

# 고고보드를 이용한 능동적 참여 모의실험을 위한 NetLogo 확장 모듈

정회원 응 홍 우\*, 종신회원 소 원 호\*\*

## NetLogo Extension Module for the Active Participatory Simulations with GoGo Board

Xiong Hongyu\* *Regular Member*, So Wonho\*\*<sup>o</sup> *Lifelong Member*

### 요 약

본 논문에서는 기존 HubNet 기반의 참여 모의실험의 한계를 극복하기 위한 능동형 참여 모의실험 (Active Participatory Simulation; APS) 학습 구조를 제시하고, 이를 위한 고고보드용 NetLogo 확장 모듈을 자바로 개발한다. NetLogo는 복잡하게 보이는 과학현상의 이면에 존재하는 복잡계를 모델링할 수 있는 에이전트 기반 모델링 (Agent Based Modeling) 언어다. 이것과 HubNet을 이용하면 모의실험이 수행되는 동안 학생은 하나의 에이전트로서 이 실험에 참여할 수 있다. 하지만 HubNet에서는 서버만이 외부장치와 연결된다. 따라서 고고보드를 이용한 환경 데이터 및 사용자 입력을 다수의 클라이언트를 통하여 수신할 수 없어 이중초점 모델링 기반 학습이 불가능하다. 이에 클라이언트에 연결된 고고보드의 입력 정보를 TCP/IP 소켓을 이용하여 수신하고 보드를 제어하는 자바 확장 모듈을 개발한다. 또한 HubNet과 이 확장 모듈을 사용한 APS 학습 구조 모델링 방법과 이를 위한 NetLogo 프로그래밍을 소개한다. 마지막으로 다양한 APS 학습 구조에 따른 예시를 제시하고 응답처리 지연 시간 관점에서 평가하여 과학분야에 활용될 수 있는 방안을 모색한다.

**Key Words** : NetLogo 확장 모듈, 참여 모의실험, 에이전트 기반 모델링, HubNet, 복잡계 학습

### ABSTRACT

Flooding based routing protocols are usually used to disseminate information in wireless sensor networks. Those approaches, however, require message retransmissions to all nodes and induce huge collision rate and high energy consumption. In this paper, HoGoP (Hop based Gossiping Protocol) in which all nodes consider the number of hops from sink node to them, and decide own gossiping probabilities, is introduced. A node can decide its gossiping probability according to the required average reception percentage and the number of parent nodes which is counted with the difference between its hop and neighbors' ones. Therefore the decision of gossiping probability for network topology is adaptive and this approach achieves higher message reception percentage with low message retransmission than the flooding scheme. Through simulation, we compare the proposed protocol with some previous ones and evaluate its performance in terms of average reception percentage, average forwarding percentage, and forwarding efficiency. In addition, average reception percentage is analyzed according to the application requirement.

※ 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2009-0086676)

\* 순천대학교 과학정보융합학과 (hongyu.xiong@gmail.com)

\*\* 순천대학교 컴퓨터교육과 통신망시스템연구실(whso@sunchon.ac.kr), (°:교신저자)

논문번호 : KICS2011-07-298, 접수일자 : 2011년 7월 15일, 최종논문접수일자 : 2011년 11월 4일

## I. 서 론

기존 7차 과학교과 교육과정은 탐구학습을 강조하여 탐구과정과 탐구활동을 통한 과학의 기본 개념을 이해하고 탐구 능력과 태도를 길러 일상생활의 문제를 창의적이고 합리적으로 해결하는 것을 목표로 한다. 이것은 구성주의 시각에서 다양한 감각 경험을 통해 학습자들이 자율적으로 참여할 수 있는 실험활동이 필요함을 의미한다. 또한 최근 첨단기술을 통한 학문간, 기술간 융합을 활용하여 수학, 과학, 예술, 문화, 그리고 기술 분야를 융합적, 수렴적, 전체적으로 접근하는 시도가 있다. 이러한 요구를 수용하기 위하여 2009년 개정 교육과정의 과학교육은 융합과학을 지향하고 있으며 과학교육의 목표를 인성을 기반으로 한 창의성 개발에 두고 있다<sup>[1][2]</sup>.

실험활동을 통한 과학교육의 중요성을 알면서도 현실적으로 사전에 결정된 결과를 얻기 위해 지시 사항에 따라서 실험을 진행한다. 따라서 아와 같은 요리책식의 과학실험은 단지 학생들로 하여금 특수한 도구를 사용하는 학습이며 수업에서 목표하는 진정한 과학학습에 방해가 된다. 뿐만 아니라 과학 실험의 데이터를 수집하고 분석 및 해석하는 과정이 많이 요구되어 무의미한 실습으로 인식될 수 있다. 따라서 과학수업을 위한 실험학습에 치밀한 계획과 조정이 요구된다<sup>[3]</sup>.

ITC(Information & Communication Technology)를 활용한 실험 수업은 전통적인 실험을 수행할 때보다 변인통제가 수월하고 실험 데이터를 쉽게 수집하고, 결과를 도표나 그래프로 나타낼 수 있다. 특히, 로봇을 활용한 교육은 전자공학적인 설계와 각종 센서의 활용과 물리적 사고를 바탕으로 한 로봇 제어 프로그래밍이 가능하다. 따라서 수학과 과학 교육이 정보 및 IT 기술과 효과적으로 결합되면 학생들의 문제해결력, 탐구 및 실험의 정교성, 창의력, 그리고 논리적 사고력 등 통합적 능력 배양될 것으로 기대하고 있다. 특히, 복잡계 과학분야에서는 IT-BT 등의 융합분야에서 통계를 사용한 모델링과 모의실험의 중요성이 부각되고 있다. 국내의 과학교육에도 이러한 변화를 수용하고 국내외적으로 그 사용이 확대되고 있는 공개 SW의 교육적 활용 방안을 모색할 수 있다. 특히, 공개 SW를 이용한 모델링/모의실험을 로봇과 결합하는 연구는 에이전트 기반의 모델링(Agent based Modeling; ABM) 기법과 이와 결합되는 로봇, MCU (Micro Control

Unit) 보드와 같은 외부모듈의 활용에 많은 관심을 보이고 있다. 하지만 실습할 수 있는 외부모듈이 한 명의 사용자만을 지원하기 때문에 실험 참여자가 동시에 협력할 수 있는 교육환경을 제공할 필요성이 있다<sup>[4][5]</sup>.

HubNet은 자바로 개발된 참여 모의실험(Participatory Simulation) 교육 도구로서 NetLogo와 연동되어 사용된다. 망으로 연결된 학생들은 HubNet 클라이언트로서 서버에 접속한다. 다수의 참여자가 NetLogo로 구성된 ABM HubNet 서버에 참여하여 하나의 에이전트로서 데이터를 입력하고 그 결과를 공유할 수 있다. 따라서 미생물, 환경 생태, 물리적 작용, 그리고 분자 활동과 같은 과학현상의 이면에 존재하는 복잡계를 이해하기 쉽고 재미있게 모델링하여 많은 학생들이 동시에 참여할 수 있다. 그러나 학습자가 직접 데이터를 입력해야 하는 불편함으로 교육 시간 이후, 또는 장시간 모니터링해야 하는 실험은 불가능하다. 또한 HubNet 클라이언트 시스템과 연결된 외부모듈을 사용하지 못하는 한계가 있다<sup>[6][10]</sup>.

본 논문에서는 로봇 교육에 활용되는 기존에 한 개의 고고보드와 연결되는 NetLogo 모델과 학습자의 직접 입력에 의한 참여 모의실험만이 가능한 HubNet의 한계를 극복할 수 있는 방안을 제시한다. HubNet 클라이언트에 연결된 고고보드를 원격으로 연결하는 TCP/IP 소켓 기반 고고보드를 NetLogo 확장 모듈을 자바로 개발한다. 따라서 이것을 이용하면 HubNet 클라이언트는 외부 모듈인 고고보드를 사용하여 데이터를 입력할 뿐만 아니라 다수의 학습자가 참여하는 모의실험의 진행과 결과를 같이 공유할 수 있다. 뿐만 아니라 HubNet 서버에 의한 실험 결과를 적용하여 클라이언트에 연결된 보드를 제어할 수 있다.

논문의 구성은 다음과 같다. 먼저 2장에서는 참여 모의실험의 핵심 프로그램 언어인 NetLogo와 HubNet을 소개한다. 3장에서는 세 가지 능동적 참여 모의실험을 제시하고 이를 위한 HubNet 기능을 확장할 수 있는 TCP/IP 소켓 기반의 고고보드를 NetLogo 확장 모듈을 설계하고 구현한다. 4장에서는 APS 기반 다양한 학습 모델과 그 평가를 수행하고, 마지막으로 5장에서 본 연구에 대한 결론과 향후 연구 및 활용 방안을 논한다.

## II. 관련 연구

이 절에서는 자연과학 및 사회과학의 현상을 모

의실험하기 위한 프로그램 가능한 모델링 환경인 NetLogo를 소개한다. 또한 NetLogo 기반의 참여 모의실험이 가능한 두 종류의 HubNet을 설명하고 그 단점을 살펴본다. 마지막으로 외부 모듈인 고고보드와 이에 대한 확장모듈을 소개한다.

최근 생태계에서 종들의 상호작용과 화학 반응에서 분자의 충돌, 그리고 곤충들의 먹이 모으기 등과 같은 다양한 자연현상을 과학자들이 학습하기 위하여 에이전트 기반 모델링을 상당히 많이 이용하고 있다. NetLogo는 이러한 현상을 묘사할 수 있는 기능을 제공한다. NetLogo에서는 대상이 되는 모델체계에서 기본 구성요소 (예를 들면 포식자, 분자, 개미)를 에이전트로 표현하고 요소에 대한 기능과 상호 관계에 따른 규칙을 정의한다. 따라서 다중행동과 요소간의 상호작용에 의한 누적된 영향으로 이루어진 체계를 복잡계 (complex system)라 하는데 기존의 슈퍼컴퓨터에서 수행하던 작업을 손쉽게 접근할 수 있도록 개발되었다. 에이전트에 제공되는 많은 명령어와 에이전트를 연결지어주는 링크를 통하여 돌연변이, 자기 구성, 랜덤화가 간단히 수행된다. 또한 학생들이 화학 반응을 좀 더 섬세하게 학습하고 싶다면, 화학반응의 구성요소인 분자 개개의 움직임과 특성을 관찰하는 기능도 제공한다. 따라서 모델링을 하는 사람인 모델러는 그림 1과 같이 NetLogo가 제공하는 GUI를 통하여 각 요소(에이전트)의 지역적, 핵심적 규칙을 프로그램으로 정의하면, 모델링 시스템이 지정되어 동작하고 모델러와 학습자는 전체 모델링 시스템의 동작 패턴과 모델링 시스템의 관찰하면 된다<sup>11)</sup>.

참여 모의실험(Participatory Simulation)은 다중 에이전트 모의실험과 유사하여 가상의 에이전트의 임무를 학습자가 수행한다. 또한 학습자가 협력하여 실험을 진행할 수도 있다. HubNet 서버에서 모델이 실행되며 망으로 연결된 HubNet 클라이언트 또는 계산기기(Calculator<sup>1)</sup>)를 통하여 학생들은 실험에 참여한다. 이러한 모델은 전체적으로, 예측하기 어려운 개인 참여자들의 행위가 서버에서 모델링되며 그 결과가 망을 통하여 각 클라이언트의 화면에 표시된다. 예를 들면 “Gridlock”과 같은 참여 모의실험에서 각 학생들은 복잡한 교통흐름이 발생하는 도시에서 한 개의 신호등을 개별적으로 제어한다.

“Disease Spread”에서는 각 학생들이 의사 또는

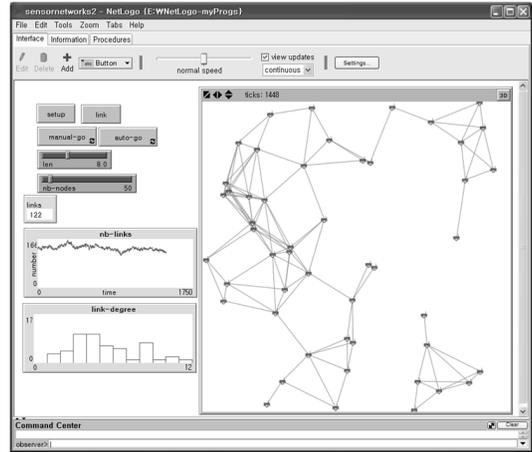


그림 1. NetLogo 예

환자로서 역할을 수행한다.

지금까지 소개한 방법이 HubNet을 이용한 첫 번째 방법이라면 최근에는 실제 환경에 대한 데이터를 센서를 통하여 수집하는 방법이 소개되었다. Calculator를 이용하는 방법이 하나의 예가 되는데, 물리적인 센서를 통하여 경험적인 데이터 기반 모델이 가능하고 가상 모델이 아닌 실제 모델을 사용할 수 있어 이중초점 모델링 (bifocal modeling)으로 정의되어 이용된다. 이것의 특징은 로봇을 이용한 교육에 적용되어 실제세계에 있는 장치를 설계하고 제어함으로써 가상 환경에서는 얻을 수 없는 예리, 잡음, 기계적 장점과 고장 등을 이해하고 학습할 수 있는 기회를 제공한다<sup>7)</sup>.

NetLogo 소프트웨어와 물리적 실세계를 연결하기 위하여 사용된 기술적 기반구조로 NetLogoLab이 소개되었다. NetLogoLab은 로봇을 제어하고 로그 데이터를 수신한다. 현재 NetLogo와 연결할 수 있는 로봇은 고고보드(GoGo Board<sup>2)</sup>)로 자바 확장 모듈 (gogo extension)이 개발되어 공개되었다. 두 가지 방식으로 고고보드를 사용할 수 있는데 먼저, 하나의 NetLogo 시스템에 연결하여 사용하는 방법이다. 고고보드를 통해 입력되는 센서데이터를 이용하여 모델링이 가능하고 모의실험도 할 수 있다. 또한 모델링의 결과에 따른 반응을 고고보드의 출력으로 보내는데 경보음, 발광다이오드, 그리고 모터 (서보모터, DC 모터)를 통한 제어가 가능하다. 두 번째 방법은 HubNet 서버에서 고고보드를 연결하

1) Texas Instruments사에서 개발한 기기로 Calculator HubNet 확장모듈을 사용하여 NetLogo와 결합되며 모의 참여실험에 이용될 수 있다.

2) MIT Media Lab의 Arnan Sipitakiat에 의해서 설계되고 개발되었다. 오픈 소스기반으로 쉽고 저가로 구성이 가능하며 현재 4.x 버전까지 공개되었다. (<http://gogoboard.org>)

는 것이다. 앞에서 설명하였듯이 서버의 프로세싱으로 클라이언트는 서버의 고고보드에서 입력된 정보를 수신하고 확인할 수만 있다.

위에서 소개한 HubNet과 고고보드는 각각 몇 가지 한계를 갖고 있다. 먼저 HubNet의 경우에, 클라이언트는 서버의 모델링에 의존한 데이터 입력만이 가능하고 단순히 결과만을 화면을 통하여 확인한다. 또한 NetLogoLab과 같은 외부 시스템을 HubNet 클라이언트 시스템에 연결할 수 없어 망으로 분산된 실험 환경 구축이 불가능하다. 또한 서버에 연결된 외부 시스템과 직접 상호작용하지 않고는 모델링에 참여할 수 있는 방법이 없기 때문에 진정한 의미의 이중초점 모델링을 구현할 수 없다. 마지막으로 NXT, Calculator는 상용 제품은 NetLogo 확장모듈이 제공되지만 추가적인 비용부담이 있기 때문에 범용으로 사용될 수 없다.

따라서 본 논문에서는 HubNet 클라이언트 시스템에 고고보드를 연결하여 이중초점 모델링이 가능하도록 NetLogo 확장모듈을 개발한다. 따라서 HubNet 서버에서 구성된 참여 모의실험에 클라이언트 학습자가 좀 더 능동적으로 관여하여 학습할 수 있는 방안을 제공한다.

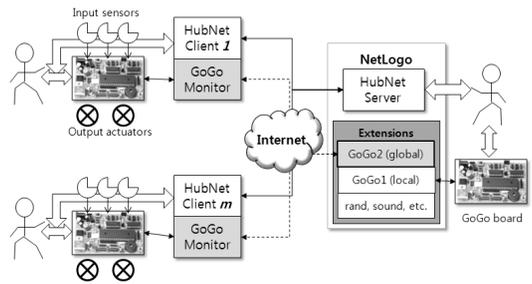
### III. 제안된 학습 모델 및 확장 모듈

본 절에서는 기존 참여 모의실험의 한계를 극복하고 이중초점 모델링이 가능한 새로운 개념의 참여 모의실험인 능동적 참여 모의실험 (active participatory simulation; APS) 학습 구조를 제시한다. 또한 이를 위한 NetLogo 확장 모듈을 자바로 개발하고 실험 학습자료 개발에 사용 방법을 제시한다.

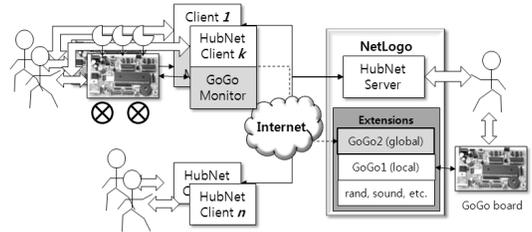
#### 3.1 능동적 참여 모의실험

이를 위하여 개발될 모듈의 요구사항을 먼저 분석한다. 기본적으로 본 논문에서 고려하는 모든 학습 자료와 실험은 ABM이 가능한 NetLogo로 개발되며 첫 번째 요구사항은 학습자가 에이전트로서 모델의 실험에 참여할 수 있어야 한다. 즉, 개발된 모델은 가상의 에이전트<sup>3)</sup>로 실행될 뿐만 아니라 학습자가 하나의 에이전트로서 실험에 참여할 수 있어야 한다. 따라서 학습자가 필요한 데이터를 입력해야 한다. 예로 신호등 끄기/켜기, 이동, 파라미터 값 변경을 들 수 있다.

3) NetLogo 프로그램 언어로 개발된 ABM의 기본 에이전트는 일반적으로 거북(turtle)이 모양을 하고 있다.



(a) 학습 구조-I (학생수 제한)



(b) 학습 구조-II (혼합형)

그림 2. 제시된 능동적 참여 모의실험 전체 구성도

둘째, 참여 모의실험의 서버에 입력되는 모든 데이터에 대한 처리 결과와 실험의 진행과정을 클라이언트인 학습자가 원격으로 모니터링 할 수 있어야 한다. 뿐만 아니라 서버의 처리 결과에 따른 제어가 가능해야 한다.

셋째, 클라이언트에 외부기기를 연결할 수 있어야 한다. 외부기기를 통하여 다양한 입출력이 가능해야 한다.

넷째, 연결된 외부기기를 통하여 참여 모의실험 서버로 다양한 환경 데이터 입력과 학습자의 추가적인 참여가 가능해야 한다.

다섯째, 다양한 실험학습을 수행하기 위해서는 실험 환경의 공간 제한성이 없어야 한다. 그래서 하나의 교실에서 뿐만 아니라 서로 멀리 떨어진 학습 환경에도 적용할 수 있어야 한다.

여섯째, 개발된 참여 모의실험 구조가 안정적인 것으로 수행되어야 한다. 망으로 연결된 학습 환경에서 실험이 진행되는 경우에도 연결이 끊고 데이터 전송 및 처리 속도가 빠르게 처리되어야 한다.

위와 같은 여섯 가지 요구사항을 만족시키기 위하여 그림 2(a)와 같은 능동적 참여 모의실험 (APS) 학습 구조-I를 제시한다. 인터넷을 중심으로 왼쪽은 학습에 참여하는 학생과 이들이 이용하는 PC, 고고보드, 그리고 관련 SW를 보이고 있다. 그림 오른쪽은 제안된 APS 구조의 중심이 되는 부분

으로 교수자 (또는 관리자)가 위치한다. HubNet 서버를 통하여 학습자의 클라이언트와 연결되며 확장 모듈 GoGo1 (local)에 이해서 자신의 고고보드와 연결된다. 또한 앞으로 설명할 확장 모듈 GoGo2 (global)에 이해서 학습자의 GoGo Monitor와 연결된다. 각 시스템이 인터넷으로 연결되며 학습자가 고고보드와 HubNet 클라이언트를 통하여 상호작용할 수 있다. 따라서 위에서 언급한 첫째, 둘째, 그리고 다섯 번째 요구사항을 만족한다. 또한 외부기기를 연결을 통하여 센서를 통하여 외부환경 데이터를 입력할 수 있을 뿐만 아니라 보드에 연결된 출력장치를 통하여 제어와 처리 결과 등을 확인 할 수 있다. 따라서 셋째와 네 번째 요구사항을 만족한다. 다만 모든 학습자는 공개 SW를 이용하여 개발된 두 개의 TCP/IP 소켓의 연결이 요구되며, 서버의 경우에는 참여 학습자의 수에 따라서 2배의 연결이 서버에 집중되는 단점을 갖고 있다.

따라서 참여 모의실험에 동시에 참여할 수 있는 학생 수를 제한하거나 그림 2(b)의 학습 구조-II와 같은 혼합형으로 구성할 수 있다. 이 구조는 HubNet 서버와 클라이언트의 안정적 특성을 이용하여 다수의 학습자를 실험에 참여시키고 반면에 APS 구조로 참여하는 학습자 수는 제한하는 방식이다 (4절에서 다양한 학습 모델 제시). 따라서 그룹 단위의 실험 학습에서 그룹 대표만 APS 구조를 사용하고 나머지 학습자는 기존 HubNet을 사용하여 참여 모의실험에 참여하는 모델이다.

마지막으로 그림으로 나타내지 않았지만 학습자가 GoGo Monitor와 고고보드를만 이용하여 참여 모의실험 학습에도 참여하는 학습 구조-III이 있다. 즉, 학습자가 HubNet 클라이언트를 사용하지 않고 고고보드를 PC에 연결하여 GoGo2와 연결하는 것이다. TCP/IP 소켓 연결 수가 줄어들고 한정된 공간에서 학습자가 참여 모의실험 학습에 동시에 참여할 수 있다.

제안된 APS 구조는 기존 HubNet에 의한 참여 모의실험과 Calculator를 이용한 이중초점 모델링 학습구조와 표 1과 같이 비교할 수 있다. HubNet의 경우에, 클라이언트 PC에 외부기기를 연결하여 학습에 참여할 수 없고 학습자에 의한 데이터 입력만이 가능하다. 이용하는 SW는 설치하여 사용이 가능하지만 소스를 공개하지는 않았다. Calculator의 경우에는 한정된 공간에서 하나의 출력 장치 (서버에 연결된 빔 프로젝터)를 통하여 실험 결과를 확인한다. 따라서 외부기기(calculator)를 연결하여 사용

표 1. 참여 모의실험 구조 비교

평가요소 \ 구조	HubNet	상용 (Calculator)	제안된 구조
에이전트 참여	○	○	○
원격 모니터링 및 제어	△	×	○
외부기기 연결 (client)	×	○	○
환경 데이터 입력/확장	×	○	○
분산 환경	△	△	△
안정성	○	○	△
공개 SW	△	×	△

이 가능하지만 원격 모니터링은 불가능하며 공개 SW도 사용하지 않는다.

제안된 APS 구조를 처음 구상할 때, HubNet SW가 공개 SW이고 소스를 수정할 수 있다면 HubNet과 GoGo Monitor의 일부 기능을 결합하려고 하였다. 하지만 소스를 공개하지 않았기 때문에 본 논문에서처럼 NetLogo 확장 모듈을 개발하여 GoGo Monitor와 연동하게 되었다.

### 3.2 NetLogo 확장 모듈 개발

앞에서 제시한 APS 구조를 사용하기 위해서는 GoGo Monitor의 GoGo Server (이후 고고서버로 표기)의 기능을 이용해야 한다. 따라서 본 절에서는 고고서버의 기능을 살펴보고 이를 이용한 NetLogo GoGo 확장 모듈 (그림 2에서 GoGo2 (global)로 표현)을 구현하고 사용방법을 제시한다.

GoGo Monitor는 PC에 연결된 고고보드와 연결되어 동작하는 SW이다. 고고보드의 입력포트를 통하여 센서 값 (sensor1~sensor8)을 읽고 표시하거나 출력 포트 (a,b,c,d)로 제어를 한다. 일반적으로 출력포트에는 DC 모터나 서보 모터가 연결되는데 모터의 방향과 각도 조절이 가능하다. 고고서버 탭은 그림 3의 오른쪽과 같이 실행되며 TCP 기반의 스트림 소켓을 통하여 포트번호 9873을 이용하여 연결 설정을 대기한다. 고고서버는 연결이 설정되면

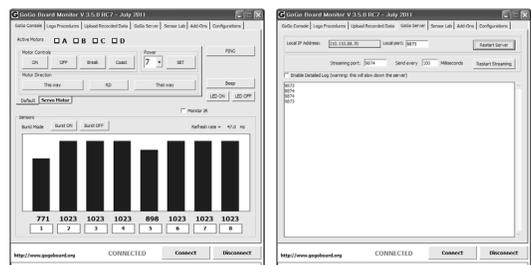


그림 3. 고고 모니터 콘솔 및 서버 실행

표 2. 고고 서버가 처리하는 명령어 (문자열로 처리)

Command	Parameter	Function
setcom c connect disconnect reconnect	$c \in \{1,2, \dots\}$	set com port connect to GoGo board disconnect from GoGo board connect to GoGo board
talktoport p setpower l on/off break coast thisway/thatway rd	$p \in \{a,b,c,d\}$ $l \in \{1,2,\dots, 7\}$	select a motor to be controlled set power level of motor as [On][Off] as [Break] as [Coast] as [Thisway][Thatway] as [RD]
beep		as [Beep]
ledon/ledoff burston/burstoff		as [LED ON] [LED OFF] as [Burst ON] [Burst OFF]
sensori sensormaxi sensormini	$i \in \{1,2,\dots,8\}$	get the value of sensor i

소켓으로 입력되는 표 2와 같은 명령어를 보고 그에 해당하는 내부명령어를 고고보드에 전송하여 그 결과를 클라이언트로 전송한다 (그림 4). 즉, 고고서버는 GoGo2(global) 확장 모듈로부터 연결 설정이 요구되어 소켓이 생성되면 수신되는 명령어에 따라서 고고보드를 제어하고 그 결과를 응답 메시지로 확장 모듈 GoGo2로 전송한다.

따라서 개발된 GoGo2는 NetLogo 서버에서 확장 모듈로서 사용되며 위에서 언급한 연결 설정과 해지, 명령어 전송과 수신을 담당해야 한다. 다음은 개발된 GoGo2 확장모듈 기본함수와 기능이다.

```
.ggs-connect [string_host, int_port]
.ggs-disconnect [connection_id]
.ggs-send [connection_id, command]
.ggs-recv [connection_id]
```

개발과정은 NetLogo 설치 디렉토리 아래 기존의

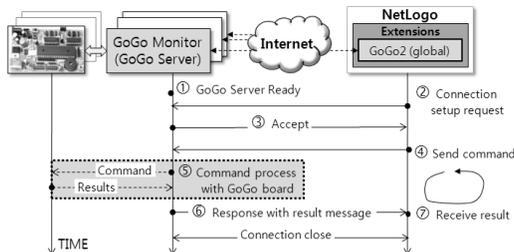


그림 4. 자바 확장 모듈 기능

```
1: let cid gogo:ggs-connect "210.110.88.77" 9873
2:
3: ifelse cid != 0
4: [output-print "connected"]
5: [output-print "failed"]
6:
7: gogo:ggs-send cid "sensor1"
8: print gogo:ggs-recv cid
9:
10: gogo:ggs-send cid "talktoport a"
11: gogo:ggs-send cid "setpower 6"
12: gogo:ggs-send cid "on"
13:
14: gogo:ggs-disconnect
```

그림 5. 개발된 확장모듈을 이용한 예시

확장 모듈이 있는 extensions/gogo의 기존 자바 소스 파일을 확장한다. 이를 위하여 추가되는 메소드에 대한 원시코드를 작성하고 NetLogo에서 프로그램을 작성하기 위한 classManager를 추가한다. 또한 JAR(Java ARchive)로 클래스를 모으기 때문에 Manifest.txt 파일을 생성한다. 마지막으로 jar 파일을 생성하여 extensions/gogo 디렉토리의 .jar 파일을 갱신한다. 새로운 확장모듈 개발은 NetLogo 사용자 매뉴얼을 참고하기 바란다<sup>[12]</sup>.

그림 5은 개발된 확장모듈을 이용한 NetLogo 프로그래밍 예시이다. 간단히 설명하기 HubNet을 사용하지 않고 그림 5와 같이 하나의 NetLogo 서버와 고고보드를 연결한 클라이언트가 고고서버를 실행하여 대기하는 것으로 가정한다. 1번 라인에서 클라이언트의 IP 주소와 대기 포트 번호 9873을 이용하여 연결을 설정한다. 에러가 없이 3번~5번을 지나면 7번과 8번 라인에서 고고보드 센서1번의 값을 읽어와 NetLogo 콘솔에 출력한다. 10번~12번은 출력 포트 a를 통하여 세기 6으로 모터를 구동하고 14번을 통하여 연결을 종료한다.

#### IV. 능동적 참여 모의실험과 평가

본 절에서는 GoGo2 확장모듈을 이용하여 제시된 능동 참여 모의실험 (APS)에 대한 모델링 방법을 소개한다. 먼저 기본적인 모델링 방법을 소개하고 이를 통한 다양한 예를 제시한다.

##### 4.1 APS 학습 모델링 방법

그림 6은 개발된 GoGo2 확장 모듈을 이용하여 기본적인 APS 모델링을 수행하는 접근 방법을 소개한다. 먼저 수업에 맞는 학습주제를 결정한 후 NetLogo 프로그래밍을 이용하여 모델링을 진행한다. 앞에서 제시한 학습 구조-I,II,III을 참고하여 종류를 선택한다. 학습 구조-I과II는 기본적으로

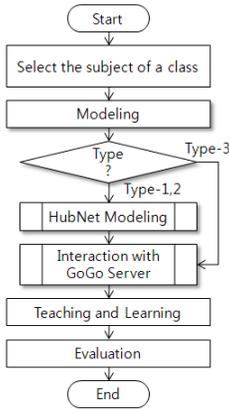


그림 6. APS 학습 모델링 개발 및 적용

HubNet을 사용하는 구조임으로 HubNet 모듈이 먼저 초기화되고 프로그램되어야 한다. 학습 구조-III은 고고 서버와 GoGo2만이 사용되기 때문에 HubNet 모듈없이 모델링을 진행한다. 작성된 모델은 학습을 진행하면서 평가와 수정을 통하여 완성된다.

그림 7은 구체적인 모델링 방법을 소개한다. 전체적인 구성은 교사와 다수의 학생들이 APS에 참여하는 것으로 가정한다. 또한 교사 시스템을 중심으로 모든 수업이 진행되며 APS 수업 모델은 이 교사 시스템을 중심으로 개발됨을 전제로 한다. 따라서, 개발된 GoGo2 확장모듈이 설치된 NetLogo의 procedures 탭에서 모델링의 시작은 HubNet 서버의 초기화와 준비를 위한 코드의 작성을 의미한다 (①). 다음으로 HubNet 클라이언트 프로그램을 학생들이 실행하여 HubNet 서버에 로그인하면 HubNet 서비스는 연결이 완성된다. 이때 학생들은 교사의 HubNet 서버 IP주소를 이용하고 모델링을 주제를 선택하여 로그인한다 (②). 모델링은 다음으

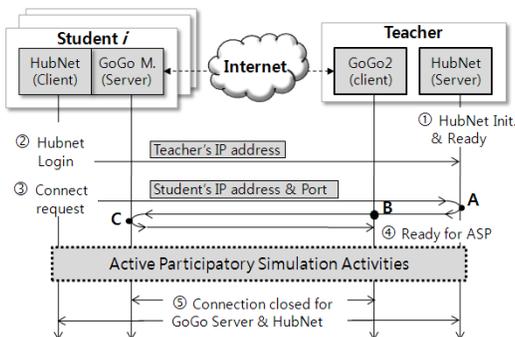


그림 7. APS모델링 초기화

```

1: if hubnet-message-tag = "Connect" [
2:   ask players with [player-id =
3:     hubnet-message-source] [
4:     ifelse not (conn-id > 0)
5:     [ set conn-id p-connect-ggs => call GoGo2
6:       hubnet-send hubnet-message-source "GoGo
7:         Server Connection" "Connected" ]
8:     [ hubnet-send-message hubnet-message-source
9:       "GoGo Server is already connected" ]
10:  ]
11: ]
21: to-report p-connect-ggs
22: set conn-id gogo:ggs-connect host port
23: ifelse conn-id > 0 [
24:   notice-server word "ggs-connect " word host
25:                                     word ":" port
26:   report true ]
27: [ report false ]
28: end
    
```

그림 8. HubNet과 GoGo2에 의한 고고 서버 연결

로 이벤트를 처리하면서 클라이언트의 요구 및 데이터를 수신하는 코드를 작성해야 한다. 이때, GoGo Monitor (그림에서 GoGo M.)가 실행되는 곳이 학생들의 시스템임으로 본 논문에서는 학생들이 자신이 이용하는 컴퓨터의 IP 주소를 교사 시스템으로 전송하게 한다. 이렇게 함으로써 교사는 모든 학생들의 연결을 직접하지 않고 학생들이 자신의 고고 서버 연결을 HubNet을 통하여 진행될 수 있도록 한다 (③). 연결과정이 순조롭게 진행되면 (④) APS 학습을 진행하고 모든 과정이 끝나면 연결을 종료한다 (⑤).

그림 8는 그림 7의 A→B→C로 이어지는 과정을 NetLogo로 프로그램 한 예를 보인다. 먼저 1번 라인에서 HubNet은 학생이 요구한 연결 이벤트를 확인한다 (A). 다음으로 5번 라인에서 p-connect-ggs를 호출함으로써 개발된 GoGo2 확장 모듈의 함수를 이용한다 (B). 즉 21번~28번에 정의된 함수를 통하여 연결을 설정하고 이에 적당한 동작을 6번부터 진행하여 처리한다 (C).

#### 4.2 APS 모델 적용 예시

- 환경 모니터링 앞에서 제시된 APS 학습 구조-I를 이용하면 원격으로 멀리 떨어진 지역간에 환경데이터를 고고보드로 수집하여 데이터를 모을 수 있다. 또한 한 곳에 모인 데이터는 NetLogo가 제공하는 monitor, plot, 그리고 output과 같은 툴 기능을 이용하여 결과를 표현한다. 뿐만 아니라 HubNet 서버의 환경설정을 통하여 HubNet 클라이언트로 동일한 결과를 모든 학습 참여자가 같이 공유할 수 있다. 기존 HubNet만을 이용한 학습에서는 진행할 수 없는 수업 모델로서 이중초점 학습이 가능

하고 현실과 밀접하게 관계된 모델 구성이 가능하다. 단점으로는 현재 TCP 기반의 연결지향형 소켓을 이용하기 때문에 동시에 많은 학생들을 수업에 참가시키지 못한다.

- **에너지 절약** 자동차와 자전거를 비교하여 같은 거리를 이동할 때, 에너지 소비와 시간과의 관계를 학습한다. 자동차에 대한 모델은 NetLogo turtle을 이용하고 자전거의 회전은 고고보드의 회전수 감지 센서를 이용하여 직접 학생들이 바퀴를 돌려 측정한다. 에너지와 시간과의 관계는 일반적인 차량의 에너지 소비효율과 주행속도 데이터를 이용한다. 실험의 진행과정은 고고 보드의 회전수를 측정하여 고고 보드 모터 a의 파워세기를 가변적으로 변화시킨다 (학습 구조-II 또는 III).
- **게임형 학습** 터치 센서를 이용한 레이싱 게임을 통하여 속도와 가속도 변화를 측정한다. 두 명의 학습자 또는 두 개의 그룹을 선정하여 고고 보드를 HubNet 클라이언트와 연결한다. 각 대표가 터치 센서를 누르는 속도를 감지하여 두 대의 자동차가 경주를 한다. 시간과 변화량을 모델링하여 주어진 거리를 이용한 속도와 가속도를 표현할 수 있다 (학습 구조-I). 그림 9(a)는 NetLogo를 작성된 HubNet 서버

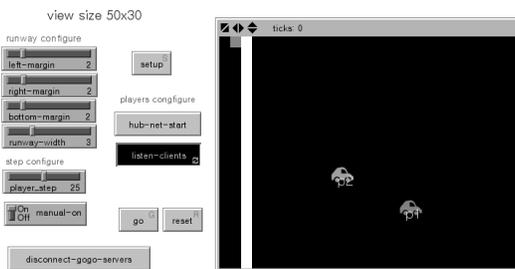
의 화면이며 현재는 두 명의 학생이 HubNet을 통하여 로그인하고 고고 서버를 연결하여 게임을 진행하고 있는 화면이다. 그림 9(b)는 학생들이 보고 있는 HubNet 클라이언트 화면이다. HubNet 서버와 같이 동일한 경기화면을 볼 수 있고 화면 아래에는 고고 서버를 학생들이 직접 접속할 수 있도록 툴바를 구성하였다.

### 4.3 APS 학습 모델의 성능 평가

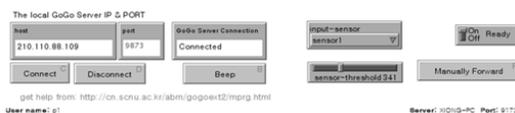
이 절에서는 제시된 ASP 학습 모델을 기존의 HubNet 학습 모델과 응답처리 지연 관점에서 비교한다. 또한 두 종류의 ASP 학습 모델을 구성하여 같은 관점으로 비교 평가한다. 먼저 구성된 HubNet 모델은 NetLogo 모델 라이브러리에서 제공하는 'template.nlogo'를 이용하였으며, 클라이언트가 약 0.5초 단위로 이벤트(버튼 클릭)를 발생하여 서버로 전송하고 서버는 그 메시지를 수신하여 결과를 화면에 표시하는 구조이다. 제시된 모델이 실행될 때, 수신되는 메시지와 처리 루틴의 응답처리 지연을 측정하기 위하여 매 100번 처리 시 (100 ticks<sup>4)</sup> 동안 루틴의 끝부분에서 지연시간을 측정한다. 이를 위하여 NetLogo 명령어인 reset-timer 명령과 timer 명령어를 이용한다.

제안된 ASP 학습모델은 학습 구조-I을 바탕으로 두 가지 모델을 사용한다. 첫 번째 모델은 클라이언트의 요구 없이 서버에서 클라이언트의 고고보드에 센서 데이터를 요청하여 그 결과를 서버 화면에 출력하는 구조이다. 두 번째 모델은 클라이언트가 약 0.5초 간격으로 센서 값을 요청하면 서버가 그 요청 메시지를 수신하여 클라이언트의 고고보드로 센서 요청 메시지를 다시 전송하여 그 값을 읽은 후에 화면에 표시하는 구조이다. 즉, 첫 번째 모델과 두 번째 모델의 차이는 클라이언트에 연결된 고고보드의 센서 값을 클라이언트의 요청 없이 수신하는 경우와 그렇지 않은 경우로 나눌 수 있다.

그림 10(a)는 HubNet모델과 두 번째 ASP 학습 모델을 비교한 결과이다. 서버에 연결된 클라이언트의 수가 증가됨에 따라서 전체적으로 응답처리 지연이 증가됨을 알 수 있다. 현실적으로 HubNet모델과 ASP 모델은 사용하는 방법이 다르기 때문에 동일한 조건으로 비교하기는 불가능하다. 다만 결과에서 ASP 모델은 클라이언트 요청에 대한 처리를 하기



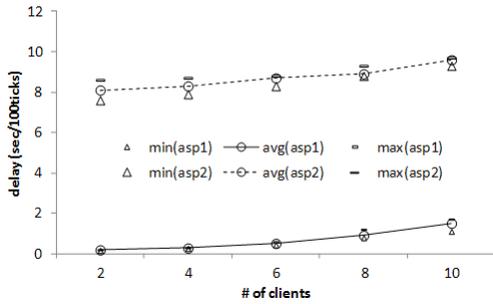
(a) HubNet 서버



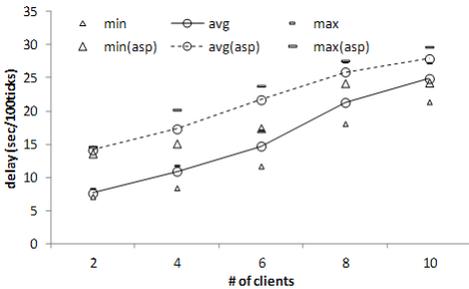
(b) 고고 보드가 연결된 HubNet 클라이언트

그림 9. 게임형 학습 모델

4) NetLogo에서 특정 루틴이 한 번 수행되는 것을 tick 단위로 수행 횟수를 측정할 수 있다.



(b) ASP 기반 모델 1과 모델 2 비교



(a) HubNet 모델과 ASP 모델의 비교

그림 10. HubNet과 ASP 모델에 대한 응답처리 지연 비교

위하여 고고보드로 다시 요청 메시지를 보내고 그 결과를 화면에 출력하기 때문에 각 요청마다 추가적인 round-trip time (RTT) 지연이 발생된다. 결과적으로 100회 수행 후에 누적된 지연이 ASP 방식이 다소 증가됨을 알 수 있다. 그러나 매 회 (tick) 수행되는 시간은 기존 HubNet 모델에 비해서 약 0.08s 이하로 증가하는 것으로 나타난다. 이 실험에서는 ASP 모델을 기존 HubNet 모델의 메시지 처리 방식과 동일하게 모델링하고 비교하였다. 따라서 그림 8과 같은 ASP 모델링 초기화를 위한 메시지 수신 여부를 확인하는 루틴도 같이 동작된다. 따라서 센서 요청 메시지만을 처리하는 경우 (그림 10(b))에는 성능이 향상됨을 알 수 있다.

그림 10(b)는 두 가지 ASP 모델을 비교한 것이다. 먼저 그림 8에 제시된 ASP 모델링 초기화 부분과 실제 모델 적용부분의 코드를 분리하여 모델링하였다. 따라서 클라이언트가 로그인할 때 사용하는 메시지는 검사하지 않고, 센서 값 요구 메시지만을 처리하여 비교한다. 결과에서 알 수 있듯이 첫 번째 모델(asp1)은 서버에서 고고 보드로 요청한 한 개의 센서 값을 매우 빠르게 처리하고 있음을 알 수 있다. 반면에 클라이언트의 요청을 받아서 그 요청에 따른 고고 보드 센서 값을 표시하는 두 번째

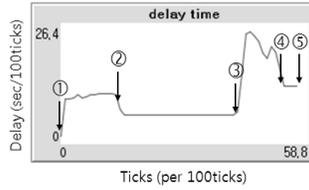


그림 11. ASP 모델링 기법에 따른 응답처리 지연 변화

모델 (asp2)은 상대적으로 지연이 증가됨을 알 수 있다. 하지만 클라이언트 수에 크게 영향을 받지 않는다. 즉, 제안된 ASP 학습 구조를 교육에 적용할 때, 설계하는 모델러가 효과적으로 메시지를 처리할 수 있는 방법을 사용하면 모델의 성능을 개선할 수 있다.

그림 11는 두 번째 ASP 모델을 응답처리 지연을 메시지 처리 방식에 따라서 분석한 결과이다. ①과 ②사이에는 10명의 클라이언트가 연결을 설정하고 약 0.5초 단위로 센서 요청 메시지만을 서버에 전송하는 경우의 응답처리 지연 시간을 나타낸다. 얼마 후 ②부터 ③까지 요청 메시지를 발생하지 않았다. 따라서 센서 값 요청에 대한 고고보드 접근이 발생하지 않기 때문에 시간이 감소되어 일정하게 유지된다. ③에서 ④사이에는 연결 설정, 종료, 기타 메시지 수신, 처리, 그리고 확인하는 루틴을 센서 요청 메시지 처리와 같이 적용한 경우이다. 따라서 성능이 크게 낮아짐을 알 수 있다. 다시 ④부터 ⑤까지는 메시지를 전혀 발생시키지 않는 경우로서 메시지 확인 루틴만 수행되는 경우의 결과이다. 즉, 모델의 종류에 따라서 필요한 메시지를 적합하게 처리할 수 있도록 모델을 구성하면 성능을 향상시킬 수 있음을 알 수 있다.

기술적인 문제로서 NetLogo 모델은 적용되는 시스템과 망 상태에 따라서 다양한 성능을 보인다. 따라서 학습 모델을 개발하는 모델링 방법과 모델러의 설계 방법이 제안된 ASP 학습 모델을 보다 안정적으로 만들 뿐만 아니라 확장성도 증가시킬 수 있음을 알 수 있다.

## V. 결론

본 논문에서는 기존 참여 모의실험의 한계를 극복하는 이중초점 모델링이 가능한 능동적 참여 모의실험 (APS)을 제안하고 이를 위한 고고보드를 NetLogo 확장 모듈을 자바로 개발한다. 기존 HubNet에서는 서버만이 외부장치와 연결되기 때문에 고고보드를 이용한 환경 데이터 및 외부 입력을 다수의

클라이언트를 통하여 수신하는 참여 모의실험 학습을 진행할 수 없었다. 따라서 제시된 APS 학습 구조 I, II, 그리고 III을 바탕으로, 개발된 GoGo2 확장 모듈을 이용하면 이러한 문제를 해결할 수 있다. 제시된 학습 구조를 위한 모델링 방법을 소개하고 적용 예시를 보였으며 간단히 개발된 확장 모듈을 기존 HubNet 모델과 비교평가 하였다. 따라서 APS 영역에 적합한 모델링을 개발하는 어려움이 있지만 제시된 학습구조와 확장 모듈은 초·중등 과학교육을 받는 학생들에게 간단한 실습부터 복잡한 영역까지 활용될 뿐만 아니라 최근 개정된 과학 융합교육에도 이용될 수 있을 것으로 기대된다.

향후 연구로는 제안된 ASP 학습 모델을 적용한 안정성 있는 모델을 개발하여 실제 교육에 적용하여 그 기능과 특성을 분석하는 것이다. 또한 HubNet을 제외한 부분이 공개 SW인 특징을 이용하여 GoGo Monitor의 고고 서버 TCP 소켓 연결을 UDP 방식으로 개선하여 실시간성을 증대할 예정이다.

참 고 문 헌

[1] 교과과학기술부, “7차 개정 교육과정 (초·중·고등학교 과학),” 2007

[2] 광영진 외 7인, “고등학교 과학 Integrated Understanding of Science,” (주)덕텍스트, 2011. 3

[3] Friedler, Y., Nachmias, R., and Linn, M.C., “Learning scientific reasoning skills in microcomputer-based laboratories,” *Journal of Research in Science Teaching*, 27(2), 173-191, 1990.

[4] Wilensky, U., NetLogo, [Computer software]. Center for Connected Learning and Computer-Based Modeling: Evanston, IL. 1999.

[5] Sipitakiat, A., Blikstein, P., and Cavallo, D. P. “GoGo board: Augmenting Programmable Bricks for Economically Challenged Audiences,” *In Proceedings of the 6th international Conference on Learning Sciences*, pp.481-488, June, 2004.

[6] Blikstein, P., Rand, W. & Wilensky, U., “Participatory, embodied, multi-agent simulation,” *the Fifth international Joint Conference on Autonomous Agents and Multiagent Systems*, pp.1457-1458, May, 2006.

[7] Blikstein, P., Rand, W., & Wilensky, U. (2007). Just a Cog in the Machine,“ Participatory Robotics as a Powerful Tool for Understanding Collaborative Learning,” *Computer Supported Collaborative Learning (CSCL)*, Rutgers University, Rutgers, NJ, USA.

[8] Uri Wilensky, Walter M. Stroup, “Participatory Simulations: Envisioning the networked classroom as a way to support systems learning for all,” *Presented at the Annual meeting of the American Educational Research Association*, New Orleans, LA Apr., 2002.

[9] Wilensky, U., Stroup, W. “NetLogo HubNet Gridlock and Disease models,” <http://ccl.northwestern.edu/netlogo/models/HubNetGridlockand/HubNetDisease>, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

[10] Blikstein, P., Rand, W., & Wilensky, U., “Examining group behavior and collaboration using ABM and robots,” *the Agent 2007 Conference on Complex Interaction and Social Emergence*, pp.159-172, Nov., 2007.

[11] Brigitte Denis, Sylviane Hubert, “Collaborative learning in an educational robotics environment,” *Elservier, Computers in Human Behavior* 17 (2001) 465-480.

[12] Wilensky, U., Stroup, W, “NetLogo manual,” <http://ccl.northwestern.edu/netlogo/>

응 홍 우 (Xiong Hongyu) 정회원  
 현재 순천대학교 과학정보융합과 박사과정  
 <관심분야> 무선 센서망 MAC 프로토콜, 임베디드 시스템 기반 교육 콘텐츠 개발

소 원 호 (Wonho So) 종신회원



한국통신학회 논문지 제25권  
 1A호 참조  
 현재 순천대학교 컴퓨터교육과  
 부교수  
 <관심분야> 무선 및 광통신망  
 MAC 프로토콜, 라우팅 프로  
 토콜 설계, QoS 및 트래픽  
 제어, 임베디드시스템 기반 교육