

## 에이전트 기반 교수시스템의 설계

최 영 미

성결대학교 컴퓨터학부

장 영 희

성결대학교 컴퓨터학부

### A Design of Tutoring System Based on Agents

Choi, Young-Mee

Sungkyul Christian University, Division of Computer Engineering

Jang, Young-Hee

Sungkyul Christian University, Division of Computer Engineering

#### Abstract

A design of Advisory Tutoring Agent(ATA) is proposed to help a student on the basis of a student-oriented discovery learning. The agents adaptively build a student model and interact with a student as well as each other to give a piece of advice at a proper time. The appropriateness of advice depends on the agents' reasoning and interaction performed based on the student model, domain knowledge, teaching strategy, and user-friendly interface. The design has been applied to a simple English learning environment.

## I. 서론

일반적으로 코칭시스템은 정규과정에서 뿐만 아니라 비정규과정에서 학생의 발전 학습을 성취하기 위하여 개발되었다. 이러한 시스템들은 주로 학생의 현재 행위에 기초한 학생 학습의 지역적인 결함들 개선하려고 하였다[5]. 그러나 이 방식은 목표를 향한 학생의 전반적인 성향을 간과함으로써 학생 고유의 학습절차를 파악하기 위한 의사결정에 오류를 범하게 된다. 시스템은 학생을 진단하고 교정함에 있어서 학생 고유의 특성이 반영되도록 충분히 융통성이 있어야 한다. 이에 부응하여 적용력 있는 학생 중심 학습을 시도하는 지능형 교육시스템에 관한 연구가 활발해졌다. 그 예로 다계층 진단기법이 각 계층에서 학생의 지식을 처리하여 학생과 상호작용하는 개념을 자원하기 위해서 고안되었다. 다계층 진단기법은 학생의 전반적인 지식을 인식론적으로 획득하여 보다 효과적인 교수환경을 제공한다[2].

최근 에이전트 분야 연구의 활성화로 현장 교사를 시뮬레이트한 교육시스템 개발에 에이전트 개념을 도입하는 연구가 시작되고 있다. 대부분의 시스템들은 학생의 전반적인 수행보다는 학생의 오답에 근거해서 직접적인 해결안을 제시함으로써 학생을 교정하려고 한다. 그러한 보조자(assistant) 유형의 에이전트는 단지 시스템에 의존하는 수동적인 학생을 만들기 쉽기 때문에 실질적인 의미에서의 탐구 학습을 저지하고 있다. Selker의 COACH는 발견학습에 에이전트 개념을 도입한 인식론적인 도움말 시스템이다[9]. CO-

ACH는 학생의 행위를 관찰하고 학생 모델에 기초하여 적절한 충고를 준다. COACH는 명시적인 요청없이 개별적으로 도와주기 위해서 사용자 행위를 예측하는 조언자(advisory) 유형의 에이전트를 사용한다.

여기에서는 학생 중심의 발견학습을 유도하여 학생을 적용력있게 도와주는 조언자 교수에이전트 (ATA : Advisory Teaching Agent)를 제안한다. ATA는 기본적으로 전문가 에이전트, 적용형 에이전트, 교수 에이전트, 인터페이스 에이전트의 네 개의 에이전트로 구성되며 각 에이전트는 영역지식, 추론엔진, 통신기능으로 구성된다. 적용형 에이전트는 개별 학생 모델을 구축하기 위하여 학생의 행위를 기록하면서 적용력있게 상호작용한다. 교수 에이전트는 학생모델을 참조하여 적절한 교수내용을 생성하여 학생에게 제시한다. 전문가 에이전트는 영역지식 상에서 적용형 에이전트가 현상황에 적절한 주제 내용을 교수 에이전트에게 제공하고 최적해를 생성하여 학생의 행위를 추론하기 위한 기초를 마련한다. 인터페이스 에이전트는 학생과 상호작용하는 화면을 학생이 보다 사용하기 쉽도록 해준다. 조언의 적절성은 학생모델, 영역지식, 교수전략, 계획기, 사용자와 친숙한 인터페이스 등에 기초하여 수행되는 상호작용과 에이전트의 추론에 의존한다. 여기서 사용하는 에이전트들 사이의 통신은 시스템 내부에서 일어나므로 본 논문에서는 대화와 상호작용을 동일한 의미로 차용할 것이다.

본 논문은 다음과 같이 구성되었다. 2

장에서는 에이전트 정의, 구조 등 에이전트의 개요에 대하여 설명한다. 3장에서는 ATA설계에 있어서 각 구성 요소의 에이전트의 역할을 제시한다. 4장에서는 적용사례로서 초등학교 영어를 제2외국어로 학습하는 간단한 시뮬레이션된 환경을 제시한다. 5장에서는 앞으로의 연구방향과 결론을 맺는다.

## II. 에이전트 개요

에이전트가 보다 지능적이기 위해서는 에이전트를 사용하는 사용자에게 적용할 수 있는 적용성이 필요하다. 에이전트는 당연히 사용자의 취향을 알아내고 이를 작업 수행에 적용할 수 있는 기능을 필요로 한다. 이러한 적용성을 사용자 수준과 각 사용자 마다의 작업 수준에 따라서 가늠해야 한다. 대부분의 적용형 에이전트는 사용자와 상호작용을 하면서 사용자의 분야별 취향을 학습하고 이를 활용하는 용도로 귀납적 기계학습 방식을 이용하고 있다. 일반적으로 에이전트는 사용자 행위의 관찰을 통한 학습, 사용자의 피드백을 통한 학습, 훈련을 통한 학습, 다른 에이전트의 충고를 통한 학습 등의 방법을 통하여 필요한 지식을 습득해 나간다[1].

### 1. 에이전트 정의

에이전트는 지식과 추론 능력을 가지고 사용자를 대신해서 사용자가 원하는 작업을 자동적으로 해결해 주는 독립적인 컴퓨터 프로그램이라고 할 수 있다. 인공

지능 분야의 거장 마빈 민스키(Marvin Minsky)는 에이전트를 '어떤 문제를 해결하는 블랙박스과 같은 것으로 인간을 대신해 일을 처리해 주는 프로세스'라고 정의하고 있다[8]. 이러한 관점에서 보면 에이전트는 응용 프로그램의 하나로 보다 편리한 컴퓨터 사용환경을 제공한다는 의미가 설득력이 있다.

에이전트는 사용자가 직접 수행하여야 하는 각종 작업을 대신 수행해주는 소프트웨어이다. 에이전트의 중요한 기능은 사용자의 행위를 관측하면서 사용자 작업(task) 단위의 취향을 추출하고 이를 이용하여 사용자에 적용할 수 있는 기능이다. 사용자가 수행하고 싶은 작업은 경우에 따라서 복합적인 과정을 필요로 하기도 하고 단순히 하나의 작업만을 수행하기도 한다. 따라서 에이전트는 사용자가 요구하는 작업을 이해하고 이를 효과적으로 수행하기 위한 계획(planning) 기능을 필요로 하며 복잡한 작업을 효율적으로 수행하기 위해서 여러 개의 에이전트가 협동으로 문제를 해결하기 위한 구조가 요구된다. 에이전트는 다중 에이전트가 서로 협동하여 가장 효과적으로 사용자의 요구를 충족시킬 수 있는 계획을 수립하고 이를 실현하는 소프트웨어이다.

또한 에이전트는 대행성, 지능성, 이동성의 3가지 차원으로 정의된 3차원 개념으로 정의할 수 있다. 물론 이러한 성질들을 모두 갖추어야 에이전트가 된다는 것은 아니며 단지 이러한 성질을 많이 만족할수록 더욱 인간과 유사한 에이전트에 근접할 수 있다고 할 수 있으며 에이전트의 연구와 개발이 이러한 방향으로 진행

되어야 할 것이다[3].

## 2. 에이전트 구조

하나의 에이전트는 기본적으로 에이전트 엔진, 영역지식, 통신모듈 등으로 구성된다. 에이전트 엔진은 에이전트 생성, 작업수행, 에이전트 종료 등의 일련의 작업을 조정하기 위한 제어지식과 추론능력 등을 가진다. 영역지식은 특정 응용 분야의 작업 수행에 필요한 지식으로써 에이전트의 역할을 특정지워주는 부분이다. 에이전트는 생성시에 자신의 영역지식과 관계된 작업능력을 공개함으로써 다른 에이전트와의 작업공유를 시도한다. 에이전트 통신은 다른 에이전트와의 메시지 교환을 담당하는데 대부분 자신이 해결하지 못하는 문제에 대하여 다른 에이전트의 도움을 청하는데 많이 이용된다. 에이전트 통신의 주된 목적은 다른 에이전트와의 협동을 통한 문제 해결이다[4,8]. 에이전트 구조 하에서는 인터페이스 에이전트, 작업 에이전트, 그리고 정보 에이전트의 크게 세 가지 유형의 에이전트가 존재한다.

### (1) 인터페이스 에이전트

인터페이스 에이전트는 사용자와 상호작용을 하며 사용자의 질의를 받아 분석하고 결과를 보여주는 역할을 수행한다. 좀더 지능을 갖춘 인터페이스 에이전트는 사용자의 습성과 기호 등의 데이터를 취득하여 이를 모델링하고 이를 바탕으로 시스템의 조정작업에 활용하는 기법을 가진다.

### (2) 작업 에이전트

작업 에이전트는 주어진 작업에 대한 영역지식과 함께 다른 작업 에이전트나 정보 에이전트의 정보를 가지고 사용자가 요구한 작업을 실제 수행하는 에이전트이다. 이와 더불어 작업 에이전트는 여러 곳에 흩어져 있는 이형질의 정보 소스를 지능적으로 접근할 수 있는 기능을 제공한다.

### (3) 정보 에이전트

정보 에이전트는 원하는 정보를 찾는 역할 뿐 아니라 정보의 변화상황을 감시하는 모니터링도 수행한다.

## III. ATA의 설계

ATA는 독립 환경에서 학습이라는 하나의 목적하에 여러 개의 에이전트들이 상호 협동하는 개념을 도입한다. 즉 에이전트들이 함께 협동작업을 하여 주어진 교수환경에서 학생이 자신의 오류를 스스로 수정하도록 도와준다. 이 개념은 어떤 환경 하에서 상위 에이전트의 목표를 달성하기 위해 하위 에이전트들이 상호 협동한다는 면에서 마빈민스키의 에이전시(agency) 개념을 그대로 따르고 있다[7]. <그림 1>은 ATA의 에이전트들 사이의 상호작용을 보여준다. 인터페이스 에이전트는 학생과 시스템 사이의 입출력 정보변환과 관련한다. 인터페이스 에이전트는 학생이 선호하는 유형을 선택하는 적응형 에이전트의 도움으로 학생이 사용

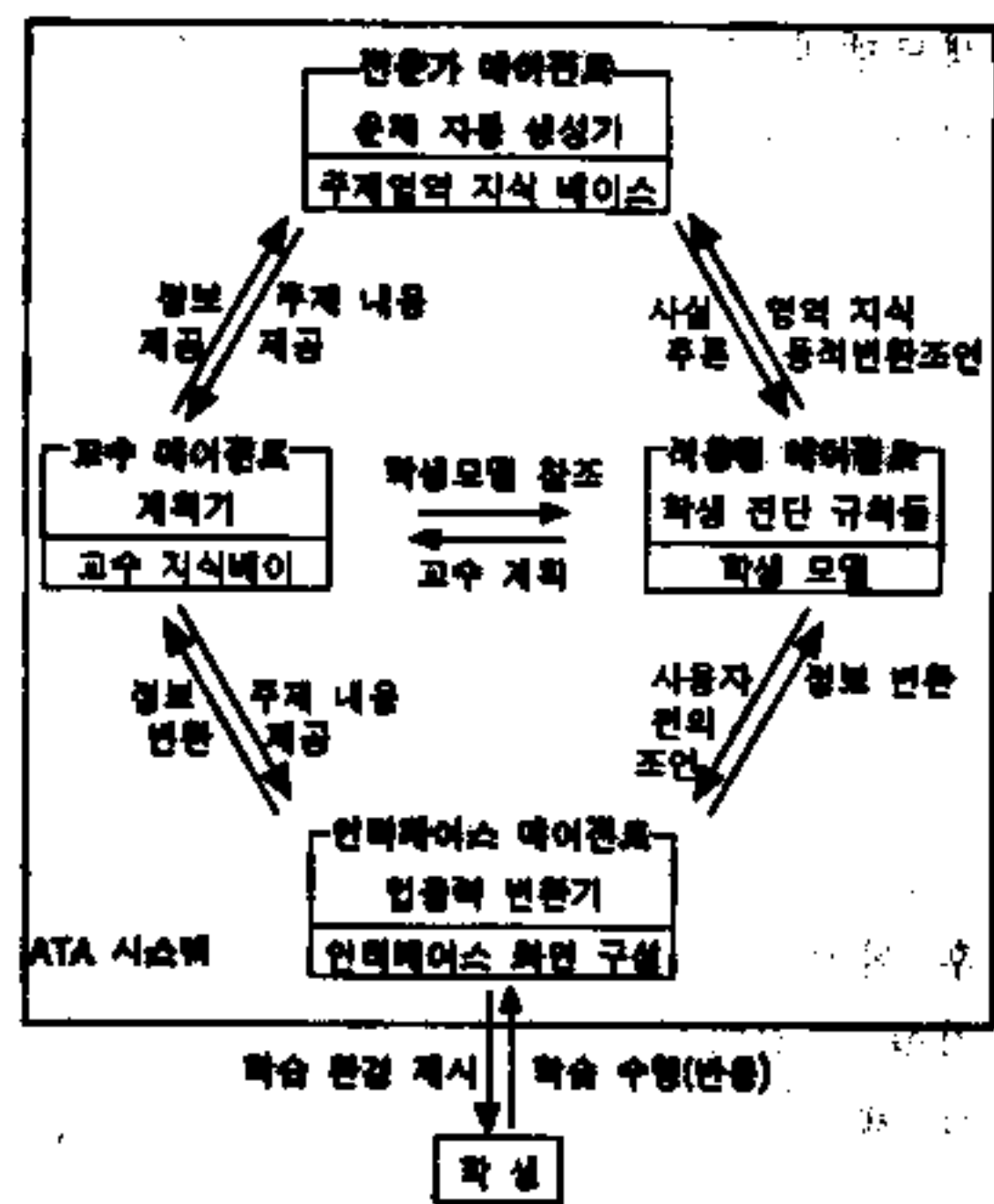
하기 쉬운 인터페이스를 제공한다.

교수 에이전트는 교수내용을 생성하여 학습할 상황을 제공하면서 인터페이스 에이전트와 상호교류한다. 이때 적응형 에이전트에 의해서 구축된 학생 모델을 참조한다. 전문가 에이전트는 가르치기 위한 배경 지식을 제공한다. 적응형 에이전트는 교수 에이전트에게 교수내용 선택 시 조언을 주고 학생지식 표현의 변화를 전문가 에이전트에게 보고한다. 적응형 에이전트는 전문가 학생 행위로 부터 사실을 추론할때 에이전트의 도움도움을 받으면서 학습전반에 걸쳐서 학생모델을 유지보수한다.

다. 적응형 에이전트는 교습과정에서 나타난 학생의 입력자료를 분석하여 학습 진행 과정 및 학생의 오류를 진단하여 ATA가 학생중심 교습이 되도록 지원한다.

적응형 에이전트 영역은 학생모델이다. 적응형 에이전트는 개별 학생모델을 구축한다. 학생모델은 학생과 전문가 에이전트와의 상호작용에서 추론된 사실들의 집합으로 형성된다. 이 학생모델은 학습도중 계속 변경되며 교수에이전트와 적응형 에이전트가 의사결정을 할 때 근거로 사용된다.

적응형 에이전트 엔진은 학생 진단규칙으로 구성되어 학생행위를 분석하여 학생모델을 유지관리한다. 범하기 쉬운 오류에 대한 라이브러리를 작성하고 이에 대한 학생의 수준을 표현하는 방법이다. 즉 오류 라이브러리의 각 항목에 대한 인지정도를 표현하고 히스토리를 기록하는 방법을 사용한다. 시스템이 학생에게 보다 유효성 있게 적용하기 위하여 다음과 같이 시행한다. 학생과의 상호작용에 있어서 적응형 에이전트는 확신이 서지 않은 중간지식을 가지고 학생모델을 만드는 의사결정을 지연한다. 몇가지 제한요소인 소요시간, 상호작용 횟수가 학생의 시간차 오류나 에러를 여과하기 위해서 적용된다. 적응형 에이전트는 학생의 오개념을 확신하였을때 진단하여 학생모델을 수정한다. 적응형 에이전트는 학생모델의 진단규칙 집합을 동적으로 수정함으로써 학생의 학습과정에 반응한다. 예를 들면 만일 학생이 더 이상 오개념을 나타내지 않으면 그것을 나타내는 규칙을 모델에서 삭제함으로써 공간을 절약할 수



<그림 1> ATA 시스템

### 1. 적응형 에이전트

시스템은 각 개인의 특성에 부합하기 위해 충분한 유효성을 가지고 있어야 한

있다.

**적용형 에이전트 통신**은 인터페이스 에이전트로부터 입력되는 학생의 입력으로부터 몇몇 사실을 추론하기 위하여 전문가 에이전트를 참조하고 학생의 학습진전에 따라서 영역을 관리할 수 있는 정보를 제공한다. 이 학생모델을 이용하여 적용형 에이전트는 교수 에이전트가 교수 내용을 결정할 때 의사결정의 근거를 제공한다. 또한 적용형 에이전트는 인터페이스 에이전트에게 학생의 선호하는 유형을 제공하여 학생이 사용하기 쉬운 인터페이스를 제공할 수 있도록 한다.

## 2 전문가 에이전트

전문가 에이전트는 가르치기 위한 주제의 배경지식을 제공한다. 전문가 에이전트는 주제영역 지식베이스를 관리하는 목적지향 문제풀이 에이전트이다.

전문가 에이전트 영역인 주제영역 지식베이스는 학생에게 가르치고자 하는 주제 영역에 대하여 시스템이 가지고 있는 영역 전문기술을 표현한다. 이 영역 전문기술은 가르칠 주제의 자원으로 사용될 수 있고 학습 수행을 평가하기 위한 지침으로써 사용될 수 있다.

전문가 에이전트 엔진은 학습상황과 학생수준에 적합한 문제를 자동 생성한다. 이 자동 생성기가 사용되는 경우는 크게 두 가지 경우이다. 첫째 학생이 공부하고 싶은 내용을 선택했을 경우 과거 학생의 지식수준에 대한 기록을 참조하여 학생의 수준에 적합한 문제를 검색하여 제시하는 경우이며, 둘째는 학습도중 학

생의 대답이 틀려서 이를 수정하기 위한 문제를 제시하여야 할 때, 또는 학습을 자연스럽게 계속하기 위하여 유사한 내용의 문제를 제시하여야 할 경우이다. 이를 원할히 수행하기 위해서 가장 중요한 것은 문제의 특성에 따른 지식베이스 구축 즉 인덱싱문제이다.

전문가 에이전트 통신은 다른 에이전트와 담화하면서 학습상황과 관련한 적절한 문제를 생성하여 적용형 에이전트에게 주고 적용형 에이전트는 영역정보를 관리할 수 있는 학생행위를 제공함으로써 전문가 에이전트는 영역정보를 관리한다. 전문가 에이전트는 적용형 에이전트의 학생모델과 인터페이스 에이전트로부터 현재의 학습 상황을 입력받아 문제를 생성하고 교수 에이전트에게는 학습 내용을 적용형 에이전트에게는 진단의 근거를 제공한다. 문제의 현실성을 위해서 의사결정트리와 깊이를 제한하고 탐색 자원 한계를 제한한다.

## 3. 교수 에이전트

교수 에이전트는 학생의 상태를 감지한 후 어떤 내용을 언제 어떻게 제시할 것인가를 결정하여 학생에게 적절한 교수자료를 제공한다. 즉 가르칠 내용과 순서 그리고 방법을 결정하고 실행한다.

교수 에이전트 영역인 교수전략은 언제 무엇을 어떻게 제시할 것인가에 대한 책임있는 교수 기술이다. 교수전략의 상태전이는 전체 의미네트워크의 링크를 따라서 동적으로 만들어진다. 링크는 그들의 역할에 따라서 두 가지로 나눌 수 있

다. 첫째는 두 노드를 연결하고 다른 하나는 서로 다른 상황에 속하는 노드들을 연결한다. 후자를 '메타링크'라고 한다.

교수 에이전트 엔진인 교수계획기(planner)는 실지로 학생들에게 제시될 교수 내용을 전문가 에이전트의 주제 영역지식으로부터 재구성하여 생성한다. 이러한 학습을 컨트롤하는 교수지식은 크게 둘로 나눌 수 있다. 하나는 가르치는 영역에 무관한 일반적 교수지식(Global tutoring knowledge)으로 각 상황에 따른 교수전략(방법)을 결정하는 지식이고 또 다른 하나는 가르치는 영역과 밀접한 지역적 교수지식(Local tutoring knowledge)으로서 이는 다시 두 가지로 분류된다. 교수도중에 가르칠 내용에 대한 계획수립(Lesson planning)을 위한 지식과 오류 라이브러리의 각 오류에 대하여 교수방법을 결정하는 지식이다. 지역적 교수지식은 학생모델의 정보가 포함되어 있으나 일반적 지식은 학생모델과는 무관하다. 교수 에이전트에 관련된 이러한 모든 지식은 가시적으로 지식베이스에 표현되어 있어야 하고 각 상황에 적합한 규칙이 선택되어 실행된다. 교수에이전트의 궁극적인 목적은 학생수준에 가장 적합한 내용을 가장 적절한 시기에 제시하고 자연스럽게 학습이 진행될 수 있도록 컨트롤하는 것이다.

교수 에이전트 통신은 교수에이전트가 전략을 변경하기 위해서 적응형 에이전트와 상의하고 적응형 에이전트는 교수에이전트에게 조언을 준다. 교수 에이전트는 학생을 위해서 언제 무엇을 제시할 것인가를 결정하면서 전문가 에이전트의 영역

지식으로부터 계획기가 재구성한 학습 가능한 상황을 제공한다. 이러한 상황 변화는 학생의 요구에 의해서 행해지거나 필요에 의해 자동으로 병행될 수 있다.

#### 4. 인터페이스 에이전트

인터페이스 에이전트는 학생과 시스템 사이의 입출력 정보교환을 해준다. 따라서 인터페이스 에이전트는 학생과 ATA 사이에 실질적인 의사교환을 다룬다.

인터페이스 영역은 색, 크기, 객체의 선택, 메뉴바 사용 유무, 입력방식, 각 객체의 위치 등과 같은 사용자에게 편리한 인터페이스와 관련이 있다. 이것은 특히 쉽게 좌절하고 산만해지는 초보학생에게는 중요하다[6,10]. 초기화면에서 메뉴바는 모두 롤다운방식이고 학생들은 자신이 배우고자하는 주제와 난이도를 선택할 수 있다. 조언의 방법도 학생의 선호도를 고려할 수 있도록 학생에게 선택권을 줄 수도 있고, 교수에이전트가 선택할 수도 있다. 조언은 음향효과를 곁들인 2차원 그림이거나 영화일 수도 있고, 또는 단순히 글자로 설명을 해놓은 배열일 수도 있다. 화면의 표현상자(display-box)는 상황을 영화나 그림으로 설명해 주는 부분이고, 실제 구현하는 방법에 따라 학생의 참여를 유인할 수도 있다. 간단한 그래픽 객체로 질문을 할 수도 있다. 인터페이스 에이전트는 지금까지 학습한 자취를 기록상자(history box)에 저장하여 표현상자에 표시할 수도 있다. 팝업윈도우(interaction box)는 학생의 피드백을 위한 부분이다. 학생은 글씨를 쳐넣거나

스크린에 접촉함으로써 입력을 할 수 있다. ATA 에이전트들은 학생의 모든 일거일동을 현재 문안에 저장한다. 활성창(active box)은 자발적으로나, 외부의 요구에 의해 출력을 하게되어 있다. 초기 화면 하단에 나와있는 상태바는 현재 활성화 되어 있는 윈도우를 나타내고 있다.

인터페이스 엔진은 외부의 입력과 시스템의 출력을 처리하고 화면을 관리한다. 인터페이스 에이전트는 인간과 컴퓨터 상호작용의 심리학적인 측면을 고려한다. 학생의 호기심을 보다 잘 유도하기 위해서는 메뉴형식과 그래픽영상을 도입한다. 초기 화면은 학생이 자신의 배경지식을 얻을 수 있도록 고안되었다.

인터페이스 에이전트 통신은 입력정보나 학생 행위를 필요에 따라 각 에이전트에게 전달한다. 학생의 입력을 인식하여 전문가 에이전트와 적용형 에이전트에게 보내고 교수 에이전트로부터 교수내용을 학생에게 전달한다. 이 때 적용형 에이전트의 도움으로 학생이 선호하는 사용하기 쉬운 인터페이스를 제공한다. 인터페이스 에이전트는 또한 전문가 에이전트와 직접적으로 상호작용하면서 영역지식에 영향을 준다.

#### IV. 적용 사례

이 장에서는 제2외국어를 배우는 초등학교 4학년 영어학습을 사례로하여 ATA의 수행과정을 구체적으로 보인다. 영어는 음성언어가 아니기 때문에 아무리 영어를 많이 공부하여도 제 2 외국어로서 그 정

확한 발음을 갖기가 매우 힘들다. 통계에 의하면 학생이 베틀 뜻은 이해할 수 있어도 쓰기에서 철자를 모르는 경우가 90% 이상이 되었다. 영어 발음을 정리해 놓은 사전을 참고한다면 영역지식을 쉽게 얻을 수 있다. 현장 교사로부터 수집 정리한 학생이 볼라기 쉬운 발음이 <표 1>에 요약되어 있다.

주제영역 지식	학생의 오류
nice	nais
cream	krim
sky	ski/skai
pease	pica/pis
mother	mada
doctor	da??
brother	bla
bye	bai
hello	helo

<표 1> 학생 철자표현의 오류 예

이 적용사례에서는 편의상 미국영어의 발음 교정에만 집중을 했고 문법, 억양, 사용법들은 생략하였다. <표 2>도 현장 교사가 직접 경험에 의해 수집한 정보로서 왼쪽 열의 단어를 원했을 때 학생의 응답은 오른쪽 열의 단어들로 나왔음을 보여주고 있다. 'doctor' 나 'brother' 같은 몇몇 단어들은 아예 학생이 포기할 해 버린 경우로 '?'나 빈 칸으로 나타나 있다.

#### 1. 교습 시나리오

이제 음향효과와 더불어 그래픽 객체를



이용하는 ATA가 어떻게 영어 발음 교정/학습에 도움이 되는가 살펴보자.

<그림 2>는 이 글에서 설명할 교수 에이전트의 원하는 대답과 학생의 응답을 표현한 것이다.

학생 반응	교사의도
mada	mother
da??	doctor
dadod	father
	brother

<표 2> 학생과 ATA와의 영어 반응 교정/학습 자료

라는 입력을 하였다. ATA는 다음 6가지 사실을 추론한다. 첫째 학생은 'mother'라는 단어를 모른다. 둘째 영어가 음성 언어의 법칙을 따르지 않는다는 것을 알고 있다. 셋째 모음 'o' 발음을 정확히 모른다. 넷째 'th' 발음을 모른다. 다섯째 'd'와 'th'를 구분하지 못한다. 여섯째 'er' 발음을 모른다. 그러나 ATA는 이 사실들을 임시 지식으로 두고 학생의 잘못된 개념을 확신할 때까지 학생과의 대화를 계속한다. 본 연구에서는 편의상 ATA가 세 번까지 동일한 반응을 하면 추론된 사실들은 학생모델에 삽입한다고 가정한다.

user action 'mada'  
user action 'da?'  
& request.

인터페이스  
에이전트

provide context  
: conversing with  
teaching agent

전문가 에이전트

teach "mother"  
: after conversing with  
expert agent.

planning

- consult intermediate knowledge and/or user model
- consult expert agent
- decide to teach the difference between "th" and "d"

teach "doctor"

교수 에이전트

reasoning on 'mada'

- know not the word 'mother'
- probably knows phonetic rule of the English
- misconception of 'er'
- misconception of 'th' and 'd' possibly knows /d/, that is, confused with 'th'
- misconception of 'o' and 'a'
- possibly knows /ou/ of 'o'

reasoning of 'da?' & on request

- probably is not patient
- knows 'd' correctly
- link with 'mother' given before
- know not the word 'doctor'
- misconception of 'o'
- links with 'mother' given before

추론형 에이전트

<그림 2> ATA의 각 에이전트 대화

대화1 : 어떤 한 시점에서 학생은 'mother'를 입력하도록 하였으나 학생은 'mama'

대화2 : 이제 '지금까지' 모아온 정보를 바탕으로 ATA는 교수 에이전트의 계획에

따라 다음 질문을 택한다. 여기 ATA가 'th'와 'd'의 차이를 가르친다고 결정했다고 가정한다. ATA는 'doctor'를 입력하도록 학생을 유인한다. 학생 입력은 'da?'였다. 여기서 ATA는 또다시 다음과 같은 추론을 한다. 첫째 학생이 'd' 발음을 안다. 둘째 'mother'에서 관찰되었던 'o' 발음의 잘못된 사용을 또다시 관찰할 수 있다. 이때 학생은 전혀 상상을 못했기 때문에 시스템에게 직접 답을 달라고 요청을 할 수도 있다. 교수 에이전트는 자신의 교수전략에 따라 요청을 받아들여 직접적으로 답을 줄 수도 있고 다른 비슷한 경우를 제시함으로써 간접적으로 가르칠 수도 있다. 이 예제에서는 ATA가 편의상 이제 막 시작한 학생에 대해 많은 정보를 가지고 있지 않으므로 교수 에이전트가 요청을 받아들여 직접적으로 가르친다고 가정한다.

대화3: ATA는 'father'를 묻고 'fod-or'를 입력으로 받았다. ATA는 또다시 다음과 같은 추론을 한다. 첫째 학생이 'o'가 /oo/ 외에는 다른 발음이 있다고 배웠다. 둘째 'er'와 비슷한 형태의 'or' 발음을 배웠다. 이것은 'mada'에 비해서 학습이 진보된 상황이다. 셋째 학생이 아직 'th' 발음을 배우지 못했다.

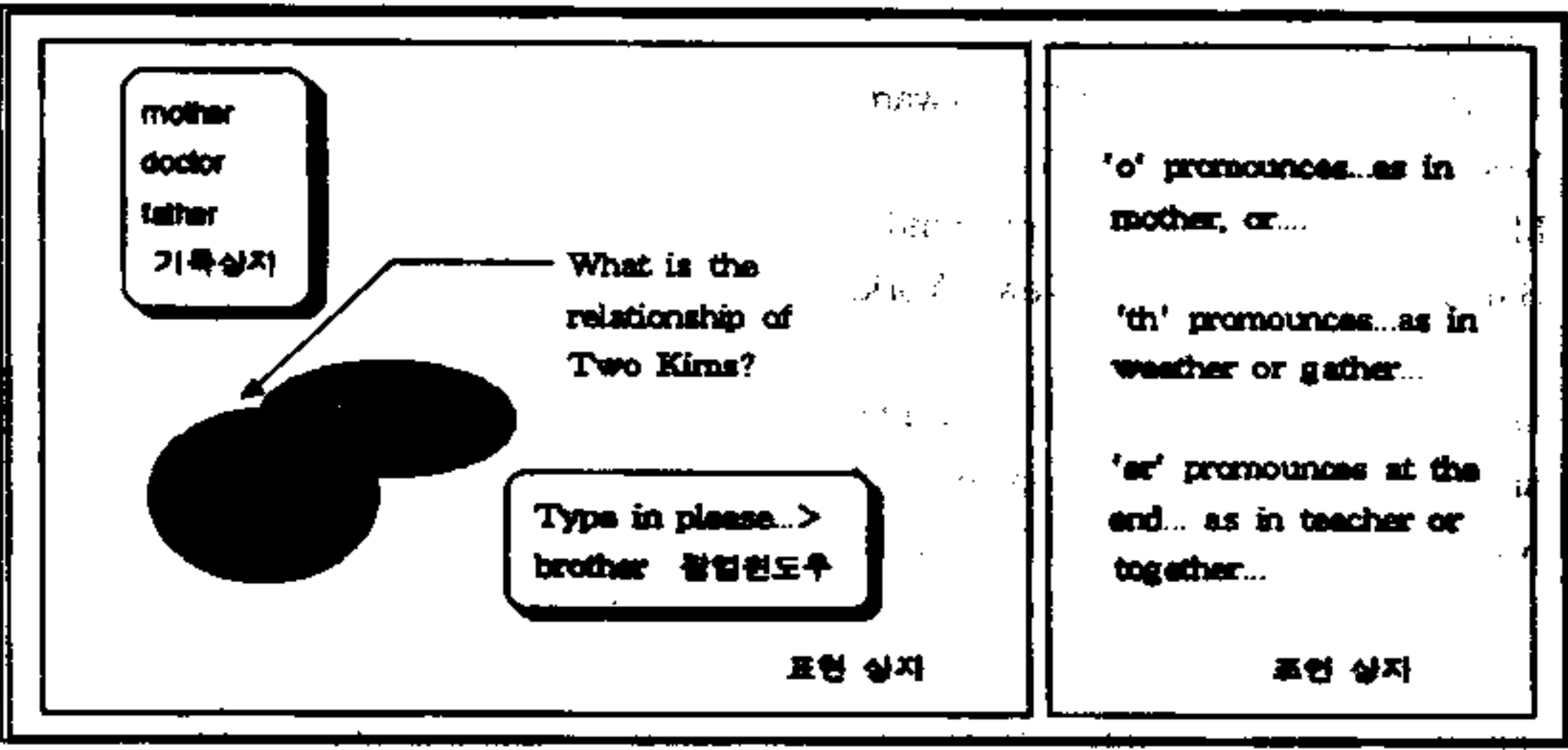
대화4: 세 번의 대화가 있은 후 ATA는 학생이 거듭해서 범한 실수들을 학생의 잘못된 지식이라 확신하고 거기에 맞춰 학생 개인 모델을 만든다. 이 상태에서 ATA가 조언상자(advice box)에 설명을 제시한다. 이후 ATA는 다른 비슷한

상황을 구현함으로써 학생의 진도를 확인한다. 예로써 학생은 'brother'를 입력하도록 유도되고 학생의 답에 따라 학생 모델과 교수전략이 상황을 반영하기 위해 동적으로 변한다. 이런 식으로 거듭되며 시간이 지남에 따라 ATA는 추론을 더욱 정확히 할 수 있는 지식을 축적하게 된다. 위에 묘사한 과정이 부분적으로 <그림 2>에 나타나 있다. ATA내부에서 각 에이전트가 하는 일들이 묘사되었다. 그림에서 점선은 의미망에서의 연결선을 나타낸다. 각 추론된 지식 객체들은 그 신빙성에 따라 가중치를 가지고 있다. <그림 2>에서 probably와 possibly는 이를 나타낸다. 표 1에 나타난 다른 경우들도 비슷한 경위를 거쳐 ATA가 학생에게 가르칠 수 있다. 복잡도(complexity) 문제는 미국영어 발음규칙이 제한된 이상 현실적이다.

<그림 3>은 위의 시나리오에서 마지막으로 학생의 진도를 확인하는 화면이다. 표현상자에 기록상자가 나타나 있다. 조언상자에는 학생이 잘못 알고 있는 발음 부분이 설명되어 있다. 표현상자에는 학생으로하여금 'brother'를 입력하도록 상황이 묘사되어 있다. 학생과 교사와의 대화는 팝업 윈도우를 통해서 이루어짐을 보여주고 있다.

## V. 결론

ATA는 학생과 병행해서 학생에게 조언을 주고, 절대 방해나 제지를 하지않는 교수스타일이다. ATA는 학생모델을 만



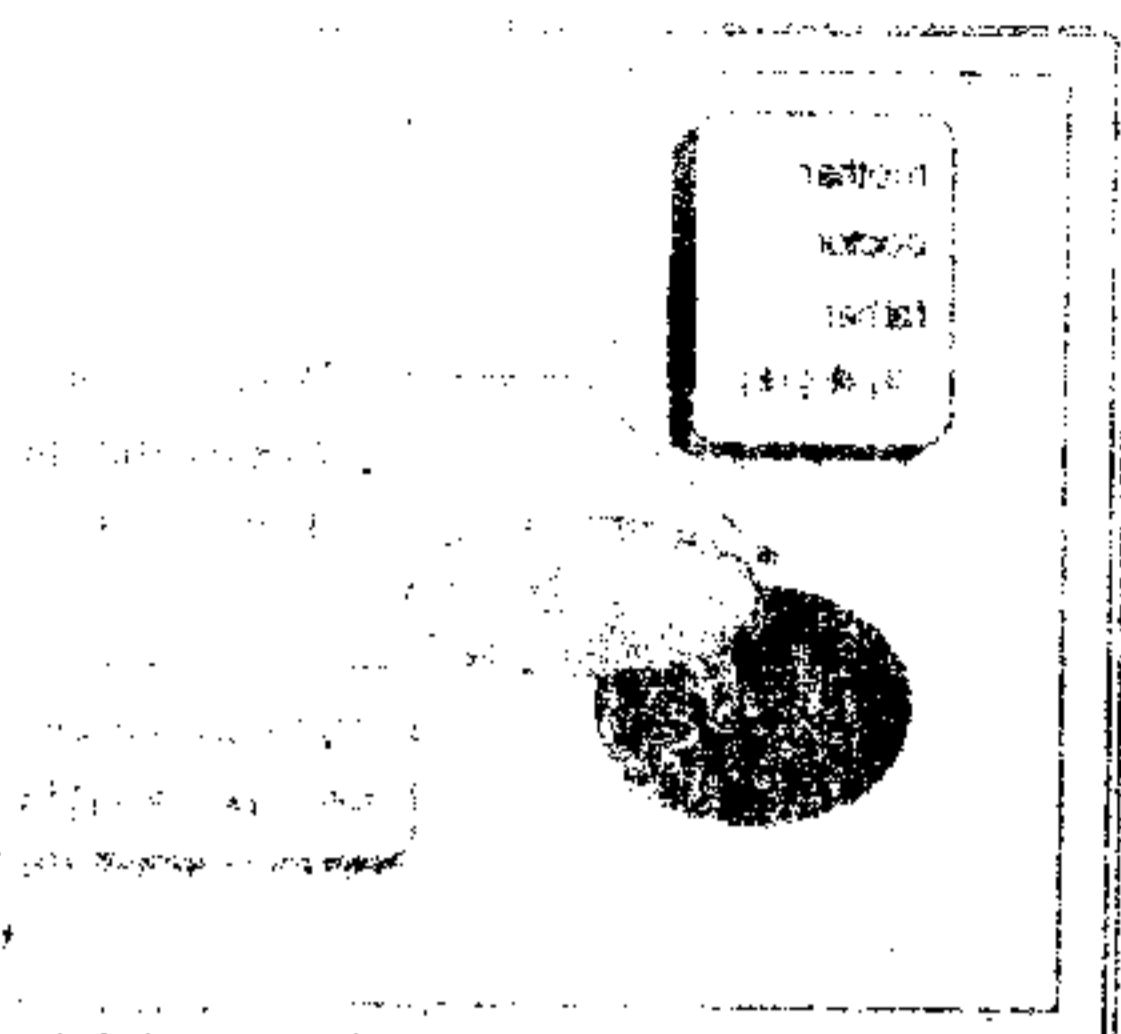
<그림 3> ATA의 학습 환경

드는데 있어 융통성을 가지고 상황을 판단하고 적용해 간다는 면에서 적응력과 인식력이 있다. ATA는 또한 상황 변동에 따라 내부 지식구조를 변경해 가는 동적인 구조를 하고 있다. 단지 영역 지식만을 바꿔줌으로써 ATA는 다른 상황으로의 변환이 용이하다. 여러 에이전트가 협동하여 하나의 목표를 이룬다는 점에서 다중에이전트의 개념을 가지고 있으므로 네트워크의 도입도 가능할 것으로 본다. ATA는 학생의 배우는 과정을 중요시하여 학생의 흥미를 유지하고 학생에게 조언을 강요하지 않고, 학생의 자발적인 학습으로 생산성을 향상함으로써 보다 효과적인 교수스타일 실현에 기여할 것이다. 앞으로 자연어 처리기를 도입함으로써 컴퓨터의 문외한들에게도 접근을 용이하게 할 수 있을 것이다. 멀티미디어 교육과 정보통신 발전에 힘입어 ATA는 앞으로 중요한 역할을 할 수 있을 것이다.

참고 문헌

- [1] 백해정, 박영택, "적용형 에이전트", 정보과학회지, 제15권, 제3호, pp. 29-38, 1997.
- [2] 최영미, 김민구, "게임 환경에서 학습자 모델링을 위한 다계층 진단기법", 정보과학회논문지, 제20권, 제4호, pp. 572-582, 1993.
- [3] 최충민, "에이전트의 개요와 연구방향", 정보과학회지, 제15권, 제3호, pp. 7-16, 1997.
- [4] Barbara H., "An architecture for adaptive intelligent systems", Artificial Intelligence, Vol.72, No.1-2, pp. 329-365, 1995.
- [5] Burton, R. and Brown, J., "An investigation of computer coaching for informallearning acitivities", Int. J. Man-Machine Studies, Vol.11, pp. 5-24, 1979.

- [6] Chang, S., "Visual Languages : A tutorial and survey", IEEE Software, Vol.4, No.1, pp. 29-39 1987.
- [7] Minsky, M., The society of mind , Simon and Schuster, New York, 1986.
- [8] Moises, L. and Thomas, D., "A Framework for the Development of Multiagent Architectures", IEEE Expert, pp. 47-59, December 1996.
- [9] Selker, T., "Coach : A teaching Agent that learns", Comm. ACM Vol. 37, No.7, pp. 92-99, 1995.
- [10] Treu, S., User interface design : A structured approach, Plenum Press, New York, 1994.



이 연구는 인공지능 기반 교육 소프트웨어 개발을 위한 프레임워크를 제시한다. 특히, 다중 에이전트 시스템의 구조를 설명하고, 이를 통해 학습 에이전트가 어떻게 상호작용하며 지식을 습득하는지를 보여준다. 또한, 사용자 인터페이스 설계의 중요성을 강조하고, 이를 지원하는 구조적 접근법을 소개한다. 연구의 주요 결과는 다음과 같다: 첫째, 학습 에이전트의 역할과 기능을 정의하고, 둘째, 에이전트 간의 협력적 학습 과정을 모델링하며, 셋째, 이러한 시스템을 구현하기 위한 소프트웨어 아키텍처를 제안한다. 마지막으로, 실제 교육 환경에서의 적용 가능성과 향후 연구 방향을 논의한다.