

이종의 소프트웨어 시스템들의 연동을 지원하기 위한 블랙보드 구조의 적용

조 대 호[†] · 고 종 영^{††}

요 약

본 논문에서는, 이형질의 분산된 소프트웨어 시스템들의 연동을 위해 각 시스템을 담당하는 에이전트들을 할당하고 블랙보드 구조를 이용하여 그들간의 협력을 제공하는 시스템을 제안하였다. 이 시스템을 분산 컴퓨팅 환경의 하나인 멀티미디어 원격교육 시스템에서 이미 개발되어 있는, 특성이 서로 다른, 소프트웨어 서버 시스템들의 연동을 지원하는데 적용하였다. 각 에이전트들에 대한 메타지식을 제어기에 분리 시킨과 더불어, 제어 지식과 문제해결 지식을 명확히 분리하여 분산, 협조형 문제해결의 기존 시스템들이 갖는 문제점을 개선하고 기존의 소프트웨어 시스템들의 재사용성을 높였다.

An Application of the Blackboard Architecture for the collaboration among heterogeneous software systems

Tae Ho Cho[†] · Jong Young Koh^{††}

ABSTRACT

This paper presents a multiagent system, based on the blackboard architecture and its mechanism of coordination and communication, developed for the collaboration of distributed heterogeneous software systems. The system provides the environment for the collaboration among pre-existing heterogeneous subsystems within the context of multimedia distance education application. The meta-knowledge for each agent is isolated in the global controller and the control knowledge and the problem-solving knowledge are clearly separated within the system. As a result, the coordination problem that typically found in the conventional distributed problem solving systems are greatly enhanced and the reusability of the pre-developed systems are improved.

1. 서 론

컴퓨터의 성능향상, 개방화 및 사용자 요구의 다양화에 따라 시스템환경이 급격히 변화하고 있다. 사람들 간, 시스템들간, 사람과 시스템사이의 상호작용에 있어서 협동작업(collaboration)이 증가함에 따라 각각의

인터페이스에서 처리할 태스크가 증가하고 있다. 이에 따라, 시스템 개발자나 관리자의 입장에서 보면 시스템 개발, 운용, 유지, 보수 등의 부담이 커지고, 시스템 사용자의 입장에서는 사용자의 자주적 노력으로 시스템 사용 지식을 획득해야 하므로 사용자 부담이 커지게 되었다.

위와 같은 상황에 대하여 인간사회의 조직구조와 유사한 지능형 인터페이스를 구현함으로써 적절히 대처할 수 있는데, 이를 구현함에 있어서 분산 인공지능, 에이

†정 회 원 : 성균관대학교 전기전자 및 컴퓨터공학부 조교수
††준 회 원 : 성균관대학교 전기전자 및 컴퓨터공학부 석사과정
논문접수 : 1997년 12월 4일, 심사완료 : 1998년 3월 10일

전트, 객체지향[1,2] 등의 개념을 바탕으로 블랙보드 구조를 적용하여 다음과 같은 이점을 얻는다. 서로 다른 개발환경에서 개발되어 특성이 다른 소프트웨어 시스템들을 담당하는 분산된 다수개의 에이전트들과 블랙보드를 이용한 그들의 협동(cooperation), 조정(coordination), 통신(communication)으로 전체 시스템을 연동시킨다. 각각의 시스템을 직접 수정하기 보다는 에이전트 그룹의 확장으로 해결함으로써 시스템의 모듈별 독립성을 보장하여 추후의 시스템 확장 및 수정을 용이하게 한다. 분산된 각 시스템이 갖는 다양한 기능을 에이전트 그룹을 통해 효과적으로 통합 서비스함으로써 사용자에게 전체 시스템에 대한 투명성을 제공한다. 이로 인해 시스템과 사용자간의 원활한 연동을 지원하고, 사용자가 갖는 시스템 사용지식의 부담을 덜어준다.

멀티 에이전트 시스템을 이용한 분산 문제해결(Distributed Problem Solving)에서 에이전트간 조정 및 통신 기법에 대한 기존 연구들에서 대표적인 시스템으로, 특정 영역내를 이동하는 이동체(자동차등)의 궤적을 추적하는 시스템인 DVMT(Distributed Vehicle Monitoring Testbed)[3]가 있다. DVMT는 전형적인 블랙보드 기반 구조로 문제해결 측면보다 광범위한 제어에 중점을 두었고, 특정 도메인을 제어하기 위한 조정 방법 연구 기구라는 특징이 있다. 또한, DVMT는 단일의 제어 스레드(thread)를 갖는다.

DVMT 계열의 연구로는 DVMT 발상의 모체인 Hearsay-II [4] 음성 인식 시스템, 최근의 연구로 센서(sensor) 해석의 불확실성을 해결하기 위한 DRESUN [5], 멀티 에이전트 환경을 다단계로 추상화하여 모델링할 수 있고 초점, 스케일링, 매커니즘, 열계를 지원하는 프레임워크인 TMS [6] 등이 있다. 이러한 연구들은 대부분, 데이터 구조적이고, 블랙보드를 이용한 통신기법과 제어를 이용한 조정기법을 추구한다.

위와 같이 다양한 분야에서의 분산, 협조형 문제해결을 위한 기존의 연구들에서 블랙보드 구조를 적용하였는데, 본 논문에서 블랙보드 구조를 적용하고자 하는 대상은 소프트웨어 시스템들의 연동에 있어서 이들의 이형질성을 극복하는 것이다. 즉, 이형질의 소프트웨어들에 각각 담당 에이전트를 할당하고, 이 에이전트들간의 통신 및 조정 매커니즘으로 블랙보드 및 제어를 구현하는데, 각 에이전트들에 대한 메타지식은 제어에

만 분리 표현하고, 문제해결 지식은 각 에이전트에, 제어 지식은 제어기에 명확히 분리하는 것이 본 논문의 목표이다. 아울러, 기존의 연구들이 대체로 복수개의 문제해결 모듈과 그들간의 조정 및 통신을 위한 모듈이 개발 시점에서 통합되었던 것에 반하여 본 논문의 접근 방법은 기존의 이형질적인 문제해결 모듈들의 약간의 변형만으로 성공적인 통합을 이룰 수 있는 조정 및 통신 모듈을 구현하는 것이다.

2. 분산 문제해결

기존의 인공지능분야가 추구한 대상은 하나의 독립된 에이전트로써 주로 인지적인 측면(cognitive aspect)에서 에이전트의 지식 표현과 추론, 자연어 처리 등에 대한 연구개발로 응용 시스템으로는 전문가 시스템이나 지식 베이스 시스템 등이 있다. 그러나, 근래 컴퓨팅 환경의 향상과 인간생활의 문제해결에 대한 관심 증대로 인해 날이 다양해지는 사용자의 요구를 해결하기 위해서는 단독 에이전트의 능력으로는 한계가 있으므로, 여러 에이전트가 분산 협동 처리하는 개념의 분산 인공지능이 대두되었다(7).

분산 인공지능(DAI, Distributed Artificial Intelligence) 연구의 핵심으로 두 가지 관점이 있는데, 그 하나는 분산 문제해결이다. 즉, 특정한 하나의 문제를 해결하는 작업을 여러 개의 모듈들로 나누고, 나눠진 모듈들이 문제에 대한 지식, 해를 발전시키는데 대한 지식을 나눠 갖거나 공유하거나 하여 서로 협동한다. 또 하나의 관점은 멀티에이전트 시스템(Multi-Agent System)(8,9)이다. 문제해결 능력에 있어 이형질적인 다수의 자율적인 지능형 에이전트들이 협력해 문제를 해결하기 위해, 문제와 그의 해에 대한 지식을 공유하고, 단일의 전체적 목표나, 분리된 각각의 목표를 향해 상호작용을 한다.

위와 같은 분산 문제해결을 위한 멀티에이전트 시스템에 관입회 대형 시스템에 비하여 갖는 이점(10)은 우선, 시스템의 모듈화가 용이하다. 그리고, 모든 태스크를 해결하기 대상 시스템의 지식(Domain knowledge) 전체가 필요치 않기 때문에, 대상 시스템의 지식을 모듈화함으로써 문제해결 시스템의 노력을 투입하고자 하는 부분에만 초점을 맞추어 효율적인 문제해결이 가능하다. 또한, 서로 다른 능력과 지식을 가진 에이전트들이 동적으로 팀을 이루어 문제를 해결함

으로써, 다양한 관점에서 유연성과 신뢰성 있는 문제해결 방법을 적용할 수 있다. 그러나, 총괄적인 제어와 일관성 있는 지식체계, 전체적 공유 목표와 문제해결 성공의 판단 기준 등이 결여된 분산환경에서 에이전트들간의 협동 및 통신 행동을 조정하는 것은 매우 어렵다(6.11). 특히, 에이전트간 조정에 있어서, 가장 큰 걸림돌은 에이전트간 이행질성인데, 이행질성에 대하여는 2.2절에서 좀더 자세히 기술한다.

결국, 분산 문제해결 시스템의 기반구조가 되기 위해서는, 문제의 표현, 재구성, 분배 및 할당, 부분해들의 조합, 등을 분산된 방식으로 행하는 문제해결 기법, 에이전트들간의 정보를 주고 받을 수 가능한 통신 구조, 시스템내의 다양한 에이전트들의 행동이 서로 모순이나 충돌없이 효율적으로 조정되어 전체 문제해결 시스템이 목표(goal)를 달성할 것을 보장하는 조정 기법 등을 갖춰야 하는데, 2.1절의 블랙보드 구조가 이에 적합한 구조중의 하나이다.

2.1 블랙보드 구조 (Blackboard Architecture)

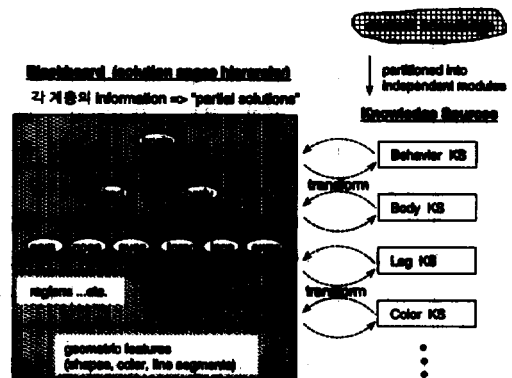
소위 문제해결 객체의 역할을 하는 독립된 에이전트로서 지식원(KS, knowledge source)들과, 이들의 통신 통로인 블랙보드, 지식원들의 활성화 순서를 결정하는 제어 모듈 등의 세가지 요소로 구성되는 시스템으로 [12,13] 지식원들 간에 직접적인 통신을 하지 않고 모든 데이터의 흐름은 블랙보드를 통해서만 이루어지므로, 메시지 교환 시스템에 비해 제어와 데이터의 흐름을 분리한 모듈화 설계가 용이하다.

- 지식원 문제 해결에 필요한 전체 시스템의 도메인 지식(domain-specific knowledge)를 가능한 전문적이고 논리적으로 독립된 모듈로 분할하여 갖고, 지식과 마찬가지로 분할된 하위 문제들에 대한 부분해를 구하는 각자의 목표를 향함으로써, 결국 전체해를 이루는데 기여한다. 이러한 접근방법이 유용한 경우는 조개진 하위 작업들이 해결방법, 해결시간, 필요한 자원 등에 현격한 차이가 있을때이다.
- 블랙보드 - 문제 해결 작업시, 에이전트들 간의 협동을 위한 메시지를 주고 받는 통신로이자, 그 문제에 대한 해결합(solution space) 즉, 문제의 모든 가능한 입력 데이터, 부분해, 또는 전체해를 갖는 계층 구조화된 공유 지식 구조(shared knowledge structure)이다. 블랙보드 계층화란,

에이전트들의 관심영역에 따른 논리적 계층으로, 레벨의 변화는 블랙보드의 상태변화 과정인데, 레벨이 진전됨에 따라 각 레벨을 자기의 관심으로 하는 에이전트들이 활성화되어 그 중에서도 자기의 관심영역만 보고 일을 한다. 일을 마치면 블랙보드를 갱신하여 레벨을 진전시킨다. 즉, 다음 에이전트들이 관심 대상으로 하는 데이터로 변화하는 것이다. 어떤 작업 에이전트는 현재 레벨과 하위 레벨의 데이터도 필요한 경우가 발생하므로 단계적인 블랙보드 변화는 남겨두어야 한다. 각 에이전트들은 제어기로부터 태스크를 할당받을 때 현재 블랙보드의 레벨을 지시받으므로 현재 레벨이하의 레벨들에서 자기의 관심영역이 해당되는 레벨을 탐색한다.

- 제어기 문제의 해를 점진적으로 구하기 위한 지식 적용의 각 단계에서 어느 지식원을 적용할 것인가를 결정하는 제어모듈로, 항상 블랙보드의 내용(problem-solving state)을 주시하여 현재의 블랙보드 상태를 개선할 수 있는 에이전트를 활성화 시킨다

(그림 1) [12]은 블랙보드 구조를 기반으로한 분산 문제해결의 한 예로 숲의 사진안에 숨겨진 코알라의 모습을 찾는 문제이다.



(그림 1) 사진을 통하여 코알라를 찾는 문제에 대한 블랙보드 모델도
(Fig. 1) The blackboard model for the problem of finding koalas in the woods

2.2 소프트웨어 시스템들간 이행질성의 극복
분산 문제해결을 위한 에이전트들로서 소프트웨어

시스템들간에는 이형질성(Heterogeneity)이 존재하게 되는데, 이는 H/W, 운영체제, 프로그래밍 언어 등 뿐만 아니라, 데이터 구조, 문제해결 및 제어 기법, 등 다양한 원인에 기인한다. 이러한 이형질성을 극복하려면, 하나의 표준화된 기법이나 언어를 정의하고 동일한 환경하에서 모든 에이전트를 새로 개발하면 되는데, 이는 소프트웨어 재사용성 문제뿐 아니라 상식적으로 생각해도 비합리적이다. 따라서, 각자의 작업 수행만을 고려하여 개발된 기존의 소프트웨어를 하나의 에이전트 그룹의 구성원으로 통합시킬 때 다른 구성원과의 상호작용이 가능하도록 해 주는 것이 기존 자원의 재활용성 및 시스템의 확장성 부여를 보장하는 합리적인 이형질성 극복 방안이라 할 수 있으며, 이때는 상호작용의 정확성 및 일치성을 보장하기 위해 정보공유(Information Sharing), 조정(Coordination), 모순 해결(Conflict Control) 등의 기능이 요구된다[14].

다시 말해, 분산환경에서 이형질성의 에이전트간 협동 및 조정, 통신을 위해 필요한 것으로는 메타지식, 통신기법, 제어기 등이다.

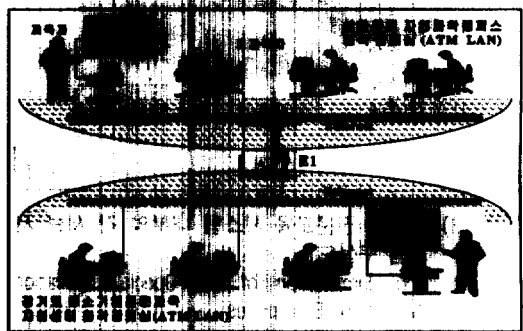
- 메타지식: 여러 에이전트간의 상호작용 제어에 필요한 지식으로서, 각 에이전트가 수행할 수 있는 기능, 에이전트의 목표, 작업 수행 상태, 문제 해결에 필요한 지식, 통신 주소 등을 포함한다.
- 통신기법: 에이전트 시스템간의 지식 공유를 위한 방법론으로, 서로 다른 지적 표현 기법이나 추론 기법을 가진 시스템간의 정보나 능력 공유를 위해 표준화된 메시지 프로토콜을 제시한다.
- 제어기: 여러 에이전트에 대한 메타지식을 갖고 에이전트들간의 상호작용을 제어하여 문제를 분산적으로 해결한다. 즉, 에이전트간 교류를 위한 장소 및 조정 기능(문제를 해결하는 동안 어느 에이전트에게 어떤 작업을 수행시킬 것인가를 판단하고 해당 에이전트를 선택하여 작업을 넘겨주는 기능)을 제공한다.

통신 기법으로 블랙보드 구조를 기반으로 하는 멀티 에이전트 시스템은 제어기에 해당하는 블랙보드와 전문가에 해당하는 여러 응용 에이전트로 구성되는데, 이들 각각은 독립된 프로세스로 존재하고, 개개의 프로세스는 통신을 위한 프로토콜에 따라 메시지를 교환한다. 제어기는 문제를 분산하여 전문가에게 분배하고, 각 에이전트는 자신에게 부여된 문제의 답을 블랙보드에 제

시함으로써 다른 에이전트와 상호 협력한다. 여기서, 응용 에이전트란 특정 도메인의 작업을 전문적으로 처리하는 에이전트로, 자신의 기능을 미리 메타지식으로 표현하여 제어기에게 등록함으로써, 추후의 작업 분배를 가능케 한다. 응용 에이전트는 제어기를 통해서만 다른 에이전트와 정보를 교환할 수 있다. 바꾸어 말하면, 다른 에이전트가 어떤 일을 할 수 있는 지에 대하여 알 필요가 없고, 단지 필요시 통신 메시지를 제어기에게 보내어 맡기면 된다.

3. 지능형 인바텍이스의 구조 및 설계

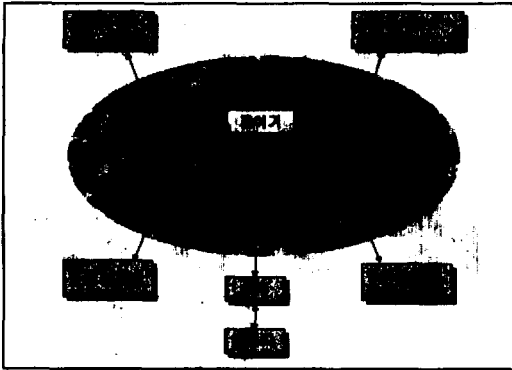
본 논문에서 구현된 시스템은 실제로 정보통신부에서 시행한 96년도 초고속 정보통신 시범사업의 하나인 열린대학 실현을 통한 산업체 전문교육 시스템의 일부로서 '지능형 사용자환경'이라는 명칭으로 적용되었다. 이 사업의 목표는 초고속국가정보통신망, 대학 및 산업체의 인트라넷 및 인터넷으로 구성되는 사이버스페이스에서 CBM(Computer Based Multimedia)기반의 통합 멀티미디어 원격 교육 기술을 이용해서 열린 대학을 실현하고, 열린 대학의 개방된 교육 환경을 이용해서 시간과 공간적인 제약을 받지않고 교육 수요자 위주의 교육을 실현함으로써 열악한 환경에 있는 중소기업의 경쟁력 강화에 기여하기 위한 (그림 2)와 같은 산업체 전문 교육 개발 시스템을 구축하는 것이다.



(그림 2) 열린대학 실현을 통한 산업체 전문교육 시스템 개요도
(Fig. 2) The overview of the Special Education System for Enterprises based on Open University

내부적으로는 기존의 멀티미디어 원격 교육 시스템 S/W, 멀티미디어 교안 S/W, 인증 S/W, 데이터베이

스 등을 각 서브 시스템들로 재사용하여 그들의 다양한 기능을 통합적으로 재구성하고, 이를 바탕으로 다양한 사용자층(학생, 교수, 시스템 관리자 등)의 요구에 대한 적절한 서비스를 제공함을 궁극적인 목표로 하였다. 이를 위한 지능형 인터페이스 구현에 블랙보드 구조가 적용되었는데, 전체 시스템(산업체 전문교육 시스템)을 구성하는 여러 종류의 서브 S/W 시스템들간의 연동을 위한 인터페이스로서의 에이전트 그룹이다. 에이전트 그룹은 사용자를 대신하여 서브 시스템들의 서비스를 검색, 접속하는 과정에서 그들간의 데이터 교환과 실행 순서를 제어하는 기능을 갖고, 여러 에이전트들의 협동 작업에 의해 결과를 처리한다. 이러한 에이전트 그룹은 사용자가 원하는 작업을 자율적으로 대리 수행함으로써 컴퓨터 시스템이나 통신 환경, 멀티미디어 서비스 등에 대해 전문적 지식이 없는 사용자들이 이러한 시스템에 대한 부담을 갖지 않게 해준다. 또한, 추후의 시스템 확장이나 수정시에는 에이전트 그룹의 확장이나 수정을 통하여 전체 시스템의 서비스를 구성함으로써, 각 서브 시스템들의 수정을 최소화할 수 있도록 해준다. 전체 시스템의 내부 구조는 (그림 3)와 같다.



(그림 3) 산업체 전문교육 시스템 내부의 각 서브 시스템 구성도
 (Fig. 3) The composition of the Special Education System for Enterprises

3.1 각 서브 시스템 개요

- 멀티미디어 원격 교육 시스템 : 실시간/비실시간으로 양방향 오디오 및 비디오 정보 전송을 지원하여 컴퓨터 네트워크를 통한 학습 및 질의/응답을 가능케 한다.
- 멀티미디어 교안 시스템 : 자습이 가능하도록 과

목별, 주제별, 장별로 정리된 하이퍼 미디어 형식의 교육 자료 및 실시간 강의시 사용되는 강의용 자료 등이다.

- 인증 시스템 : 상용 서비스 및 불필요한 사용자의 접근 권한을 제한하기 위해 사용자마다 부여한 패스워드의 적절성 검사.
- 사용자 및 개설교육 DB : 교육의 개설 및 참여가 이루어지기 전에 개설되는 모든 교육에 대한 정보로써 개설교육 DB(과목명, 교육모드, 교재, 개설자, 강의계획 등)를 구성하고, 이에따라 수강신청을 받아서 사용자 DB(이름, 나이, 학력, 전공, 경력, 직장, 주소 등의 인적사항 및 수강신청한 과목)를 구성한다. 이러한 내용을 바탕으로 교육의 개설이 이루어지는 시기에 사용자와 산업체 전문교육 시스템을 지원한다.
- GUI : 2/3 차원 그래픽을 이용하여 사용자에게 보다 친숙한 인터페이스를 제공한다. 새로운 사용자입력을 받아 블랙보드에 저장하고, 사용자입력에 대하여 처리된 결과를 사용자에게 제시한다.
- 에이전트 그룹
 - 1) 인증 에이전트 : 인증 시스템을 담당하여, 전체 시스템에 대한 사용자 인증과정을 대리한다.
 - 2) DB관리 에이전트 : 사용자/개설교육 DB를 담당하여, 교육 및 사용자 관리에 필요한 정보들을 저장하거나 검색하는 일을 대리한다.
 - 3) 교재관리 에이전트 : 멀티미디어 교안 시스템을 담당하여, 사용자가 원하는 과목의 교재를 선택할 수 있도록 해준다.
 - 4) 교육연결 에이전트 : 멀티미디어 원격교육 시스템을 담당하여, 실시간 교육 개설이나, 진행중인 교육에 참여 및 출결관리를 지원한다.
 - 5) 블랙보드 및 제어기 : 블랙보드는 에이전트간 간접 통신 수단으로 계층화된 공유 작업 공간(shared working memory)이다. 제어기의 역할은 블랙보드 감시(monitoring), 데스크 할당(allocation), 에이전트 스케줄링이다.

3.2 사용자 서비스

사용자 분류는 교육개설자(교수), 참여자(학생), 시스템 관리자로 분류하고, 각 서비스 항목에 대한 Type-id, 접근권한, 사용자 입력사항 등을 <표 1>과

같이 부여하였다.

<표 1> 에이전트 그룹의 제공 서비스
<Table 1> Services of the agent group

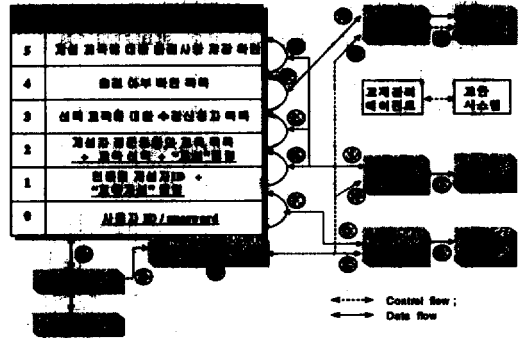
A	사용자 인증	사용자전체	사용자 ID, password
B	교육 개설 정보의 갱신	개설자	과목, 교육모드, 분반, 개설자, 수강신청가능 여부, 강의계획서 등
C	강의 계획 열람	참여자	열람을 원하는 교육 선택
D	수강신청/사용 정보 등록	참여자	수강할 교육 선택, 원격사항 등
E	실시간 교육 개설/출결관리	개설자	개설할 교육 선택
F	지행중인 교육에 참여	참여자	참여할 교육 선택
G	비실시간 교육(자습) 개설	참여자	자습할 교육 선택
H	사용자 등록 및 삭제	관리자	사용자 정보, 삭제대상 사용자
I	시스템 사용상황 감시	관리자	

3.3 시스템 내부의 제어 및 데이터 흐름

<표 1>에 구성된 여러가지 서비스 항목에 따른 서브 시스템, 에이전트 그룹의 제어 및 데이터 흐름으로 에이전트간 협동 및 조정을 나타낼 수 있는데, 다음의 (그림 4)와 같다.

블랙보드를 주시하던 제어가 GUI를 통한 사용자의 새로운 서비스 요청을 감지하면, 이를 해결하기 위한 에이전트 활성화 제어 순서를 결정하고, 이에 따라 적절한 작업 에이전트를 활성화 시키고, 블랙보드의 상태변화 즉, 상태 레벨이 진전되기를 기다린다. 블랙보드의 현재 상태 레벨의 데이터에서 문제해결에 필요한 정보를 얻은 작업 에이전트는 각자의 문제해결 지식과 자기의 연결된 서브 시스템을 이용해서 할당된 작업을 수행한 후, 결과를 블랙보드에 갱신하고, 상태 레벨을 진전시킨다. 이를 기다리던 제어가 다시 제어를 반환받고, 에이전트들에 대한 메타지식 및 제어지식으로 사용자가 요청한 서비스의 최종결과가 나올 때까지 제어를 계속한다. 제어기와 작업 에이전트들이 제어 메시지 및 문제의 중간 처리결과를 간접적으로 통신하

는 수단이 되는 블랙보드의 현재 상태는 사용자가 요구한 서비스의 처리 상태를 나타낸다. 즉, 사용자에 대한 지식(사용자 입력사항들 + 그에 대한 처리결과)들의 집합으로써 최종 처리 결과는 사용자가 원하는 서비스의 제공이다.



(그림 4) 실시간 교육 개설에 대한 제어/데이터 흐름도
(Fig. 4) The control & data flow for the 'on-line education initiation'

- A1: 사용자가 새로운 서비스를 요구할 목적으로 GUI를 통해 입력한 사용자 ID와 패스워드가 블랙보드에 반영된다.

- B: 새로운 서비스 요구를 기다리던 제어가 활성화되어 블랙보드의 현재 상태레벨을 초기화 한다.

- C: 제어기는 블랙보드를 통하여 인증에이전트에게 활성화 메시지를 보낸다.

- D: 제어기의 활성화 메시지를 기다리던 인증에이전트가 활성화되어, 현재 블랙보드 상태레벨의 데이터를 읽는다. 데이터의 종류에 따라 문제해결 행동을 취한다.

- E: 인증에이전트는 사용자 대신에 인증시스템에 연결되어 사용자 인증에 필요한 정보인 사용자 ID 및 패스워드를 가지고 사용자 인증을 대리한다. 인증시스템의 검증과정에서 사용자ID는 적절한 개설자ID임이 입증된다.

- F: 인증에이전트는 사용자 인증의 결과로 보다 개선된 형태로 인증된 개설자ID를 블랙보드에 갱신하고, 블랙보드의 레벨을 한단계 진전시킨다.

- G1-A2: 블랙보드의 레벨이 진전되기를 기다리던 제어가 활성화되어 GUI를 통한 새로운 사용자 입력으로부터 사용자가 요구하는 서비스가 서비스 타입 E

인 실시간 교육 개설임을 판별하고 그에 해당하는 제어 순서에 따라 에이전트들을 활성화 시킨다.

- H1 : 제어기는 블랙보드를 통하여 DB관리 에이전트에게 활성화 메시지를 보낸다.
- I1-J1-K1 : 제어기의 활성화 메시지를 기다리던 DB관리 에이전트가 활성화되어, 현재 블랙보드 레벨의 데이터를 읽는다. 데이터의 종류가 인증된 개설자ID + 교육개설 명령임을 판별하여, 해당하는 문제해결 행동으로서 개설교육 DB로부터 개설자 접근권한의 교육목록을 검색한다. 검색을 위한 질의 정보는 블랙보드의 기록내용에서 얻는다. 개설교육 DB 검색의 결과로 얻어진 목록으로 블랙보드를 갱신하고, 블랙보드의 레벨을 진전시킨다.
- G2-H2 : 블랙보드의 레벨이 진전되기를 기다리던 제어기가 활성화되어 GUI를 통한 새로운 사용자 입력을 받고, 다음의 제어 순서에 따라 다시 DB관리 에이전트를 활성화 시킨다.
- I2-J2-L : DB관리 에이전트가 활성화되어, 현재 블랙보드 레벨의 데이터의 종류가 개설자 접근권한의 교육목록 + 교육선택 + 개설명령임을 판별하여, 해당하는 문제해결 행동으로서 사용자 DB로부터 선택 교육에 대한 수강신청자 목록을 검색한다. 검색을 위한 질의 정보는 블랙보드의 기록내용에서 얻는다. 개설교육 DB 검색의 결과로 얻어진 목록으로 블랙보드를 갱신하고, 블랙보드의 레벨을 진전시킨다.
- G3-M : 제어기가 활성화되어, 다음 제어 순서에 따라 교육연결 에이전트를 활성화 시킨다.
- N-O-P : 교육연결 에이전트가 활성화되어, 블랙보드 내용이 수강신청자 목록임을 판별한다. 이에 따라, 멀티미디어 원격교육 시스템에 수강신청자 목록과 제어권을 전달하면서 사용자와 멀티미디어 원격교육 시스템을 연결하여 교육이 진행될 수 있도록 한다. 교육종료후 원격교육 시스템으로부터 얻어진 출결여부 확인목록으로 블랙보드를 갱신하고, 레벨을 진전시킨다.
- G4-H3 : 제어기가 활성화되어, DB관리 에이전트를 활성화 시킨다.
- I3-J3-Q : DB관리 에이전트가 활성화되어, 블랙보드의 데이터가 출결여부 확인목록임을 판별하여, 개설교육 DB를 갱신하고, 저장확인 정보로 블랙

보드를 갱신하고, 블랙보드의 레벨을 진전시킨다.

- G5 : 제어기가 활성화되어, 사용자 서비스의 최종 처리결과가 얻어졌음을 인식하고, 사용자로부터 새로운 서비스 요청이 있을때까지 기다린다.

3.4 각 에이전트별 기능

여러 서비스 항목들의 처리에서 블랙보드의 변화과정에 따라 관련된 에이전트들이 제어기의 제어순서대로 활성화되는데, 이번에는 초점을 바꾸어, 각 에이전트별로 해결할 수 있는 하위작업(subtask)들의 범위를 명시하고자 한다. 하나의 에이전트가 처리할 수 있는 블랙보드에 기록되는 데이터의 종류와 그 데이터가 얻어질 수 있는 블랙보드 상태 레벨, 데이터 종류에 따른 관련된 에이전트의 문제해결 행동, 처리 결과로서 블랙보드에 갱신되는 데이터의 내용을 정리하면 다음의 <표 2>와 같다. <표 2>에서 블랙보드의 데이터 내용중 밑줄 그은 부분은 GUI를 통한 사용자 입력 사항이다.

3.3절에서 기술했던, 사용자 서비스의 하나인 실시간 교육 개설에 대한 에이전트 그룹의 제어/데이터 흐름을 블랙보드의 상태변화와 이를 야기시키는 에이전트의 행동의 관점에서 나타내기 위해, <표 2>의 내용에서 어둡게 칠해진 부분을 이용하여 아래와 같이 기술하였다. AT, DB, DL 등의 기호는 에이전트 종류를 나타내고, A ~ K, Q 등의 기호는 블랙보드 데이터의 종류를 나타낸다.

제어기가 새로운 사용자 서비스 요청을 받으면 블랙보드의 데이터는 Q이다.

- ① Q → A : AT agent가 활성화되어 인증시스템을 연결한 결과, 블랙보드를 A로 갱신하고, 블랙보드 상태레벨을 1로 증가시킴.
- ② A → A1 : 제어기는 현재 블랙보드 상태레벨에 GUI를 통한 사용자 입력을 추가.
- ③ A1 → B : DB agent가 활성화되어 개설교육 DB를 검색한 결과, 블랙보드를 B로 갱신, 블랙보드 레벨은 2로 증가.
- ④ B → B1 : 제어기는 현재 블랙보드 상태레벨에 GUI를 통한 사용자 입력을 추가.
- ⑤ B1 → F : DB agent가 개설교육DB를 검색하여 블랙보드를 F로 갱신, 레벨은 3으로 증가.
- ⑥ F → K : DL agent가 활성화되어 원격교육 시스템을 연결한 결과로 블랙보드를 K로 갱신하고

〈표 2〉 각 에이전트별 기능 분류표
 (Table 2) The classification of agents' functions

인증 (■)	사용자 ID/passwor	■	0	■	인증된 ID	■	
DB 관리 에이 전트 (■)	인증된 개설자 ID + "교육개설" 명령	■	1	■	개설자 접근권한의 교육목록	■	
	인증된 개설자 ID + "교육개설" 명령	A2		개설교육 DB 검색	개설자 접근권한의 교육목록	B	
	인증된 사용자 ID + "강의계획열람" 명령	A3		개설교육 DB 검색	전체 개설교육 목록	C	
	인증된 사용자 ID + "수강신청" 명령	A4		사용자 DB 검색	사용자 인적사항	D	
	인증된 사용자 ID + "교육개설" 명령	A5		개설교육 DB 검색	자습용 교육 목록	E	
	개설된 접근권한 교육목록 + 교육선택 + "개설"명령	■	2	■	선택교육에 대한 수강신청자 목록	■	
	개설된 접근권한 교육목록 + 교육선택 + "개설"명령	B2		개설교육 DB 검색	개설자 접근권한의 갱신된 교육목록	B	
	개설된 접근권한 교육목록 + 교육선택 + "출결관리"명령	B3		개설교육 DB 검색	선택교육에 대한 수강신청자 목록	F	
	전체 개설교육 목록 + 교육선택 + "강의계획열람"명령	C1		개설교육 DB 검색	선택교육에 대한 강의계획서	G	
	사용자 인적사항 + "확인 및 변경"명령	D1		개설교육 DB 갱신 및 검색 / 개설교육 DB 검색	사용자 접근권한 수강신청 목록 + 전체 교육목록	H	
	전체 개설교육 목록 + 교육선택 + "수강신청"명령	H1		3	사용자 DB 갱신 및 검색	갱신된 수강신청 목록	I
	개설교육에 대한 출결여부 확인 목록	■		4	■	출결사항 저장확인	■
교육 연결 (■)	인증된 사용자 ID + "교육연결" 명령	A6	1	일괄교육 시스템 연결	■		
개설자 선택요청 명령	■	■	■	출결여부 저장확인	■		
교재 관리 (TL)	자습용 교육목록 + 교육선택	E1	2	교안 시스템 연결			

레벨을 4로 증가시킨.

⑦ K → J: DB agent가 활성화되어 개설교육DB를 갱신하고, 결과로 블랙보드를 3로 갱신, 레벨은 5로 증가.

서비스가 종료되고, 제어기는 다음 서비스 요청을 기다린다.

'실시간 교육 개설' 이외의 나머지 서비스들에 대해서도 위와 같은 방법으로 블랙보드의 변화과정을 통해서 에이전트 그룹의 제어 및 문제해결 행동을 기술할 수 있다.

4.1.1. 구현 모듈 명세

이 장에서는 3장에서 기술된 지능형 인터페이스 설계 방법론에 대한 구현 및 실행 내용으로서 의사코드에 의한 주요 모듈 명세, 구현 환경, 유사연구와의 비교, 사용자 인터페이스 화면을 중심으로한 실행예제 등을 기술한다.

4.1 구현 모듈의 명세(specification)

● 제어기 명세

제어기의 구문론적 형태는 (그림 5)와 같은 명세로 나타낼 수 있다. 위의 명세에서 <BODY>부분은 서비

스 타입 E인 실시간 교육 개설에 대한 내용만 명시하고 나머지 부분은 생략하였다. <HEADER> 부분의 서비스 형태 식별자 리스트에서 전체 시스템이 제공할 수 있는 서비스 형태들을 정의하고 <BODY>에서 서비스별 에이전트 활성화(조정) 순서를 명시함으로써 제어기의 제어 지식을 표현한다. 또한, <HEADER> 부분의 에이전트 형태 식별자 리스트에서 에이전트별 기능을 명시하고, <BODY>에서 각 에이전트의 작업 수행 상태를 모니터링함으로써 메타지식을 표현한다. 제어기는 외부 사건(event) 모니터링과 그에 대한 메시지를 에이전트에게 전달하는 기능의 데몬(demon) 역할을 한다.

```

<HEADER>

ID(S) : type identifications of overall systems services
* Type-id list : <표 1> 참조.
Agent(type-id) : a agent of type-id type
* Type-id list : <표 2> 참조.
BCL : the Blackboards Current Level

<BODY>

Loop
Wait until the user requests a new service message.
Delete the service message
Reset BCL
Issue the activation message to agent(AT)
Wait until BCL is increased
Case kind of ID(S) is
A: .....
:
E: Issue the activation message to agent(DB)
Wait until BCL is increased
Issue the activation message to agent(DB)
Wait until BCL is increased
Issue the activation message to agent(DL)
Wait until BCL is increased
Issue the activation message to agent(DB)
Wait until BCL is increased
:
F: .....
End case
End loop
    
```

(그림 5) 제어기 명세
(Fig. 5) The specification of the controller

• DB관리 에이전트 명세

각 에이전트의 구분론적 형태는 일례로 (그림 6)의 DB관리 에이전트와 같은 명세로 나타낼 수 있다. <HEADER> 부분의 블랙보드 내용 형태 식별자 리스

트에서 처리할 수 있는 블랙보드의 내용 형태들을 정의하고 <BODY>에서 그의 처리방법을 명시함으로써 각 에이전트의 문제해결 능력 및 도메인, 문제해결 지식을 표현한다. 또한, <BODY> 부분에서 블랙보드를 통한 에이전트간 통신으로써 통신 프로토콜을 표현한다.

```

<HEADER>

ID(d) : type identifications of Blackboards contents
* Type-id list : <표 2> 참조.
d(type-id) : a document of type-id type
QI(type-id) : Query information extracted from
                d(type-id)
DB : 사용자 및 개선편교육 데이터베이스
BCL : the Blackboards Current Level
Blackboard(BCL) : a partitioned slot of the blackboard

<BODY>

Loop
Wait until the controller issues a new activation message
Delete the activation message
Read (ID(d), d(type-id)) from Blackboard(BCL)
Case kind of ID(d) is
A1: retrieve new d(B) from DB with QI(A1)
A2: retrieve new d(B) from DB with QI(A2)
A3: retrieve new d(C) from DB with QI(A3)
A4: retrieve new d(D) from DB with QI(A4)
A5: retrieve new d(E) from DB with QI(A5)
B1: retrieve new d(F) from DB with QI(B1)
B2: update old d(B) in DB with QI(B2)
    retrieve new d(B) from DB with QI(B2)
B3: retrieve new d(F) from DB with QI(B3)
C1: retrieve new d(G) from DB with QI(C1)
D1: update old d(D) in DB with QI(D1)
    retrieve new d(H) from DB with QI(D1)
H1: update new d(I) in DB with QI(H1)
    retrieve new d(I) from DB QI(H1)
k: update new d(K) from DB QI(K)
End case
Insert (ID(d), d(type-id)) into Blackboard(BCL+1)
Increase BCL
End loop
    
```

4.2 시스템 구현환경 및 유사연구 비교

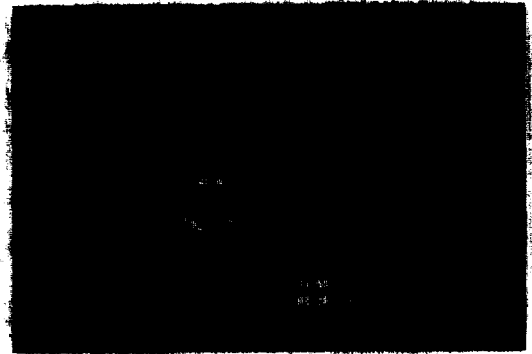
지능형 인터페이스의 구현환경은 TCP/IP 네트워크를 기반으로 MS Windows 플랫폼에서 C++로 프로그래밍 되었다. 연동을 지원하기 위한 기존 소프트웨어 시스템들의 기반구조가 TCP/IP 네트워크 및 Windows 플랫폼이다. 대표적인 APL(Agent Programming Language) [9]로는 이기종 플랫폼간의 이식성이 뛰어나고, 간결성있는 Java, Perl, Tcl, Lisp 계열 언어 등의 interpreted language가 있지만 이들보다 실행 속도 및 특정 플랫폼에서 신뢰성있는

성능보장 면에서 뛰어난 C++을 선택하였다.

특성이 서로 다른 기존 시스템들간의 협동을 지원하기 위한 대표적인 연구로 기존의 여러 전문가 시스템 (Expert system)들을 통합하기 위해 common Lisp 와 C++로 개발된 ARCHON[15] 시스템이 있다. 이는 이형질적이고 다양하게 특화되어 분산된 산업 프로세스(industrial process)를 제어 및 감시하기 위한 시스템으로, 미리 정의된 서비스들과 그 서비스들에 정의된 메시지 타입을 이용한 통신을 한다. 그리고, 문제 영역이 한정된 기존의 시스템위에 에이전트 계층을 두어 메타지식을 표현하고, 개개의 에이전트가 각자의 제어권을 갖고 다른 에이전트와의 상호작용을 전적으로 관리하게 하였다. ARCHON의 문제점은, 블랙보드를 두지 않음에도 불구하고 에이전트 통신을 위한 표준화된 언어의 확실한 정립이 결여되었다는 것과 제어를 두지 않고, 에이전트들이 작업 할당을 위해 서로 협상을 하는데, 이것은 한 에이전트가 다른 에이전트들과 통신하기 위해서 자신에 대한 메타지식과 더불어 상대방 모두에 대한 메타지식을 가져야 한다는 오버헤드가 있다.

4.3 실험 예

3.3, 3.4절에서 기술했던 사용자 서비스 타입 E인 "실시간 교육개설"에 대한 실제 구현된 시스템의 실험 예를 사용자 인터페이스를 중심으로 다음과 같이 나타내었다. (그림 7) [16]의 화면은 현재 시스템의 초기화면으로 제어가 새로운 사용자 요구를 기다리고 있는 상황이다. (그림 7)의 GUI를 통해 개설자가 ID/password를 입력하여 새로운 요청을 하게 되면, 인증에이전트가 활성화되어 인증시스템에서의 인증을 대리해 주고, 인증된 개설자ID를 블랙보드에 갱신한다. 다음으로 제어가 사용자의 입력을 받으려 하는데, 이때 개설자는 (그림 8) [16]의 GUI를 통해 교육개설 명령을 입력한다. 제어의 제어행동으로 DB관리 에이전트가 활성화되어 블랙보드의 개설자ID를 검색정보로 하여 개설자 접근권한의 개설교과 목록을 검색한다. 다시 제어가 사용자 입력을 필요로 하여 사용자는 (그림 9) [16]의 GUI를 통해 개설할 교육을 선택하고, 개설명령을 입력한다. 이로 인해, 제어기는 DB관리 에이전트를 활성화 시키고, 이번에 DB관리 에이전트는 선택교육에 대한 수강신청자 목록을 검색한다. 다음으



(그림 7) 산업체 전문교육 시스템의 초기화면
(Fig. 7) The Initial user-interface of the Special Education System for Enterprises



(그림 8) 인증된 사용자의 서비스 요청 사용자 인터페이스
(Fig. 8) The approved user's interface for the service request



(그림 9) 개설자 접근권한의 교육목록과 교육개설을 위한 사용자 인터페이스
(Fig. 9) The listing of education items and user interface for the lecture initiation

로 교육연결 에이전트가 제어기에 의해 활성화되어 원격교육 시스템을 연결하고 수강신청자 목록을 전달한

다. 교육이 종료되면 교육원형 에이전트는 원격교육 시스템을 모니터링 할 것이며, 학습 목적을 달성하고, 학습 결과에 대해 통신한다. 마지막으로, 원격교육 시스템을 할 수 있도록 운영체제 원격교육을 지원하고, 원격교육을 하고 이를 위한 정보로 원격교육을 전송한다. 제이기를 서비스의 최종결과가 인력들을 인식하고, 다음의 새로운 사용자 서비스 요청을 기다린다.

결론

서로 다른 환경에서 개발되어 각기 다른 기능을 가진 소프트웨어들을 하나로 묶어 사용자에게 복합적인 서비스를 제공하는 새로운 시스템을 구성하기 위해서는, 시스템 내부적으로 이형질적 구성요소들의 원활한 연동을 지원하고, 사용자가 가져야할 시스템 사용지식의 부담을 덜어주고, 아울러 전체 시스템의 확장 및 수정의 용이성을 확보해주는 지능형 인터페이스의 구현이 절실히 필요하다. 이러한 지능형 인터페이스를 구현하기 위해 가장 범용성있고, 적용이 용이한 구조중의 하나가 분산 인공지능, 에이전트, 객체지향등의 개념을 바탕으로한 블랙보드 구조이다.

본 논문에서는, 기존에 개발되어 있는 인공/w, 원격교육/w, 멀티미디어 교안/w, 데이터베이스 등에 담당 에이전트들을 할당하고 그들의 통신수단으로서 블랙보드와 제어수단으로서 제이기를 이용하여, 기존 소프트웨어들의 이형질성을 극복하고 서로간의 원활한 연동을 지원하여 사용자에게 투명성있는 서비스를 제공하였다. 추후의 연구 내용으로는 데이터베이스의 분산화와 같은 에이전트 그룹이 해결해야 할 문제의 복잡도가 더해질 것에 대비하여, 각 에이전트들에게 이동성(mobility)을 부여해야 하고, 아울러 에이전트 각각의 해결가능도 보다 지능형으로 개선해서, 에이전트 특성에 보다 충실한 시스템이 되도록 해야 한다.

참고 문헌

[1] B.P. Zeigler, "Objects and Systems", Springer-Verlag New York, inc., 1997.
 [2] David Brown, 'An Introduction to Object-Oriented Analysis', John Wiley & Sons, Inc., 1997.
 [3] V.R. Lesser, D.D. Corkill, "The Distributed

Vehicle Monitoring testbed: A Tool for Investigating Distirbuted Problem Solving Networks", The AI Magazine, pp.15-33, Fall, 1983.

[4] L.D. Erman, et al., "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty", Computing Surveys, Vol.12, No.2, June 1980.
 [5] N. Carver, et al., "Sophisticated cooperation in FA/C distributed problem solving systems", Proc. Nat. Conf. Artif. Intell., 9th Anaheim, CA, pp.191-198, 1991.
 [6] G. M. P. O'hare, N. R. Jennings, 'Foundations of Distributed Artificial Intelligence', John Wiley & Sons, Inc., 1996.
 [7] 최중민, "에이전트의 개요와 연구방향", 한국정보과학회지, 제15권, 제3호, pp.7-16, 1997.
 [8] M. Wooldridge, N.R. Jennings, "Intelligent Agents: Theory and Practice", Knowledge Engineering Review, October 1994.
 [9] J. Williams, "Bots and other internet beasts", Sams.net Pub., 1996.
 [10] E. Rich, K. Knight, "Artificial Intelligence", pp.429-445, McGraw-Hill, Inc., 1991.
 [11] A. H. Bond, L. Gasser, 'Readings in Distributed Artificial Intelligence', pp.3-35, Morgan Kaufmann Publishers, Inc., 1988.
 [12] H. Penny Nii, "Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures", The AI Magazine, pp.38-53, summer, 1986.
 [13] Grantham K.H. Pang, "A Framework for Intelligent Control", Journal of Intelligent and Robotic Systems 4: pp109-127, 1991.
 [14] 민병희의, "분산 환경 기반 개방형 에이전트 구조", 한국정보과학회지, 제15권, 제3호, pp.39-46, 1997.
 [15] N.R. Jennings, T. Wittig, "ARCHON: Theory and practice", 'Distributed Artificial Intelligence: Theory and Praxis', pp.179-195, Kluwer Academic Press, Boston, 1992.

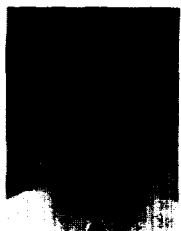
- [16] "초고속 정보통신 시범사업: 열린 대학 실현을 통한 산업체 전문교육 시스템: 사용자 지침서", 기아정보시스템㈜, 1997.
- [17] L. Cavedon, G. Tidhar, "A Logical Framework for Multi-Agent Systems and Joint Attitudes", Lecture Notes in AI, vol.1087, Springer, pp.16-30, 1996.
- [18] Orlando Belo, Jose Neves, "Cooperation Among Opportunistic Agents on a Distributed Environment", Lecture Notes in AI, vol.1087, Springer, pp.204-215, 1996.
- [19] H. Schifdt, "Teach yourself C++", 2nd. Ed., McGraw-Hill, 1994.



고종영

1997년 성균관 대학교 정보공학과 졸업 (학사)
 1997년-현재 성균관 대학교 전기전자 및 컴퓨터공학부 석사과정

관심분야: 컴퓨터시뮬레이션, 지능형시스템



조대호

1983년 성균관대학교 전자공학과 졸업 (공학사)
 1987년 University of Alabama 전자공학과 졸업 (공학석사)
 1993년 University of Arizona 전기 및 컴퓨터공학과 졸업 (공학박사)

1993-1995년 경남대학교 전자계산학과 전임강사

1995년-현재 성균관대학교 전기전자 및 컴퓨터공학부 조교수

관심분야: 컴퓨터시뮬레이션, 공장자동화, 지능형시스템

TOAST