

# M&S와 AI간의 유기적 통합을 위한 시간기반 전문가 시스템 설계

신석훈 · 지승도<sup>†</sup>

## Time-based Expert System Design for Coherent Integration Between M&S and AI

Suk-Hoon Shin · Sung-Do Chi<sup>†</sup>

### ABSTRACT

Along with the development of M&S, modeling research utilizing AI technology is attracting attention because of the fact that the needs of fields including human decision making such as defense M&S are increased. Obviously AI is a way to solve complex problems. However, AI did not consider logical time such as input time and processing time required by M&S. Therefore, in this paper we proposed a “time-based expert system” which redesigned the representative AI technology rule-based expert system. It consists of a rule structure “IF-THEN-AFTER” and an inference engine, takes logical time into consideration. We also tried logical analysis using a simple example. As a result of the analysis, the proposal Time-based Expert System proved that the result changes according to the input time point and inference time.

**Key words** : Rule-based Expert System, AI Modeling, Time-based Inference Algorithm

### 요약

M&S의 발전과 더불어 국방 M&S 등 인간의 의사결정을 포함하는 분야의 요구가 증대되는 현실에서 AI 기술을 활용한 모델링 연구가 각광받고 있다. AI는 복잡한 문제 해결을 위한 방법임은 분명하나 M&S에서 요구되는 입력시점과 처리시간 등의 논리적 시간을 고려하지 않았다. 따라서 본 논문에서는 대표적인 AI 기술인 규칙기반 전문가시스템을 논리적 시간을 고려한 규칙구조 “IF-THEN-AFTER”와 추론엔진으로 재설계한 시간기반의 전문가 시스템을 제안하고, 기존의 규칙기반 전문가 시스템과의 차이를 설명하기 위한 간단한 예제를 들어 논리적 분석을 시도하였다. 그 결과로 제안하는 시간기반의 전문가 시스템 모델은 일반적인 규칙기반 전문가시스템과 다르게 입력시점과 추론시간에 따라 다른 결과를 보임을 알 수 있으며, 이는 M&S에서 요구되는 논리적 시간을 고려한 AI의 문제해결이 가능함을 의미한다.

**주요어** : 규칙기반전문가시스템, 인공지능 모델링, 시간기반 추론 알고리즘

## 1. 서론

최근 M&S(Modeling & Simulation) 환경은 고도로 복잡한 대상시스템을 모델링하고 시스템을 구성하는 개체간의 관계들과 이로 인해 발생하는 다양한 상황을 모

의하는 수준에 이르렀다. 과거 M&S의 대상은 건물, 기계, 전자제품 등의 인공물들로 대상의 입/출력의 분석이 주된 주제가였다면, 현재 이슈가 되는 대상은 공장, 집단, 사회 등의 기존 주제와 상호작용하는 외부환경, 인간, web 등의 관계의 분석을 요구하고 있다<sup>[1]</sup>.

많은 연구에서 M&S의 복잡성을 해결하기 위한 방법으로 인공지능(AI: Artificial Intelligent) 기술을 제안하였다. 특히 인간의 의사결정 과정이 포함되는 문제 영역에서 AI는 그 가치를 증명하고 있으며, 국방M&S(DM&S: Defense M&S)가 대표적이다.

인간의 의사결정<sup>[2,3]</sup>을 모델링 한다는 것은 인간의 지

\* 본 연구는 한국항공대학교 교내 학술연구비(no. 2012-01-009)의 지원으로 수행되었습니다.

Received: 25 January 2017, Revised: 18 May 2017,  
Accepted: 22 May 2017

<sup>†</sup> Corresponding Author: Sung-Do Chi

E-mail: sdchi@kau.ac.kr

Korea Aerospace University, Dept. of Software Engineering

식과 이에 기반을 둔 사고의 과정을 시간의 흐름에 따른 상태의 변화와 해당 상태에서의 입력 대한 출력을 모델링 하는 것과 크게 다르지 않다. 때문에 의사결정의 시점과 소요시간, 그에 따른 상태의 변화가 고려되어야 함은 당연하다. 예로 DM&S의 경우 적의 위협상황에서 지휘관의 전술 선택 및 명령의 타이밍에 따라 전술의 효과도가 달라진다<sup>4)</sup>.

그러나 인공지능 기술은 기본적으로 문제해결을 목적인 알고리즘을 기반으로 한다. 즉 시점과 소요시간을 고려하지 않는다. 단지 입력된 정보를 기반으로 알고리즘을 이용한 추론결과 만을 구한다<sup>5)</sup>.

본 논문은 AI의 기법들 중 규칙기반 전문가 시스템(Rule-based Expert System)에 대하여 입력 시점과 추론 소요시간을 고려한 시간기반 규칙구조와 이에 기반을 둔 시간기반 추론알고리즘을 제안, 그 차이점에 대하여 설명한다. 또한 제안한 규칙구조와 추론알고리즘을 구현한 시뮬레이션 모델을 제안하고 사례 연구를 통해 검증한다.

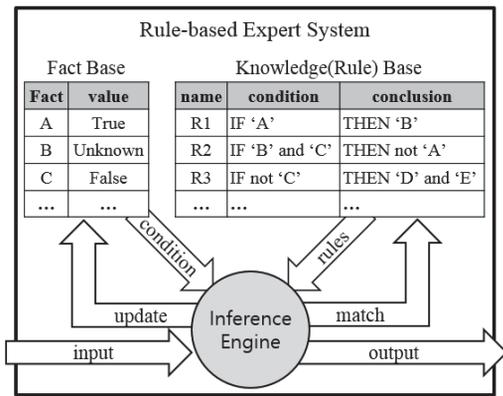


Fig. 1. 규칙기반 전문가 시스템

## 2. IF-THEN-AFTER 규칙구조와 추론알고리즘

규칙기반 전문가 시스템은 입력 또는 추론을 통해 현재 알려진 사실(Fact)들의 집합(Fact Base)에 대하여 지식베이스(Knowledge Base)의 규칙들 중 조건부(IF)의 조건(condition)이 만족하는 규칙을 탐색(match)하여 해당하는 규칙의 결론부(THEN)의 결론(conclusion)을 새로운 사실로 갱신(update)하는 연쇄과정을 통해 목표하는 결론을 추론해내도록 고안된 AI 시스템이다<sup>6)</sup>(Fig. 1 참조). 때문에 새로운 사실이 입력되면 규칙을 검색하여 결론을 도출하도록 추론엔진의 알고리즘이 구성되어 있

으며, 따라서 M&S에서 요구되는 시간을 고려한 추론과정이 불가능하다. 다시 말해 입력된 정보가 추론을 거쳐 출력되는 사이의 시간은 존재 하지 않는다. 때문에 아무리 찰나의 시간일 지라도 시간차를 두고 입력된다면, 앞선 입력에 대한 추론 이후에 뒤의 입력에 대한 추론을 처리한다.

그래서 본 논문에서는 시간 요소를 포함한 규칙구조인 IF-THEN-AFTER 규칙구조를 제안한다(수식1).

**Rule = IF<condition> THEN<conclusion> AFTER<time>**

### 수식 1. IF-THEN-AFTER 규칙구조

```

algorithm Time-based inference
variables:
list<rule>      RULEBASE //rule-base
list<fact>      FACTBASE //fact-base
fact           F         //external fact
list<fact,time> HOLDLIST //holding fact list to be fired
time           Tcur      //current time
time           Th-next    //minimum time among HOLDLIST
/*----- forward chaining -----*/
when receive Fext
  update FACTBASE Fext withtimeTcur
  for each rule R in RULEBASE
    if is matched condition of R with FACTBASE
      /*----- time management -----*/
      for each fact Fe in conclusion of R
        if is exist Fh(= F)inHOLDLIST
          update time of Fh = min(Tfh | Tfc)
        else
          add Fe to HOLDLIST
      /*----- time management end -----*/
      /*----- forward chaining end -----*/
      /*----- global clock management -----*/
      when Th-next - Tcur = 0 //is over holding time some Fh(inHOLDLIST)
        update FACTBASE Fh
        for each rule R .....
          update HOLDLIST
      when no external fact until next scheduled time
        Tcur = Th-next //advance the clock to next scheduled time
      /*----- global clock management end -----*/
end algorithm
    
```

### 알고리즘1. 시간 기반의 규칙 전방 추론 알고리즘

제안한 규칙구조는 일반적인 IF-THEN 규칙에 AFTER 항목을 추가하고, 시간 값을 부여한다. 본 IF-THEN-AFTER규칙 구조로 작성된 규칙은 “<condition>이 만족할 경우 <time>의 시간이 지난 후 <conclusion>이 만족됨”으로 해석된다.

또한 제안하는 규칙 구조의 해석을 통해 AFTER <time>항목을 고려한 추론을 수행하는 시간기반 전방 추론 알고리즘(Time-based Forward Chaining Inferencing Algorithm, 알고리즘1 참조)을 제안한다. 제안하는 시간기반 알고리즘은 규칙 리스트에서 현재 <condition>을 만

족하는 규칙 ‘R’을 임시로 선택하고, ‘R’의 <conclusion>을 대기사실목록(HOLDLIST)에 저장한다. 이후 전역시간(global clock)을 기준으로 ‘R’의 <time>이 지나면 HOLDLIST의 <conclusion>을 업데이트 한다. 만일 <time>이 지나기 전에 <condition>의 변화가 발생하면 다시 규칙을 검색하는 과정을 반복한다. 알고리즘1은 제안하는 시간기반 전방 추론 알고리즘의 의사코드이다.

한편 T.G Kim과 B.P. Zeigler는 규칙기반 전문가 시스템에 시간 개념을 적용하여 효과적인 시스템 제어 스케줄링이 가능함을 보였다<sup>7)</sup>. Kim의 연구에서는 규칙을 통해 수행하는 행위(activity)의 시작시간, 종료시간, 다음 실행시간, 실행기간의 요소를 포함하였다. 따라서 규칙기반의 전문가 시스템을 이용하여 즉각 적인 입력에 대한 추론 결과가 아니라, 시기적절하고 지속적인 제어가 가능함을 보였다. 그러나 제어 스케줄링 관점에서 시기와 지속시간을 고려하였기 때문에 본 논문에서 제안하는 의사결정에 소요되는 추론시간에 대한 고려는 포함하지 않는 차이가 있다.

### 3. 시간기반 전문가 시스템 설계

앞서 정의한 IF-THEN-AFTER 규칙 구조와 시간기반의 추론 알고리즘을 적용한 시간기반 전문가 시스템의 설계 및 모델 구현은 이산사건형식론<sup>9,10)</sup>(DEVS: Discrete Event System Specification)기반 M&S 환경에서 진행하였다. DEVS M&S 환경에서 모델은 입력 이벤트 집합(X), 출력 이벤트 집합(Y), 상태의 집합(S), 입력 이벤트에 의한 상태 전이 함수( $\delta_{Ext}$ ), 상태의 지속시간의 종료로 인한 내부 상태 전이 함수( $\delta_{Int}$ ), 상태 전이에 의한 출력 함수( $\lambda$ ), 상태 전이에 따른 시간 진행 함수( $\alpha$ )로 정의한다.

이를 위하여 DEVS의 모델의 정의를 상속하여 시간기반 추론엔진의 알고리즘을 구성하였고, 내부 변수로 IF-THEN-AFTER 규칙구조를 반영한 지식베이스를 포함한 DEVS 서브모델로 구현하였다(Fig. 2). Fig. 2의 DEVS로 구현된 시간기반 전문가 시스템 모델은 UPDATE, MATCH, FIRE의 상태를 갖는다. 초기 상태는 UPDATE이며, 외부에서 사실이 입력( $\delta_{Ext}$ )되거나 추론을 통해 새로운 사실이 알려지면( $\delta_{Int}$ ) MATCH 상태로 천이한다. MATCH 상태에서는 현재 알려진 사실들의 조합을 기준으로 조건이 만족하는 규칙을 탐색하여 규칙의 결론을 HOLDLIST에 등록하고, HOLDLIST의 결론들의 remaining time을 FIRE 상태의 지속시간으로 설정하고 FIRE 상태

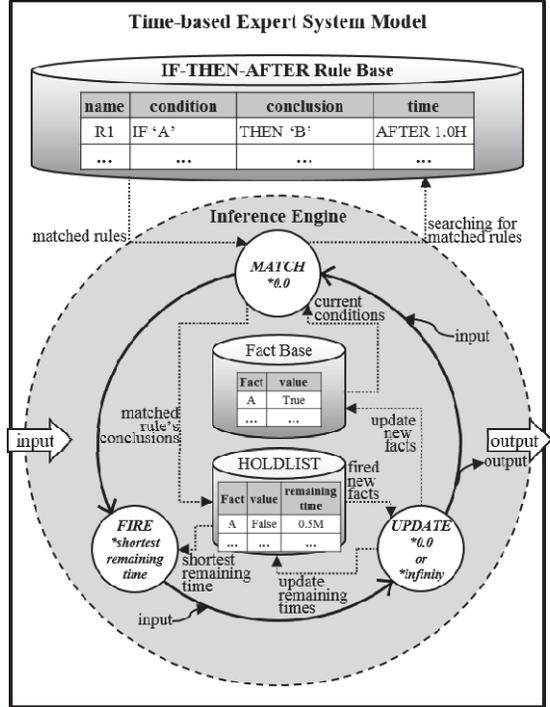


Fig. 2. DEVS를 이용한 시간기반 전문가 시스템의 시뮬레이션 모델 설계

로 천이 ( $\delta_{Int}$ )한다. FIRE 상태에서는 지속시간이 종료될 때 까지 외부 입력이 없으면 HOLDLIST의 가장 작은 remaining time의 결론을 FactBase에 갱신하고 UPDATE 상태로 천이( $\delta_{Int}$ )한다. 따라서 FIRE 상태의 지속시간이 현재의 알려진 사실들의 집합, 즉 조건에 따라 탐색된 규칙들에 의해 변경된다. 한편 FIRE 상태의 지속시간이 종료되기 전에 외부 사실이 입력되면( $\delta_{Ext}$ ) 즉시 진행된 시간을 HOLDLIST에 반영하고, UPDATE 상태로 천이한다. 상기한 바와 같은 상태 천이 알고리즘을 통해 시간기반 전문가 시스템 모델이 작동한다.

한편 모델링하는 관점에서 시간기반 전문가 시스템 모델은 모델링 대상 규칙들을 입력하는 것만으로 모델링 과정을 마칠 수 있다. 그 이유는 각 모델의 입/출력은 모든 대상에서 동일하게 구현된 시간기반 추론엔진이 탑재된 규칙에 따라 추론을 통해 결정하기 때문이다. 따라서 시간기반 전문가 시스템 모델을 상속받아 제안하는 형식의 규칙 입력만으로 의사결정 과정에 대한 DEVS 모델링이 가능하다. 또한 ‘규칙’의 일반화된 모델링 방법은 기존의 프로그래밍 코드 작성 방법에 비해 GUI 기반의 모델링 도구 개발, 유지/보수 및 재사용 관점에서 이점으로

작용한다.

#### 4. 사례연구

일반적인 규칙기반 전문가 시스템과 본 논문에서 제안하는 시간기반의 규칙기반 전문가 시스템의 차이에 대하여 규칙기반 전문가 시스템을 활용한 산불 발생 예측의 예를 통해 논리적 비교 실험을 진행 하였다.

Fig. 3은 숲(Forest)에서 강우(raining/stop raining)와 낙뢰(lightning)에 의해 산불(연기, smoke)이 발생하는 조건을 상태와 이벤트 입력의 관계로 표현한 예이다. Fig. 2의 상태와 이벤트 입력의 관계를 설명하면 다음과 같다.

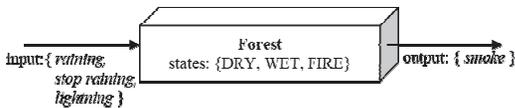


Fig. 1. 숲 시스템 - 강우/변개에 따른 산불발생 여부<sup>[8]</sup>

- 마른 상태의 숲은 강우가 지속되면 젖은 상태가 된다.
- 젖은 상태의 숲은 강우가 멈추면 천천히 마른 상태가 된다.
- 젖지 않은 숲에 낙뢰 발생하면 산불 상태가 되고, 연기가 발생한다.
- 산불 상태의 숲은 강우가 지속되면 마른 상태를 거쳐 젖은 상태가 된다.
- 강우가 지속되는 젖은 상태의 숲은 낙뢰가 발생해도 산불이 발생하지 않는다.
- 젖은 상태에서 강우가 멈춘 숲에 낙뢰가 발생하면 마른상태를 거쳐 산불 상태가 된다.

Fig. 3의 이벤트와 상태변화의 관계를 바탕으로 산불 발생을 예측하는 시스템을 가정할 때의 일반적인 규칙기반 전문가 시스템과 제안하는 IF-THEN-AFTER 규칙기반 전문가 시스템의 규칙을 각각 설계 하였다(Table 1). 일반적인 규칙기반 전문가 시스템을 위한 규칙은 Table 1의 (a)와 같다. 같은 내용에 대하여 우리가 제안한 IF-THEN-AFTER 규칙은 Table 1의 (b)와 같다. 이때 R1부터 R6까지는 숲의 입/출력 및 상태변화에 관한 규칙이고, R7부터 R11까지는 정보간의 반의관계에 대한 규칙이며, R12는 낙뢰 발생이 순간적임을 나타내기 위한 규칙이다.

Table 2는 Table 1의 규칙을 각각 탑재한 일반적인 규칙기반 전문가 시스템(Table 2의 왼쪽)과 IF-THEN-AFTER

규칙기반 전문가 시스템(Table 2의 오른쪽)에 대하여 동일한 조건에서의 추론결과를 나타낸 것이다. Table에서 'matched rules'는 해당시각에의 조건을 만족하는 규칙을 의미하고, value는 Fact의 값(True/False/Unknown)을 의미하며, HOLDLIST의 remaining time은 IF-THEN-AFTER 규칙기반 전문가 시스템에서 탐색된 규칙의 결론부가 value의 값으로 알려져 Fact Base에 갱신되기 까지 남은 시간을 의미한다.

Table 1. 숲 시스템을 기준으로 작성된 전문가 시스템의 규칙 예제

(a) <IF-THEN> rule structure

name	condition	conclusion
R1	IF 'DRY' and 'raining'	THEN 'WET'
R2	IF 'WET' and 'stop raining'	THEN 'DRY'
R3	IF 'DRY' and 'lightning'	THEN 'FIRE' and 'smoke'
R4	IF 'FIRE' and 'raining'	THEN 'DRY'
R5	IF 'WET' and 'lightning'	THEN 'WET'
R6	IF 'stop raining' and 'lightning'	THEN 'FIRE' and 'smoke'
R7	IF 'DRY'	THEN not ('WET' and 'FIRE')
R8	IF 'WET'	THEN not ('DRY' and 'FIRE')
R9	IF 'FIRE'	THEN not ('DRY' and 'WET')
R10	IF 'stop raining'	THEN not 'raining'
R11	IF 'raining'	THEN not 'stop raining'
R12	IF 'lightning'	THEN not 'lightning'

(b) <IF-THEN-AFTER> rule structure

name	condition	conclusion	time
R1	IF 'DRY' and 'raining'	THEN 'WET'	1H
R2	IF 'WET' and 'stop raining'	THEN 'DRY'	24H
R3	IF 'DRY' and 'lightning'	THEN 'FIRE' and 'smoke'	0.1H
R4	IF 'FIRE' and 'raining'	THEN 'DRY'	1H
R5	IF 'WET' and 'lightning'	THEN 'WET'	0H
R6	IF 'stop raining' and 'lightning'	THEN 'FIRE' and 'smoke'	0.2H
R7	IF 'DRY'	THEN not ('WET' and 'FIRE')	0H
R8	IF 'WET'	THEN not ('DRY' and 'FIRE')	0H
R9	IF 'FIRE'	THEN not ('DRY' and 'WET')	0H
R10	IF 'stop raining'	THEN not 'raining'	0H
R11	IF 'raining'	THEN not 'stop raining'	0H
R12	IF 'lightning'	THEN not 'lightning'	0.1H

Table 2의 Simulation Time 0H는 초기 상태를 의미한다. 숲은 상태는 'DRY'이고, 연기(smoke)는 발생하지 않았다(False). 초기상태를 기반으로 조건이 만족된 규칙은 R7이며, 'DRY'상태에 대한 반대 상태('WET', 'FIRE')를 정의하기 위한 규칙이다. 따라서 0H에서 Fact Base는 'DRY' = True, 'WET' = False, 'FIRE' = False임을 알 수 있다.

Table 2의 1H는 입력으로 강우('raining')가 발생한

**Table 2.** 일반적인 규칙기반 전문가 시스템과 IF-THEN-AFTER 규칙기반 전문가 시스템의 추론 결과 비교  
 <강우시작 0.5시간 후 낙뢰 발생 조건>

Simulation Time	IF-THEN Rule-based Expert System		IF-THEN-AFTER Rule-based Expert System																																	
	input	matched rules	Fact Base	HOLDLIST	output																															
0H	input	-																																		
	matched rules	R7																																		
	Fact Base		<table border="1"> <thead> <tr><th>Fact</th><th>value</th></tr> </thead> <tbody> <tr><td>DRY</td><td>True</td></tr> <tr><td>smoke</td><td>False</td></tr> <tr><td>WET</td><td>False</td></tr> <tr><td>FIRE</td><td>False</td></tr> </tbody> </table>	Fact	value	DRY	True	smoke	False	WET	False	FIRE	False	<table border="1"> <thead> <tr><th>Fact</th><th>value</th><th>remaining time</th></tr> </thead> <tbody> <tr><td>-</td><td>-</td><td>-</td></tr> </tbody> </table>	Fact	value	remaining time	-	-	-																
	Fact	value																																		
DRY	True																																			
smoke	False																																			
WET	False																																			
FIRE	False																																			
Fact	value	remaining time																																		
-	-	-																																		
output	not smoke			not smoke																																
1H	input	raining																																		
	matched rules	R1, R8, R11																																		
	Fact Base		<table border="1"> <thead> <tr><th>Fact</th><th>value</th></tr> </thead> <tbody> <tr><td>DRY</td><td>False</td></tr> <tr><td>smoke</td><td>False</td></tr> <tr><td>WET</td><td>True</td></tr> <tr><td>FIRE</td><td>False</td></tr> <tr><td>raining</td><td>True</td></tr> <tr><td>stop raining</td><td>False</td></tr> </tbody> </table>	Fact	value	DRY	False	smoke	False	WET	True	FIRE	False	raining	True	stop raining	False	<table border="1"> <thead> <tr><th>Fact</th><th>value</th><th>remaining time</th></tr> </thead> <tbody> <tr><td>WET</td><td>True</td><td>1.0H</td></tr> </tbody> </table>	Fact	value	remaining time	WET	True	1.0H												
	Fact	value																																		
DRY	False																																			
smoke	False																																			
WET	True																																			
FIRE	False																																			
raining	True																																			
stop raining	False																																			
Fact	value	remaining time																																		
WET	True	1.0H																																		
output	not smoke			not smoke																																
1.5H	input	lightning																																		
	matched rules	R5, R12																																		
	Fact Base		<table border="1"> <thead> <tr><th>Fact</th><th>value</th></tr> </thead> <tbody> <tr><td>DRY</td><td>False</td></tr> <tr><td>smoke</td><td>False</td></tr> <tr><td>WET</td><td>True</td></tr> <tr><td>FIRE</td><td>False</td></tr> <tr><td>raining</td><td>True</td></tr> <tr><td>stop raining</td><td>False</td></tr> <tr><td>lightning</td><td>False</td></tr> </tbody> </table>	Fact	value	DRY	False	smoke	False	WET	True	FIRE	False	raining	True	stop raining	False	lightning	False	<table border="1"> <thead> <tr><th>Fact</th><th>value</th><th>remaining time</th></tr> </thead> <tbody> <tr><td>WET</td><td>True</td><td>0.5H</td></tr> <tr><td>FIRE</td><td>True</td><td>0.1H</td></tr> <tr><td>smoke</td><td>True</td><td>0.1H</td></tr> <tr><td>lightning</td><td>False</td><td>0.1H</td></tr> </tbody> </table>	Fact	value	remaining time	WET	True	0.5H	FIRE	True	0.1H	smoke	True	0.1H	lightning	False	0.1H	
	Fact	value																																		
DRY	False																																			
smoke	False																																			
WET	True																																			
FIRE	False																																			
raining	True																																			
stop raining	False																																			
lightning	False																																			
Fact	value	remaining time																																		
WET	True	0.5H																																		
FIRE	True	0.1H																																		
smoke	True	0.1H																																		
lightning	False	0.1H																																		
output	not smoke			not smoke																																
1.6H	input	-																																		
	matched rules	-																																		
	Fact Base		<table border="1"> <thead> <tr><th>Fact</th><th>value</th></tr> </thead> <tbody> <tr><td>DRY</td><td>False</td></tr> <tr><td>smoke</td><td>False</td></tr> <tr><td>WET</td><td>True</td></tr> <tr><td>FIRE</td><td>False</td></tr> <tr><td>raining</td><td>True</td></tr> <tr><td>stop raining</td><td>False</td></tr> <tr><td>lightning</td><td>False</td></tr> </tbody> </table>	Fact	value	DRY	False	smoke	False	WET	True	FIRE	False	raining	True	stop raining	False	lightning	False	<table border="1"> <thead> <tr><th>Fact</th><th>value</th><th>remaining time</th></tr> </thead> <tbody> <tr><td>DRY</td><td>True</td><td>1.0H</td></tr> </tbody> </table>	Fact	value	remaining time	DRY	True	1.0H										
	Fact	value																																		
DRY	False																																			
smoke	False																																			
WET	True																																			
FIRE	False																																			
raining	True																																			
stop raining	False																																			
lightning	False																																			
Fact	value	remaining time																																		
DRY	True	1.0H																																		
output	not smoke			smoke																																

상황이다. 우선 일반적인 전문가 시스템의 경우를 보면, 규칙 R1의 조건이 만족되어 Fact Base의 ‘WET’이 True가 되고, ‘WET’ = True로 인해 규칙 R8이 연쇄적으로 조건이 만족되어 ‘DRY’=False가 된다. 또한 ‘raining’으로 인해 R11의 조건이 만족되어 ‘stop raining’ = False가 된다.

반면 제안하는 시간기반 전문가 시스템의 경우 같은 시점(1H)에서 규칙 R1의 조건이 만족되지만 해당 결론인 ‘WET’=True가 즉시 Fact Base에 갱신되지 않고, R1의 AFTER<time>항목의 값 1H만큼의 remaining time을 가지고 HOLDLIST에 등록된다. 따라서 현재 ‘DRY’=True는 상황에 대한 변화가 없다. 때문에 규칙 R8의 조건도 만족하지 않는다.

Table 2의 1.5H는 강우가 0.5시간 지속되는 상황에서 낙뢰(‘lightning’)이 발생한 경우이다. 먼저 일반적인 전문가 시스템의 경우를 보면, 규칙 R5의 조건이 만족되어 ‘WET’=True의 상황이 유지되고, R12의 조건 만족에 의해 ‘lightning’=False가 된다.

반면 제안하는 시간기반 전문가 시스템의 경우 우선 앞선 1H에 비하여 0.5H의 시간이 진행되었으므로 HOLDLIST의 ‘WET’=True의 remaining time값을 0.5H만큼 감해준다. 또한 1.5H현재 ‘DRY’=True가 유지되고 있기 때문에 규칙 R3의 조건이 만족되고, R3의 결론 ‘FIRE’와 ‘smoke’가 HOLDLIST에 R3의 time=0.1H을 가지고 등록된다. 마지막으로 규칙 R12의 조건(‘lightning’)으로 인해 ‘lightning’=False도 0.1H의 값으로 HOLDLIST에 등록된다.

마지막으로 Table 2의 1.6H를 보면, 일반적인 전문가 시스템의 경우 1.5H 시점에 비교하여 입력 및 추론결과에 변화가 없음을 알 수 있다. 한편 같은 시점의 시간기반 전문가 시스템의 경우 HOLDLIST의 ‘FIRE’=True, ‘smoke’=True, ‘lightning’=False의 remaining time 값이 ‘0’이 되어 FactBase에 반영되었고, 이로인해 연쇄적으로 규칙 R4 조건(‘FIRE’=True and ‘raining’=True)이 만족되어 결론(‘DRY’=True)이 HOLDLIST에 반영되었으며, 최종적으로 출력또한 not smoke에서 smoke(산불발생)으로 변화하였다.

Table 2의 일반적인 전문가 시스템의 시점별 추론 결과는 입력 순서에 따른 결과 도출만이 고려되는 AI의 특성과 일치한다. 만일 ‘lightning’이벤트의 입력 시점을 2.0H 이후로 가정할 지라도 일반적인 전문가 시스템의 경우 Table 2와 동일한 결과를 보인다. 그러나 우리가 제안하는 시간기반 전문가 시스템을 적용할 경우 ‘WET’

상태가 되어 Table 2와 다르게 not smoke의 결과를 나타낸다. 다시 말해 동일한 입력 순서일지라도 강우의 지속 시간과 낙뢰의 발생 시점에 따라 다른 예측 결과를 보인다. 이는 입력 순서를 고려함은 물론 입력의 시점과 내부 상태 즉 <condition>의 지속 시간<time>이 반영된 결과이다.

## 5. 결론 및 향후연구과제

본 논문에서 제안한 시간기반 규칙구조와 이를 이용한 추론 알고리즘은 입력시점과 시간 진행에 따라 달라지는 출력을 표현할 수 있다. 이를 기반 하여 시뮬레이션을 위해 설계된 시간기반 전문가 시스템 모델은 추론을 통한 결론도출은 물론 앞서 제안한 시간을 고려한 추론이 가능하게 한다.

사고하는 시간과 입력되는 사건의 인지를 통해 결과가 달라질 수 있는 인간의 의사결정을 모델링하는데 있어 본 논문에서 제안하는 규칙구조와 알고리즘을 기반한 시간기반 전문가 시스템의 설계와 추론과정의 적합성을 산불 예측에 대한 논리적 시뮬레이션 사례를 통해 검증하였다.

또한 본 논문에서 제안한 시간기반 전문가 시스템의 모델링 방법은 DEVS 형식론 및 프로그래밍 기술 의존을 없애고 일반적인 규칙작성을 통해 모델링할 수 있는 장점이 있으며, 차후 모델링 도구 개발 및 이를 이용한 생산성 향상의 이점이 예상된다.

향후 규칙의 조건과 결론을 구성하는 사실 정보의 가치에 대한 시간에 따른 평가를 반영하는 연구를 진행할 계획이다.

## References

1. Li, Bo Hu. “Some focusing points in development of modern modeling and simulation technology.” *Systems Modeling and Simulation: Theory and Applications*. Springer Berlin Heidelberg, 2005. 12-22.
2. Svenson, Ola, ed. *Time pressure and stress in human judgment and decision making*. Springer, 1993.
3. Hogarth, Robin M. *Decision time as a function of task complexity*. Springer Netherlands, 1975.
4. Fewell, M. P., and Mark G. Hazen. *Network-Centric Warfare-Its Nature and Modelling*. No. DSTO-RR-

0262. DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION SALISBURY (AUSTRALIA) SYSTEMS SCIENCES LAB, 2003.
5. Erdem, Ali, W. Lewis Johnson, and Stacy Marsella. "Task oriented software understanding." Automated Software Engineering, 1998.
  6. Joseph, Giarratano, and Riley Gary. Expert systems principles and programming. PWS Publishing Company, 1998.
  7. Tag Gon Kim and Bernard P. Zeigler, "An AI-Based, Distributed Environmental Control System for Self- Sustaining Habitats," Journal of Artificial Intelligence in Engineering, Vol. 5, No. 1, pp. 33-42, 1990.
  8. Zeigler, Bernard P., Herbert Praehofer, and Tag Gon Kim. Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems. Academic press, 2000.
  9. Zeigler, Bernard P. "Object-Oriented Simulation with Hierarchical." Modular Models-Intelligent Agents and Endomorphic Systems Academic Press, Boston (1990).



**신 석 훈** (ev4shin@naver.com)

2009 한국항공대학교 컴퓨터공학 학사  
 2011 한국항공대학교 컴퓨터공학 전공 석사  
 2011~ 현재 한국항공대학교 컴퓨터공학 전공 박사과정  
 2014~ 2017 한국항공대학교 부설 IT 연구소 연구원  
 2017~ 현재 (주)하이브로 게임사업본부 테크니컬 디렉터

관심분야 : 자율에이전트, 국방 M&S, CGF/SAF 모델링, 자율 무인 시스템



**지 승 도** (sdchi@kau.ac.kr)

1982 연세대학교 전기공학과 학사  
 1984 연세대학교 전기공학 전공 석사  
 1991 아리조나대학교 전기·전산공학 박사  
 1992~ 현재 한국항공대학교 소프트웨어학과 교수

관심분야 : 자율 무인 시스템, 이산사건 시뮬레이션, 국방 M&S