

□ 기술해설 □

지능형 교육시스템

한신대학교 심 임 섭*

● 목	차 ●
1. 서 론	5.3 Bug Library Modeling
1.1 CAI	5.4 진단과 분석
1.2 ICAI와 ITS	5.5 Information about the Student
1.3 교육학과 심리학	5.5.1 Source of Information
2. ITS의 일반적 구조	5.5.2 Noise
3. Domain Expert	6. Situation-Based Instruction과 Multimedia를 이용한 ITS
3.1 Domain Knowledge	6.1 Story-Based Instruction
3.2 Problem Solver	6.2 Cognitive Apprenticeship과 Anchored Instruction
4. 교수 계획자 (Pedagogical Planner)	6.2.1 Cognitive Apprenticeship
4.1 Pedagogical Planner의 구조	6.2.2 Anchored Instruction
4.2 교수 계획시의 고려 사항	7. 일반적인 교육 시스템의 여러 형태
5. 학습자 모델 (Student Model)	8. 결 론
5.1 Student Model과 Student Modeling	
5.2 Overlay Modeling	

1. 서 론

1.1 CAI

CAI는 Computer Aided Instruction 또는 Computer Assisted Instruction의 약자로서 계획했던 소기의 성과는 거두지 못했지만 1960년대 초부터 시작된 미국 일리노이 주립 대학의 PLATO[14] 프로젝트를 CAI의 본격적인 시작으로 볼 수 있다. 그후로 많은 교육학자들과 컴퓨터 과학자들 사이에서 교육용 컴퓨터 소프트웨어의 연구 개발이 본격적으로 시작되었다[3]. 30여년 전의 컴퓨터는 교육 그 자체를 담당하기 보다는 학과목 시간표 작성, 교육보조재 관리, 수험채점 등에 사용되었으며, 교수와 학생들과의

통신 도구로서 쓰였다. 또한 컴퓨터와의 이질감을 줄이기 위한 교육과 Simulation을 보여주기 위한 것들이었으며, 자가 학습은 부차적인 경우가 많았으며, CAI는 Helper, 전자식 Page Turner와 같이 사전에 준비된 다수의 문제를 제시하고 학습자의 응답에 대한 정오답을 비교해 주고 경우에 따라서는 추가적인 설명이 제공되었으며, 학습자의 응답을 저장하여 두기도 하였다. 이것은 차후에 프린트하여 교사가 교수에 참조할 수 있도록 하였다. 학습자의 전체적인 수준에 맞추어 문제의 난이도를 다르게 제시하고 다수의 연습 문제를 제공하는 기능이 전통적인 CAI(Computer Aided Instructions)의 형태이었다. 이는 Courseware로서 현재 국내에서도 상업적으로 많은 제품들이 나와 있다. 이러한 제품들은 시스템 구조상으로 아직 분화가 이루어지지 못하여, Interface의 문제, 문제의 제시 방법, 그리고

*증신회원

오답에 대한 정정 설명 등이 문제별로 고정되어 있어서 개인의 차에 따르는 교수가 될 수 없었다.

1.2 ICAI와 ITS

인공지능 분야의 발전에 따라 인공지능 기술이 CAI에 도입되으니, 그것이 1970년대의 이른바 ICAI(Intelligent CAD)이다[5]. 우선 CAI에서는 시스템에 의해서 정해진 순서에 따라 문제가 주어지고 학습자의 응답에 대하여 고정된 한 가지의 설명이 주어지는 데 비하여 ICAI에서는 학습자가 필요에 따라서는 재학습을 할 수도 있고 뛰어 넘거나 중지할 수도 있는 이른바 Mixed-Initiative 개념이 도입되었다. 또한 ICAI에서는 학습 자료가 교수 부분과 분리되어서 학습자 개인에게 다른 종류의 설명이 제공될 수 있어서 개인의 수준에 따라 다른 학습 교정이 이루어질 수 있으며 대량의 학습 문제를 보유하고 있는 Database로부터 적당한 문제를 추출할 수도 있게 되었다. 이 두가지의 도입으로 학습자에게는 학습상에 상당한 자유가 주어졌으며, Reactive Learning Environment[7]가 조성되었다. ICAI는 과학자들에 따라서는 ITS(Intelligent Tutoring System)[32]와 별다른 구분없이 같은 의미 명칭으로 쓰이기도 한다. 1980년대에 들어서 본격적인 인공지능 기술이 접합되면서 ITS는 학습자 개개인의 특성을 강조하는 개인 교수 시스템이 강조되며 시스템의 구조도 세분화되고 Module화 되고 있다. Adaptive Instruction[35]이라는 새로운 개념이 도입되었는데, 이것은 학습자의 주제에 대한 지식 정도, 학습 진척도, 관련 지식의 정도에 맞추어 다음의 학습 주제와 교수의 방법을 결정하여 각 개인에게 최적합한 교육을 제공할 수 있는 교육 시스템을 추구하고 있다. 따라서 개인 교수 시스템 (Individualized Instruction)은 같은 의미를 갖는다고 할 수 있겠다. 당연히 학습자의 지식 상태와 학습의 효율성을 분석 평가하는 학습자 모델이 중요하게 되었으며, 이에 관련한 여러가지의 연구가 이루어져 왔다. 대표적인 것으로 뒤에 논의 할 Overlay Model, Bug Library 등이 있다. 현재 다양한 인공지능 배경을 가진 사람들의 연구가 진행 중

인 바, de Kleer, B. Kuiper, K. Forbus 같은 Qualitative Reasoning 그룹, B. Woolf같은 Information Science 그룹, W. Clancey 같은 Cognitive Science그룹, R. Shank 같은 Case-Based Reasoning 그룹의 과학자 등 여러 인공지능 분야의 과학자들 대거 참여하면서 ITS가 빠른 발전을 보이고 있다.

1.3 교육학과 심리학

지능형 교수 시스템은 이름에서 보는 바와 같이 전산학을 이용해서 교육을 이행하고자하는 분야이다. 다시 말하자면, 교사나 강사라는 전문가를 컴퓨터 소프트웨어로 실현하고자 하는 것이다. 따라서 인공지능의 전문가 시스템의 구조와 특성을 가진다. 전문가 시스템에서와 같이 Knowledge Acquisition, Knowledge Engineering 등의 과정을 거치며, 결과에 대한 설명과 근거를 제시할 수 있어야 한다. 또한 인간의 지식 상태를 변화시키는 것을 목적으로 하므로, 인간 지능, 인식론(Epistemology), 심리학(Psycho-

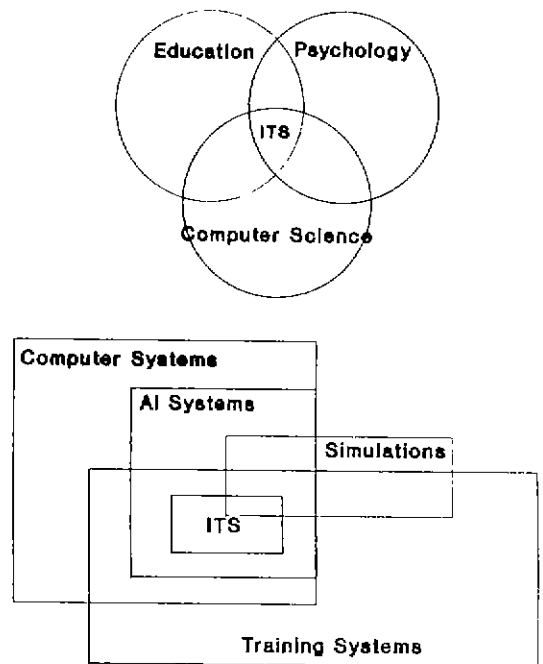


그림 1 ITS와 관련 학문

logy), 인지학(Cognitive Science) 등이 연관되어 있다. 사람들이 어떠한 형태로 지식을 보지하고 있으며, 어떻게 새로운 지식을 습득하며, 전문가와 초보자는 어떻게 지식을 실행에 옮기며 차이점은 무엇인가 등이 연구 과제가 된다. 교육과 교육 심리학은 ITS에 필요한 전문적 교육 이론, 교육 경험자의 실험적 경험과 효과적인 교수 계획의 실체를 제공한다. 교육은 또한 교육자와 피교육자간의 지식 교환을 위한 통신(Knowledge Communication)으로서 언어(Linguistics)라는 매개체를 필요로 한다.

2. ITS의 일반적인 구조

그림 2는 ITS에서 취급해야 할 세 분야의 지식을 보여주고 있다. 즉, 학습 주제에 대한 지식, 교수법에 대한 지식, 학습자에 대한 지식 등이다. ITS는 이러한 분야의 지식이 복합적으로 진행되어서 교육이라는 목적을 달성하게 된다. 이를 ITS의 구성 요소로 바꾸어 망하면 다음과 같다. 첫째는 학습하고자 하는 주제 또는 영역에 대한 지식(Domain Knowledge Base)이다. 무엇을 가르칠 것인가 하는 것이다. 이 지식은 문제 해결의 추론과정을 거쳐 결론을 도출하는 기능(Problem-solver)을 포함하며, 영역 전문가 (Domain Expert)의 기능이다. 둘째는 교수 방법에 대한 것이 된다. 어떻게 가르칠 것인가 하는 것이다.

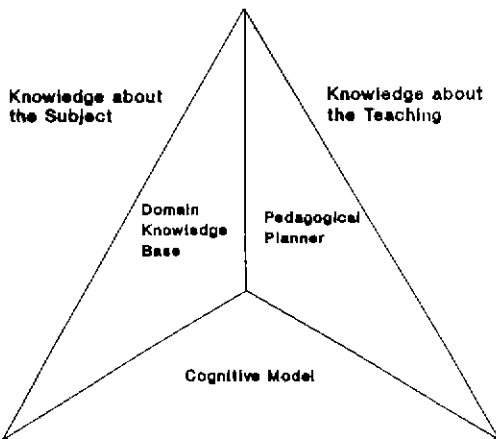


그림 2 ITS관련 지식 분야

이것은 교수 계획자(Pedagogical Planner)의 기능이다. 여기에는 교수 전 과정 동안의 내용적인 일관성 유지(Discourse Management)를 포함한다. 셋째는 학습자에 대한 지식(Student Model)이다. 피교육자가 무엇을 알고 있으며, 무엇을 장차 알아야 하는가 하는 것이다. 피교육자의 학습 영역에 대한 과거의 지식 상태 및 현재의 상태, 그리고 그 변환 과정으로서 학습자 모델링(Student Modeling)과정을 거쳐 학습자 모델에 보관된다. 여기에 부수적으로 학습자와의 정보 교환 방법(Interface)이 포함된다. 어떠한 정보 교환 방법이 가장 효율적인가 하는 것이다. 이 네 가지를 정보를 교환하는 통신(communication)의 관점에서 본다면 학습 영역은 통신의 목적물이고, 학습자 모델은 통신의 수혜자이며, 교수 방법은 통신의 숙련도를 의미하며, Interface는 통신의 형태를 의미하는 것이다. ITS의 일반적인 구성 요소는 그림 3과 같다.

3. Domain Expert

Expert module은 교육 전문가인 교사가 학습 주제에 대해서 가지고 있어야 할 모든 학습 정

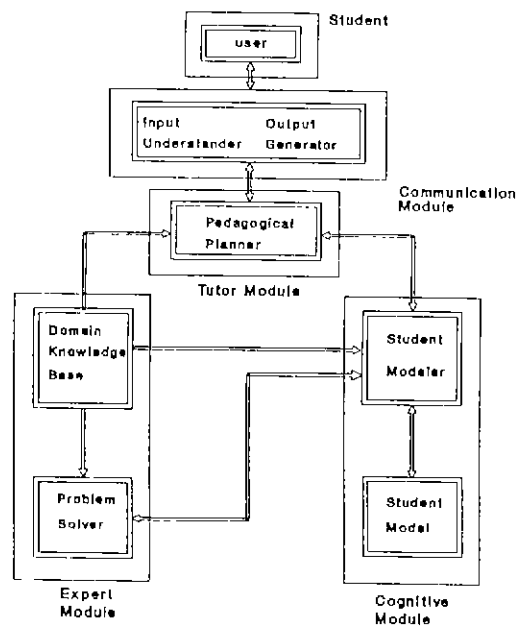


그림 3 ITS의 구성도

보를 가지고 있어야 하며, 학습자 모델이 학습자의 행동을 분석 평가하는 데 필요한 학습 주제에 대한 모든 자료를 제공할 수 있는 능력이 요구된다. 강력한 Domain Expert는 다량의 충분한 지식을 보유하고 있어야 하며(Knowledge Base Module의 기능), 그 지식을 이용하여 각종의 문제에 대하여 최적의(Optimal) 결론(Solution)을 추출할 능력을 갖추어야 한다(Problem Solver의 기능).

3.1 Domain Knowledge

지식은 여러가지 방식으로 분류할 수 있다. W. Clancey[16]는 결론을 도출하기 위한 추론 과정(Reasoning Process)의 특성에 따라 지식 모델을 분류하였다. Classification 모델은 질병에 대한 것과 같이 Object, Event와 Process를 규정하는 형식을 말한다. MENO[37]의 경우를 보면 최종 코딩이 패턴에 합치되는가를 평가할 뿐 Programming 과정에 대해서는 모델링을 할 수 없다. Simulation은 인과관계(Causal-Effect)를 표현하는 것으로 특정한 상황에서의 시스템을 설명하는 Behavioral 모델과 일반적이고 개념적인 기능을 의미하는 Functional 모델로 나뉘었다. MYCIN과 GUIDON[17]은 전자의 범주에 속하며, LISP TUTOR[30], SOPHIE 등은 후자에 속한다. P. Fink[20]은 인공지능에서 통상 분류하는 Declarative Knowledge와 Procedural Knowledge로 구분하였다. Declarative 지식은 사실이나 그림, 물건(Objects), 또는 Object 사이의 관계를 표현하는 것이며, Procedural 지식은 특정한 문제를 해결하거나 추론할 때 쓰이며, 적절한 규칙의 선택이나 우선 순위를 지정한다. Procedural 지식은 대개 Rule-Based 방식을 이용하며 LISP TUTOR, BUGGY, LEEDS 등이 좋은 예이다. Rule-Base 방식은 학습자가 문제 해결에 사용한 Rule을 추적하므로써 학습자의 학습 양태(Student Behavior)를 분석하여 학습자 모델의 구축에 좋은 자료를 얻을 수 있다. WUSOR와 SPADE 등이 좋은 예이다. Declarative 지식은 원칙이나 사실의 나열 등의 특성을 가지므로 Frame 방식이나 Semantic Nets, Scripts 등을 사용한다. 남

아메리카의 지리학에 관한 SCHOLAR는 Semantic Net의 예이며, 지리학적 사실들간의 관계를 정의하고 정보를 추출하는 데 유연한 구조를 갖고 있다. 그러나, Declarative 지식은 사실들과 추론 과정이 분리되어 있어서 학습자가 사용한 지식이 바로 학습자가 그것을 알고 있다는 증거는 아니라는 사실이다. 지식의 표현방식에 있어서는 1960년대에는 일반적이고 광범위한 문제를 풀기위한 다양한 지식의 표현이 주관심사였으나, 1970년대와 1980년대를 거치면서 특정한 문제에 대하여 집중적이고 상세한 지식의 표현을 주목표로 하고 있다. 이러한 추세에 따라, ITS도 좁은 범위의 주제를 가장 집중적이고 효과적으로 교수할 수 있는 방향으로 가고 있다.

3.2 Problem Solver

Domain Expert의 두 번째 기능은 문제 해결 능력이다. 문제가 주어졌을 때 Knowledge Base의 지식을 사용하여 최적의 해답을 도출하는 기능이다. 이것은 학습자의 해답과 비교 분석되어 차후의 교수 계획에 반영된다. J. Anderson[4]은 Blackbox Model, Glassbox Model, Cognitive Model의 세 가지 방법론을 제시하고 있다. 전자회로의 고장수리를 교육하는 SOPHIE에는 수학 공식을 이용하는 Simulator가 내장되어 있으며, 계산 과정은 학습자에게 설명되지 않고 다만 계산의 결과만이 제시되어 학습자의 Solution과 비교 평가된다. WEST에서도 같은 방법이 사용된다. 즉, 학습자의 이동에 대한 옳고 그름과 최적의 계산자 이용에 대한 조언이 학습자에게 주어지지만, 어떠한 과정을 거쳐서 그 조언이 만들어졌는지에 대한 이유는 제시되지 않는다. 따라서 Blackbox Model의 이용은 이른바 Surface-Level 교습으로 Deep-Level 교습보다는 학습면에서 효과적이지 못할 수 있다. Glassbox Model은 전통적인 Expert System과 같은 투명한 구조로 추론 과정의 설명과 정당성(Justification)이 그 결론(Hypothesis)과 같이 동시에 제공된다. 추론의 과정은 인간 전문가들의 그것과 동일한 것이 되며, 따라서 보다 효과적인 교육의 자료로 사용될 수 있다.

4. 교수 계획자(Pedagogical Planner)

4.1 Pedagogical Planner의 구조

교수 계획자(Pedagogical Planner)는 학습자가 정해진 학습 목표를 완성할 수 있도록 교수 계획을 수정 발전시켜야 한다. 초기의 교육 시스템들은 사전에 지정된 계획만 실행되는 정적인 Pre-planned Planning으로부터 동적(Dynamic)인 Opportunistic Planning으로 발전해 왔다. 1960년대의 Courseware에서는 특정한 조건에 따라 가능한 경로가 정해진 시나리오를 따르는 Conditional Planning이 이용되었으며, BIP에서는 인간 교사의 교육에 주요한 역할을 하는 동기부여나, 조직화된 교수보다는 부분적인 숙달의 정도에 의존하는 Algorithmic Planning으로 발전하였다. 1980년대에 들어서 GUIDON, MENO-TUTOR 등에서는 Means-Ends Planning 방법이 쓰여졌다. 이는 일단의 규칙(Rules)을 사용하여 현재의 학습 과정과 전의 학습 상태를 관련지어 교수 계획을 작성하는 것이다. 그림 4는 MENO-TUTOR의 Discourse Management Network로

서 정상적인 흐름(Default Flow)에서는 각 Node에 연결된 Link를 따르며, Interruption 상황이 발생하였을 때에는 Network내의 어느 Node로도 전이될 수 있다. 이것은 물론 전체적인 관계(Global Context)에서는 이상이 없는 한도 내에서이며, Meta-Rule이 조정하게 된다. 이러한 방법은 매우 복잡한 Domain에 적용하기가 부담스러우며, 예측 불명한 학습자의 행동에 대해서 융통성이 부족하다. SIIP에서는 BB1, HEARSAY 등에서 사용되었던 Blackboard를 이용한 Planner를 제시하고 있다. Plan Blackboard는 Iobjective, Istrategy, Iprocedure, Iaction 등의 Level을 가지며, 각각은 총체적인 학습 목표로부터 구체적인 행동의 단계까지를 계획한다. Control Blackboard는 Problem, Strategy, Tactic, Focus 등의 Level로 구성되어 각각의 Level에서 Invoke되거나 Trigger되어야 할 사항들의 List와 순서 등을 결정하게 된다. 이러한 시스템은 Opportunistic Planning에 명시적이고 융통성있는 Planning 기능을 제공할 수 있다. 위의 여러 예에서 보는 바와 같이 어떠한 형식의 구조를 갖든, Pedagogical Planner는 일반적으로 3단계 이상의

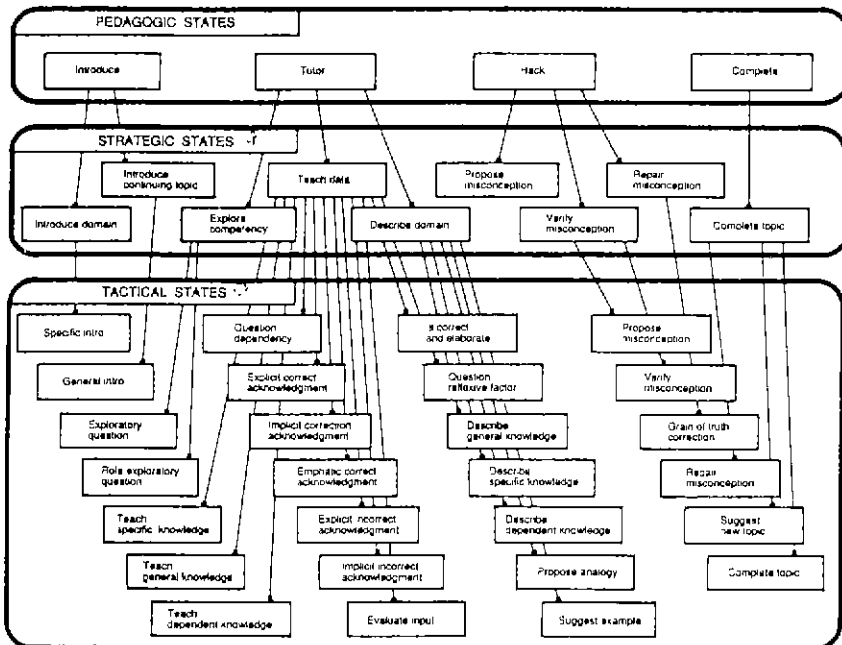


그림 4 MENO-TUTORP

Layer를 갖는다. 즉, 학습 목표(Objectives or Goals), 학습 전략(Strategy Plan), 학습 자료 제공(Action Plan) 등이다. 여기에 Curriculum의 단계가 최상위에 별도로 추가되기도 한다. 물론 모든 교수 계획에는 흥미와 동기의 유발이 학습 효과 증진에 필수 조건이다. 이러한 다단계 구조는 결론적으로 다음의 몇 가지의 기능을 제공하기 위한 것이다.

- 전체적인 학습 목적에 맞도록 각 단계별 학습 주제를 선정한다.
- 단계별 주제에 맞는 학습 전략 및 행동을 정한다.
- 학습자 모델에 수집된 것들 중에서 불확실한 정보를 해소할 수 있는 계획을 수립한다.
- 학습자의 Learning이나 학습 자료의 제공도중 예상치 못한 특이한 사항을 검증한다.
- 학습자가 예상과 다르게 행동할 때에는 교수 계획을 수정해야 한다.
- 교수 계획은 교수 중에도 끊임없이 자체 수정을 거쳐 발전되어야 한다.

4.2 교수 계획시 고려 사항

또한 여기서 간과해서는 안될 사실은 학습 과정에서의 교사와 학생의 정보 교환상의 제약이다. 교사의 측면에서 보면 다음과 같다.

- 교사는 학생의 지식 상태를 정확하게 측정하거나 관찰할 수 없다. 교사는 단지 교수를 통하여 간접적으로 학생의 지식 상태에 영향을 미치고, 시험 등을 통하여 그것을 측정할 수 있다.
- 교사는 교수 행위에 대한 학생의 반응을 확실하게 예견할 수 없다. 학습이 어떻게 이루어지는가에 대한 정확한 이론은 아직 정립되어 있지 않으며, 학습 전의 학생에 대한 지식 상태를 알 수도 없다.
- 교사가 교수하는 목표와 학생이 학습하고 있는 목표는 대개의 경우 서로 상이하다. 가장 효과적인 교수와 우수한 학습의 경우가 가장 접근한 경우라 할 수 있다.
- 교사는 차후 또는 차차후의 교수 계획이

있어야 하며 이는 학생의 행동에 대한 예측을 포함한다.

- 교사는 광범위하고 다양한 교수 전략을 준비하고 있어야 한다.

학습자의 측면에서 보면 다음과 같다.

- 학생의 지식 상태는 교사의 교수 방법을 포함하여 여러 요인에 의해 영향을 받게 된다. 교사의 교수 방법에 따라 긍정적, 부정적 영향, 또는 전혀 영향을 미치지 않을 수도 있다. 교수 행위에 의해서 학생의 지식이 항상 향상되는 것은 아닌 것이다.
- 학습하기 전의 학생의 지식 상태는 다르며, 각 개인의 사전 지식과 학습 요령에 맞는 개별화된(Individualized) 교수가 이행될 때 가장 효과적인 학습이 이루어질 수 있다.
- 학습의 결과로서 학생의 지식이 항상 향상되는 것은 아니다. 새로운 지식의 습득으로 인하여 혼동이 일어날 수도 있으며, 관련 지식의 추론 과정의 곡해나 중요성의 오해로 인해서 왜곡되거나 편향된 지식을 얻을 수도 있다.
- 새로 습득한 지식은 기존의 지식과 분리되어 저장되는 것이 아니라 학습 과정 중에서 함께 종합(Integration)되어 저장된다.

5. 학습자 모델 (Student Model)

5.1 Student Model과 Student Modeling

ITS는 학습자 개개인의 능력과 수준에 맞는 개별화된 교과 내용이 제공되며, 매단계에서의 학습자의 반응에 따라서 적절한 교수 계획이 수립되고 그에 따른 학습 내용을 제공한다. 따라서 항상 학습자의 지식 상태와 행동 양식을 분석 저장해야 한다. 학습자가 갖고 있는 Misconception이나 추론 과정상의 오류 등을 분석하여 Pedagogical Planner에게 제공하고 Pedagogical Planner는 학습자에게 그러한 잘못된 지식을 지적해 주고 수정할 수 있도록 하는 과정을 밟게 된다. 따라서 ITS에서의 Student Model은 시스템의 질적인 성능을 결정하는 가장 중요한 구성 요소가 된다. Student Model은 더 정확히 표현

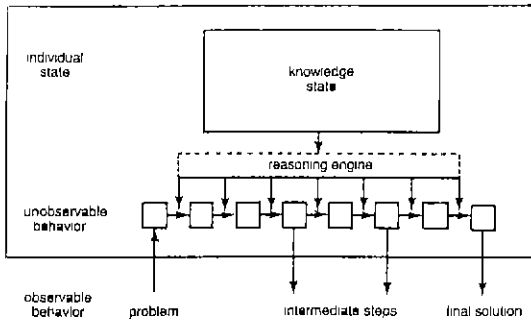


그림 5 개념적 Cognitive Framework

하자면 Cognitive Student Model로서 정적인 Database가 아니고 Student가 무엇을 하려고 하는가를 설명할 수 있는 생각하는 존재인 것이다. Student Modeling은 학습자의 응답으로부터 Student Model을 구축하는 작업을 말하며, 진단(Diagnosis), 모델 갱신(Update) 및 모델 검색(Retrieval) 등의 작업이 순차적으로 또는 동시에 이루어진다. 응답의 진단(Diagnosis)은 추출 과정의 역추론(Inferencing), 행동 양식의 해석(Behavior Interpretation), 이상의 결과와 과거의 상태와를 결합한 종합적인 판단(Classification) 등의 과정을 거친다. 이에는 모든 응답은 학습자의 지식과 추론 능력의 결과라는 가정이 전제되어 있다. 그림 5는 Cognitive Student Model의 개념을 설명하고 있는데 Behavioral Level은 현재 시도되고 있는 대부분의 시스템들은 이 수준

에서 머물고 있다. 학습자의 Unobservable Behavior는 학습자 두뇌의 내부에서 일어난 모든 추론 과정에 필요한 지식의 사용 양태이며 시스템 상으로는 관측이 안되는 것으로 Knowledge 자체와는 분리된다. 실제로 그 차이는 미약하며, Communication상 야기되는 것으로 간주한다. 그림 6은 이 Level에서의 여러 가지 구체적인 방법과 예를 보여 준다. Individual Level은 개인의 특성으로서, 지식의 습득 양태(Acquisition of Knowledge), 개성(Personality), 학습 동기(Motivation: 의욕, 관심, 피로함 등등), 환경(또는 경험), 기계에 대한 거부감 등을 의미하며, 현재로서는 매우 어려운 과제이며, 교육심리학자들의 도움이 필요한 분야이다. Student Model은 크게 Overlay Model 방법과 Bug Library 방법의 두 가지로 구분할 수 있다.

5.2 Overlay Modeling

Overlay Model은 전문지식을 여러 조각의 독립적인, 그러나 서로 연관된 세부적인 지식으로 나눌 수 있으며, 그 각각의 세부적인 지식 조각들이 학습되어 평면적인 집합으로 더하여지면 원래의 전문지식을 습득한 것으로 간주하는 데에서부터 출발한다. 전문가와 비전문가 또는 초보자와의 구별은 세부적으로 분리된 지식들의 전체집합과 부분집합을 각각 소유하고 있는 것

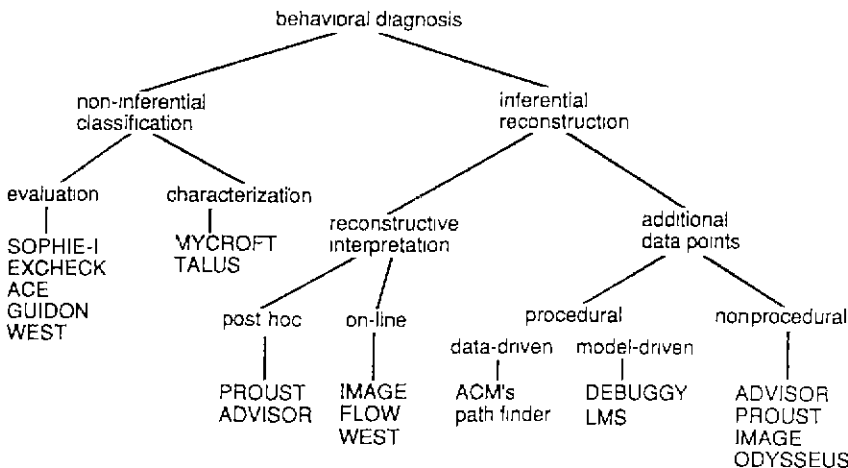


그림 6 학습 양태 분석

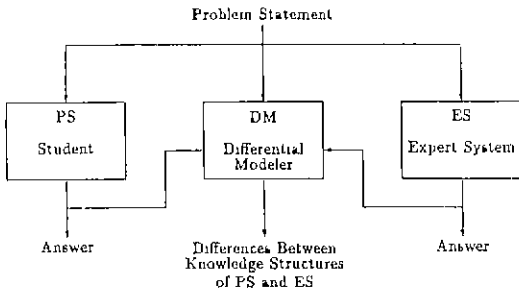


그림 7 Differential Model

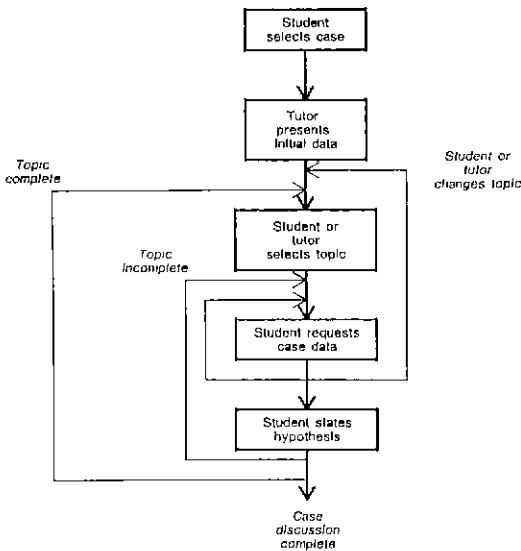


그림 8 GUIDON의 교수 과정도

으로 한다. 비전문가나 초보자의 부분집합을 전문가의 전체집합으로 발전시키는 지식의 불완전성의 완성(Incompleteness-to-Completeness)이 바로 이방식을 이용한 교육의 목적인 것이다. 부분집합과 전체집합과의 차이라는 의미에서 Differential Model(그림 7)이라고도 부르는데

SCHOLAR, WUSOR, GUIDON 등이 대표적인 시스템이다. 그림 8에서 보듯이 모든 topic이 완료될 때까지 교수는 계속된다. 그림 9는 GUIDON의 Student Model의 구조와 흐름을 보여준다. Student Model은 Domain Rule들과 Goal들로 모여서 이루어진 Database이며, 'USE-HISTORY', 'SAPPLIED', 'USED'의 부분으로 되어 있다. 'USE-HISTORY'는 학습자의 현재의 지식상태를 나타내며, 과거의 응답의 분석 결과이다. 따라서 교수 계획자(Pedagogical Planner)는 교수 계획을 수립할 때에 여기에서 정보를 얻게 된다. 'SAPPLIED'는 학습자가 특정한 문제 해결 과정에서 어떠한 지식들이 필요한지를 알고 있는가에 대한 정보를 간직하고 있다. 'USED'는 학습자가 문제를 해결하는 과정에서 실제로 그러한 지식들을 사용했는가를 보여준다. 'SAPPLIED'가 필요한 지식들을 알고 있다는 것을 보여주고, 'USED'가 실제 사용을 보여주게 되면, 'USE-HISTORY'은 학습자가 그 지식들을 알고 있는 것으로 기록을 경신하게 되고 이것은 이어지는 교수 계획에 기본이 된다. 이 과정에서 지식의 Misconception이나 Incorrectness라는 개념은 존재하지 않는다. 다만 생략되거나 빠뜨린 지식들만이 교수의 대상이며, 'USE-HISTORY'에 모든 Domain Rule들이 학습자에 사용된 것으로 기록되면 학습은 완료되고, 학습자는 전문가와 동일한 지식과 문제 해결 능력을 가진 것으로 간주한다. 이러한 개념은 지식의 정체를 사실의 나열이나 평면적인 서술로 너무 과다하게 단순화시킨 근본적인 약점을 지닌다. 그림에도 불구하고 Overlay Modeling은 Program 실현상의 단순함과 Knowledge Engineering의 용이함이라는 장점을 가지고 있어서 복잡하고 광대한 양의 지식을 내포한 영역(Do-

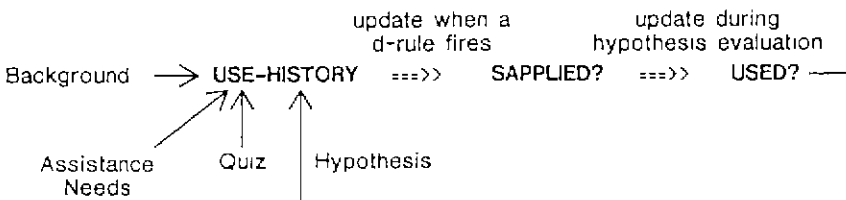


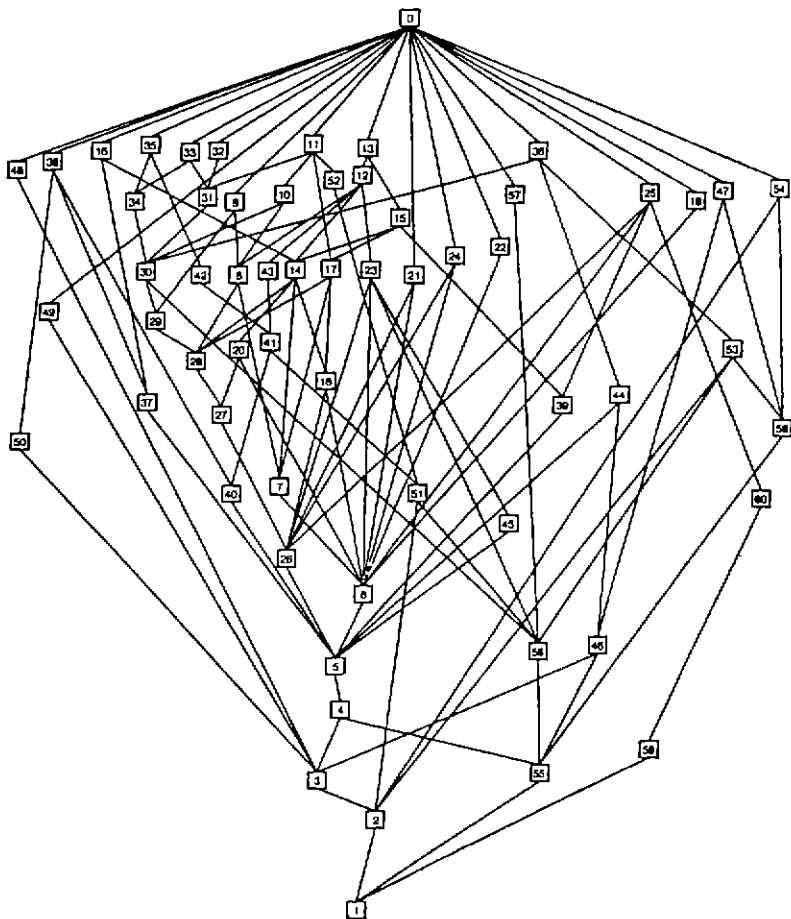
그림 9 GUIDON의 Student Modeling

main)을 대상으로한 ITS에서 주로 쓰이는 방법이다. 또한 최근에는 Overlay Model속에 Buggy (Misconception 또는 Incorrect Knowledge)적인 요소를 가미하는 경우도 있다. 원래의 Overlay Model을 Strict Overlay라 부르는 것과 비교하여 Extended Overlay라고 하기도 하며[35], ACTP 등이 좋은 예에 해당한다.

5.3 Bug Library Modeling

문제의 해답(Optimal Solution)이나 결론(Hypothesis)의 도출(Reasoning)에 있어서의 오류는 종종 잘못된 세부 지식(Pieces of Knowledge)이나 문제와는 전혀 무관한 지식을 관련 시킴으로써 야기된다. 잘못된 지식들, 즉 Misconception

이나 세부 지식들간의 상관관계에 대한 잘못, 문제 해결 과정에서의 잘못 적용되는 Algorithm이나 Reasoning Paths 등을 수정하는 것이 학습의 요체가 되는 것이다. 이러한 것들을 Bug라고 부르며, Bug를 수집하여 놓은 것이 Bug Library이다. 학습의 과정에서 학습자에게서 이러한 Bug를 발견하여 교정함으로써 학습의 목적을 달성하는 것이며, Overlay Model에서 발생하는 근원적인 문제점을 해결하려는 시도이다. Bug Library Modeling에서는 Bug의 양과 질이 ITS의 질적 성능에 결정적 요소가 되며, 동시에 해결해야 할 가장 중요한 난관이다. 유용한 Bug는 주제 영역에 대한 실험 데이터의 세심한 분석에 의해서만 가능하다. Overlay Model의 세부 지식은 전문가의 의견만으로 취득이 가능



Skill lattice for Subtraction
그림 10 (계속)

- 0 Correct skills
- 1 No subtraction skills
- 2 Subtract 0 from a number whose right-most digit is 0
- 3 Subtract 0 from a number
- 4 Subtract columns which are n-n or n-0
- 5 Subtract without borrowing
- 6 Borrow from some columns with a digit on the bottom
- 7 Borrow from columns whose answer is the same as their bottom digit
- 8 Borrow from a column with a 9 or a larger number on top
- 9 Borrow from a column with a larger top digit
- 10 Borrow from a column with zero on top when blank on bottom
- 11 Borrow from a column with zero on both top and bottom
- 12 Borrow from a column unless one is on top and a nonzero digit is on the bottom
- 13 Borrow from a column unless digit one is on both top and bottom
- 14 Borrow in two consecutive columns
- 15 Borrow from columns with the same digits on top and bottom
- 16 Borrow from columns that have nine as the bottom digit
- 17 Borrow from columns with the same nonzero digits on top and bottom
- 18 Borrow from columns which have an answer of zero
- 19 Borrow from the leftmost column when it has a non-blank in the bottom
- 20 Borrow more than once per problem
- 21 Can borrow then not borrow
- 22 Borrow from columns with two on the bottom
- 23 Borrow from leftmost columns or columns that have a non-one on top
- 24 Borrow from columns with top digits smaller than the bottom digits
- 25 Borrow from columns that have a top digit one less than the bottom digit
- 26 Borrow from the leftmost column
- 27 Borrow once in a problem
- 28 Borrow from columns that have the top digit larger than the bottom digit
- 29 Borrow from a column with the top digit greater than or equal to the bottom digit
- 30 Borrow from a column with a zero on top
- 31 Borrow from a middle (not leftmost) column with a zero to the left
- 32 Borrow from leftmost column of a problem whose form one followed by one or more zeroes
- 33 Borrow from a column with a zero on top and a zero to the left
- 34 Borrow from or into a column with a zero on top and a zero to the left
- 35 Borrow into a column with a zero on top and a zero to the left
- 36 Borrow from a column with a zero on top and a blank on bottom
- 37 Borrow into a column with a nonzero digit on top
- 38 Borrow into a column with a one on top
- 39 Borrow when difference is 5
- 40 Subtract columns with a one or a zero in top that require borrowing
- 41 Subtract columns with a zero in the top number that require borrowing
- 42 Subtract a column with a zero on top that was not the result of decrementing a one
- 43 Borrow into a column with a zero on top when next top digit is zero
- 44 Borrow from a column with a blank on the bottom
- 45 Borrow from a column with a one on top
- 46 Subtract numbers of the same lengths
- 47 Subtract a single digit from a large number
- 48 Subtract columns unless the same digit is on top and bottom
- 49 Borrow all the time
- 50 Subtract when one is in top
- 51 Subtract when neither number has a zero unless the 0 is over a 0
- 52 Subtract columns when the bottom is a zero and the top is not zero
- 53 Subtract when a column has a zero over a blank
- 54 Subtract numbers which have a one over a blank that is not borrowed from
- 55 Can subtract a number from itself
- 56 Subtract numbers with zeros in them
- 57 Subtract problems that do not have a zero in the answer
- 58 Subtract numbers when the answer is no longer than the bottom number
- 59 Subtract leftmost columns that have top and bottom digits the same
- 60 Subtract columns that have top and bottom digits the same*the same ones.

그림 10 Bug Diagram

하나, Bug의 수집은 원칙적으로 비전문가나 초보자를 교육하면서 얻는 실험적인 방법이 유일한 방법이다. DEBUGGY의 경우 수 천개의 감산(Subtraction)문제를 학생들에게 시험을 치루도

록한 후 그 결과를 직접 하나씩 손으로 분석하여 수십개의 Bug를 추출하였다[11](Enumerative Bug). 그림 10은 그 일부를 보여주며, 되풀이되는 문제 해결과정을 통하여 이러한 Bug가 모두 교

정되었을 때 학습자는 전문가와 같은 문제 해결 능력을 가진 것으로 평가된다. Bug Library가 존재할 경우에는 그것을 이용하여 학습 과정에서 Bug를 예측할 수 있으며, 분석 (by Repair Theory[8])에 의하여 학생들의 응답중에서 새로운 Bug를 추출하기도 한다(Generative Bug). 또한 ACM에서와 같이 모든 Bug가 사전에 정해지는 대신에 학습자의 응답을 진단하는 과정에서 추출할 수도 있다(Reconstructive Bug). Predicate Library에는 P개의 Predicate가 있으며 Action Library에는 A개의 Action이 있다면, $A \times 2^P$ 개의 bug가 가능하게 된다. 이러한 경우를 일반적인 Bug Library와 구별하여 Bug Part Library라고도 부른다[34]. Bug Library 방법은 전문지식 교육의 현장에 근접한 효과적인 수단임에는 틀림이 없으나, Bug 수집상의 난점으로 인하여 비교적 단순하고 쉽게 정의될 수 있는 영역(Domain)에 응용되고 있다.

5.4 진단 및 분석(Diagnosis)

ITS에서 진단과 분석이라함은 학습자의 지식, 지식의 결함, 표면적인 Bug 및 그와 관련되어 함축된 오류(Misconception)를 지능적으로 추출해 내는 것을 의미한다. 이에는 여러가지의 방법들이 제안되고 있으며, 그중의 몇 가지를 소개하면 다음과 같다.

Model Tracing: 부여된 문제로부터 결론(Hypothesis)까지의 추론 과정을 추적하여 전문가의 추론 경로(Paths)와 비교하는 방법이다. 추론 중의 각 단계에서 다음 단계로 추적할 때에 추론 전개가 가능한 다수의 경로중에서 학습자의 것에 해당하는 경로를 추적하여 학습자의 지식 상태 및 추론 과정을 분석하는 방법이다. 추론 경로는 대개의 경우 결론에서부터 원인을 찾아가는 역방향(Backward) 경로가 구축된다. 그림 11은 ACM[23]의 탐색도(Search Tree)로서 '93-25=72'라는 학습자의 오답을 분석한 것이다. 이러한 방법은 문제의 유형(Problem Space)이 투명하고 명확하게 정의될 수 있어야 하며, 추론 경로가 다양하고 실질적으로 제공되고 같은 종류의 문제에 대해서는 동일한 경로와 순서가 적

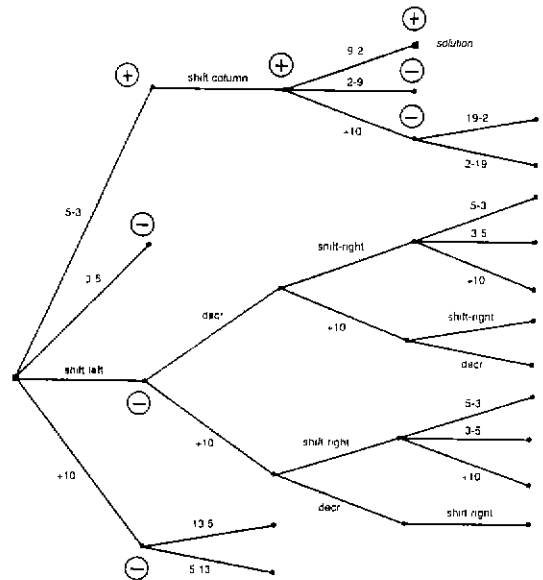


그림 11 ACM의 탐색 트리 경로

용될 수 있어야 한다. 이러한 조건이 만족되더라도 한 단계에서 다음 단계로의 경로가 복수인 경우는 선택의 문제가 주어질 수 있다.

Path Finding: DPF[27]에서는 위에서와 같은 Model Tracing의 문제점을 해소하고 특정한 Domain에만 적용되지 않도록(Domain-Independent) 'Best-First' 알고리즘을 적용하였다. 실행중에 적시에 적절한 최상의 경로를 찾을 수 있는 'Psychological Heuristics'를 제안하고있다. 경로의 적합성의 가부를 묻는 절대적 평가 기준(Assumptions)과 적합한 경로들의 우선 순위를 가리는 상대적 평가기준(Criteria)로서 다음과 같다.

Assumptions으로는 영역의 닫힘성(Causal Closure), 목적성(Purposefulness), 비반복성(No Duplication) 등이며, Criteria는 간결성(Memory Load), 세부 목표(Subgoalng), 생산성(Productivity), 최소의 오차(Minimal Error), 최단 경로(Minimal Length) 등이다.

Condition Induction: Model Tracing 방법에 있어서 한 단계에서 다른 단계로의 전환은 정해진 Rule에 의해서 이루어진다. 최소량의 Bug를 가진 Bug Library(혹은 Bug Part Library)로서 많은 종류의 Bug를 검출하고자 하는 것이 이

방법으로, ACM, REPAIR 등에서 시도되었다. Operator Library와 Predicate Library 등의 2개의 Library가 각각 행동과 그행동의 조건을 가지고 있어서 Production Rule을 형성하여 학습자의 지식에 내포된 Bug를 생성한다.

이 밖에도 Plan Recognition, Issue Tracing, Decision Tree, Expert System, Generate and Test, Interactive Diagnosis[34] 등 여러가지의 분류가 있을 수 있으며, 이러한 방법들은 대개의 경우 어느 한 가지만 독립적으로 쓰이는 것이 아니라 Domain의 특성에 따라서 복합적으로 쓰인다.

5.5 Information about the Student

5.5.1 Source of Information

ITS와 같이 정보를 분석하여 그 결과를 이용하는 시스템에 있어서, 취득 가능한 데이터의 범위와 질적인 특성은 그 시스템의 성능을 결정적인 요소이다. ITS는 기계와 인간의 통신을 정보 교환의 수단으로 사용하기 때문에 인간과 인간의 직접적인 정보 교환에서 얻을 수 있는, 부차적이지만 대단히 유효한 얼굴 표정, 음성의 Tone이나 속도, 또는 반응 시간 등의 정보를 얻을 수 없는 결점이 있다. 최근에 개발된 Multimedia 시스템을 이용하여 초보적인 발전을 보고 있으나, 인간 사이의 직접 통신에 비하면 아직은 무시되어도 될 정도의 수준이다. 현재의 ITS에서는 학습자에게 주어진 문제와 그 대답을 분석하여 학습자에 대한 모든 교육 자료(Diagnostic Data)가 얻어진다. 학습자의 직접적인 대답의 맞고 틀림에 의한 정보는 명시적인 증거(Explicit Evidence)이며, 학습자의 질문이나 도움 요청도 학습자의 지식 상태에 대한 간접적인 증거(Implicit Evidence)로서 이러한 경우는 학습자가 자신의 지식에 대해서 확신을 갖지 못한 것으로서 부정적인 증거(Negative Credits)가 된다[17]. 학습자의 응답이 옳을 경우는 그 지식에 관련된 세부 지식(Pieces of Knowledge)을 알고 있다는 해석이 가능하며, 간혹은 잘못된 중복에 의해서 우연히 생성되는 경우도 있다[31]. Student Mo-

del의 데이터를 추출하여 이전의 지식 상태를 분석하면 판단에 도움이 될 수 있다. 오답의 경우는 ITS의 학습의 필요 이유가 되며, 분석의 주목표가 된다. 오답의 원인은 여러가지가 될 수 있으며, 잘못된 지식(Misconception) 자체가 원인이 될 수도 있고, 문제 추론 과정상의 오류에 의할 수도 있다. 다시 말하면, 잘못된 조건이나 데이터를 적용하거나, 전혀 관계가 없는 사항들을 연관시킬 수도 있다. 좀 드물기는 하지만, 학습자 스스로 새로운 Algorithm을 개발하는 수도 있다. 여기서의 새로운 Algorithm은 대부분의 전문가들이 결론 추출과정에서 인용하는 단계가 아닌 것을 의미한다. 이것은 전문가들이 세부 단계를 분리할 때의 과정이나 방식과 상당히 거리가 있는 경우를 포함하며, 정답인 경우는 추적이 가능할 수 있으나, 오답인 경우는 추적이 거의 힘들다. 이러한 분석 과정에서 유의할 것은 분석을 위한 데이터 수집은 교수의 일부분으로서 행하여져야 하며, 학습을 지연시키거나 지장을 주어서는 안된다는 것이다. 시스템내의 Student Model과 학습자간의 불일치점을 명확히 한 후, 최소의 테스트 문제를 선정하여, 학습의 계속성이나 집중 등의 교수 계획에 장애를 주지 않도록 교수 계획의 순서에 삽입해야 한다. 소크라테스식 대화법을 사용하는 WHY, GUIDON 등에서는 학습자의 지식을 탐색하기 위한 목적으로 관련된 예측이나, 가정(Hypothesis)을 확인하는 질문을 한다. 한 걸음 더 나아가, MACSYMA ADVISOR [22]에서는 가장 타당성이 있는 가정을 도출하기 위해서 학습자의 의도나 행위에 대한 직접적인 질문 대신에 시스템 자신의 가정에 대한 대화를 하며 그 결과로서 진단한다.

5.5.2 Noise

학습자로부터 얻어지는 정보는 한 사람의 학습자에게서 생성된 것으로 생산 근거인 학습자의 지식은 동일한 것이다. 따라서 학습 중의 학습자에 대한 모든 정보가 일관성이 있어야 한다는 것은 아주 당연한 일이다. 그러나, 실제로 ITS에 취득되는 정보는 그렇지가 못하다. 그 원인은 여러가지가 있으며, 몇 가지를 살펴보면 다음과 같다[31].

◦ 학습에 의한 변화: 학습에 의해서 새로운 지식을 습득하게 되면 학습 전에 학습자가 옳다고 믿었던 지식이 틀린 것을 알게 되며 학습자는 전과는 다른 응답을 하게 된다. 경우에 따라서는 학습 전의 지식이 모두 바뀔 수도 있으나, 학습자가 받는 감동의 정도에 따라서 지식의 편향이 이루어질 수도 있게 된다. 그러나, 학습이 항상 학습자의 응답의 변화를 일으키지는 않는다. 이해를 못하거나, 곧 잊거나, 또는 자신의 현재의 지식에 대한 믿음이 매우 강할 때에는 학습에 큰 영향을 받지 못할 수도 있다.

◦ 학습자 자신의 지식에 대한 확신이 부족할 때도 응답에 일관성이 없게 된다. 이러한 경우는 질문 환경 즉, 질문 형태나 분위기에 따라 응답이 다를 수 있게 된다.

◦ 학습자는 자신이 해결할 수 없는 문제에도 단순한 추측으로 답을 만드는 경우가 종종 있다. 특히 선택형의 질문이 주어지면 대개의 경우, 근거가 있는 답을 구하지 못하더라도 어느 하나는 선택하게 된다. 후에 또같은 문제가 주어지면 전혀 다른 선택을 할 수 있다.

◦ 학습자의 응답을 정확하게 해석하는 것은 상당히 어려운 일이다. 여러가지 제약이 따르는 컴퓨터 시스템에서는 더욱 어려운 일이다. 응답은 학습자의 내부 지식이 만들어 낸 결론이지만, 우리는 이 내부 지식중에 어떤 것이 어느 만큼 관련되어 있고 또 어떠한 과정을 거쳐서 만들어졌는지 알 수 있는 방법을 가지고 있지 못하다. 따라서, 우리는 현재의 Student Model에 보관되어 있는 정보와 현재의 응답을 함께 분석한 결과가 학습자가 실제 가지고 있는 내부 지식 상태와의 차이(Gap)가 최소화되도록 노력할 뿐이다.

◦ 인간들은, 특히 대부분의 학생들은 자신이 알고 있는 것을 모두 표현하지를 못한다. Burton과 Brown[12]은 이를 Cognitive Tear라고 부르기도 했다.

학습 과정에서 얻어진 정보, 특히 일관성이 결여된 정보중에는 학습자에 대한 정확한 정보가 아닌 것이 상당히 많이 존재하며 이것을 Noise라 한다. ITS의 연구하는 많은 학자들이 자신들의

시스템을 설명하는 과정에서 교사의 믿음(In Tutor's Belief)라는 용어를 쓰는 것도 이 때문이다. 이에 대한 해결책으로는 여러가지 방법이 시도되고 있으나 현재까지는 크게 두 가지 정도로 분류할 수 있다[35]. 첫째는 단정적인 증거가 없는 경우로서, Model Tracing 형태의 ITS에서 볼 수 있다. ACTP 경우에서처럼 자기추론에 의한 선택(Heuristic Selection)으로 결론에 이르는 여러 경로들을 서로 비교하여 각기 다른 비중치를 주어서 선택하는 것이다. 가장 비중치가 무거운 추론 경로를 선택하여 학습자의 응답에 대한 해석으로 간주하는 것이다. 다음으로는 여러 번의 응답을 축적하여 통계 처리하는 방법이다. GUIDON에서와 같이 학습이 진행되는 과정에서 동일한 항목에 대한 학습자의 응답을 정답에는 긍정적인 값(Credits)을, 오답에는 부정적인 값(Blame)을 누적시킨 후, 누적된 값이 일정한 범주(Range)내에서 처리하는 것이다.

6. Situation-Based Instruction과 Multimedia를 이용한 ITS

최근에 와서 교육학자들 사이에 Situation Instruction의 중요성이 강조되면서 ITS에도 응용이 되고 있다. Situation Instruction이란 학습 주제에 알맞는 종합적인 상황을 설정하여 학습자가 그 상황 속에서 경험을 할 수 있도록 하는 교육 방법이다(Experiential Learning). Adams[1] 등에 의하면 지식에는 사실로서의 지식(Information as Facts)과 문제 해결의 도구로서의 지식(Information as Tools) 등의 두가지가 있다. Situation Instruction은 도구로서의 지식의 학습에 중점을 두며, 각각의 사실들을 주어진 상황에 적절히 적용하여 문제를 해결할 수 있는 능력을 키우게 된다. 이 방법의 이론적인 근거는 Case-Based Reasoning에서와 같은 것으로, 인간이나 기계나 모두 문제 해결능력이나 이해력의 근거는 사전에 겪었던 특정한 경험이라는 것이다. 여기에서는 이러한 부류에 속하는 여러 방법론들 중에서 최근에 많이 시도되고 있는 Story-based Instruction, Cognitive Apprenticeship과 Anchored Instruction 등에 대해서 간략하게 설명하고

자 한다.

6.1 Story-based Instruction

어릴적에 우리는 우리의 어머니 할머니로부터 교훈적인 옛이야기들을 많이 들어왔다. 옛이야기는 사실적인 분위기를 흥미롭게 구성하여 여러가지의 함축된 지식을 자연스럽게고도 효과적으로 전달하는 교수법의 일종인 것이다. 이것은 이솝식 대화법(Aesopic Dialog)라고 하여 소크라테스식 대화법(Socratic Dialog)과는 구별된다. 적절한 상황 설정하에서 이루어지는 적절한 이야기는 정보의 전달이라는 목적외에 비슷한 상황에 대한 회상을 유도하는 부수적인 효과도 있다. 즉 그 이야기속에 내포된 정보와 학습자의 기억속에 이전에 내장되어 있던 것과의 연결(Related Index)이 이루어진다. 또한 이솝식 대화에서는 학습자가 대화의 전후 내용을 연결 추적하여 그 속에 내포된 교훈을 추출하는 사고 능력을 배양한다. 여기서 Aesop식 대화는 그 속에 이러한 목적의 Follow-Up질문이 포함되어야 하며 교사는 이에 대한 답변은 물론 새로운, 때로는 무관한 질문에도 답할 준비가 되어 있어야 하는 것이다. 이러한 방법은 ITS에서도 응용이 되어 Ferguson[19] 등은 ASK에서 신탁회사 상담관 등의 교육에 적용하고 있다.

6.2 Cognitive Apprenticeship과 Anchored Instruction

6.1.1 Cognitive Apprenticeship

Cognitive Apprenticeship[18]은 중세의 도제식 수업을 응용한 것으로 학습자는 전문가가 생각하는 것과 똑 같이 생각하고, 전문가가 서술하는 것과 똑 같이 서술하는 것이다. 이것은 전문가가 이용하는 똑 같은 도구와 자료를 사용하면서 그리고 같은 형태의 일을 하면서 배우므로, 일단 학습이 완료된 후에는 다른 형태의, 익숙지 않은 문제에는 쉽게 적응치 못한다는 결점이 있다. 이러한 학습법은 읽기, 쓰기, 수학풀이 등에 효과적이다. 교수의 내용으로는 문제 해결에 필요한 사실과 개념 및 그의 적용과정으로 구성된

영역 지식(Domain Knowledge), Heuristic Strategy, Metacognitive인 제어 전략(Control Strategy), 학습 전략(Learning Strategy) 등이다. 교수방법으로는 전문가의 문제 해결 과정을 실연하는 Modeling, 학습자의 행동을 관찰하여 즉석 Feedback을 제공하는 Coaching, 학습자가 문제 해결 능력이 부족할 때 조언을 주는 Scaffolding 등의 방법이 쓰인다[36].

6.2.2 Anchored Instruction

Anchored Instruction[6]은 학습자의 흥미를 유발하고 학습자들이 자신의 인식력(Perception)과 이해력을 이용하여 문제를 정의하고 이해하도록 하는데 도움을 주는 Anchor를 이용한 교수 방법이다. 학습자는 Anchor를 통하여 문제의 핵심을 이해하고 상황을 새로운 관점에서 관찰하여 인식의 변환을 스스로 경험케 하는 것이 목적이다. Anchor는 특성상 흥미로워야 하며 학습자가 관련 세부 문제와 그에 대한 세부 목표를 포함한 모든 문제를 처리할 수 있도록 도와 줄 수 있어야 한다. 예를 들면 남미의 정글지대를 여행하면서 발생하는 문제들을 해결한다든가 비즈니스 케이스를 다룬다든가 하는 것이다. 이상적인 Anchor는 여러 개의 다른 행동들이 서로 연관을 갖도록 설정되어야 하며 문제 상황의 특성을 학습자가 이해하기 쉽도록 할 수 있어야 한다. 가령 비서를 교육한다고 하면, 비서가 외부 손님을 초청 단계에서부터 회의 일정, 면담 계획 등의 회사내 일정, 호텔 예약, 식사 장소 및 시간, 교통 수단 등을 실제 상황으로 설정하여 각각의 단계를 경험케 하는 것이다. 이러한 시스템에서는 1990년대에 들어서 급격한 발전을 하고 있는 Multimedia 컴퓨터 기술을 이용하여 효과적인 교육 효과를 얻고 있다. Multimedia 컴퓨터는 모든 정보가 텍스트화된 전통적인 컴퓨터로부터 시각과 청각을 그대로 이용할 수 있게 되므로서 ITS에도 새로운 형태와 방법론의 등장을 유도하게 되었다. Anchored Instruction은 이의 좋은 예로서 Multimedia 컴퓨터를 이용하여 시제와 같은 상황을 설정하여 학습자가 차후에 그와 유사한 상황에 직면했을 때 스스로 응용할 수 있게 하는 것이다. Anchored Instruction의 기본 개

넓은 학습자의 사고력을 유도하여 여러 관련된 지식을 교육한다는 것이다. 즉, 문제를 풀 수 있는 능력을 키워 주자는 것이다. 이것은 많은 교육 학자들의 주장으로 학습에 있어서 교사는 학생이 스스로 생각하고 문제를 해결하도록 유도하므로써 교사의 역할은 단지 조력자이어야 한다는 것이다[24]. HANDY[26]는 text와 video 화상을 이용하여 학습자의 흥미를 유도하면서도 학습자 스스로 문제를 설정하고 관련된 지식을 종합하여 해답을 구하는 훈련을 할 수 있도록 하고 있다.

7. 일반적인 교육 시스템의 여러 형태

지능형 교수 시스템을 포함하여 현재 개발되고 사용되는 교육용 시스템의 형태는 매우 다양하며, 어떤 명확하고 일반화된 기준에 따라 분류한다는 것은 상당히 어려운 일이다. 분류가 모호하며 중간쯤의 특성을 가진 것도 있고, 여러 가지의 특성을 복합적으로 가지고 있는 것도 많다. 일반적으로 교육 시스템은 정보의 제시, 학습자 인도, 학습자에 의한 연습, 학습자의 성취도 평가 등의 네 단계를 필요로 한다. 그림 12는 이러한 교육 시스템의 일반적인 단계와 흐름을 보여준다. 정보 제시 단계에서는 규칙이나 예의 제시, 그림의 제공 등을 글이나 소리, 음성, 정지 및 이동 화상의 방법으로 학습자에게 제공한다. 학습자 인도 단계에서는 학습자를 관찰한 후 오류의 수정, 제안 및 힌트의 제공 등이다. 이 과정에서는 Discovery Learning[28] 이론을 응용하여 학습자가 교육 시스템의 도움으로 실험과 연습의 과정을 통하여 스스로 원리를 발견하거나 숙련이 되도록 하는 Guided Discovery의 방법도 효과적이다. 반복적인 연습은 빠르고 유창하게

할 수 있도록 하는 효과뿐 아니라 오래 기억하는 데도 도움이 된다. 평가는 학습 수준, 교수의 질을 평가하고 차후의 교수 계획에 참고가 된다. 학습자에 대한 정보 전달의 형태에 따라 몇 가지로 분류하여 보면[2], Tutorial, Drill, Simulation, Instructional Game, Test 등으로 나눌 수 있다. Tutorial은 위의 네 단계중에서 정보의 제시 및 학습자 인도에 중점이 주어진 것으로 반복 연습이나 평가는 종종 생략된다. 이것은 거의 모든 분야의 주제에 이용되고 있으나, 사실적인 정보의 제공, 규칙이나 원리의 설명, 문제 풀이 전략의 교육 등에 효과적이다[21]. Drill은 반복 연습에 의한 숙련에 비중이 주어진 것으로서 기초적인 산술 계산, 외국어 공부, 철자 및 단어 암기, 역사, 과학적인 개념 등의 교육에 효과적이다. 이러한 Drill 시스템에서는 학습자가 기초적인 지식은 이미 습득한 것으로 가정하며, 다량의 문제를 생성하고 그것들을 가장 효과적인 순서와 난이도로 제시하는 것이 중요하다. Tutorial에서는 정보 제시에 이은 학습자와의 질문과 응답이 진행되는 반면에, Simulation에서는 현실과 같은 환경을 만들고 학습자가 그 속에서 일어나는 작용이나 과정을 직접 관찰하고 실험하는 과정에서 학습하는 것이다. 이는 또 두가지로 나눌 수 있는데, 물체의 운동역학을 주제로 한 Mechanics [13] 등과 같은 물리적인 것에 대하여서는 그것이 무엇인지에 대하여 학습하게 되며, 학교에서 일어날 수 있는 상황을 주제로 하는 School Transactions[25]같은 진행 절차나 상황적인 것은 그것에 어떻게 대응하거나 또는 그것을 어떻게 처리할 것인가에 대한 학습이 된다. 이러한 Simulation 시스템은 여러가지의 장점을 갖고있다. 위험을 동반한 교육의 경우에 안전성을 확보할 수 있으며, 현실 세계에서는 경험하기 어려운 경험을 축적할 수도 있고, 장시간을 요하는 실험의 경우는 시간을 단축할 수 있으며, 교육을 위해서 매우 복잡한 학습 환경이 필요한 경우에는 통제가 용이하며, 경비를 절약하는 등의 장점을 제공한다. Instructional Game은 오락을 즐기면서 학습을 한다는 면에서 가장 강력한 교육 시스템 중의 하나이다. 게임에는 몇가지의 특징이 있게 된다. 목표가 주어지고, 게임의 규

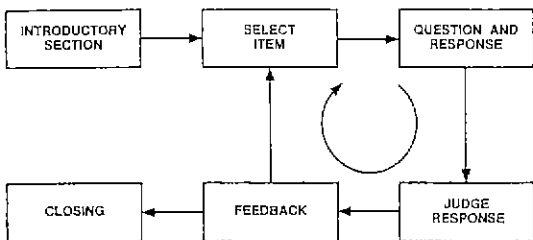


그림 12 교육 시스템의 단계

칙이 있고, 경쟁자가 존재하고, 도전적인 기회가 주어지기도 하고, 흥미를 도발할 수 있도록 환상적인 상황이나, 물체를 제공하기도 한다. 또한 정오답에 따라서 상벌점이 주어지기도 한다. 게임의 종류를 살펴보면, Oregon과 같은 탐험 게임(Adventure game), Pinball같은 아케이드형 게임, Chess같은 보드 게임(Board Game), Phizquiz같은 카드 게임, Moonwar같은 전투 게임, Rocky's Boots같은 논리 게임, Diffusion Game 같은 역할 게임(Role-Playing Game) 등 다양한 형태가 개발되고 있다. Test는 위의 네 단계 중에서 마지막 평가에 주목적이 주어지며, 평가 문제 은행(Item Bank)를 생성하고, 프린트하고, 채점에 쓰이며, 또한 평가 결과를 동시에 기록 처리하고 수험자에게 즉시 통고하는 데에도 사용된다. 지리학 테스트용인 Exambase, The Examiner[25] 등이 좋은 예이다. 최근에는 인공지능을 이용한 Adaptive Test라는 개념이 도입되고 있다. 수험자(학습자)의 응답에 따라서 차후에 제시되는 문제가 선택되는 것이다. 질문에 대한 응답이 맞을 경우에는 이어지는 문제가 주어지고, 응답이 틀린 경우에는 그와 관련되어 오류를 수정하는 데 도움이 되는 문제가 선택되는 것이며, 이러한 Adaptive Testing에는 Latent Trait 이론, Item Characteristic 이론, Item Response 이론 등이 도입되었다. 또한 평가 결과의 분석에는 APM(Admissible Probability Measurement)[9] 또는 IRT(Information Referenced Testing) 등이 응용되어 수험자(학습자)의 응답의 부분적인 정오를 추정하기도 한다.

8. 결 론

미국의 국민학교 교실에서의 학습을 분석한 한 보고서[10]에 의하면 각 개인의 학생에 대한 교사들의 1대1(One-To-One)의 직접 대면(Face-To-Face)을 통한 의사소통(Teacher-Student Interaction)은 하루에 고작 2분 정도에 지나지 않는다고 한다. 또다른 한 보고서[15]에 의하면 2000년대 까지는 현대 산업의 모든 분야에서 95% 이상은 컴퓨터를 이용해야만 직무를 수행할 수 있을 것으로 예측하고 있다. 그 보고서는 아

울러 다음 세대의 근로자들을 일생에 적어도 다섯번 이상 직업을 바꾸게 될 것으로 전망하고 있다. 미국 항공 우주국(NASA)에서는 2005년까지 8명 정도의 우주선 승무원이 거주하는 달착륙 정거장을 세울 계획을 하고 있다. 그들은 장기간 달에 머무르며 임무를 수행하게 된다. NASA의 계획에 의하면 장차의 우주선의 임무는 장기화, 복잡화되어 승무원의 교육이 지상에서 100% 이루어질 수 없으며, 우주 여행 중에도 계속되어야 하리라고 한다. 위의 예들은 ITS의 장래를 예측하는데 도움이 될 수 있는 많은 실예들 중의 몇이다. 앞의 장들에서 토의한 바와 같이 ITS는 짧은 기간에 완전하지는 않지만 많은 발전을 이루었으며, 앞으로도 급격한 발전과 상용화를 이루리라 보인다. 미래의 ITS는 크게 무엇을 가르칠 것인가, 어떻게 가르칠 것인가, 그리고 누구를 상대로 가르칠 것인가에 초점이 모아질 것이다 [29]. 첫째 과제에서는 앞의 Domain Knowledge부분에서 논의 한 바와 같이 Declarative Knowledge, Procedural Skills 등과 Metacognitive Knowledge, Reasoning Ability, Problem-Solving Skill과 같은 Higher Order Thinking Skill 등을 어떻게 구현할 수 있을 것인가가 관건이 될 것이다. 어떻게 가르칠 것인가는 인간의 교수 방법에 대한 연구가 모델이 될 것이며, 정보의 적절한 제시, 학습자에 대한 Monitoring, 학습자의 취약점 파악, 교안의 운영 등을 포함한 교수 기법의 개발이 주과제가 될 것이다. 교육의 대상에 대해서는 학습자의 특성을 살리는 방향이며, 학생의 진척도, 개인 성격, 사전 교육 및 지식 상태, 교육 환경, 학습 주제 분야와 학생의 관계 등이 영향을 미치는 요인이 될 것이다. 장래의 ITS는 교수적인 면에서 뿐만 아니라, 컴퓨터 시스템으로서의 일반적인 요구도 무시할 수 없는 사항이다. 즉, 개인뿐 아니라 팀별 교육의 가능, 주제 영역의 풍부한 데이터 제공, 학습자의 시간 스케줄에 대한 유연성, 교육 환경에 따른 이동 및 분리 작동 가능, 교육 수준의 난이도 조정, Multimedia 등 첨단 컴퓨터의 성능 적절한 이용, Upgrade 및 수정 용이, 컴퓨터 프로그램으로서의 신뢰성있는 작동 등도 앞으로의 ITS 상용화에 결정적인 영향을 미칠 것으로 전망된다.

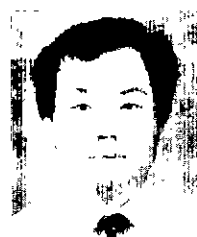
20여년전 IBM mainframe 컴퓨터가 출시될 때 현재와 같은 PC의 이용을 예측한 사람은 없었다. 더구나, 반 세기전 최초의 컴퓨터가 시도될 때, 누가 금세기내에 이러한 정보화 시대를 예측할 수 있었을까. 10년 후, 20년 후의 ITS는 어떠한 모습을 하고 있을지, 인공지능은 어떠한 모습일지, 교육학과 심리학의 발전 및 ITS에의 응용은 어떠한 것인지, 저자 또한 매우 궁금한, 누구도 장담할 수 없는 과제이다.

참고문헌

- [1] Adams, L., Kasserman, J., Yearwood, A., Perfetto, G., Bransford, J. and Franks, J. (in press). The Effects of Facts versus Problem-oriented Acquisition. *Memory and Cognition*
- [2] Alessi, S. and Trollip, S. (1991). *Computer-Based Instruction: Methods and Development*, Prentice Hall, Englewood Cliffs.
- [3] Alpert, D. and Bitzer, D. L. (1970). *Advances in Computer-Based Education*. *Science*, 167, 1582~1590.
- [4] Anderson, J. (1988). The Expert Module. In M. Polson and J. Richardson (Eds.), *Foundation of Intelligent Tutoring Systems*, Lawrence Erlbaum Assoc. Inc., Hillsdale, 21~53.
- [5] Barr, A. and Feigenbaum, E. (1982). *The Handbook of Artificial Intelligence*. William Kauffman, Inc. Los Altos.
- [6] Bransford, J. D., Sherwood, R. D., Hasselbring, T. S., Kinzer, C. K., and Williams, S. M. (1990). *Anchored Instruction: Why we need it and How Technology can Help*. In D. Nix(Ed.). *Cognition, Education, Multimedia*, Lawrence Erlbaum, Hillsdale, 115~141.
- [7] Brown, J. S. (1977). Uses of Artificial Intelligence and Advanced Computer Technology in Education. In R. J. Seidel and M. Rubin(Eds.), *Computers and Communications: Implications for Education*. New York, Academic Press, 253~270.
- [8] Brown, J. S. and VanLehn, K. (1980). Repair Theory: A Generative Theory of Bugs in Procedural Skills. *Cognitive Science*, 4, 379~426.
- [9] Bruno, J. (1987). Admissible Probability Measures in Instructional Management. *Journal of Computer-based Instruction*, Vol. 14(1), 23~30.
- [10] Bunderson, C. V. and Inouye, D. K. (1987). The Evolution of Computer-aided Educational Delivery Systems. In R. M. Gagne(Ed.), *Instructional Technology: Foundations*, Lawrence Erlbaum Assoc., Hillsdale, 283~318.
- [11] Burton, R.B. (1982). DEBUGGY: Diagnosis of Errors in Basic Mathematical Skills. In D.H. Sleeman and J.S. Brown(Eds.), *Intelligent Tutoring Systems*, New York. Academic Press, 157~183.
- [12] Burton, R. and Brown, J. (1982). An Investigation of Computer Coaching for Informal Learning Activities. In D.H. Sleeman and J. Brown (Eds.), *Intelligent Tutoring Systems*, Academic Press, London, 79~98.
- [13] Cabrera, B. (1986). *Mechanics-Physics Simulations I(computer program)*. Stanford University, Palo Altos.
- [14] CERL(Computer-based Education Research Laboratory) (1977). *Demonstration of the PLATO IV Computer-based Education System*. Final report, University of Illinois, Urbana-Champaign.
- [15] Cetron, M. J. (1988). Class of 2000: The Good News and the Bad News. *The Futurist*, Vol. 21(6).
- [16] Clancey, W. (1986). Qualitative Student Models. In Traub, J.(Ed.), *Annual Reviews of Computer Science*, Vol. 1, Annual Reviews, Inc. Palo Altos, 381~450.
- [17] Clancey, W. (1987). *Knowledge-based Tutoring: the GUIDON Program*. MIT Press, Cambridge.
- [18] Collins, A., Brown, J. S. and Newman, S. (1989). Cognitive Apprenticeship: Teaching the Crafts of Reading, Writing, and Mathematics. In L. B. Resnick(Ed.), *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser*, Lawrence Erlbaum Assoc., Hillsdale, 453~494.
- [19] Ferguson, W., Bareiss, R., Birnbaum, L. and Osgood, R. (1992). *The Journal of Learning Sciences*, Vol 2(1), Lawrence Erlbaum Assoc., 95~134.

- [20] Fink, P. K. (1991). The Role of Domain Knowledge in the Design of an Intelligent Tutoring System. In H. Burns, J. W. Parlett, and C. L. Redfield(Eds.) Intelligent Tutoring System, Lawrence Erlbaum Assoc. Hillsdale, 195~224.
- [21] Gagne, R. M., Wager, W. and Rojas, A. (1981). Planning and Authoring Computer-Assisted Instruction Lessons. Educational Technology.
- [22] Geneserth, M. R. (1982). The Role of Plans in Intelligent Tutoring Systems. In D. H. Sleeman and J. S. Brown(Eds.), Intelligent Tutoring Systems, Academic Press, London, 137~155.
- [23] Langley, P., Wogulis, J. and Ohlsson, S. (1987). Diagnostic Monitoring of Skill and Knowledge Acquisition, Lawrence Erlbaum Assoc., Inc., Hillsdale.
- [24] Linn, M. C. (1986). Establishing a Research Base for Science Education: Challenges, Trends, and Recommendations. University of California, Berkley.
- [25] Lunetta, V. N. (1984). School Transactions (computer program). CONDUIT, Iowa city.
- [26] Nix, D. (1990). Should computers know what you can do with them? In D.Nix(ed.), Cognition, Education, Multimedia: Exploring Ideas in High Technology, Lawrence Erlbaum Assoc., 143~162.
- [27] Ohlsson, S. and Langley, P. (1988). Psychological Evaluation of Path Hypothesis. In H. Mandle and A. Lesgold(Eds.), Learning Issues for Intelligent Tutoring Systems. Springer-Verlag, New York, 42~62.
- [28] Osborne, R. and Fryeberg, P. (1985). Learning in Science: The Implications of Children's Science. Heinemann Publishers, New Zealand.
- [29] Redfield, C. L. and Steuck, K. (1991). The Future of Intelligent Tutoring Systems. In H. Burns, J. W. Parlett, and C. L. Redfield(Eds.), Intelligent Tutoring Systems: Evolutions in Design, Lawrence Erlbaum Assoc., Hillsdale, 265~284.
- [30] Reiser, B. J., Anderson, J. R., and Farrell, R. G. (1985). Dynamic Student Modeling in an Intelligent Tutor for LISP Programming. Proceedings of the Ninth International Joint Conference on Artificial Intelligence Conference, Los Angeles, 8~14.
- [31] Shim, L. (1991). Student Modeling for an Intelligent Tutoring Systems: Based on the Human Tutoring Sessions. Doctoral Dissertation, Department of Computer Science, Illinois Institute of Technology, Chicago.
- [32] Sleeman, D. H and Brown, J. S. (1982). Introduction: Intelligent Tutoring Systems. In D. H. Sleeman and J. Brown(Eds.), Intelligent Tutoring Systems Academic Press, New York, 1~12.
- [33] Trollip, S. R. and Brown, G. C. (1989). The Examiner V. 2.1 (computer program). Media Computer Enterprise, Mendota Heights.
- [34] VanLehn, K. (1988). Student Modeling. In M. C. Polson and J. J. Richardson(Eds.), Foundations of Intelligent Tutoring Systems, Hillsdale, Lawrence Erlbaum Assoc. Inc., 55~78.
- [35] Wenger, E. (1987). Artificial Intelligence and Tutoring System. Morgan Kaufmann, Los Altos.
- [36] Williams, S. M. (1992). Putting Cased-based Instruction into Context: Examples from Legal and Medical Education. The Journal of Learning Sciences, Vol 2(4) Lawrence Erlbaum Assoc., Hillsdale, 367~427.
- [37] Woolf, B. P. (1984). Context-dependent Planning in a Machine Tutor. Doctoral dissertation, Department of Computer and Information Science. University of Massachusetts, Amherst.

심 임 섭



1954 서울 출생
 1978 서울대학교 공과대학, 공학사
 1987 Illinois Institute of Technology, 전산학 석사
 1991 Illinois Institute of Technology, 전산학 박사
 1992 한국IBM Software연구소 연구원
 1994 한신대학교 정보통신학과 조교수
 관심 분야: Artificial Intelligence, Expert Systems, Intelligent Tutoring Systems, Cognitive Science, Multimedia Applications
