

# 시뮬레이션을 통한 지식의 자동 획득

## Simulation based Automatic Knowledge Acquisition

이 강 선\*, 김 명 희\*

Kang-Sun Lee , Myoung-Hee Kim

### Abstract

본 연구에서는 시뮬레이션을 통한 지식의 자동 획득(Simulation-Based Automatic Knowledge Acquisition) 가능성을 제시한다. 이를 위한 작업 단계는 다음과 같다. 첫째, 지식 제공자에 의한 대상(domain) 관련 초기 인과 관계 정보 입력 단계, 둘째 경험 베이스 탐색에 의한 확장된 정보 생성 단계, 세번째로 생성되어진 정보를 사용하여 대상 반영 모델을 구축하는 단계, 네번째로 구축된 모델을 시뮬레이션하고 수행 결과의 분석을 통해 새로운 지식을 획득하는 단계로 구성된다.

제안된 지식 획득 방법은, 대상에 관계된 개념과 개념들의 인과 관계를 바탕으로 모델을 자동 생성하여 이를 지식 획득 표현틀로 이용하는 유연한(flexible) 구조를 사용하였고, 또한 생성된 모델의 시뮬레이션 결과를 분석함에 의해 새로운 지식을 획득함으로써 획득된 지식이 동적 세계를 잘 반영할 수 있도록 하였다.

## 1.서론

대상에 대한 전문 지식은 인간 전문가로부터 얻어진다. 지식 획득(Knowledge Acquisition)이란, 어떤 대상에 대한 지식을 생성해내고 이것을 컴퓨터 프로그램상에서 실행 가능한 형태로 변화시키는 과정이다.[1] 그러나 대상이 복잡해 질수록 인간 전문가로부터 관련된 모든 지식을 정확하게 얻어내는 것은 어려운 작업이다. 이러한 지식 획득의 어려움을 부분 혹은 완전 자동화된 지식 획득 시스템을 통해 해결하려는 많은 연구가 있어 왔다. 지능적 프로그램에 의해 생성된 대화(dialog)등을 통해 전문가의 지식을 얻어내는 지식 추출(Knowledge Elicitation) 부

류의 방법이나, 대상에 대한 예를 귀납적으로 학습하여 지식을 생성하는 기계 귀납법(Machine Induction) 부류의 방법등이 그 예이다. [1,2]

지식 획득을 위한 많은 연구들이 만족할만한 성과를 거두지 못한 주요 원인으로 아래와 같은점이 지적되었다 [3,4,5]. 첫째, 지식 획득 시스템은 지식 제공자의 초점을 대상 자체가 아닌 특정 형식의 표현 구조나 도구로 치우치게 하였다는 지적이다. 이런 부적당한 초점으로 지식 제공자는 잘못된 지식을 제공하기도 하며, 주요 지식을 생략하는 오류를 보이기도 한다.[4] 둘째, 시간을 배제한 정적 관계(static relationship)에 의해서만 지식을 획득함으로써 시간흐름에 따라 변화하는 면이 생략된다는 것이다.[5]

\* 이화여자대학교 전자계산학과

따라서 획득된 지식이 대상에 대한 완전한 표현인가를 확신하기 어렵다.

본 연구에서는 시뮬레이션을 통한 자동 지식 획득 방법을 제시한다. 이 방법은 대상에 관련된 개념과 개념들의 인과 관계를 통하여 이를 반영하는 시뮬레이션 모델을 내부적으로 자동 생성하고, 생성된 모델로부터 여러 세부적인 지식을 이끌어 내는 방법으로 시뮬레이션 실험을 수행한다. 본 연구의 시뮬레이션을 통한 자동 지식 획득 방법은 아래와 같은 시나리오를 통해 설명될 수 있다.

의사가 어느 환자에 대한 지식을 얻는 과정을 생각해 보자. 의사가 환자의 상태에 관한 지식을 얻고자 할 때 그는 환자에 대한 “환자 모델”을 머리속에서 만들게 된다. 이 모델에서 묘사하는 것은, 인간 신체 구조와 구성 요소들이 어떻게 서로 영향을 미치는가를 표현한 인과 관계 모델이다. 의사는 환자에 관한 지식을 얻기 위해 머리속에 작성된 기본적인 모델(mental model)을 이용하여, “이런 경우라면 과연 어떠한 일이 생겨날까”를 실험해 보게 된다. 즉 정상적 상황이나, 병적 상황등 여러 상황에서 이 모델의 변화 내용을 시뮬레이션하게 되고 환자에 대한 지식은 이 결과로부터 획득 되어 진다. [7]

위의 시나리오에서 시뮬레이션 모델은 지식 획득을 위한 도구(device)<sup>1)</sup>로서 사용되며, 실험(혹은 시뮬레이션)은 모델의 시간에 따른 세부적인 변화내용을 이끌어 내기 위한 방법으로 사용된다.

모델과 시뮬레이션을 통해 지식을 획득하려는 많은 연구가 시뮬레이션 영역과 인공지능 분야에서 있어 왔다. Kuiper는 그의 질적 시뮬레이션(qualitative simulation) 이론을 통해 신장염에 대한 지식을 획득하였다. 그는 먼저 신장염에 관계된 객체(object)와 이들의 원인 결과 관계 정보를 바탕으로 질적 모델을 구성하였다. 그 다음으로 구성된 질적 모델을 여러 상황에서 질적 시뮬레이션하여, 이 시뮬레이션 결과를 통해 신장염에 관한 지식을 획득하였다.[7] 그러나 질적 시뮬레이션에서는 불확실한 초기 지식들을 바로 모델화 하는데 사용하므로 복잡한 현상에 대한 정확한 대답을 주기란 어렵다.[6,8,9] 따라서 처음 정보로부터 바로 모델을 생성하는 것이 아니라 이를 결과

지식으로 이끌어 가기 위한 메타 지식(meta knowledge)이나 초기 추정(initial guesses)으로 보는 것이 유리하다.[6,9] Fishwick은 시스템에 대한 초기 불완전한 인과 관계 지식을 메타 지식으로 보고, 이를 바탕으로 더 정제된 전문 지식을 얻는데 있어 현상에 대한 전문 지식을 시뮬레이션의 고유 모델에서 구하였다. 또한 시뮬레이션 모델에 지식 제공자가 가진 불확실한 정보가 포함될 수 있는 방법인 퍼지 시뮬레이션(fuzzy simulation)을 제시하였다. [6] 그러나 지식을 생성하기 위한 도구로서의 시뮬레이션 모델이 지식 제공자들에 의해 별도로 제공되어야 하기 때문에, 기존 지식 도구가 갖는 문제점인 지식 제공자의 초점이 대상자체가 아닌 표현 구조나 도구로 치우치는 현상을 해결하지 못하였다. 한편 Pradip Dey는 시뮬레이션의 동적 모델링 개념이 지식 획득을 위한 효과적인 방법이 될 수 있음을 지적하면서, 시뮬레이션 환경(simulation environment)을 통해 지식 획득에 필요한 모델을 자동 생성하는 지식 획득 시스템 구조를 제시한 바 있다.[3]

본 연구에서는 지식 획득을 위해 다음 4단계를 거친다.

- 1) 지식 제공자에 의해 대상에 관한 초기 인과 관계 정보를 입력받는다.
- 2) 지식 제공자에 의해 입력된 정보를 일정한 프레임에 따라 구조화시켜 경험 베이스를 구성하고, 구축된 경험 베이스를 탐색하여 대상과 관련된 확장된 정보를 얻는다.
- 3) 위의 1,2단계를 통해 얻어진 정보를 분석하여 주어지는 입력에 대해 다양한 출력을 낼 수 있는 시뮬레이션 모델을 생성한다. 이 모델은 다음 단계의 지식 생성을 위한 도구가 되며, 구간값과 같은 지식 제공자의 불확실한 정보를 포함할 수 있는 구조이다.
- 4) 생성된 모델을 시뮬레이션한 결과를 분석하여 지식으로 획득한다.

위의 과정에서, 지식획득을 위한 도구인 시뮬레이션 모델이 지식 제공자의 개입 없이 내부적으로 자동 생성됨에 따라 지식 제공자는 특정 형식으로 인한 부담없이 대상

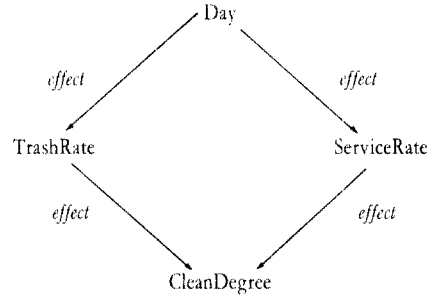
1) Minsky, M. Society of Mind, 1986 by Simon and Schuster, New York, NY.

“A model is a device for answering questions about an object”

지식 자체에 집중할 수 있으며, 생성된 모델의 시뮬레이션 수행 과정에서 보이는 시간에 따른 변화 내용을 분석하여 지식화 함으로써 획득된 지식이 동적인 실세계에 대한 보다 정확한 내용이 될 수 있도록 하였다.

## 2. 지식 획득 과정

단계별로 설명해 가는데 있어 예로 들것은 도시의 깨끗함 정도를 이와 연관된 여러 요인들로부터 분석해 내는 시스템이라 하자. 어느 상황에서 도시의 깨끗함 정도와 관련된 요인으로 지식 제공자 쉽게(상식적으로) 생각할 수 있는 정보로, 요일에 따라 쓰레기 수거 정도와 시민들의 쓰레기를 버리는 양이 다르며 이들에 따라 깨끗함 정도가 결정된다는 사실이라 가정한다.



〈그림 1〉 초기 인과 관계 그래프

### 2.1 대상에 대한 인과 관계 정보 입력

지식 제공자가 제공하는 입력으로 대상에 관련된 개념 (concept)과 이 개념의 파급관계(effect)만을 받아들인다. 입력내용은 바로 모델 구성에 사용되는 것이 아니라, 대상에 대한 초기 추정 정보로 이용한다. 입력을 바탕으로 대상에 대한 인과 관계 그래프를 구축한다.

예에서 지식 제공자가 제공한 입력은 아래와 같다.

- “If Day is different then TrashRate and ServiceRate are different”
- “If TrashRate is different, then CleanDegree is different”
- “If ServiceRate is different, then CleanDegree is different”

이를 통해 그림 1의 인과 관계 그래프를 구성한다.

### 2.2 경험 베이스 탐색 및 대상에 관련된 확장된 인과 관계 정보 생성

경험 베이스는 지식 제공자에 의해 입력된 경험적인 지식들을 인과 관계에 따라 계층적으로 구조화하여 모아 놓은 것으로, 일정 프레임(frame)에 따라 구성된다. 이 단계에서 수행하는 작업은, 2.1 에서 형성된 인과 관계 그래프

를 경험 베이스에 적용하여 대상에 대한 확장된 인과 관계 그래프를 얻는 것이다. 확장된 인과 관계 그래프에 담긴 정보를 통해서 대상에 대한 시뮬레이션 모델이 자동 생성되어지므로, 경험 베이스를 구성하는 프레임과 이에 대한 슬롯은 아래와 같이 구성한다.

- 1) Concept - 해당 프레임이 담고 있는 개념의 이름
- 2) ISA - Concept가 속한 클래스(class)명  
(StaticConcept, DynamicConcept, Server, Customer, Time, Law)
- 3) Behavior - Concept가 하는 활동  
(Service, Delay, Split, Queuing, Gathering, Selection)
- 4) Effect - Concept가 영향을 주게 되는 다른 개념들의 이름
- 5) Procedure - Concept와 Effect가 연결되는 방식으로 두 개념사이의 관계가 명확하게 알 수 있는가의 여부에 따라 KNOWN, UNKNOWN 으로 명시된다.

경험 베이스의 계층적으로 연결된 프레임들과 2.1에서 얻어진 초기 인과 관계 그래프가 연결되어지는 방식은 아래의 확장된 BFS(Breadth First Search) 알고리즘을 따른다.

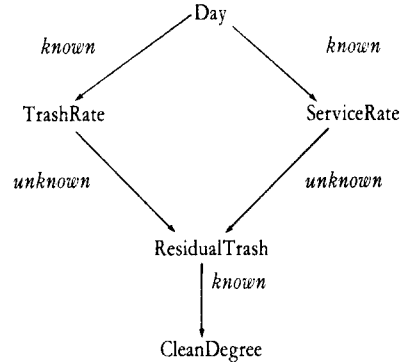
```

Refined Causal Graph Generator Algorithm
{
    Input: 1st Causal Graph

```

```

Output: Refined Causal Graph
Function : Generate Refined Causal Graph after Searching Experience Base
}
While 1st Causal graph not empty do
begin
  Choose an effect edge(v,w) from 1st Causal Graph;
  Perform Breadth First Search(BFS) from v to w in Experience Base;
  Add replacement of (v,w) to Refined Causal Graph including new concept founded in BFS;
  delete effect edge (v,w) from 1st Causal Graph;
end
    
```



〈그림 3〉 확장된 인과 관계 그래프

위의 알고리즘을 적용하여 생성되어진 예에 관한 프레임들의 내용을 구체적으로 보면 다음 그림 2와 같다.

### 2.3 시물레이션 모델 생성

이 단계에서는, 앞서 2.2 단계를 통해 얻어진 확장된 인과 관계 그래프를 통해 대상을 반영하는 모델을 자동 생

|           |               |
|-----------|---------------|
| Concept   | Day           |
| ISA       | StaticConcept |
| Behavior  | None          |
| Effect    | TrashRate     |
| Procedure | Known         |

|           |               |
|-----------|---------------|
| Concept   | Day           |
| ISA       | StaticConcept |
| Behavior  | None          |
| Effect    | ServiceRate   |
| Procedure | Known         |

|           |               |
|-----------|---------------|
| Concept   | TrashRate     |
| ISA       | Customer      |
| Behavior  | TimeDelay     |
| Effect    | ResidualTrash |
| Procedure | Unknown       |

|           |               |
|-----------|---------------|
| Concept   | ServiceRate   |
| ISA       | Server        |
| Behavior  | Service       |
| Effect    | ResidualTrash |
| Procedure | Unknown       |

|           |                |
|-----------|----------------|
| Concept   | ResidualTrash  |
| ISA       | DynamicConcept |
| Behavior  | Queueing       |
| Effect    | CleanDegree    |
| Procedure | Known          |

|           |               |
|-----------|---------------|
| Concept   | CleanDegree   |
| ISA       | StaticConcept |
| Behavior  | None          |
| Effect    | None          |
| Procedure | Known         |

〈그림 2〉 경험 베이스의 내용

그림 2의 내용에 기초하여 생성된 확장된 인과 관계 그래프는 그림 3과 같다.

성하게 된다. 과정은 아래와 같다.

- 1) 모델의 입출력 변수 결정

개념간의 연관관계가 불분명한 경우, 프레임의 PROCEDURE 슬롯은 UNKNOWN으로 명시된다. 이 단계에서 PROCEDURE 슬롯값을 조사하여 UNKNOWN으로 명시된 개념을 찾아내고, EFFECT 슬롯에 의해 이 프레임과 연관된 개념들을 추적하여 발견된 개념들을 입출력 변수로 취하게 된다.

2) 활동 개체(Active Entity) / 자원(Resource) 결정

모델을 작성하기 위해 입력 변수는 다시 활동 개체의 개념이 되는 것과 자원의 개념이 되는 것으로 나누어지며, 이것은 슬롯의 ISA 값에 의해 결정된다

3) Event List 결정

모델이 가지는 Event를 결정하기 위해 각 개념 슬롯의 BEHAVIOR 값을 분석한다.

4) 모델 생성

앞서 지식 획득에 사용되는 시뮬레이션 모델 코드는 불확실성을 수용할 수 있는 구조여야 함을 지적하였다. 이를 해결하기 위해 기존의 시뮬레이션 패키지에 불확실성을 부과 하려는 많은 연구가 이루어지고 있다.[2, 7] 본 연구에서는 기존의 시뮬레이션 형태를 최대한 유지하면서 불확실성을 수용하기 위해 Fishwick의 퍼지 시뮬레이션 알고리즘[2]을 따른다. 따라서 생성되는 시뮬레이션 모델 코드는 이를 지원할 수 있는 코드이다. 퍼지 시뮬레이션 알고리즘은 아래와 같다.

For  $j \in \{1, 2, 3\}$ :

- (a) Let  $p[j] = f[j]$ .
- (b) SIMULATE.
- (c)  $\forall i$ , obtain  $x_{it_e}[j]$ .

where  $p$  : fuzzy simulation component such as a parameter  $p$  in a discrete event simulation.  
 the component is defined as a fuzzy number  $f = (a, b, c)$

$t_e$ : the end time for the simulation

$x_i$ : the state variables of interest

퍼지 시뮬레이션 알고리즘을 수행하기 위해 본 연구에

서는 MIT에서 개발된 C언어 기반의 사건 중심 시뮬레이션 언어 *simpl*[10]에 퍼지 처리를 위한 필요 함수를 첨가하였으며, 시뮬레이션 모델은 확장된 *simpl* 언어에 기반하여 먼저 위의 1,2,3 단계를 통해 얻어진 분석된 정보를 통해 필요한 구조의 모델 조각을 생성한 후 각 조각들을 합성하여 전체 모델을 만들게 된다.

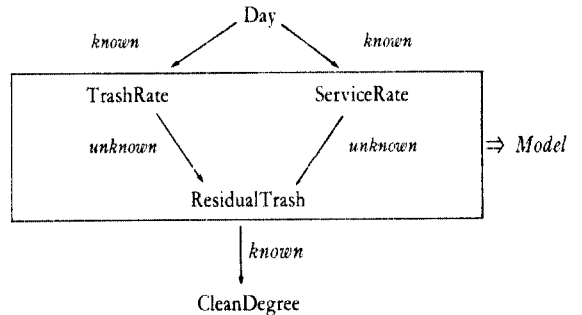
5) 실험수 결정

각 개념들은 표 1과 같이 구간값을 가지는 적당한 퍼지 언어 변수로 표현되어 진다. 퍼지 변수들이 가진 퍼지 값의 종류에 따라, 실험 횟수는 입력 변수들이 취할 수 있는 값들의 모든 조합 수로 결정된다.

〈표 1〉 퍼지 언어 변수

| Variable      | Value  | Fuzzy Number  |
|---------------|--------|---------------|
| TrashRate     | fast   | 100, 150, 200 |
| TrashRate     | medium | 200, 250, 300 |
| TrashRate     | slow   | 300, 350, 400 |
| ServiceRate   | short  | 50, 70, 100   |
| ServiceRate   | medium | 180, 230, 280 |
| ServiceRate   | long   | 280, 330, 380 |
| ResidualTrash | small  | 1, 2, 3       |
| ResidualTrash | medium | 4, 5, 6       |
| ResidualTrash | large  | 7, 8, 9       |

그림 3에서 UNKNOWN의 관계를 갖는 개념들은 모델로 구성되어진다. (그림 4 참조)



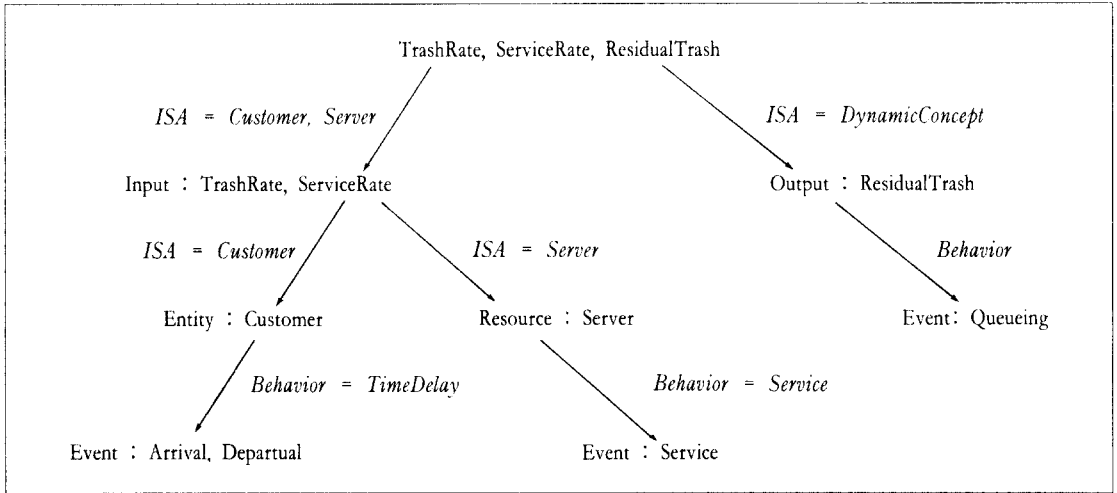
〈그림 4〉 모델 구성 부분

모델링을 위한 분석 과정을 도식화하면 그림 5와 같다. 다음으로, 분석된 각 사건에 대한 부분 모델을 구축하고,

이들을 합성하여 전체 퍼지 시물레이션 모델 코드를 생성하게 된다. 생성된 모델 코드는 그림 6과 같다.

is medium

- If Day = Friday then Trashrate is slow and ServiceRate



〈그림 5〉 모델 생성을 위한 분석 과정

## 2.4 시물레이션 및 지식 획득

앞서 단계에서 UNKNOWN관계를 통해 구축된 시물레이션 모델에 대해 결정된 수 만큼의 시물레이션을 행한 후, 생성된 결과를 통해 지식을 획득하는 단계이다. 시물레이션이 가진 특성에 따라 획득된 지식은 시간 변화에 따른 지식이라는 성격을 갖는다. 또한 KNOWN관계에 의해 연결된 개념들은 시물레이션을 거치지 않고 바로 지식으로 생성된다. 이때 지식은 규칙의 형태로 표현되도록 하였다.

앞서든 예에서 지식 획득 과정을 통해 획득된 지식은 아래와 같다.

- If Day = Monday then TrashRate is fast and ServiceRate is fast
- If Day = Tuesday then TrashRate is medium and ServiceRate is medium
- If Day = Wednesday then Trashrate is fast and ServiceRate is slow
- If Day = Thursday then Trashrate is fast and ServiceRate

is medium

- If Day = Saturday then Trashrate is fast and ServiceRate is fast
- If ResiduaTrash is small then CleanDegree is excellent
- If ResidualTrash is medium then CleanDegree is good
- If ResidualTrash is large then CleanDegree is poor
- If TrashRate is fast and ServiceRate is fast then ResidualTrash is small
- If TrashRate is fast and ServiceRate is medium then ResidualTrash is medium
- If TrashRate is fast and ServiceRate is slow then ResidualTrash is large
- If TrashRate is medium and ServiceRate is fast then ResidualTrash is small
- If TrashRate is medium and ServiceRate is medium then ResidualTrash is medium
- If TrashRate is medium and ServiceRate is slow then ResidualTrash is large
- If TrashRate is slow and ServiceRate is fast then ResidualTrash is small

- If TrashRate is slow and ServiceRate is medium then ResidualTrash is small
- If TrashRate is slow and ServiceRate is slow then ResidualTrash is medium.

획득된 결과들은 TrashRate과 ServiceRate이 CleanDegree에 구체적으로 어떻게 연관되어 있는가에 관한 지식으로, 2.1의 지식 제공자의 초기 입력 정보와 비교해 볼 때 영역에 관련된 새로운 많은 구체적인 지식들이 자동 획득되었음을 보여 준다.

레이션의 사용 가능성을 제시하였다. 제안된 지식 획득 방법이 가지는 의미는 다음과 같이 요약될 수 있다.

- 1) 지식 표현틀로서 모델을 도입하고, 이의 시뮬레이션 결과에 의해 새로운 지식이 획득되게 함으로써 지식 제공자에 대한 지식 획득 의존도가 낮아질 수 있다.
- 2) 지식 표현틀으로써 사용된 모델의 생성이 자동적으로 이루어지도록 함으로써 지식 제공자는 대상 자체에 대한 지식에 집중 할 수 있다.
- 3) 생성 모델은 불확실성이 포함된 지식 표현이 가능하

```
#include <stdio,h>
#include "f.h"

main()
{
    FILE *dfpt;
    char s[S_MAX],s2[S_MAX];
    int i;
    float data[TRIAN],data2[TRIAN];
    int customer = 1,event,server,num_points;

    dfpt = fopen("data2.txt","r");
    fscanf(dfpt, "%s %s\n", s, s2);
    fzvocab("trashrate");

    makefz(s,data);
    fzvocab("serviceamount");
    makefz(s2,data2);

    for (i =0; i < TRIAN ; i++) {
        num_points=0;
        smpl(1,"M/M/1 Queue");
        server=facility("server",1);
        schedule(1,0,0,customer);

        while(num_points < FZ_MAX ) {
            cause(&event,&customer);
            a[num_points][i]=Lq(server);
            num_points++;
            switch(event) { /* event loop */
                case 1: /* arrival */
                    schedule(2,0,0,customer);
                    schedule(1,expntl(data[i]),customer);
                    break;
                case 2: /*request server */
                    if ( request(server,customer,0) == 0)
                        schedule(3,expntl(data2[i]),customer);
                    break;
                case 3: /* completion */
                    release(server,customer); break;
            } /* end of event loop */
        }
        report();
    }
    fzvocab("residualtrash");
    defuzzy(a,b,num_points);

    fclose(dfpt);
} /* end main */
```

〈그림 6〉 생성된 시뮬레이션 모델 코드

### 3. 결론

본 연구에서는 지식 획득을 위한 새로운 방법으로 시뮬

도를 하였다.

- 4) 구축된 모델로부터 지식을 이끌어 내는 방법으로 시뮬레이션을 수행하도록 함으로써 지식 획득 대상의

동적 측면에 대한 표현이 가능하도록 하였다.

제안된 방법은 경험 베이스가 풍부해짐에 따라 지식 제 공자의 불완전한 정보로부터 보다 많은 새로운 지식이 자 동적으로 생성될 수 있다는 장점을 갖는다. 그러나 경험 베이스가 커짐에 따라 개념들간의 연관 관계에 불일치 (inconsistency)나, 고립, 순환등이 없도록 유지해야 한다는 것을 문제점으로 지적할 수 있다. 또한 본 연구에서 제시 한 지식 획득 방법을 여러 복잡하고 다양한 대상에 적용 하여 그 타당성을 조사해 보는 작업이 필요하다.

## Reference

- [1] Gunnar Johannsen and James L. Alty, "Knowledge Engineering for Industrial Expert Systems", Automata, vol. 27, no. 1, pp 97 - 114, 1991
- [2] Chaya Garg-Janardan and Gavriel Salvendy, "A Conceptual Framework for Knowledge Elicitation", Knowledge-Based Systems, vol. 2, pp 119 - 129, 1988
- [3] Pradig Dey, "ACKNOWLEDGE 2: A Knowledge Acquisition System", Simulator IV, vol.18, no. 4, 1987, pp 1-5
- [4] Vincent J. Kovarik Jr., "Using Discrete Event Qualitative Simulation as a Knowledge Acquisition Tool", Proceeding: AI, Simulation and Planning in High Autonomous Systems, 1991
- [5] Bernard P. Zeigler, "Object-Oriented Simulation with Hierarchical Modular Model", Academic Press, 1990
- [6] Paul A. Fishwick, "Extracting Rules From Fuzzy Simulation", Expert System with Applications, vol 3, 1991, pp 317 - 327
- [7] Benjamin Kuipers and Jerome P. Kassairer, "Knowledge Acquisition by Analysis of Verbatim Protocols", Knowledge Acquisition for Expert Systems, pp 45 - 71
- [8] SukHan Lee, Judy Chen, "Modeling and Simulation for Minimum Uncertainty in Qualitative Physics", Proc. of the 1991 IEEE Int. Conf. on Tools for AI, pp 322 - 329
- [9] Paul A. Fishwick, "A Study of Terminology in Qualitative Simulation", Simulation, vol. 52, no. 1, 1989, pp 5 - 9
- [10] M.H. MacDougall, "Simulating Computer Systems : Techniques and Tools", MIT Press, 1987

## 저자소개



김명희(金明姬)

1974.2 이화여대 졸, 사회학전공(문학사)  
 1979.8 서울대학교 대학원졸  
 전산학전공(이학석사)  
 1986.2 독일 Göttingen 대학교 대학원졸  
 전산학 전공(이학박사)  
 1986.3~1987.2 서울대. 이화여대 강사  
 1987년 3월 이후~현재 이화여대 전산과 교수  
 관심분야: 시뮬레이션 방법론 및 응용, 성능평가.