

논리 시뮬레이션을 기반으로한 체험형 자동차 정비 훈련 시스템

박길식¹ · 박대성¹ · 박기현² · 김준태^{1*}

An Experience-Type Car Maintenance Training System based on Logic Simulation

Gil-Sik Park · Dae-Sung Park · Ki Hyun Park · Juntae Kim

ABSTRACT

Recently, researches on the application of IT technology to various fields including traditional industries are becoming more popular. One challenge in the field of education is to understand the way how technology may support learning, and research on self-directed learning has been accelerated by integrating education and IT technology. The process of self-directed learning in e-learning applications such as Car Maintenance Training is very difficult and complicated. Previous studies on car maintenance training applications provided simple training scenarios with predetermined action sequences. To incorporate self-directed learning in car maintenance training, however, trainees must be able to perform various maintenance operations himself and experience various situations. To provide such functionality, it is necessary to obtain an accurate response for various operations of trainees, but it requires complicated calculations with respect to varieties in the electrical and mechanical processes of a car. In this paper, we develop a logic simulation agent using JESS inference engine in which self-directed learning is achieved by capturing the behavior of trainees and simulating car operations without complicated physical simulations in car maintenance training.

Key words : E-training, E-learning, Logic simulation, JESS, Car maintenance

요약

최근 IT 기술을 전통 산업 등 다양한 분야에 적용하려는 연구가 많이 진행되고 있다. 교육 분야에서는 기술을 이용하여 학습을 개선시키는 방법에 대한 관심이 높아지고 있으며, 교육에 IT 기술을 접목하는 자기주도적 학습에 대한 연구가 활발하게 진행되고 있다. 자동차 정비 훈련 분야에서도 이러닝 형태의 교육이 이루어지고 있지만, 자기주도형 학습을 하기에는 어려움이 많다. 기존에 만들어진 자동차 정비 훈련 프로그램은 개발자에 의해 정해진 훈련 시나리오에 따라 미리 정해진 동작만을 차례로 수행하게 되어있다. 그러나 훈련자가 자기주도적으로 정비 훈련을 수행하기 위해서는 훈련자 스스로 여러가지 정비 동작을 수행하고 다양한 상황을 경험할 수 있어야 한다. 그러한 기능을 제공하기 위해서는 훈련자의 다양한 동작에 대해서 실제와 같은 결과가 나오도록 프로그램이 개발되어야 하지만, 그러려면 자동차의 여러가지 복잡한 전기적, 기계적 동작에 관한 매우 복잡한 계산이 수행되어야 한다. 본 논문에서는 자동차 정비 훈련에서 복잡한 물리적 시뮬레이션 없이 훈련자의 다양한 동작에 따라 자동차의 동작을 시뮬레이션 함으로써 자기주도적인 학습을 할 수 있는 JESS 추론 엔진을 이용한 논리 시뮬레이션 에이전트를 구현한다.

주요어 : 이트레이닝, 이러닝, 논리 시뮬레이션, JESS, 자동차 정비

* 본 연구는 산업통상자원부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였습니다.(No.10040102, 자동차 정비 훈련을 위한 자기주도 체험형 e-training 서비스 기술 개발)

접수일(2014년 5월 28일), 심사일(2014년 6월 10일),

게재 확정일(2014년 6월 19일)

¹⁾ 동국대학교

²⁾ (주)테크빌닷컴

주 저 자 : 박길식

교신저자 : 김준태

E-mail; jkim@dongguk.edu

1. 서론

컴퓨터와 인터넷의 발전과 확산으로 교육장소가 오프라인에서 온라인으로 확장되고 있는데, 이는 오프라인 교육에서 제공하는 교육 서비스를 온라인에서도 제공할 수 있다는 가능성 및 현실에 기인하고 있다. 정보통신기술을 이용하여 시간과 장소에 구애받지 않는 교수-학습이

가능한 이-러닝은 기존의 교육이 가지고 있는 한계에서 벗어나 쌍방향적 의사소통 및 활동을 가능하게 한다.

하지만, 이-러닝은 훈련자의 집중도와 태도에 대해 교수자가 직접적으로 관여하기 어렵고, 자발적인 수업 참여가 이루어지기 어려운 결과를 초래하기도 한다. 이에 따라 이-러닝은 기존 동영상 위주의 이론 강의에서 탈피하여 훈련자가 마치 산업 현장과도 같은 가상 환경에서 기술 훈련을 하는 체험형 학습(e-training)으로 발전했다^{4, 7)}. 체험형 학습은 훈련자의 흥미를 이끌어 낼 뿐만 아니라, 고위험 장비에 대한 안전 문제, 비용 문제, 공간 및 장소의 문제, 장비 고장의 우려 등을 해소할 수 있는 많은 장점이 있다.

이러한 체험형 학습을 위한 가상환경에는 훈련자의 동작을 해석하고 그에 상응하는 결과를 보여줄 수 있는 시뮬레이터의 구현이 필수적이다. 하지만 기존에 연구 개발된 훈련 시스템들은 훈련자의 미리 정해진 동작에 대해서는 결과가 도출되지만, 정해져 있지 않은 다양한 동작에 대해서는 대응하지 못한다⁶⁾. 예컨대, 훈련 시스템에 구현되어 있지 않은 훈련자의 예외적인 행동에 대해 일일이 적절한 결과를 보여주지 못하며, 다양한 훈련자의 동작에 대해서 적절한 반응을 모두 구현하기에는 시간적, 경제적으로 어려움이 따른다. 따라서 훈련 시스템 개발 시 훈련자의 행동과 훈련 과정에 대한 범위 등을 제한하여 개발할 수밖에 없었다. 이러한 훈련 시스템들은 정해져 있지 않은 다양한 훈련자의 동작에 대해서는 시뮬레이션을 하지 못하므로, 훈련자의 학습 효과가 높지 않다. 또한 틀(template)에 짜인 시나리오에 따른 획일적인 교육훈련이 될 수밖에 없으므로 현실감과 흥미가 떨어지게 되어 자기 주도적인 학습을 하기에 어려움이 따르게 된다.

본 논문에서는 자동차 정비 훈련 교육을 수행하는데 있어, 훈련 시나리오에 포함되어 있는 동작뿐만 아니라 포함되어 있지 않은 훈련자의 다양한 동작에 대해서도 시뮬레이션 결과를 도출해 낼 수 있도록 지능형 논리 시뮬레이터를 설계하고 구현하였다. 훈련 시나리오에 포함되어 있지 않은 훈련자의 다양한 동작은 추론을 통해 인식하고 결과를 도출할 수 있는데, 이를 위해 JESS(Java Expert System Shell) 추론 엔진을 사용하였다. JESS 추론 엔진은 훈련자의 동작들이 입력되는 즉시, 지식베이스(Knowledge Base)에 저장되어있는 규칙(rule)들에 의해 다양한 동작에 대한 새로운 결과를 사실(fact)형태로 도출해 낼 수 있고, 이 사실들은 다시 입력되어 저장되어 있는 규칙(rule)들에 의해 또 다른 사실들을 연쇄적으로 도출해 낸다. 이와 같은 형태로 동작하는 JESS 기반의 추론 모델을 설계하고 이를 접목한 시뮬레이터가 현재 상황을 인지

하고 연쇄적으로 동작하여 새로운 결과를 추론해냄으로써 예외적인 상황과 다양한 동작에 대해 대응할 수 있다.

본 논문의 2장에서는 관련 연구, 3장에서는 체험형 훈련 시스템 설계, 4장에서 구현 및 실행 결과를 보여주며 5장에서 구현된 결과를 토대로 결론을 맺고, 향후 연구를 제시한다.

2. 관련연구

2.1 e-Learning과 e-Training

이-러닝은 전자적 수단에 의한 학습, 훈련, 교육의 전달을 위해 가상공간에서 이루어지는 데이터, 정보, 기술, 혹은 지식을 획득하기 위해 사용되는 모든 가상적 행위나 프로세스로서, 학습을 촉진하고 강화하기 위한 방식을 의미한다¹⁵⁻¹⁷⁾. 이-러닝은 더 이상 선택이 아닌 필수가 되어 가고 있으며, 교육의 효율성과 효과성을 높일 수 있다는 기대 속에서 미래 교육의 대안이 되고 있다. 그러나, 이-러닝의 급속한 발전에도 불구하고, 여전히 면대면 교육에 대한 선호도가 더 높게 나타나는 것은 그 중심에 학습을 촉진시키는 인간 교사가 있기 때문이다²¹⁾.

아-트레이닝은 업무에 필요한 수행능력을 습득, 향상시키기 위해 정보통신 기술과 다양한 디바이스를 활용한 학습훈련으로서 기존의 이-러닝에서 한 단계 더 나아가 학습자가 체험할 수 있는 실전위주의 교육훈련을 가능하게 하는 학습 모델이다²⁵⁾. 아-트레이닝은 학습을 위한 가상 환경을 제공하여 훈련자의 동작에 따라 결과를 보여줄 수 있어 현실감과 흥미를 올려줄 수 있고 더 나아가 자기주도적 학습을 이끌어 낼 수 있는 교육 트렌드로서 고위험 장비에 대한 안전 문제, 비용 문제, 공간 및 장소의 문제, 장비 고장의 우려 등을 해소할 수 있는 많은 장점이 있다. 이러한 학습을 위한 가상환경에는 훈련자의 동작에 따라 결과를 보여줄 수 있는 시뮬레이터의 구현이 필수적이다.

이외에도 컴퓨터를 이용한 학습 형태는 학습 과정을 통제하는 시스템인 컴퓨터 기반 교육부터 자신의 학습에 대해 효과적인 결정을 내릴 수 있도록 하는 적응적 이-러닝 및 기존 이-러닝보다 학습에서의 이동성과 학습자의 학습상황에 맞는 진단 및 학습 처방이 강조된 환경으로서, 학습자의 상황정보 수집과 이에 따른 진단 및 서비스 제공을 하는 학습 모델인 유-러닝까지 다양한 접근법이 존재한다^{8, 9, 14, 18, 20)}.

최근 자동차 정비 관련 선행연구로는, 이-러닝 형태로 자동차 정비를 수행할 수 있도록 하는 환경을 Visual Basic 언어를 이용해 프로그램을 만들고, Microsoft Access 프

로그를 사용하여 지식베이스를 구축하여 운용한 사례가 있다^[1]. 이 연구에서 제안한 자동차 정비 이-러닝 프로그램은 개발자가 구현한 훈련자의 동작에 대해서는 진단이 가능하나, 개발자가 구현하지 않은 훈련자의 동작에 대해서는 진단이 어렵다. 따라서 교사가 의도한대로 수행만 가능하므로, 자기주도적인 학습이 어렵다는 단점이 있다.

이처럼, 기존에 연구 개발된 훈련 시스템들은 학습 훈련자의 정해진 동작에 대해서는 어느 정도 시뮬레이션이 가능하나, 정해져 있지 않은 다양한 동작에 대해서는 정확한 시뮬레이션을 하는 데에 어려움이 있었다. 예컨대, 개발자가 구현하지 않은 훈련자의 예외적인 행동에 대해 일일이 대응하기 어렵고, 다양한 훈련자의 동작에 대해서 모두 구현할 수 없으므로 학습자의 행동과, 학습 과정, 시뮬레이션 범위 등을 제한하고 있다. 그러나 이는 현실감과 흥미를 떨어뜨리게 되어 자기주도적 학습을 하기 어렵다^[24].

2.2 JESS

2.2.1 JESS 추론 엔진

JESS는 Java Expert System Shell의 약자로서 JAVA 플랫폼에서 동작하는 규칙 기반 추론 엔진으로 1995년 미국의 Sandia National Laboratories에서 개발되었으며, 추론 엔진이자 지식 기반 시스템 개발 환경이다^[10]. 규칙 기반 전문가 시스템(rule-based expert systems)인 CLIPS의 개념을 이어받고 있으며, LISP와 유사한 규칙 기반 시스템의 개념을 갖는 언어이나, 규칙을 정의하거나 코딩하기에 간단하며 이러한 규칙은 룰이라고 부른다.

JESS는 JAVA 플랫폼에서 동작하기 때문에 JAVA의 네트워킹, 그래픽, 데이터베이스 등과 같은 JAVA의 강력한 API를 이용할 수 있다. 이것은 JESS를 Java Program, Java Applet, JSP(Java Server Page) 등과 같은 JAVA기반의 여러 환경에도 적용 될 수 있다는 것을 의미한다. 또한 JAVA로 제작된 프로그램은 자바 가상 머신(Java Virtual Machine)이 바이트코드를 해석하여 다양한 플랫폼에 실행 될 수 있다는 장점이 있다. 따라서 JAVA 플랫폼에서 동작하는 JESS 또한 JAVA를 지원하는 다양한 플랫폼에서 동작한다는 장점이 있다. 또한 컴파일의 필요하지 않는다는 장점도 있으며, 규칙을 작성하는 문법만 알면 누구든 프로그램의 수정 없이 지식베이스를 구축할 수 있는 장점도 있다. JESS와 같은 규칙 기반 추론 엔진은 워킹메모리에 저장된 사실을 바탕으로 지식베이스에 저장된 규칙을 이용하여 새로운 사실을 도출하여 워킹메모리에 저장한다.

2.2.2 JESS기반 전문가 시스템

기존의 JESS를 활용한 연구들은 대부분 JESS를 이용하여 전문가 시스템을 구축하는 것이었다. Java 기반의 JESS를 이용하면 플랫폼에 독립되어 어떠한 시스템 환경에서도 사용이 가능하다는 점과, 시스템과 지식베이스를 구성하는 룰이 분리되어 있으므로, 지식베이스의 수정과 확장이 용이하다는 장점이 존재하기 때문이다.

Ha et al.^[28]은 JESS를 이용한 소화기 질환 진단용 의료 전문가 시스템을 구현하였다. 소화기 질환을 진단하기 위하여 지식베이스를 환자가 나타나는 증상들을 결정트리 형태로 표현하여 룰을 설계했다. 그 다음 Java Servlet 모듈을 이용하여 웹상으로 사용자에게 아픈 부위와 증상에 대해 물어본 다음 추론하여 진단하였다. 이 논문은 JESS로 작성된 전문가 시스템을 웹 플랫폼에 접목하여 활용했다는 점을 보여주고 있다.

근래 차량 시스템도 온톨로지와 같은 기술을 접목시키는 연구들이 이루어지고 있다. Lee et al.^[27]은 차량의 상태 진단을 위해 차량 내부의 장치와 센서들의 연결 관계를 파악하여 온톨로지를 구성하며 JESS를 이용하여 결과를 실험하고 있다. 이 연구에서는 차량의 센서를 온톨로지로 구현하고 그 온톨로지를 테스트하기 위하여 JESS를 사용하였다. 이 연구에서는 차량의 센서를 온톨로지로 구현하고 그 온톨로지를 테스트하기 위하여 JESS를 사용하였다.

Ahn^[26]은 작업장의 유해가스 정보를 모니터링하기 위한 추론 규칙을 JESS로 작성하여 상황에 따른 의사결정이 가능하도록 상황인식 시스템을 설계하였다. 이 논문에서는 밀폐된 공간에서 발생하는 유해가스 정보를 작업장 내의 센서를 통해 실시간으로 모니터링하고, 위험으로부터 작업자의 안전을 보장할 수 있도록 상황에 맞는 의사결정이 가능한 모니터링 시스템을 JESS 기반으로 설계하였다.

Kil et al.^[19, 22]은 가스절연변전소에 설치되어 있는 피뢰설비의 감시와 진단이 가능한 전문가 시스템의 설계 및 구현에 대하여 기술하였다. 이 논문에서 제안한 전문가 시스템은 마이크로프로세서 기반의 데이터 취득 모듈과 진단 알고리즘으로 구성되며, 피뢰기 진단에 필요한 계통 전압, 누설전류성분 및 온도 등을 검출하고 분석한다. 전문가 시스템에서 피뢰기의 진단은 지식 베이스와 추론 엔진, 그리고 그래픽 사용자 인터페이스로 구성된 JESS를 이용하여 결정하였다.

Robindro et al.^[11]은 벼(rice plant)의 질병 진단을 위한 JESS 기반 전문가 시스템 구조를 설계하였다. 벼(rice

plant) 질병 진단 시스템은 농부가 전문가의 도움이 필요할 때, 농부에게 의사 결정을 지원해주는 전문가 시스템이다. AlHamad et al.^[2]은 JESS를 사용하여 e-learning 환경의 학습관리시스템(Learning Management System, LMS)에서 학생들의 프로필을 토대로 학습 정보를 개인화하는 방법을 제안하였다. 이 논문에서는 학생들의 수준을 파악하기 위한 문제를 템플릿(template)으로 정의하였고, 학생들의 문제에 대한 정답 및 오답의 수에 따라 추천되는 학습 시나리오로 진행되도록 설계되었다.

Rai et al.^[12]은 텍스트, 이미지, 오디오, 비디오 등과 같은 멀티미디어 데이터를 서비스하는 주문형 시스템에 대해 데이터를 검색하고 동작할 수 있는 JESS 추론 엔진을 사용하였다.

그 밖에 JESS를 이용한 학습과정 추천 이러닝 시스템 등 JESS는 다양한 전문가 시스템에 적용되어 연구되고 있으며, 전문가 시스템 이외에도 상황 인지 시스템, 온톨로지와의 통합 추론 등의 연구도 이루어지고 있다^[3, 5, 23].

본 논문에서는 복잡하고 방대한 차량 시스템과 정비 훈련 과정을 어떻게 표현하며, JESS를 이용하여 어떻게 규칙을 작성하는지에 대해 다룬다. 그리하여 현재 차량의 상태를 파악하고 추론을 통해 다음 차량의 상태 변화를 유추해 낼 수 있는 지능형 에이전트를 설계하고 구현한다.

3. 시스템 설계

3.1 자동차 정비 훈련

자동차 정비 훈련은 훈련 교사에 의해 미리 설정된 자동차 고장 상황에 따라 훈련자가 그 원인을 찾아내고 그에 상응하는 적절한 조치를 취하는 과정이다. 이를 위하여 제안한 e-training 시스템은 훈련 과정에 정의된 정비 동작에 대한 시뮬레이션 이외에도 자동차의 일반적인 동작인 시동을 걸거나, 기어를 변속하는 등의 자동차 조작과 관련된 기본 동작들과, 멀티미터나 스캔툴 등 다양한 정비 도구들에 대한 시뮬레이션도 구현하였다. 자동차 정비 훈련 과정을 크게 분류하게 되면 엔진과 차체로 나눌 수 있으며, 엔진에서 발생하는 현상을 다시 세분화 시키면 시동 불량, 엔진 부조, 시동 꺼짐, 출력 가속 성능 불량, 연비 불량으로 나누어 질 수 있다. Fig. 1은 자동차 시동 불량의 경우에 대한 점검 과정을 분류한 계층도이다.

Fig. 1을 보면, 자동차 시동 불량의 경우 정비과정은 엔진회전이 가능한지, 불가능한지에 따라 점검하는 과정이 달라진다. 엔진회전이 되지 않는 경우에는 다시 솔레노이드의 작동여부에 따라 점검과정이 더욱 더 세부적으

로 나누어질 수 있다. 먼저, 교사가 다양한 고장 점검 시나리오 중 어느 하나를 선택하여 점검을 수행할 수 있도록 고장 설정을 하게 되면, 훈련생은 각 점검 과정을 수행하여 교사가 설정한 고장 문제를 해결할 수 있을 것이다. 예컨대, 교사가 배터리를 불량으로 고장 설정했다면 훈련자는 과정이 시작되면서 시동이 걸리지 않는다는 것을 확인하게 되며, 시동이 걸리지 않으면 스타트 모터가 회전되는지 확인하고, 스타트 모터가 회전하지 않는다면 솔레노이드 작동여부를 확인하게 된다. 솔레노이드가 제대로 작동한다면 경고등 및 지시등이 점등되는지를 보게 되고, 만약 점등되지 않는다면 배터리 점검을 하기 위해 멀티미터를 이용해 전압값을 측정한 후 전압값이 정상인 아닌 걸 확인하고 배터리 교체를 할 것이다.

3.2 체험형 훈련 시스템 구성

체험형 학습 시스템은 훈련 교사(tutor), 훈련생(trainee), 자동차 정비 훈련 전문가(expert)로 구성된 사용자들과, 자동차 정비 훈련 과정에 관한 규칙(rule)이 정의된 지식 베이스(knowledge base), 추론 엔진(JESS), 훈련 시뮬레이터로 구성된다.

교사는 사전에 자동차 정비 훈련에 대한 시나리오를 입력하고 훈련자는 주어진 상황에 따라 자동차를 조작해 보며 점검을 하는 과정을 반복한다. 훈련 시뮬레이터는 훈련자의 자동차 정비 동작을 입력받아 추론 엔진으로 전달하고, 추론 엔진(JESS)은 전달된 동작에 대한 추론을 수행하게 된다. 훈련자의 행동에 대한 추론 결과에 따라 잘못된 동작에 대한 조언을 훈련자에게 전달해 주거나 훈련자의 훈련 기록을 교사에게 전달하여 교사가 훈련자에 대한 과정 수행 정보를 구체적으로 파악할 수 있도록 한다. 훈련 과정은 자동차 정비 전문가가 정한 훈련 목표와 그에 따른 훈련 시나리오들로 구성된다. 전문가가 작성한

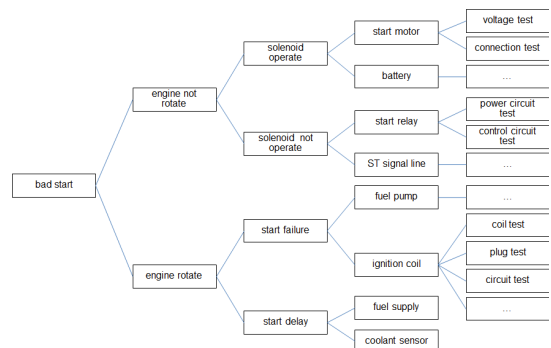


Fig 1. A part of car maintenance training process hierarchy

훈련 시나리오는 훈련자의 동작에 대한 추론 규칙으로 정의된다. 시뮬레이션 에이전트(simulation agent)는 훈련자의 행위를 모니터링하여 훈련과 관련된 도메인 지식에 따라 훈련 상황을 갱신한다. Fig. 2는 본 논문에서 제안하는 시스템의 전체 구성을 보여준다.

훈련 플랫폼에서 훈련 교사가 고장 원인 등을 설정하게 되면, 훈련자는 훈련 과정을 수행하면서 다양한 훈련 동작을 취하게 된다. 그러면 훈련 플랫폼에서는 시뮬레이션 에이전트에 훈련자의 동작 및 상태 등에 대한 정보를 HashMap<Key, Value>형태의 멤버들로 구성된 객체(object)형태로 전달하게 된다. UI(훈련플랫폼)에서 훈련자의 동작 및 교사의 설정 상태를 HashMap 멤버들로 구성된 객체에 담아두고, execute 함수를 호출하면 Simulation Agent가 수행되면서, 추론엔진에서는 객체로부터 값을 읽어와 내부 워킹메모리에 저장하게 된다. Simulation Agent는 추론을 하면서 변수의 상태 변화가 발생 시 내부 워킹메모리뿐만 아니라 HashMap에도 추론 결과를 저장한다. Fig. 3을 보면 Simulation Agent는 HashMap의 Key(훈련자 동작 또는 교사의 고장 설정을 저장하는 변수)에 대한 Value(훈련자의 동작)에 따라 지식베이스(Knowledge Base)에 저장된 사실(fact)과 규칙(rule)을 적용하여 추론을 수행하게 된다. 추론 결과는 해쉬맵 형태로 구성된 객체로 변환하여 훈련 플랫폼으로 전달되게 되고, 최종적으로 시뮬레이션 결과가 제시된다.

3.3 훈련 시뮬레이터 설계

자동차 정비 훈련 시뮬레이터에서 교사는 자동차 정비 훈련에 대해 차량의 상태나 고장 원인에 대한 초기 값을 설정 할 수 있어야 한다. 훈련자는 고장의 원인을 모르는 상태에서 차량에 나타나는 현상에 따라 자동차를 조작하고 점검을 하는 과정을 수행할 수 있어야 한다. 지능형 에이전트는 JESS Rule로 구성된 지식베이스와 데이터를 주고 받으며 교사가 설정한 상태와 훈련자가 자동차를 조작하는 행위를 가지고 자동차의 현재 상태를 추론하도록 설계해야 한다. 시뮬레이터를 제작하려면 먼저 그 해당분야에 대한 구현 범위 설정이 중요하며, 전문가의 지식을 지식베이스로 구축하는 작업이 필요하다.

자동차 정비 훈련을 위해서는 자동차 정비 과정에 대한 구현만이 아닌 정비 대상이 되는 자동차 또한 시뮬레이션 할 수 있도록 구현하여야 하며, 정비를 하기 위한 정비 도구 또한 구현해야 한다. 또한 정비를 하기위해서 단순히 자동차가 동작을 표현하는 것이 아니라, 전기 기계적인 동작 원리에 따라 추론이 되도록 구현하여야 한다.

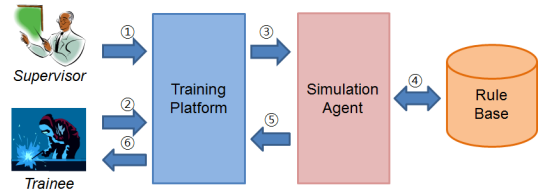


Fig 2. The system structure of car maintenance training based on logic simulation

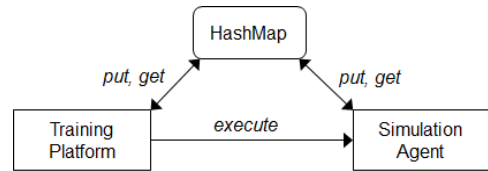


Fig 3. Data exchange between training platform and simulation agent

3.3.1 개발 범위

자동차에 나타나는 현상에 대한 현상별 정비 과정은 크게 엔진과 차체로 나눌 수 있다. 이 중 엔진 점검 과정은 현상별로 크게 5단계로 1. 시동 불량, 2. 엔진 부조, 3. 시동 꺼짐, 4. 출력 가속 성능 불량, 5. 연비 불량으로 나뉜다.

Fig. 4는 시동 불량에 해당하는 점검 과정 중의 하나로서 스타트 모터 점검 과정을 보여주고 있다. 자동차 정비 훈련 과정에서, 시동이 걸리지 않는 현상이 발생하면 훈련자는 우선 엔진 회전여부를 살펴보게 된다. 만약 엔진이 회전하지 않는다면 훈련자는 솔레노이드 상태를 확인해봐야 하며 그 상태에 따라 배터리 또는 스타트 모터에 연결된 전기 배선에 문제가 있는지 찾아보게 된다. 훈련자가 고장 원인을 찾기 위해 이러한 일련의 과정을 수행하도록 하는 것이 하나의 훈련 과정 시나리오가 된다. 훈련자는 정해진 시나리오를 따라가며 훈련을 할 수도 있지만, 시동이 걸리지 않는 현상 하나만으로도 다양한 원인이 존재 하는 가능성이 있으므로 시뮬레이터가 제공되면 훈련자는 여러 가지 가능성을 시뮬레이터를 조작하면서 훈련할 수 있다.

3.3.2 지식베이스 설계

차량의 상태나 조건들을 JESS 룰로 표현하기 위해 자동차 정비 과정에서 사용된 변수는 크게 3가지로 구분된다. Table 1을 보면 교사가 고장 설정하는 항목을 의미하는 변수명은 접두어 SP를 사용하였고, 훈련자의 정비 동

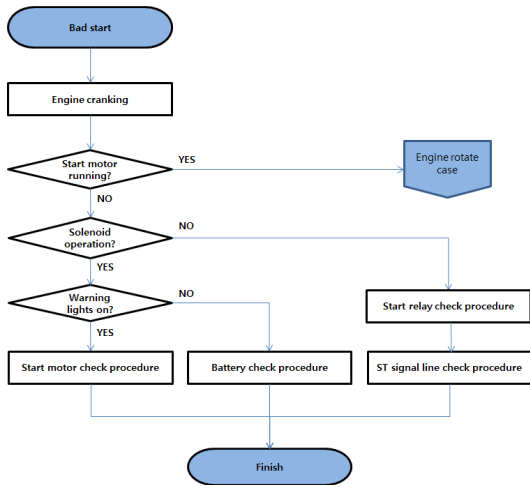


Fig 4. The flowchart of a test procedure for bad start situation

Table 1. The list of the variables for simulation

Type	Variable name	Description	Value
Supervisor (SP)	SP_ENGN_STARTMOTOR	state of the start motor	GOOD/BAD
	SP_ENGN_BATTERY	state of the battery	GOOD/BAD
	SP_ENGN_AIRBAG_FUSE	state of the air bag fuse	GOOD/BAD

Trainee (TR)	TR_BODY_IGKEY	operation on start key	REMOVE/OFF/ON/START
	TR_BODY_GEAR	operation on gear	P/R/N/D
	TR_BODY_FOOTBREAK	operation on foot brake	ON/OFF

Logic Simulator (LS)	LS_ENGN_RUNNING_SOUND	sound of the engine	ON/OFF
	LS_ENGN_SOLENOID_SOUND	sound of the solenoid	ON/OFF
	LS_BODY_RPM	engine RPM	NUMBER

작을 의미하는 변수는 접두어 TR을 사용하였으며, 로직 시뮬레이터가 추론을 통해 얻어낸 값들이나 추상적인 값들을 의미하는 변수는 접두어 LS를 붙였다. 또한 엔진에

해당되는 변수인지, 단순 차량에 관련 있는 변수인지를 구분하기 위한 구분 명을 변수명 중간에 넣었다.

예를 들어, 훈련 교사가 배터리 상태를 나쁨으로 설정하게 되면, 해쉬맵에 (“SP_ENGN_STARTMOTOR”, “BAD”)가 저장되며, 훈련플랫폼으로 그 값이 전달된다. 그런 후 훈련생이 정비를 하는 과정에서 시동키를 켜고, 기어를 중립에 놓고, 풋 브레이크를 밟아서 시동을 건다면, 해쉬맵에 (“TR_BODY_IGKEY”, “START”), (“TR_BODY_GEAR”, “N”), (“TR_BODY_FOOTBREAK”, “ON”) 이 저장된다.

이렇게 해쉬맵에 저장된 값이 훈련 플랫폼에서 추론 엔진으로 전달되어 추론이 진행된다. 훈련생이 시동을 걸기 위한 여러 가지 동작을 했으나 훈련 교사가 사전에 배터리 상태를 ‘BAD’로 설정하였으므로, 시동이 걸리지 않게 될 것이다. 따라서 추론엔진에서 추론된 결과값은 (“LS_ENGN_RUNNING_SOUND”, “OFF”)가 될 것이며, 해쉬맵 형태로 훈련플랫폼에 전달되어 시동이 걸리지 않아 시동소리가 들리지 않게 될 것이다.

또한, 훈련자의 행위에 대해 연속적으로 일어날 수 있는 모든 반응 결과들에 대한 변수들을 정의하고 구현하는 것은 복잡하고 어려워질 수 있어서, 중간 반응 결과를 저장하는 중간상태 변수를 두었다. 예를 들면 자동차 내부의 어느 전선 하나를 실수로 잘라버렸다면, 시동이 걸리지 않을 수도 있으며 시동뿐만 아니라 에어컨이나 조명도 들어오지 않을 수 있다. 이것을 구현하려면 많은 조건들을 매번 비교할 수밖에 없고, 이는 구현이 어려울 뿐더러 유지보수에도 많은 문제가 생긴다. 이러한 연속적인 추론을 하기 위해, 추론 중 발생 되는 중간단계의 현상을 저장할 수 있는 변수가 필요하다. 예를 들면, 자동차의 히터는 시동을 켜야 나온다는 조건을 가지고 있다. 자동차의 시동은 자동차의 전기계통과, 엔진 상태, 연료 계통이 모두 정상일 때 등 여러 조건이 만족해야 시동이 걸린다. 따라서, 히터가 켜지는 조건은 전기 계통, 엔진 상태, 연료 계통 등이 모두 정상일 때 히터가 켜지는 동작이 수행될 수 있다. 이처럼 히터를 켜는 동작에 조건이 무수히 많아져 규칙이 복잡해 질 수 있다.

이와 같은 문제를 해결하기 위해 시동 여부에 대한 상태를 저장하는 중간 변수를 만들게 되면 히터를 켜는 동작엔 시동여부 변수와 히터를 켜는 동작 조건 2가지만 비교하면 되므로 조건 비교가 간결해지고 유지보수에도 용이해진다. 이처럼 중간단계에서 상태를 저장하는 구조를 본 논문에서는 Internal Model이라 부르며, 이 Internal Model은 연속적으로 상태를 추론하는 추론모델에 적합하다고 할 수 있다. Fig. 5를 보면 계층적 형태의 구조로

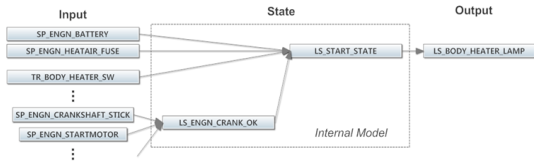


Fig 5. The hierarchy of variables according to the Internal Model

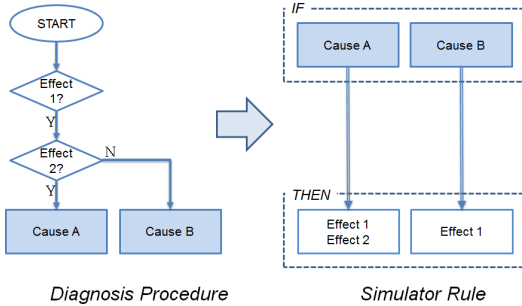


Fig 6. The relationship between a diagnosis procedure and simulator rules

```
(defrule rule_name
  <SP : car problem setting>
  <LS : current state of the simulator>
  <LS : current state of the car>
  <TR : trainee's behavior>
  ...
  =>
  <LS : new state of the car>
  <LS : new state of the simulator>
  ... )
```

Fig. 7. Logic simulation rule in JESS form

Internal Model을 설계하였는데, 각 입력에 대해 출력이 되기 전에 중간 상태를 기억하는 변수가 사용이 되었다. Fig. 5는 자동차 크랭킹이 정상인지를 파악하여 LS_ENGN_CRANK_OK에 담아두고, 시동이 걸리는 조건과 엔진 크랭킹이 정상인지 변수를 재 참조하여 내부 변수인 LS_START_STATE 변수에 중간 상태를 저장하고, 이 변수와 히터를 켜는 동작이 참인 경우 히터를 동작하는 과정을 보여주고 있다.

훈련 과정은 자동차에 나타나는 현상에 따라 훈련자가 여러 가지 점검을 통해 원인을 찾아야한다. Fig 6은 이러한 훈련 과정과 시뮬레이터 추론 규칙의 관계를 나타낸다. JESS 룰은 IF-THEN의 형태를 취하고 있으며 IF절에는 어떤 사건의 원인인, THEN절에는 그 원인에 따른 결과

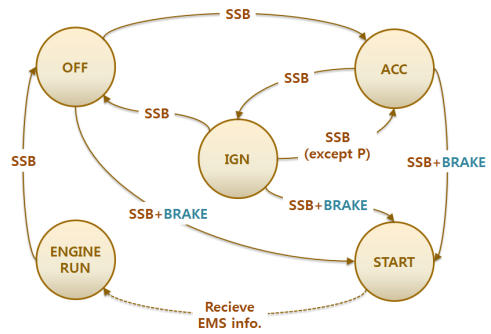


Fig 8. The start up sequence with smart key

를 적는다. 그러므로 훈련 과정 순서도에서 상위 노드에 등장하는 현상들을 THEN절에 작성하고, 말단 노드에 나타나는 원인이나 점검과정을 IF절에 작성해야 한다. 결국 훈련 과정 순서도가 뒤집힌 것과 같은 형태로 룰이 작성되게 되며 Fig 6은 이러한 일반적인 관계를 보여준다. 이러한 규칙들과 앞서 설계한 변수들을 JESS 룰로 표현하게 되면 Fig. 7과 같이 된다. Fig. 7을 보면 IF절에 SP변수인 교사가 설정하는 부품이나 자동차의 상태, TR변수인 훈련자의 행위, LS변수에는 시뮬레이터의 현재 상태가 들어가며 THEN절에는 그 결과로 나타나는 시뮬레이터의 상태 또는 엔진의 상태, 점검 도구의 측정 결과 등이 표현된다.

3.3.3 시동 시퀀스의 표현

최근에 출시되고 있는 차량들은 스마트키의 시동버튼을 누르면 시동이 걸리도록 하고 있다. 이러한 스마트 버튼 시동 시스템은 단 하나의 버튼만을 가지고 Fig. 8과 같이 특정 조건에 따라 4가지 상태가 전환되게 된다. 이 시스템에선 기존의 자동차 키를 이용한 시동 시스템과 달리 입력 값으로 ‘버튼을 누른다’라는 행위만 발생하게 되며, 이 행위만으로 룰을 구현하기 위해서는 현재 시동 모드가 어떠한 상태에 놓여 있는지를 기억하여야 한다. 본 논문에서는 LS_IGMODE 라는 시뮬레이터 내부 변수를 정의하여 Fig. 8과 같이 시동 시퀀스를 설계하였다.

Fig. 8에서 상태 전이 화살표에 있는 SSB는 스마트버튼을 의미하고, BRAKE는 풋브레이크를 밟음을 의미한다. 상태를 나타내는 도형에 있는 ACC는 최소 요구되는 전기가 흐르고 있는 상태이며, IGN은 ACC상태에서 모든 전기 장비에 전기가 흐르게 되는 ON상태를 의미한다. START는 시동이 걸린 상태를 의미하고, ENGINE RUN은 자동차가 시동이 걸린 START 상태에서 엔진의 RPM

이 일정 조건을 만족하게 됐을 때 엔진이 동작하는 상태로 넘어가는 것을 의미한다. 예컨대 시동이 걸리는 과정을 보면, 맨 처음 시동이 걸리지 않은 OFF상태에서 SSB을 누르게 되면 ACC상태가 되고, 브레이크를 밟게되면 START 상태가 되어 시동이 걸리게 된다. 이와 같은 시동 시퀀스에 대한 JESS rule은 Fig 9와 같다.

Fig. 9에서는 스마트키를 이용해 시동을 거는 차량에 대해 정비 훈련을 하는 경우, 시동 상태에 관한 rule들을 보여주고 있다. (1) on_state1 룰은 시동이 걸리지 않은 채로 ACC상태에서 시동키를 눌렀을 때 시동이 켜지도록 하는 rule로서, 배터리가 정상이고, 시동버튼이 눌러졌다 떤 상태이고, 풋 브레이크를 밟지 않은 상태에서 ACC상태인 경우 시동이 걸리도록 추론하는 rule이다. (2) on_state1 룰은 ON 상태에서 기어가 P가 아닌 경우 다시 ACC로 전환되도록 하는 rule이다.

4. 구현 및 실행 결과

4.1 JESS rules

본 연구에서 자동차 동작을 추론하기 위한 룰은 Table 2와 같이 각 기능별로 분리할 수 있다. define_facts.clp과 일은 추론을 하기 위한 rule을 작성하기 앞서, 사실을 저장하기 위한 틀(template)을 정의한 파일이다. 또한 추론에 필요한 slot 및 fact등을 정의하였다. tool_rule.clp에는 자동차 정비에 필요한 스캔툴, 멀티미터, 충전기, 게이지 바와 같은 측정 도구들에 관한 rule들이 정의되어있다. 측정 결과는 tool_rule.clp에 정의된 rule과 electric_rule.clp에 정의된 rule에 의해서 추론되어 만들어지게 된다. compression.clp파일에는 엔진 실린더에 대한 압력을 측정하기 위한 rule이 1~4번 실린더에 대해 정의되어 있다. etc_system.clp파일에는 스톱밸브를 전개하고 전폐했을 경우와 공회전 등에 대한 rule이 정의되어 있으며, igcoil.clp에는 점화코일에 대한 상태를 나타내는 rule이 정의되어 있다. 마지막으로 malfunction.clp에는 엔진 부조 현상에 관한 rule 들이 정의되어 있다.

4.2 논리 시뮬레이터 프로그램

논리 시뮬레이터 프로그램은 시뮬레이터 동작을 확인하기 위해 작성된 것으로서 Java로 구현하였으며 실행 화면은 Fig 10과 같다. 시뮬레이터 화면은 크게 좌우로 구분 할 수 있는데, 좌측패널은 교사 또는 훈련자가 변수를 설정 할 수 있는 탭들이 위치해 있고, 우측 패널은 좌측 패널에 설정된 입력값에 대해 JESS엔진이 추론을 마치고

```

(1)
(defrule on_state1
(tCar {
  SP_ENGN_BATTERY = "GOOD" &&
  TR_BODY_IGBTN = "RELEASED" &&
  TR_BODY_FOOTBREAK = "OFF" &&
  LS_IGMODE = "ACC" &&
  LS_START_STATE = "FALSE"
})
=>
  (put "TR_BODY_IGBTN" "OFF")
  (put "LS_IGMODE" "ON")
  (debug "on_state1")
)

(2)
(defrule on_state2
(tCar {
  LS_START_STATE = "FALSE" &&
  TR_BODY_IGBTN = "RELEASED" &&
  SP_ENGN_BATTERY = "GOOD" &&
  TR_BODY_FOOTBREAK = "OFF" &&
  LS_IGMODE = "ON" &&
  TR_BODY_GEAR != "P"
})
=>
  (put "TR_BODY_IGBTN" "OFF")
  (put "LS_IGMODE" "ACC")
  (debug "on_state2")
)
    
```

Fig. 9. An example of JESS rules related to car start up

Table 2. List of simulation rule files

File name	Rules related to	Number of rules
define_facts.clp	data loading, templates, user definition function	n/a
tool_rule.clp	diagnosis tools	48
start_rules.clp	ignition sequence	38
electric_rule.clp	electric system	26
compression.clp	fuel system	24
etc_system.clp	ETC system	18
igcoil.clp	ignition system	16
malfunction.clp	engine malfunction	35

난 결과가 표시된다. 예를 들어, 좌측 패널에서 Supervisor가 훈련을 위해 고장 설정을 하면, Trainee는 정비를 위한 동작을 취하게 된다. 이렇게 입력된 값에 대해 추론된 결

과는 우측 패널에서 확인할 수 있게 된다. 우측 패널은 위아래로 나누어지는데 위쪽 패널은 입력값에 대한 추론 결과가 출력되며, 아래쪽 패널은 내부 상태 변수(Internal Model)들이 출력된다. 내부 상태 변수는 실제로 훈련자가 확인할 수 없을 수도 있는 추상적인 변수로 존재하지만, 시뮬레이터가 논리적으로 어떻게 동작하는지 중간 상태를 확인할 수 있도록 구성되었다. Fig. 10의 좌측 패널 하단을 보면 멀티미터, 스캔툴, 압축압력계이지 버튼을 눌러 Fig. 11과 같이 해당 측정 도구를 출력할 수 있으며, 도구를 이용하여 측정된 결과를 화면 우측 패널에서 확인할 수 있다.

4.3 실행 결과

시뮬레이터에서 전달된 자동차 정비 과정 변수 값들에 대한 추론결과가 올바른지에 대한 여부를 확인하기 위해 시동 불량 현상에 대한 훈련자의 정비 과정을 테스트하였다. 자동차 시뮬레이터에 시동을 걸어보기 위해 풋브레이크 변수를 TRUE로 설정한다. 스마트키가 적용된 차량에서는 브레이크를 밟지 않고 스마트키를 누르면 전원만 공급되는 ON과 ACC상태로 전환된다. 실제 시뮬레이터에 풋브레이크를 밟지 않고 스타트키를 누른 경우엔 시동이 걸리지 않고 전원만 공급됨을 확인할 수 있다. 브레이크를 밟은 뒤 스타트키를 누르면 엔진 크랭킹 소리와 함께 시동이 걸리는 것을 Figs. 12~13과 같이 확인할 수 있다.

Fig. 12에서 Trainee는 풋 브레이크 동작이 OFF 상태였고, 시동이 걸리지 않은 상태이므로 RPM 수치값을 추론한 결과를 나타내는 변수 LS_BODY_RPM_GAGE의 값은 0임을 볼 수 있다. 또한 내부 상태 변수인 시동상태는 FALSE가 저장되어 있는 것을 확인할 수 있다. Fig. 13에서는 Trainee가 풋 브레이크 동작을 수행하여 ON 상태가 되었고, 시동키 START를 눌러 시동을 걸어 RPM 수치값을 추론한 결과를 나타내는 변수 LS_BODY_RPM_GAGE의 값이 600RPM이 된 것을 확인할 수 있다. 또한 내부 상태 변수인 시동상태는 TRUE가 되어 시동이 걸린 상태임을 확인할 수 있다. 시동 이외에도 히터 동작, 에어컨 동작 등 기본 동작들도 모두 작동되며, 훈련자는 스타트 모터, 배터리, 인젝터, 삼원촉매 등의 검사를 수행할 수 있도록 구현되어 있다.

Fig. 13에서 내부 상태 변수의 값을 살펴보면, 시동상태가 TRUE로 설정되고 시동모드가 ON으로 설정된 것으로 볼 수 있다. 자동차의 시동 모드는 START-ON-ACC-OFF로 총 4단계의 모드가 존재한다. 시동을 걸면 START 모드로 전환 되지만, 시동이 걸리고 난 뒤엔 ON모드로

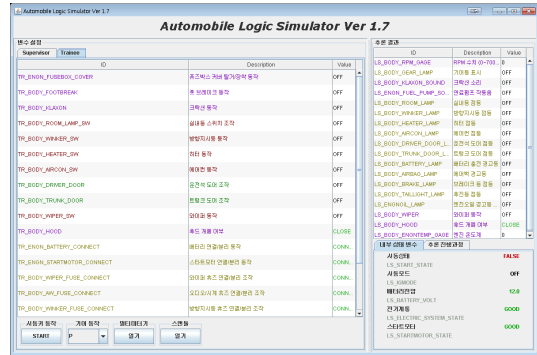


Fig. 10. The logic simulator program

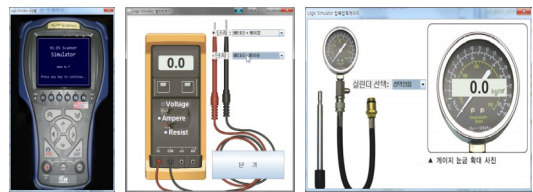


Fig. 11. Measuring tools (Scan tool, Multimeter, Compression Pressure gauge)

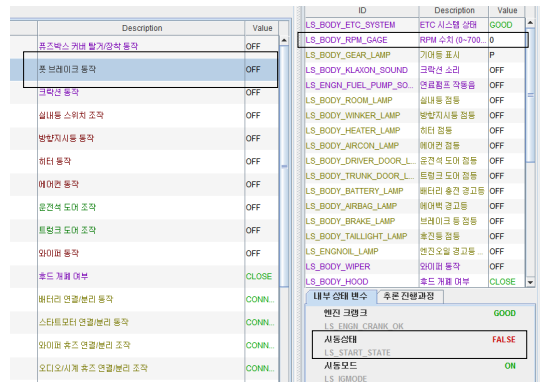


Fig. 12. The simulator state before start up

변경 된다. 내부 상태 이외에도 실제 추론된 변수들의 값을 확인하면 RPM수치가 600으로 나오고, 기어등과 엔진 온도가 추론되어 표시되고, 앞서 시동을 걸기위해 풋브레이크를 밟고 있었으므로 브레이크 등등 점등 되어있는 것을 볼 수 있다.

본 시뮬레이터는 UI와 추론모델이 독립적인 형태로 되어있다. 따라서 데이터를 전달 형태와 변수명만 일치 시켜준다면 다른 형태의 시스템에서도 활용이 가능하다. Fig. 14는 (주)테크빌에서 제작된 자동차 정비 훈련 시스템에 본 논문의 지능형 에이전트와 데이터 교환 방식은

Description	Value	ID	Description	Value
프론트박스 기변 동기화/차량 동작	OFF	LS_BODY_ETC_SYSTEM	ETC 시스템 상태	GOOD
후 브레이크 동작	ON	LS_BODY_RPM_GAGE	RPM 수치 (0-700)	600
크리크 동작	OFF	LS_BODY_XLAXON_SOUND	기어동 표시	P
비상등 스위치 조작	OFF	LS_BODY_FUEL_PUMP_BO	크랙션 소리	OFF
실내등 스위치 조작	OFF	LS_BODY_EMERGENCY_LIGHT	엔진실린 작동음	OFF
방탈지시등 동작	OFF	LS_BODY_WINKER_LAMP	비상등 점등	OFF
히터 동작	OFF	LS_BODY_AIRCON_LAMP	실내등 점등	OFF
배터리 동작	OFF	LS_BODY_HEATER_LAMP	히터 점등	OFF
문전석 도어 조작	OFF	LS_BODY_TRUNK_DOOR_L	배터리 충전 경고음	OFF
트렁크 도어 조작	OFF	LS_BODY_BATTERY_LAMP	배터리 경고음	OFF
와이퍼 동작	OFF	LS_BODY_BRAKE_LAMP	브레이크 점등	ON
후드 개폐 여부	CLOSE	LS_BODY_TAILLIGHT_LAMP	후진등 점등	OFF
배터리 충전	N/A	LS_BODY_WIPER	와이퍼 동작	OFF
배터리 고장	N/A			
배터리 연결상태 동작	CORNL			

변수	값
엔진 크랭크	GOOD
시동상태	TRUE
시동모드	ON

Fig 13. The simulator state after start up



Fig. 14. The car maintenance training system using the logic simulation

맞춰 시뮬레이션을 수행하는 화면을 보여준다. 본 논문에서 구현한 논리 시뮬레이션 모델은 3D 가상현실, 동작 인식 등과 같은 여러 가지 다양한 기술과 함께 체험형 훈련 시스템 구축에 활용될 수 있다.

5. 결론 및 향후 연구

본 논문에서는 JESS를 이용한 지능형 시뮬레이터를 설계·구현하였고, 이 지능형 시뮬레이터를 체험형 자동차 정비 훈련 시스템에 적용하였다. JESS를 이용한 소구

모의 전문가 시스템 구축에 대한 연구는 진행 되었으나, 본 연구처럼 훈련자의 다양한 동작과 예외적 행위에 대한 처리를 수행할 수 있는 체험형 훈련 시스템을 시뮬레이션 하는 연구는 기존에 없었다. 이러한 시뮬레이션을 하기 위해 Internal Model이라는 계층적 형태의 변수들을 설계 하였고, Java 기반의 어플리케이션에서 테스트 해 보았다. 또한 이 시뮬레이터를 가지고 훈련자의 행동들을 분석하여 추론 결과를 알려주는 시스템을 구축하였다. 본 논문에서는 지능형 시뮬레이터를 구현하여 자동차 정비를 위한 훈련 시스템에 접목시켰지만, 이러한 논리 모델은 자동차 뿐만 아니라 선박, 항공, 미사일 등 다른 분야에도 충분히 응용 할 수 있을 것이다.

향 후 다른 다양한 개인화 기법을 대해 연구하고 접목 시키면 더 나은 정비 훈련 시스템을 구축 할 수 있으리라 기대 되며, 전문가가 직접 룰-지식베이스를 손쉽게 구축 할 수 있도록 전문가 시스템 도구의 개발이 이루어져야 한다. JESS 룰을 작성해주는 전문가 시스템 도구가 개발 되면 차량 전문가가 시스템 전문가나 프로그램 개발자의 도움 없이도 시뮬레이터의 동작을 수정하고, 유지보수하며, 새로운 기능을 추가하는 일이 가능해 질 것이다.

References

1. Adsavakulchai, S, Ngamdumrongkiat N, Chuchirdkiatskul E(2011), "E-Learning for Car Faulty Diagnosis", International Journal of Information and Communication Technology Research(IJICT), Vol. 1. No. 1. pp. 20-26.
2. AlHamad, A.Q., Al-Omari, F. Yaacob, N(2012), "Applying JESS Rules to Personalized Learning Management System (LMS) Using/ Online Quizzes", Interactive Collaborative Learning (ICL). pp. 1-4
3. AlHamad, A.Q., Mohammed Ali Akour, Farouq Al-Omari (2012), "Personalizing Student's Profiles Using JESS in A Learning Management System(LMS)", International Conference on Interative Mobile and Computer Aided Learning(IMCL), IEEE. pp. 109-112
4. Andries van Dam, David H. Laidlaw, Rosemary Michelle Simpson(2002), "Experiments in Immersive Virtual Reality for Scientific Visualization", Computers & Graphics 26, p.535-555.
5. Arun N Nambiar, Anish K. Dutta(2010), "Expert System for Student Advising using JESS", International Conference on Educational and Information Technology(ICEIT), Vol.1, pp 312-315.
6. Brett E. Shelton(2003), "How Augmented Reality Helps

- Students Learn Dynamic Spatial Relationships”, Unpublished doctoral dissertation, University of Washington, 2003.
7. Dede, C(1995), “The evolution of constructivist learning environments : Immersion in distributed”, virtual worlds, Educational Technology.
 8. en.wikipedia.org/wiki/elearning
 9. Fernando A. Mikic Fonte, Juan C. Burguillo, Martin Llamas Nistal(2012), “An intelligent tutoring module controlled by BDI agents for an e-learning platform”, Expert Systems with Applications, Volume 39, Issue 8, Pages 7546-7554.
 10. http://www.jessrules.com
 11. Khumukcham Robindro, Shikhar Kr. Sarma(2013), “JESS Based Expert System Architecture For Diagnosis Of Rice Plant Diseases: Design And Prototype Development”, International Conference on Intelligent Systems Modelling & Simulation (ISMS), pp 674-676.
 12. Laxmisha Rai, Nongliang Sun, Fasheng Liu(2013), “Jess based On-demand Multimedia Player with Data Retrieval and Streaming Service”, International Conference on Image and Graphics. pp 628-633
 13. Lina Chen(2008), “About e-Learning Application in Communication Teaching”, International Conference on Web-based Learning, IEEE. pp 87-92
 14. McLuhan, M.(2011), “Understanding Media: The Extensions of Man, Cambridge”, USA: MIT Press, 1994.
 15. Minhong Wang(2011), “Integrating organizational, social, and individual perspectives in Web 2.0-based workplace e-learning”, Information Systems Frontiers, Springer. Vol. 13. pp 191-205
 16. Nora Houari, Behrouz Hodayun Far(2004), “Application of Intelligent Agent Technology for Knowledge Management Integration”, International Conference on Cognitive Informatics, IEEE, pp 240-249.
 17. Yin-Ho Yao, Amy J.C. Trappey, C.C. Ku, Gilbert Y.P. Lin, J.P. Tsai, P.-S. Ho, Benson Hung(2005), “Develop an Intelligent Equipment Maintenance System Using Cyber-enabled JESS Technology”, International Conference on Mechatronics, IEEE, pp 927-932.
 18. Sungho Kwon, Jungeun Lee(2010), A Development Study of Learner Profile Model for u-Learning, Hanyang University Institute of Educational Technology, pp. 1-26
 19. Gyungsuk Kil, Jaeyong Song, Ilkwon Kim, Seungbo Moon, Jangwoo Kwon(2007), “Implementation of an Expert System to Supervise GIS Arrester Facilities”, Journal of the Korean of Illuminating and Electrical Installation Engineers, Vol. 21, No. 1, pp 75-81.
 20. JeongSeok Kim Bonghoi Kim Hoiyoung Jung, Euiin Choi (2008), “Ontology Reasoning in Intelligent U-learning for adaptation of learning contents”, Korean Society for Internet Information, Vol. 9, No. 2, pp. 267-270.
 21. Jung-Hwa KIm, Young-Hee Kang(2011), “A Study on Designing and Developing a Personalized e-Tutor to Facilitate e-Learning”, THE JOURNAL OF KOREAN ASSOCIATION OF COMPUTER EDUCATION, Vol. 14, No. 1, pp 91-109
 22. Tai-Jin Kim, Keel-Soo Rhyu, Gyung-Suk Kil, Chang-Gyu Seong, Il-Kwon Kim, Jong-Il Park, La-Kyoung Weong(2006), “Design Methodology of Expert System for aging diagnosis of Arrester”, The Korean Society of Marine Engineering, pp. 69-70.
 23. Gilsik Park, Sunghul Park, Daesung Park, Juntae Kim (2012), “Ontology-based logic simulator design for car diagnosis e-training system”, Korea Intelligent Information System Society, pp 218-222.
 24. Daesung Park, Juntae Kim(2013), “Jess-based Simulator for Automobile Maintenance Training”, Korean Institute of Information Scientists and Engineers, pp 634-636.
 25. Soyoun Park, Viet Tran Hoang, Anh Nguyen Hoang, Gyohwa Bae, Jaewoon Lee, Dongho Kim(2013), “Automobile Maintenance Training System using Phased Learning based on Virtual Reality”, Journal of KISS, Vol. 19, No. 12, pp 663-667.
 26. Yoon-ae Ahn(2010), “Design of JESS-based Inference Rule for the Monitoring of Harmful Gas”, International Conference on Convergence Content, Vol. 8, No. 1, pp 396-398.
 27. Hyogap Lee, Junho Choi, Taejin Ha, Pankoo Kim(2012), “The Design of Vehicle Diagnosis Support System Using Ontology”, Journal of , Vol 10, No 3, pp. 133-140
 28. Sangho Ha, Ildae You, In-Gook Chun, Sang-Heum Park, Sun-Joo Kim(2003), “The Internet based Expert System for Gastroenteritis”, Journal of Korea Multimedia Society, Vol. 6, No. 6, pp.1079-1087.



박길식 (gilsikpark@dongguk.edu)

2009 중앙대학교 컴퓨터공학과 석사
2010~현재 동국대학교 컴퓨터공학과 박사과정
2012~현재 (주)소프트캡트 책임연구원

관심분야 : 추론엔진, 온톨로지



박대성 (basic@dongguk.edu)

2014 동국대학교 컴퓨터공학과 석사

관심분야 : 추론엔진, 이-러닝



박기현 (key@tekkville.com)

1986 서울대학교 전기공학과 학사
1996 한국과학기술원 전기 및 전자공학과 석사/박사
2009~현재 (주)테크빌닷컴 부사장

관심분야 : 이러닝, 데이터마이닝



김준태 (jkim@dongguk.edu)

1986 서울대학교 제어계측공학과 학사
1993 University of Southern California 전기공학과 석사/박사
1995~현재 동국대학교 컴퓨터공학과 교수

관심분야 : 인공지능, 기계학습, 데이터마이닝, 추천시스템