학교 기계·산업공학부
*** 부산대학교 기계공학부

계와 지식표현이 강구되어야 한다.

이들 데이터베이스는 대부분이 동작기계 특성을 구체적으로 기술하지 않은 범용성에 중점을 두고 있다. 그러나 본 연구대상인 연삭가공에 대한 고정도·고품위를 요구하는 생산을 대상으로 한 경우에 있어서는 동일한 모델의 동작기계라 할 지라도 그 결과는 경우에 따라서는 상이한 차이가 있어, 이러한 특징을 충분하고도 상세하게 반영하기 위해서는 각각의 정보를 룰 베이스에 기능적으로 분류하여 수납함으로써 신뢰성 있는 데이터의 활용 및 추론시간(Inference Time)의 단축을 기대할 수 있다.

본 연구에서는 Lot size가 비교적 적은 준전용 연삭가공을 대상으로 한 동작기계 특성을 고려하여 지식을 특정 개념으로 분류하여 룰 베이스화하고, 그 시스템 운영에는 목적지향형(Object Oriented Paradigm Systems)⁽¹⁾로 구축하였다. 따라서 이를 위한 효율적인 연삭가공용 데이터베이스 설계를 위한 기본 알고리즘을 정립하고 설계한 시스템의 효율을 검토하였다.

2. 룰 베이스 모델의 특성

본 시스템 룰의 추론은 후레임을 작업영역으로 이용하여 룰의 조건부를 조건조합(Pattern Matching)하여 실행한다. 룰의 추론은 조건조합, 경합해소, 룰 실행사이클을 반복해서 주어진 조건의 최적치를 얻을 수 있도록 각각 주어진 조건을 탐색하여 이루어진다. 조건조합은 룰의 각 조건과 작업조건과의 조합으로 룰 형태의 사이클 중에서 가장 실행시간이 걸리는 부분이다. 따라서 이는 조건조합의 효율화가 데이터 베이스 성능을 좌우하게 된다. 본 시스템에서는 조건조합의 효율화를 추구하기 위해 RETE 알고리즘⁽²⁾을 적용하고 있다. RETE 알고리즘의 특징은 다음과 같다.

(1) 룰 컴파일에 있어서 전체 룰의 조건부를 기본적으로 분해해서 룰간에 동일한 조건을 공통화하여 최적의 추론 네트워크를 생성한다. 또한 추론 Network는 동일 클래스(Class)내에 폐쇄된 클래스내의 조건조합과 클래스간에 추천된 조건조합의 2단계로 구성된다. 본 시스템에서는 각 룰을 활용하는 데이터 베이스를 Class와 Instant(또는 Slot)로 나누어 그룹관리를 하는 것은 조건부 입력에 대한 데이

터 베이스의 검색에 있어서 추론시간을 단축할 수 있도록 본 시스템에서는 RETE 알고리즘을 개선하여 구축하고 있다.

(2) 추론실행은 추론대상이 되는 후레임 전체를 대상으로 하고 추론 네트워크에 표시된 순서로 조합을 하고, 그 조건조합에 따른 실행중의 결과는 Temporary Frame에 보존된다. 이때 복수의 조건부가 진(True)이 된 경우에는 경합해소(Conflict Resolution)를 하여 한가지를 선택한다. 경합해소를 효율적으로 하기 위해서 본 시스템에서는 RETE 알고리즘에 각 룰에 대하여 가중치(Weight Value)를 부여하여 경합해소에 따른 시스템의 추론시간을 단축하고 있다.

(3) 추론사이클의 두번째 이후에는 Temporary Frame에 보존되어 있는 이전 사이클까지의 조건조합의 도중결과를 이용해서 조건조합 횟수를 감소시킨다. 한번 Pattern Matching된 룰에 대하여는 다음 룰의 추론시에는 학습기능을 부여하여 가장 새로운 룰 조합을 활용할 수 있도록 시스템을 구축하였다.

한편 현장의 기능적·경험적 지식을 연삭가공 자동화 생산시스템에 활용하기 위해서는 각종자료와 정성적 지식 및 정량적 지식을 효과적으로 데이터 베이스화하기 위한 데이터 精練過程(Refined Process)⁽³⁾이 필요하다. Fig.1은 본 시스템에서 작업 기능자의 경험적 지식으로부터 자동화 생산시스템에 대하여 데이터 베이스화하기 위한 지식의 처리 과정을 나타내고 있다. 즉 현장에서 얻은 데이터는 정량적 지식과 정성적 지식을 Class와 Instant(Slot)로 분류하여 저장하고, 각각의 클래스 지식은 가중치를 두어 우선순위를 설정하여 추론시간 단축함으로써 데이터 베이스의 효율화를 도모하였다. 또한 Fig.2는 본 시스템에서 룰 베이스 구축을 위한 지식 정보의 분류와 트러블 처리를 위한 기본적인 개념으로 진단(증상)결과를 토대로 한 트러블 처리기준을 제시하고 있다. 그리고 본 시스템의 연삭가공용 트러블 처리를 위한 기능적·경험적 지식의 자동화 생산 시스템에 활용하기 위한 경합해소 규칙으로, 경합해소(Conflict Resolution)는 조합된 복수의 Instantiation 중에서 다음에 실행해야 할 다수의 Instantiation에는 룰의 실행부를 실행하는 Rule Instantiation과 끝(Goal) 생성을 하는 Goal Instantiation이 있다. 본 시스템에 설계한 경합해소⁽⁴⁾ 전략은 다음과 같다.

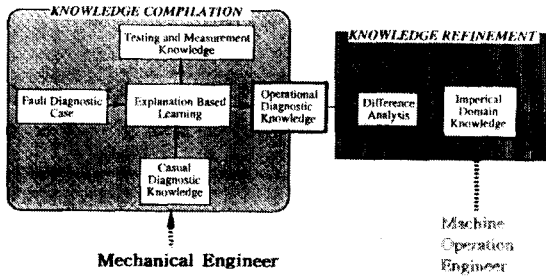


Fig. 1 Data Refinement Procedure Obtained for Mechanical Engineer

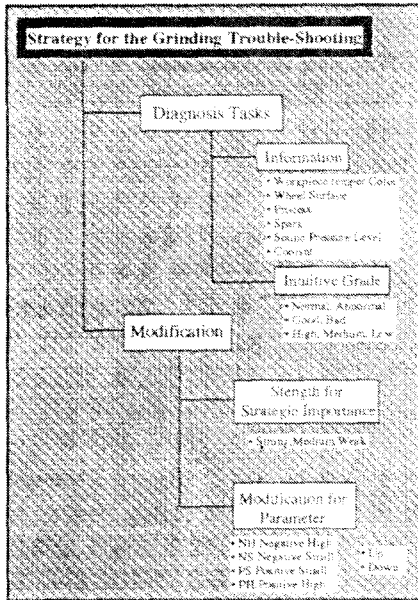


Fig. 2 Strategy for the Grinding Trouble-shooting

- (1) Goal: Goal Instantiation을 Rule Instantiation보다 우선하여 선택한다. 만약 동일한 종류의 Instantiation이 다수 대립하는 경우에는 우선 Instantiation 중 룰의 우선순위가 가장 큰 것을 선택하고, 또한 우선순위가 같은 Instantiation이 복수 존재하는 경우에는 Instantiation중의 Instance중에서 가장 최근에 경신되어 있는 것을 우선한다.
- (2) Rule: Rule Instantiation을 Goal Instantiation보다 우선하여 선택한다.

(3) Mix: Goal Instantiation과 Rule Instantiation을 구별 없이 다룬다.

그밖에 No-loop Goal, No-loop Rule, No-loop Mix 등이 설계되어 있다. 이는 기본적으로 Goal, Rule, Mix와 동일한 역할을 하지만, 차이점은 Instantiation이 경합집합으로부터 선택되어 룰의 실행부를 실행하여 과거에 만들어진 다른 Instantiation과 동일한 내용의 후레임을 작성하는 경우 동일한 내용의 Instantiation을 만드는데 대하여 No-loop Goal, No-loop Rule, No-loop Mix는 작성하지 않는다. Fig.3은 본 시스템에서 연삭숫들의 입도(Grain Size)를 결정하기 위한 대역함수(Global Function)를 활용한 룰 베이스이다.

```
(defun Determination_Grainsize
    (g_method mat heat)
  (let ()
    (loop
      (kdialog_message "Determine Grain Size")
      (cond ((get-instance 'Wheel-DAT)
              (kdialog_message3 "Determine Grain
Size from Actual Grinding Data-base."
"Select the Given Number(Hold is NIL) ")
              (setq grain_size (input_ans
(list-menu-and-select
(get-all-instance-database-variable
'Wheel-DAT 'Grainsize
(list (list 'Grinding_Type-DAT
(list 'Grinding_Type g-method))
(list 'Workpiece_Kind-DAT (list
'Workpiece_Kind mat)
(list 'Heat_Treatment heat ))
(list 'Surface-DAT (list
'Surface_Grade surface) )
)))
              ))
      (cond ((not (equal grainsize 'nil))
              (return grainsize)))
      (kdialog_message3 "Determine Grain Size
from Grinding Knowledge-base.")
      (run-ruleset 'grainsize_rule)
      (setq grainsize (input_ans grainsize))
      (return grainsize)
      ))
  )
)
```

Fig. 3 Determination of Grain Size Based on Rule-based Model

Fig.3에서와 같이 본 시스템에 사용한 대역함수로 "get-all-instance-database-variable", list-menu-and-select", "get-instance" 등을 활용하고 있다. 여기서 정의한 대역함수 "get-all-instance-database-variable"는 데이터 베이스에 있는 모든 인스턴스 변수를 검색하라는 기능을 갖고 있으며, "list-menu-and-select"는 주어진 조건부의 전제조건을 만족할 수 있는 데이터를 검색하여, 가장 가능성이 있는 데이터를 선택하기 위한 함수이며, "get-instance"는 위의 과정에서 검색된 데이터 중에서 가장 가능성이 있는 하위개념(실체적인 데이터)을 선택하여 Temporary Frame에 저장하기 위해 설계된 대역함수이다. 또한 Fig.3은 최적 연삭숫돌을 선정하기 위한 LISP언어로 구성된 룰 베이스 모델에 의한 데이터 베이스를 검색하기 위한 프로그램으로, 연삭숫돌을 설정할 때에는 공작물의 종류 및 공작물의 열처리 조건(g_method mat heat)을 고려하도록 설계되어 있다.

본 시스템에 사용한 대역함수(Global function)로 "get-all-instance-database-variable", "list-menu-and-select", "get-instance" 등을 구성함으로써 데이터베이스의 조합과 검색의 효율화를 도모하고 있다.

Fig.3에서와 같이 연삭숫돌 요소 중 숫돌입자(Grainsize)를 선정하는 흐름은 아래와 같다. 우선 공작물의 종류와 열처리 조건을 선정하기 위해 실가공 데이터 베이스(Actual Grinding Database)을 검색한다. 그 다음 연삭숫돌의 입자선정을 위한 본 시스템에서 제시한 선정조건에 관한 데이터 베이스 변수로 본 시스템에서 정의한 대역함수인 (get-all-instance-database-variable)을 기동하여, 연삭방식(Grinding_Type-DAT), 공작물의 종류(Workpiece_Kind-DAT) 및 가공물의 요구표면 거칠기(Surface-DAT)를 조합하는 조건을 탐색한다. 마지막으로 이에 대응하는 데이터가 존재하지 않는 경우에는 기존지식베이스(Grinding Knowledge-base)를 검색하여 Default값으로 연삭숫돌의 입자를 설정한다. 한편, 본 시스템의 Pattern Matching을 위한 룰의 지식표현(Knowledge Representation)은 다음과 같다.

(1) 주어진 사실로부터 새로운 사실을 이끌어내는 전향적 룰(Forward chaining Rule)⁽³⁾과 결론을 가정하여 그 과정이 성립할 수 있도록 조건을 이끌어내는 후방향 룰(Backward chaining Rule)⁽³⁾로 구성된다.

(2) 관련이 있는 룰을 Rule Set로 정리하여 룰로

표현한 지식을 체계화한다.

(3) 룰 기동은 룰 조건부와 후레임 조합에 의해 이루어지는 작업영역에 대한 후레임을 선택한다. 그리고, Forward chaining Rule의 기본구성은 <조건부> ⇒ <실행부> 로 구성된다.

여기서 조건부는 If부에 대응하고, 실행부는 Then부에 상응한다. 즉, 본 시스템의 Forward chaining은 조건부가 만족될 때 실행부에서 기술되어 순차적으로 실행되는 구조로 되어 있다. 또한, Backward chaining은 Head부, 조건부, 실행부로 구성되며, <Head부> ← <조건부> ⇒ <실행부> 로 표현된다. Backward chaining Rule의 실행은 지정된 후레임 또는 Working Memory 조건이 "진"이 되지 않는 경우에는 생성된 Goal이 Head부와 조합하여 조건부가 만족될 때 실행부에 정의된 질차에 의하여 실행된다. 조건부에는 후레임의 종류, 관련에 관한 후레임 조건, Goal Frame 조건, Group 조건, Working Memory 조건, Goal Working Memory 조건 등이 있다.

3. 룰 베이스 모델에 의한 연삭가공 트러블 슈팅

연삭가공의 트러블 진단·처리는 전문적, 경험적 지식을 If-Then 형의 Production Rule 형태로 축적한다. 연삭가공의 애매하고 상호관계가 복잡한 정보를 룰화하여 사용하는 경우에는 계층분석(AHP)⁽⁶⁾를 적용하여 지식의 타당성을 부여하여 룰의 우선순위를 결정할 필요가 있다. 본 시스템의 트러블 진단·처리에는 지식 베이스내 후레임 지식과 룰 지식을 이용하며, Fig.4와 같이 증상, 원인, 처리(제어)을 위한 룰 베이스의 Rule-set로 구성되어 있다.

후레임 지식에 대한 사실적인 데이터와 룰 베이스내의 조건과를 조합하여 조건이 성립하면 Rule-set의 실행부를 실행하여 트러블 처리를 수행한다. 본 시스템에서 설계한 연삭가공용 트러블 처리과정을 정을 위한 기본적인 알고리즘은 다음과 같다.

(1) 트러블이 발생하면 후레임 지식의 Method로 부터 룰 지식 베이스의 Rule-set가 기동한다.

(2) 주어진 Object에 대하여 지식베이스내의 모든

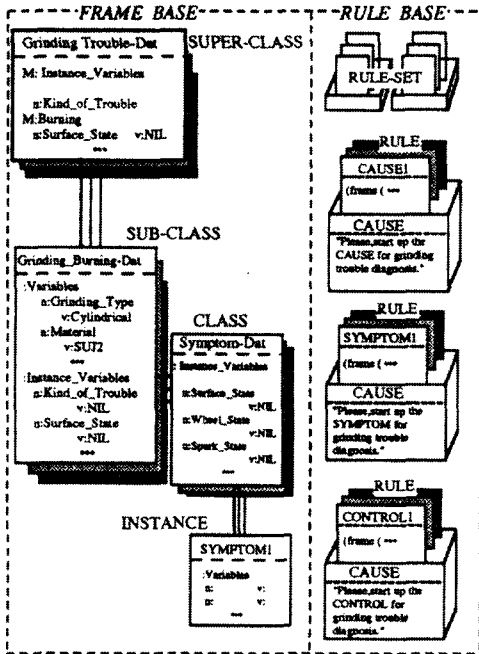


Fig. 4 Grinding Trouble Database by Rule-based Model

Instance, Goal과 Rule-set내의 모든 불조건을 조합하여 조건이 성립한 룰과 인스턴스(Instance) 조합을 구성한다. 우선 Backward chaining Rule 기동은 Forward chaining Rule에서는 그 조건이 성립하지 않으면 Goal을 생성한다. Backward chaining Rule에서는 Head-part와 동일한 후래임 조건을 갖는 Goal이 있는 경우에 조건부의 조합을 실행한다.

(3) 복수조건이 조합되는 경우에는 AHP(계층분석법)에 의해 얻어진 중요도(Weight)가 큰 순으로 선택한다. 중요도가 같은 경우에는 룰의 전제조건이 조합하는 Instance의 갱신순서의 최신 데이터를 우선으로 선택한다.

(4) 조건부가 모두 만족한 경우에는 Rule 실행부를 실행한다.

Fig.4에서 트러블 슈팅을 위한 데이터 베이스는 최상위 클래스에 연삭 트러블에 관한 전반적인 관리를 위한 후래임(Grinding Trouble-DAT)이 있고, 그 바로 아래 하위 클래스인 연삭눌음(Grinding_Burning-DAT)과 (Chatter_Vibration-DAT)

로 설계되어 있다. 또한 각각의 하위 클래스는 증상(Symptom-DAT), 트러블의 원인(Cause-DAT) 및 이를 제어하기 위한 처리 또는 제어(Control-DAT)의 최하위 클래스로 구성되어 있다. 또한 각각의 최하위 클래스에는 구체적인 데이터가 수록되어 있는 인스턴스(Instance)로 설계하여, 주어진 전제조건에 대하여 유기적으로 조합할 수 있도록 했다.

또한 Fig.5은 연삭가공용 Trouble-shooting의 시스템 운영을 위한 시스템 제어용(System Control) LISP Program이다.

```
# (rule 1)
(relation (?fault symptom1 symptom2 symptom3)
 (frame (fault ?f abnormal surface t) )
 (frame (symptom ?s1 (temper_color t) ) )
 (frame (symptom ?s2 (loading of wheel t) ) )
 (frame (symptom ?s3 (suitable to the
                    amount of coolant t) ) )
-->
(create kind of trouble (kind of trouble ?k)
 (call rule-set rule 2)
# (rule 2)
(not (frame (symptom ?? (grain size ?) ) )
<--
(not (frame (symptom ?) ) )
-->
(call (format t "~% wheel depth of cut ?")
 (call (format t "~% grain size ?")
 (bind ?x number-read) )
-->
(create rule-set 3)
# (rule 3)
(goal (frame (modification ??) )
 (frame* (grinding_conditions ?
 (grain size mesh number ?x! (>80 ?x46) ) )
-->
(create modification (modification grade-1) )
# (rule 4)
(frame (modification ?grinding_trouble
 (modification_parameter ?) ) )
->
(bind ?x (modification (modification
 ?grinding_trouble) ) )
(bind ?message (rule_addstring "main cause
 considered with trouble are" ?x " ) )
(call (display_result (list "" "" ?message) ) )
(call (print 'modification_of_parameter,
 "please, decrease a little the wheel depth of
 cut or increase a little the workplace
 velocity at the first place"))
```

Fig. 5 System Control Program for Grinding Trouble-shooting

Fig.5와 같이 본 시스템의 룰 베이스에 의한 연삭가공용 트러블 슈팅은 전방향 룰(Forward-chaining Rule)과 후방향 룰(Backward-chaining Rule)을 이용하여 효율적으로 대응할 수 있도록 설계되었다.

Fig.6은 본 시스템의 룰 베이스의 룰을 구성하기

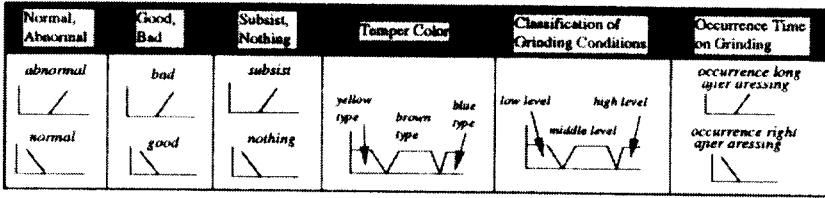


Fig. 6 Conceptual Classification for Grinding Trouble-shooting System

GW:Grinding Wheel, GC:Grinding Conditions, DC:Dressing Conditions

| no. | IF-CLAUSE | | | | | | | | | | | | | THEN-CLAUSE | | | | | | |
|-----|----------------|---------------------|---------------------|----------------|---------------------|---------------------|----------------|---------------------|---------------------|----------------|---------------------|---------------------|----------------|---------------------|---------------------|----------------|---------------------|---------------------|--|--|
| | grinding wheel | grinding conditions | dressing conditions | grinding wheel | grinding conditions | dressing conditions | grinding wheel | grinding conditions | dressing conditions | grinding wheel | grinding conditions | dressing conditions | grinding wheel | grinding conditions | dressing conditions | grinding wheel | grinding conditions | dressing conditions | | |
| GW | 1 | | | | | | | | | | | | | | | | | | | |
| | 2 | | | | | | | | | | | | | | | | | | | |
| | 3 | | | | | | | | | | | | | | | | | | | |
| | 4 | | | | | | | | | | | | | | | | | | | |
| GC | 5 | | | | | | | | | | | | | | | | | | | |
| | 6 | | | | | | | | | | | | | | | | | | | |
| | 7 | | | | | | | | | | | | | | | | | | | |
| | 8 | | | | | | | | | | | | | | | | | | | |
| DC | 9 | | | | | | | | | | | | | | | | | | | |
| | 10 | | | | | | | | | | | | | | | | | | | |
| | 11 | | | | | | | | | | | | | | | | | | | |
| | 12 | | | | | | | | | | | | | | | | | | | |

Fig. 7 Trouble-shooting Rules Refined by the Specialized Information on Burning

GW:Grinding Wheel, GC:Grinding Conditions, DC:Dressing Conditions

| no. | IF-CLAUSE | | | | | | | | | | | | | THEN-CLAUSE | | | | | | |
|-----|----------------|---------------------|---------------------|----------------|---------------------|---------------------|----------------|---------------------|---------------------|----------------|---------------------|---------------------|----------------|---------------------|---------------------|----------------|---------------------|---------------------|--|--|
| | grinding wheel | grinding conditions | dressing conditions | grinding wheel | grinding conditions | dressing conditions | grinding wheel | grinding conditions | dressing conditions | grinding wheel | grinding conditions | dressing conditions | grinding wheel | grinding conditions | dressing conditions | grinding wheel | grinding conditions | dressing conditions | | |
| GW | 1 | | | | | | | | | | | | | | | | | | | |
| | 2 | | | | | | | | | | | | | | | | | | | |
| | 3 | | | | | | | | | | | | | | | | | | | |
| | 4 | | | | | | | | | | | | | | | | | | | |
| GC | 5 | | | | | | | | | | | | | | | | | | | |
| | 6 | | | | | | | | | | | | | | | | | | | |
| | 7 | | | | | | | | | | | | | | | | | | | |
| | 8 | | | | | | | | | | | | | | | | | | | |
| DC | 9 | | | | | | | | | | | | | | | | | | | |
| | 10 | | | | | | | | | | | | | | | | | | | |
| | 11 | | | | | | | | | | | | | | | | | | | |
| | 12 | | | | | | | | | | | | | | | | | | | |

Fig. 8 Trouble-shooting Rules Refined by the Specialized Information on Chatter Vibration

위한 트러블 슈팅에 대한 기본적인 증상별 상태 구별을 위한 분류기준이다. Fig.7과 Fig.8은 연삭눌음(Burning)과 진동(Chatter Vibration)에 대하여 트러블 증상별에 따른 수정해야 할 Parameter를 정리한 룰 집합이다. 예를 들어, Fig.7의 룰 번호 201은 "IF 가공면의 간섭색이 Brown type이고, 연삭숫돌면에 눈매움>Loading)현상이 있고, 가공면이 거칠고, 결합도가 M(정상), 조직이 m(정상)이며 연삭액의 공급형태가 정상이라면, THEN 절입깊이(Depth of Cut)에 의한 연삭눌음으로 보이며, ⇒ 처리로서 "절입깊이(Depth of Cut)를 크게(NH) 줄이라"는 메시지가 전달된다.

4. 결 론

본 연구는 연삭가공 자동화 생산시스템의 구현을 위한 데이터 베이스를 룰 베이스 모델을 이용하여 설계· 구축하였다. 얻어진 결과를 요약하면 다음과 같다.

- (1) 룰 베이스 모델에 의한 연삭가공용 데이터 베이스는 다양한 숙련 경험자의 정성적 지식과 정량적 지식 등을 유기적으로 축적· 이용할 수 있도록 하였다.
- (2) 본 시스템에서 정의한 대역함수는 시스템의 관리 및 운용을 유기적으로 제어할 수 있어, 룰 베이스 모델에 의한 지식 표현의 효율성을 부여하였다.
- (3) 목적지향형 추론방식을 도입함으로써 다양한 입력조건에 대해, 본 시스템에서 정의한 경합해소 전략에 의해 효과적인 추론이 가능하게 되었다.

참고문헌

1. M.Birch and K.Whiteley: "An Object-Oriented Expert System Based on Pattern Recognition," Institute of Electrical and Electronics Engineers(IEEE), Transactions on System, Man, and Cybernetics, Jan. pp. 33-38, 1990.
2. R. Reiter; "A Logic for Default Reasoning, Artificial Intelligence," 13, pp. 81-87, 1980.
3. KBMS, Knowledge-base Management System(Software Tool for Developing Expert Systems), NTT co. in Japan, 1988.
4. Umeshwar Dayal, Hai-Yann Hwang; "View Definition and Generalization for Database Integration in a Multidatabase System," IEEE Transactions on Software Engineering, Vol. SE-10, No. 6, pp. 628-704, 1984.
5. P. Harker; "The Art and Science of Decision Making, The Analytic Hierarchy Process," pp. 3-8, 1989.