

적응형 튜토링 에이전트

주 문 원, 최 영 미
성결대학교 컴퓨터학부

요 약

해결해야 할 문제가 복잡해짐에 따라 소프트웨어의 구조가 대단히 복잡해지고 있다. 이는 데이터와 응용 프로그램 모두 동적 분산적 성격이 강해지고 있기 때문이다. 소프트웨어 시스템은 사용자간의 작업을 조정하는 역할뿐만 아니라 분산된 프로그램간의 협력기능을 관리하는 데에도 일조를 하도록 요구되고 있다. 소프트웨어 에이전트는 이러한 요구를 만족시키는 일반적인 메커니즘을 갖고 있다. 에이전트가 대단히 동적이고 분산된 환경하에서 효과적으로 주어진 작업을 수행하기 위하여 적응성을 가질 필요가 있다. 적응성은 에이전트 행동의 여러 차원에 존재하는 기능으로, 본 논문에서는 이 적응성을 에이전트의 관점에서 내부구조 살펴보고 튜토링 에이전트에 적용할 수 있는 적응형 에이전트 모델을 제시하고자 한다.

Adaptive Tutoring Agents

Monwon Choo, Youngmee Choi

SungKyul Christian University, Division of Computer Science

ABSTRACT

Recently software complexity will continue to increase dramatically in the coming decades because of the dynamic and distributed nature of both data and applications. Not only must software systems assist in coordinating tasks among humans, they must also help manage cooperation among distributed programs. Software agent may properly respond to these requirements. This paper investigates the adaptive characteristics found in adaptive tutoring agents. To function effectively throughout the range of situations presented by such a dynamic and extremely-distributed information entities, an agent must be highly adaptive. Adaptivity exists in several key aspects of agent's behavior to its dynamic situation. My goal is to show the various aspects of internal adaptivity that could be applied to build comprehensive agents that function effectively in tutoring agent system.

1. 서론

튜토리얼은 가장 흔하게 볼 수 있는 교육용 소프트웨어 중의 하나이다. 튜토리얼이라는 말 자체가 암시하고 있듯이 프로그램이 선생님처럼 뭔가를 가르쳐 주는 프로그램을 말한다. 이러한 형태의 프로그램은 새로운 기술, 정보, 지식 등과 같은 것을 선생님이 수업을 하듯이 학습자에게 가르쳐 줄 수 있어야 한다. 따라서 튜토리얼 프로그램은 한 사람의 교사가 수업을 하는 활동과 상당히 흡사한 점이 특징이다 [1].

인터넷 사용이 일반화되면서, 교육 현장에서도 이를 활용한 새로운 교육 환경 조성에 대한 다양한 노력이 진행되고 있다. 분산된 컴퓨팅 자원과 더불어 교육용 데이터와 응용을 비롯한 콘텐츠 자원에 대한 효율적인 사용과 관리가 무엇보다도 관심의 대상이 되고 있다. 인터넷의 확산에 따라 교육정보가 폭발적으로 증가하고, 이에 따라 주문형 교육 정보 검색의 필요성이 대두되고 있으며, 정보 통신 서비스의 다양화는 교육 서비스의 상호 연계 및 교육 콘텐츠의 손쉬운 제작도 요구되고 있다. 이에 따라 교육 환경에 대한 새로운 교육 패러다임이 요구되고 있는 것이다.

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 탐구학습을 저지하고 있다[3]. 학습자가 수행하고 싶은 작업은 복합적인 과정을 필요로 하기도 하고 단순한 하나의 작업만을 수행하기도 한다. 따라서 교육용 프로그램은 학습자가 요구하는 작업을 이해하고, 이를 효과적으로 수행하기 위한 계획 기능을 필요로 하며, 복잡한 작업을 효율적으로 수행하기 위하여 여러 개의 프로그램들이 협동하여 문제를 해결하기 위한 구조와 프로그램간의 조정기능도 요구된다. 지능형 에이전트는 이러한 요구를 대체적으로 만족시킬 수 있는 메카니즘을 포함하고 있으므로 근래에 교육용 소

프트웨어 개발을 위한 새로운 패러다임으로 집중적인 연구대상이 되고 있다.

교육용 응용 프로그램은 학습자에 적용할 수 있는 적응성이 필요하다. 학습자의 지능이나 태도, 취향 그리고 환경 등에 따라 교육 내용을 제시하고, 수정하며, 진행시키는 동적인 기능을 보유해야 하며, 학습자의 교육 진행의 결과를 분석하여 피드백함으로써 진행될 교육내용을 적절히 수정·보완할 수 있어야 한다. 본 연구에서 제시하는 적용형 튜토링 에이전트는 학습자에 부과된 교육적인 목적을 달성하기 위하여 사용자와 관련된 교과과정 수준을 이해하고 교수과정을 적응력있게 자동화함으로써 교육환경을 웹 환경 내에서 최적화시키는 프로그램이라고 할 수 있다.

본 논문은 다음과 같이 구성된다. 제 2 장에서는 웹 기반 튜토링 시스템에 대한 정의와 그 개선방안을 살펴본다. 제 3 장에서는 적용형 에이전트에 대한 정의를 내려보고, 제 4 장에서는 에이전트의 적응성을 5가지 차원으로 분류하여 제안하고, 5장에서는 이를 바탕으로 한 적용형 튜토링 에이전트에 대한 모델을 제시하고 6장에서 결론을 내린다.

2. 웹 기반 튜토링 시스템

오늘날 인터넷에 접속할 수 있는 가장 쉽고, 그리고 가장 인기 있는 방법인 웹의 등장과 함께 인터넷은 가장 중요한 교수도구로서 교사들에게 인식되고 있으며, 웹을 이용한 새로운 교수모형에 대한 시각이 나타나고 있다. 새롭게 출현하고 있는 이 교수모형을 웹 기반 튜토링 시스템이라고 부르고 있는데, 이는 특정한 그리고 미리 계획된 방법으로써 학습자의 지식이나 능력을 육성하기 위한 의도적인 상호작용을 웹을 통해 전달하는 활동이라고 정의 내릴 수 있다 [10].

현재 운영되고 있는 튜토링 시스템들은 대부분이 주어진 과제의 암기가 학습목표가 되고 있다. 암기를 강조하는 소프트웨어들은 많은 경우 학습자의 인지적인 정보처리과정을 고려하지 않고 있다. 따라서 이들은 학습자의 심층적 사고과정에 개입하여 적절한 자극을 줄 수 없으며, 학습자가 이미 알고 있는

내용에 새로이 암기한 정보를 의미있게 연결시키기 어려워 진정한 의미의 학습이나 심도 있는 이해를 가져오기 어렵다. 따라서 본 연구에서는 이러한 문제점의 개선방안으로 어떠한 주제에서건 학습자 주도적(self-directed)이고, 학습자의 수준에 맞는(self-paced) 교수법을 제공하며, 학습자에 부과된 교육적인 목적을 달성하기 위하여 사용자와 교과과정 수준을 이해하고 교수과정을 적응력있게 자동화시킴으로써 교육환경을 웹 환경내에서 최적화시키는 적응형 튜토링 에이전트 모델을 제시하고자 한다.

3. 적응형 에이전트(adaptive agent)

일반적으로 지능형 에이전트는 자신이 속한 환경으로부터의 동적인 자극이나 조건들을 인식(perception)하고, 인식된 데이터들을 해석하여, 문제를 해결하고 최종의 행위를 결정하기 위한 추론(reasoning)을 하며, 이를 바탕으로 부여된 작업을 수행하기 위하여 환경내의 조건들을 변화시키는 행동을 취한다[5]. 이러한 세 가지의 기능들을 지속적으로 수행하는 에이전트가 좀더 효율적으로 인식하고, 추론하며, 행동하기 위하여 적응성이 필요하다. 여기서 적응성이라는 것은 자신이 처한 상황에 따라 대처해야 하는 행동을 적절히 변화시키는 능력을 의미한다. 특히 에이전트가 수행해야 할 작업, 사용가능한 자원, 자신이 처해있는 문맥적인 조건들(contextual conditions), 그리고 성능기준에 있어 동적인 변화가 내재하는 환경 하에서는 더욱이 그러하다. 웹과 같은 분산되고 열린 시스템(open system)에서는 여러 측면에서 동적인 변화들이 예측 불가능한 상태에서 발생한다. 시스템의 환경이 시시각각 변화하고, 다양한 목적을 가진 사람들이 다른 장소와 시간에, 여러 가지 소프트웨어 개발 도구와 기술을 사용하여 만든 이형질의 프로그램들과 데이터들이 네트워크 상에 분산되어 있다[8]. 에이전트가 웹 상에서 요구된 작업을 효율적으로 수행하기 위해 이러한 열린 시스템의 환경과 조건에 반응하는 사용자의 프로파일을 학습하고, 예측하고, 제안하고, 능동적으로 대처하는 능력을 보유하는 것이 필수적이다.

4. 적응성(Adaptivity)

거시적인 측면에서 에이전트가 자신에게 주어진 환경에 적응하기 위하여 자신의 내부적인 환경을 변화시키는 내부적 적응과 에이전트 자신이 처한 외부 환경을 변화시키는 외부적 적응으로 대별할 수 있다. 에이전트의 외부적인 적응은 현재 활발하게 연구되고 있는 주제로서 본고에서는 내부적 적응에 초점을 맞추어 보고자 한다. 그러나 외부적 적응도 에이전트의 구조에서 해결해야 하는 문제이므로 내부적인 환경과 외부 환경과의 조건적 차이점을 정확히 인식하고 있으면 내부적 적응성을 기반으로 하여 외부 적응의 메카니즘을 어느 정도 예측하여 확장시킬 수 있으리라고 본다[3].

적응성은 에이전트가 보유하고 있는 여러 기능적 차원에서 발생할 수 있다(<표 1> 참조).

적용대상의 기능적 차원	적용 내용
인식전략/반응적 전략 (perception/responsive strategy)	외부로부터 입력되는 메시지와 자원 한계에 따르는 조건에 대한 적응
제어/계획 모드 (control/planning mode)	목표기반 한계조건과 환경적 불확실성에 대한 적응
추론/스케줄링 작업 (reasoning/scheduling tasks)	인식되고 추론된 조건/계획들에 대한 적응
추론/실행 방법 (reasoning/execution methods)	실행을 위한 가용 정보와 현재의 성능 기준들에 대한 적응
메타 제어 전략 (meta-control strategy)	요구와 기회의 동적인 구성에 대한 적응

<표 1> 에이전트의 행동적 적응의 차원[3]

첫째, 에이전트는 변화하는 인식적 요구사항 즉, 센서로부터 입력되는 정보 혹은 다른 프로그램부터의 메시지에 대한 요구사항과 제한된 가용 자원을 활용함에 있어 자신의 인식적 전략을 적응시켜할 필요가 있다. 에이전트는 사용자, 다른 프로그램, 다른 에이전트 혹은 환경으로부터의 모든 반응을 인식하고 문제 해결을 위한 최적의 방향으로 적응할 필요가 있다. 그러나 가용 자원이 제한되어 있는 상황하

에서는 사용자의 모든 반응, 환경과 조건 등을 충분히 인식할 수 없다. 에이전트는 반응해야 하는 대상으로서의 환경 인식에 있어 선택적일 수 밖에 없다. 에이전트는 적응적 인식 전략을 사용하여 주어진 환경의 조건과 정보를 가능한 최대한으로 인식하되, 관련성이 적고 무의미한 데이터를 제거하여 현 상태에 가장 적합한 데이터를 취함으로써 목표를 향한 다음 단계인 제어나 계획 과정을 적절히 진행시킬 수 있도록 적용하여 효율성을 극대화하여야 한다. 사용자의 입력 데이터를 추상화하고, 중요도에 따라 우선순위를 매기고, 여과시키는 작업이 인식 단계에서 수행되어야 할 것이다.

둘째, 에이전트는 동적인 목표기반 한계조건에 대한 제어모드를 자신의 행동과 환경으로부터의 불확실성에 적용시켜야 한다. 즉, 외부로부터 입력되는 메시지에 의하여 생성된 목표를 달성하기 위하여, 상위 목표를 해결 가능한 세부/하부 목표들로 분리하여 해결 가능한 절차를 수립하는 계획적 제어가 이 단계에서 이루어져야 한다. 환경의 특징들이 변화함에 따라 에이전트 자신의 제어 모드를 적용시키는 단계라고 할 수 있다. 제어 환경은 작업 환경 내에서의 불확실성, 목표를 달성하는데 효과적인 일련의 행동에 부여되는 제약요건들, 컴퓨팅 자원의 가용성과 비용 등의 여러 차원으로 설정될 수 있다. 간단한 구조를 가진 에이전트는 가장 중요한 제어 조건들이 특징 지워져 있는 것이 보통이다. 이러한 경우 에이전트는 이 제어 조건들을 만족시켜주는 제어 모드를 채택해야 한다. 예로, 어떤 에이전트에서는 가장 효과적인 제어모드가 미리 조정된 일련의 행동을 계획하고 실행하는 것이라면, 다른 에이전트에서의 제어 모드는 실행 시에 발생 가능한 이벤트에 지역적인 처리를 적응적으로 준비하고 실행하는 것일 수 있다.

셋째, 에이전트는 역동적이며 환경적 조건인 지역적 그리고 전역적인 목표가 변화함에 따라 계획된 수행가능한 잠재적인 작업들 중에서 적절한 작업을 선택할 수 있어야 한다. 에이전트는 일반적으로 여러 작업의 일정한 수준의 성능을 요구하고 이러한 작업을 별개로 혹은 일련의 관련된 작업들을 수행할 많은 기회를 갖게 된다. 이러한 기회를 적응적으로 시스템에 제공할 수 있어야 한다.

넷째, 에이전트는 현재 가용한 정보와 성능 기준에 맞추어 스케줄링이 완료된 작업에 적용시킬 최적의 실행 방법을 선택해야 한다. 예로, 특정 작업을 수행하기 위하여 에이전트는 모델 기반 방법, 구조적 선택 방법[9], 혹은 케이스 기반 방법 등과 같은 추론 방법을 활용할 수 있다. 모델 기반 방법은 모니터링된 현상의 구조 혹은 기능 모델내에 세워진 인과 링크(causal link)를 추적하여 가설을 세우고 입증하게 된다. 구조적 선택 방법에서는 관찰된 데이터들을 추상화하여 가설 공간에 경험적 매핑을 수행하여 가설을 세우고, 다시 문제의 문맥으로 되돌아가 세워진 가설을 정제하게 된다. 케이스 기반 모델은 관찰된 문제와 흡사한 케이스를 구별해내어 케이스와 관련된 작업으로 관찰된 문제를 설명하고자 함으로써 가설을 세우게 된다. 이러한 추론 방법들을 선택하기 위하여 자원의 가용성(영역 지식, 환경 변수등 환경 관련 데이터, 실시간, 계산 등), 요구되는 성능관련 속성들(인터럽트 가능성, 잠재적으로 유용한 중간 결과치 등), 요구되는 반응의 특징들(정밀도, 확신도, 질, 정당성 등)에 따라 추론 방법을 선택해야 할 것이다.

마지막으로, 에이전트는 사용자의 요구사항과 기회와 자원 등이 역동적으로 재구성되어야 하는 상황이 발생하면 메타 제어 전략을 적응적으로 적용시켜야 한다. 서로 경쟁적인 행동이 발생하면 에이전트는 메타 전략을 사용하여 계산적 혹은 물리적 자원(센서나 작동기(effectors))들을 적절히 할당하게 된다. 결과로 에이전트가 목표로 삼고 있는 대상의 종류나 그 성취 정도 그리고 부작용 등을 가능할 수 있게 된다.

이러한 다양한 차원에서의 적응력을 제고하기 위하여 각 단계별로 기계학습의 방식을 활용하여 특정 사용자의 주문형 튜토링 에이전트를 구성할 수 있을 것이다. 이러한 적응 능력을 갖고 있는 에이전트를 적응형 에이전트라고 부른다.

5. 적응형 튜토링 에이전트

요약하면, 에이전트는 주변 환경을 이해하고 인지하고 있어야 하고(perception), 그 상황들을 유지할

수 있어야 하며(belief), 현 상태의 요구 상태에 대한 적합성을 판단을 할 수 있어야 하며(desires/wants), 현 상태를 다른 상태로 변화시킬 수 있는 방법을 추론할 수 있어야 하며(planning/reasoning), 다른 상태로 변화시키는 잠재적 실행 행위를 결정할 수 있어야 하고(commitment), 결정한 행위들에 대한 의도적인 순서를 유지할 수 있어야 하고(intentions), 실제 행위를 수행해서 현상태를 원하는 상태로 변화시킬 수 있어야 한다(acting/execution). 이러한 기능을 갖춘 적응형 튜토링 에이전트에 대하여 알아보기 전에 칸[10]이 제시한 플러그인 튜토링 에이전트에 대하여 간략하게 알아보자. 이 에이전트는 그림 1과 같은 간단한 구조를 갖고 있다. 여기에서 도구(tool)는 마이크로소프트 엑셀과 같은 계산용 소프트웨어 패키지 환경을 의미한다. 이와 같은 모델이 엄밀한 의미에서 적응형 에이전트 모델로서 사용하기는 어려운 점이 있으나, 앞으로 설명할 웹을 기반으로 한 적응형 튜토링 에이전트의 모델을 제시하는데 기초적인 개념을 제공한다.

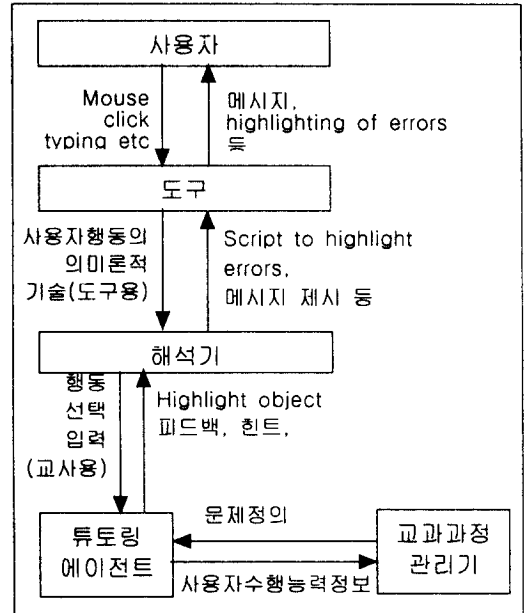
이러한 에이전트의 내부적인 본래적 기능들과 상기한 여러 차원 적응성을 고려하면 적응형 튜토링 에이전트는 다음과 같은 적응적 특성을 보유할 수 있다.

첫째, 튜토링 에이전트는 학습자의 반응을 모니터링할 때, 학습자에 관한 프로필 데이터들 가운데 인식적 활동들을 분할함으로써 학습자의 전반적인 상태를 유지하고, 발생할 수 있는 문제점에 대응할 수 있는 준비를 하게 된다. 예로, 교육 도중 학습자의 문제 해결 능력에 심각한 문제점이 드러나면 학습자의 수준에 맞는 내용에 초점을 변화시킴으로써 다른 시점에서 문제를 해결할 수 있는 처방을 내리게 된다.

둘째, 튜토링 에이전트는 동시 발생적이며 상호 작용하는 다양한 작업의 적절한 성능을 제공해야 한다. 예로, 학습자가 처한 현재의 교육 조건을 모니터링하거나, 문제해결 시 발생하는 오류를 감지하여, 진단하고 해결을 위한 계획을 세우는 것과 같은 모니터링 작업을 수행한다.

셋째, 학습자가 지식을 습득하는 것은 핵심적이고 중요한 사안이지만 지루한 과정이다. 즉, 일정한 시

간을 두고 천천히 진화하는 형태를 취하는 것이 보통이다. 튜토링 에이전트는 학습자를 위한 최적의 학습과정을 적절한 시간을 배분하여 제시해야 하며, 학습자의 반응에 따라 치료적인 기능을 발휘하도록 교육과정에 대한 계획을 세운다.



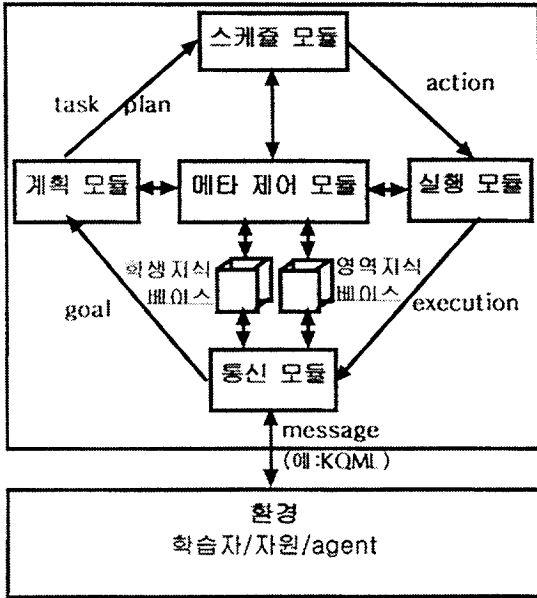
<그림 1> 플러그인 튜토링 에이전트

넷째, 튜토링 에이전트는 일반적으로 발생하는 학습자의 실수나 에러 등을 인식하고, 그에 맞는 해법을 제시하기 위하여 기존의 관련된 데이터나 처리 방법들을 활용한다. 그러나 새로운 문제가 발생하면 조직적인 진단과 적절한 처방을 내리기 위하여 학습되어 저장된 학습자의 모델을 참조하게 된다.

다섯째, 튜토링 에이전트는 상황에 따라 여러 가지 행동을 취할 수 있다. 그 행동들은 서로 관련이 없거나, 혹은 최소한의 연관성을 갖고 있다. 학습자의 문제가 발생하면 여러 가지 행동들이 표출될 수 있으며, 어떤 행동들은 서로 경쟁적인 관계에 있을 수 있다. 이러한 경우 에이전트가 보유하고 있는 메타 제어 전략을 구사하여 동시 발생적인 문제들이 요구하는 자원과 기회 등을 조절하게 된다.

이러한 내용을 기반으로 하여 본고에서는 <그림

2>와 같은 적용형 튜터링 에이전트를 제안하고자 한다. 이에 관한 전체적인 내용을 기술하면 다음과 같다.



<그림 2> 적용형 튜터링 에이전트 모델

학습자의 능력에 적용할 수 있는 레슨을 만들기 위하여 먼저 학습자의 능력을 동적으로 판단할 수 있어야 하며, 동시에 학습된 지식을 통합하여 앞으로 진행될 새로운 학습 내용을 예측할 수 있는 적용 기능을 갖고 있어야 한다. 또한 문제가 제시될 때마다 동일한 유형의 문제에 대하여 이미 저장된 플랜을 적용하여 문제해결을 제시하는 것과 함께, 문제의 특성과 학습자의 레슨 수용 태도에 따라 적용력 있는 플랜을 동적으로 생성하여 교수할 레슨을 수정하고 보완하여 설계할 수 있는 적용력도 있어야 한다.

이러한 내용을 수행하기 위한 역할을 모듈별로 구분해 보면 다음과 같다.

통신 모듈(communication module)은 주변세계에 대한 상황들의 메시지를 받아서 해석한다. 메시지 중 교수서비스의 요청은 이 에이전트가 처리하여야 할 튜터링의 목표(goal)가 된다. 또한 학습자의 학습

과정을 모니터링하여 학습자 모델을 기반으로 한 학습자 프로파일(학습자 지식베이스)을 유지관리한다. 학습자 지식베이스는 학습이 진행되는 과정을 통하여 지속적으로 변경되고 업그레이드되며, 튜터링 에이전트가 교수내용을 결정할 때 판단 근거로 활용된다. 예를 들면 학습자의 레슨 수용 태도에 따라 교육내용을 동적으로 재구성하기 위한 조건으로 사용되거나, 학습자의 학습과정에 인터럽트를 걸어 진행되고 있는 학습 환경을 재고하기 위한 메시지를 출력한다.

계획 모듈(planning module)은 일련의 목적에 관한 입력을 받고, 그 목적을 만족시킬 계획을 생성한다. 에이전트의 계획 프로세스는 계층적인 작업 네트워크(hierarchical task network) 계획 정규화에 근거를 두고 있다. 이 계층 그래프는 교수플랜이 확장되는 것과 세분화되는 것을 제어한다.

스케줄링 모듈(scheduling module)은 각 단계의 교수플랜을 스케줄한다. 일반적으로 에이전트 스케줄링 프로세스는 현재의 계획 사건(plan instance)을 입력으로 받아서 다음에 실행할 실행 가능한 행동과 행위를 계획하며, 결정된 행위들에 대한 의도적 순서를 유지한다.

실행 모듈(execution module)은 에이전트가 다음에 고려하고 있는 행동을 입력으로 받아 실행을 준비하고, 실행 과정을 모니터링하며, 실제 행위를 수행해서 현재의 상태를 좀더 효율적인 목표 지향적인 상태로 변화시키게 된다. 에이전트의 행동 과정은 실행 모듈에 의해서 조절된다. 실행 모니터는 행동을 위한 문맥(context)을 확립함으로써 실행을 위한 행동을 준비한다. 이는 또한 선택적으로 제한된 자원들에 대하여 모니터링 할 수 있다. 예를 들어 일정한 시간이 주어지고, 이 시간 안에 행동이 끝나지 않았을 경우, 계산이 중지되고, 행동이 실패했다고 처리하여 적절한 표시를 하게 된다. 실패된 행동(failed action)은 예외 처리 프로세스로 조절된다.

이와 같은 여러 모듈 외에도 에이전트는 자신이 처한 환경과 적절히 상호작용 할 수 있는 능력이 필요하다. 멀티 에이전트 환경에서의 학습은, 학습자 에이전트와 네트워크 상에서 메시지를 주고 받는 다

른 에이전트의 특성에 의해 학습자의 교육 환경이 동적으로 변해간다는 사실로 인해서 더욱 복잡해진다. 더욱이, 다른 에이전트의 행동은 종종 직접적으로 관찰할 수 있는 것이 아니고, 다른 에이전트가 전달하고자 하는 학습내용이 편향되거나 학습자의 레슨과 상충되는 정보일 수도 있다. 이러한 동적인 환경에서 튜토링 에이전트는 지능적인 행위뿐만 아니라 반응적인 행위도 할 수 있어야 하는데 이를 위해서는 다음과 같은 요소가 필요하다.

첫째, 계획과 실행이 밀접하게 결합된 복합 구조가 있어야 한다. 둘째, EBL과 같이 계획의 성능을 높여 주거나 강화 학습과 같이 더 나은 실행을 하기 위한 학습 방법이 필요하다. 셋째, 외부의 인터럽트에 적절히 대응할 수 있어야 한다. 이와 같이 동적인 환경에 적합한 에이전트를 구현할 수 있는 대표적인 에이전트 구조로는 SEPIA와 Soar를 들 수 있다. 넷째, 튜토링 에이전트는 부여된 작업을 수행하기 위하여 필요한 자원을 제공해야 한다. 예로, 진단하는 작업을 수행하기 위하여 그에 적절한 모델링 방법(예: 연상적 혹은 인과적 모델)을 제공해야 한다. 다섯째, 튜토링 에이전트는 복잡하고, 수시로 변화하는 문맥적인 조건들을 처리해야 한다. 예로, 학습자의 반응은 문제 풀이 과정에서 발생하는 여러 가지 조건들에 대한 복합적이고 문맥에 따라 진단을 달리해야 하는 변수들을 포함하고 있으므로, 에이전트가 제공해야 할 처방도 이러한 조건들을 고려하여 관련된 다른 에이전트들과 상호작용하여 자신의 결점을 보완할 수 있는 기능을 보유해야 한다.

6. 결론

지능형 에이전트 기술은 이제 소프트웨어 개발의 새로운 패러다임으로 정착되고 있다. 소프트웨어가 단순한 문제 해결을 위한 수동적이고 무감각한 프로그램에서 사용자의 요구사항을 이해하고, 자율적으로 해결하여 주기 위한 감성적인 도우미로서의 개념으로 변하고 있다. 이는 해결해야 할 문제의 성격이 복잡해지고, 다양해지고, 어려워지고 있기 때문일 것이다. 또한 다양한 정보가 분산되어 있는 인터넷이 편재해 있으며, 여러 측면에서 활용 잠재력에 대한 기

대가 높아지고 있기 때문이기도 하다. 무엇보다도 사용자의 정보 사용 요구와는 반비례로 정보 사용 능력이 부족한 일반 사용자의 범위가 점점 넓어지고 있기 때문이다. 에이전트는 이러한 배경으로 탄생한 지능적인 소프트웨어이다. 사용자가 요구하는 작업을 소프트웨어 스스로가 이해하고, 이를 효과적으로 수행하기 위한 계획을 세우고, 복잡한 작업을 효율적으로 수행하기 위하여 여러 개의 프로그램들이 협동하며, 경쟁하기도 한다. 이러한 일들을 효율적으로 수행하기 위한 사용자의 프로필을 학습하여, 사용자의 지능이나 행동 패턴 등에 따라 사용자가 원하는 작업을 성공적으로 수행하게 된다.

본고에서는 이러한 배경을 가진 적응성을 보유한 튜토링 에이전트에 대하여 알아보았다. 다분히 개념적인 수준의 내용이지만 적응형 튜토링 에이전트를 이해하는 도움이 되었을 것이다. 그러나 여기에서 제시한 기능적인 단계별 구분은 다분히 인위적인 것으로 적응성이라고 하기에는 무리가 따르는 부분이 존재한다. 그러나 정적인 상태에서 동적인 상태로 변화되는 에이전트의 내부적인 변화를 적응성이라고 정의함에 의한 것으로 좀더 연구를 필요로 하며, 차후에 외부적인 적용에 대한 검토도 할 예정이다.

참고 문헌

- [1] 박성익, 나일주 외, "소프트웨어 설계이론", 한국방송대학교출판부, 1997
- [2] 백해정, 김형훈 외, "적응형 에이전트", 정보과학회지 제15권 제3호, 1997.3.
- [3] 주문원, "지능형 에이전트의 적응성", 정보산업기술 연구소 논문집 제3집(출판예정).
- [4] 최영미, 장영희, "에이전트 기반 교수시스템의 설계", 한국정보교육학회논문지 제2권 1호, pp 45-56, 1998
- [5] Barbara Hayes-Roth, "An Architecture for adaptive intelligent systems", Artificial Intelligence, 72, p.329-365, 1995.
- [6] Charles J. Petrie, "Agent-Based Engineering, the Web, and Intelligence," IEEE Expert, Dec. 1996.
- [7] Hyacinth S. Nwana, "Software Agents: An

overview", <http://www.cs.umbc.edu/agents>

[8] Katia P. Sycara, "Multiagent Systems," AI Magazine, summer, 1998.

[9] W.J. Clancey, "Heuristic Classification," Artificial Intelligence, 27, pp. 289-350, 1985.

[10] <http://www.kmec.net/malsm/wbi/index.html>

[11] http://www.tufts.edu/as/csmt/poster/koedinger_ritter_poster.html