

□ 기술해설 □

## 지능형 저작 시스템의 현황과 발전 방향

인하대학교 윤경섭\* · 왕창종\*

● 목	차 ●
1. 서 론	3.1 교육용 소프트웨어 개발 도구
2. ITS의 고찰	3.2 IAS의 고찰
2.1 전문가 모듈	3.3 기존 IAS
2.2 학습자 모듈	3.4 IAS의 개발 현황
2.3 학습 전략 모듈	4. 결론 및 IAS 발전 방향
3. IAS의 고찰	

### 1. 서 론

1960년대 초부터 교육 환경에 컴퓨터를 도입하여 학습 내용을 전달하고 평가의 도구로 사용하려는 CAI(Computer Aided Instruction) 프로그램이 본격적으로 연구되기 시작하였다. 이러한 CAI 프로그램을 범용 프로그래밍 언어로서 작성하는 데는 많은 시간 및 개발 경비가 소요되고, 반복적인 요소들이 많이 있다. 따라서 CAI 프로그램 구축을 도와주는 지원 소프트웨어인 저작 도구가 다수 발표되었다. 그러나 CAI 기법에 의해 개발된 프로그램은 교과 과정이 고정되어 있어서 다양한 계층의 학습자의 학습 수준에 맞는 내용 전달이 어렵고, 학습자의 능동적인 학습과 학습자의 반응 누적에 대한 정보의 제공이 불가능하며 신속성의 결여로 인해 학습 방법이 획일적일 수 밖에 없었다. 또한 문제 해결 추론 기능이 없으므로 진정한 의미의 개별화 학습을 기대할 수 없다. 이러한 문제점을 보완하기 위해서 나온 기법이 CAI에 인공 지능의 원리와 기법을 적용해서 학습자에게 더욱 강력하고 정확한, 적응력 있는 교육 시스템을 개발하려고 시도한 것이 지능형 교육 시스템(ITS: Intelligent Tutoring Sys-

tem)이다[12,18]. 지난 10여 년간 수십 종류의 ITS가 개발되었으나[8,12,24]. 이러한 ITS의 폭 넓은 사용에 있어서 두 가지 큰 장애 요소는 개발 시간이 많이 걸린다는 것과 ITS 개발 전문가가 부족하다는 것이다[3,4]. 현재는 ITS 프로그램이 LISP이나 Prolog와 같은 인공지능 언어로 작성되고 있다. 따라서 하나의 ITS를 만들기 위해서는 개발자가 주제 영역에 관한 지식 외에도 인공 지능 방법론과 프로그래밍 기술의 지식이 필요하다. ITS 프로그램의 이용도를 증가시키기 위해서는 개발 시간과 인공 지능 지식에 대한 의존도를 실질적으로 감소시켜야 한다. 이것은 ITS 프로그램 생성기인 지능형 저작 시스템(IAS: Intelligent Authoring System)을 개발해야만 달성될 수 있다. 기존의 교육용 프로그램을 생성하기 위한 저작 시스템의 상당수는 CAI를 위해서 개발되었다. 이와 비슷하게 많은 프로그램 생성기가 전문가 시스템을 생성하기 위해 개발되어 왔다. 따라서 ITS를 위한 저작 도구가 개발되지 말아야 할 이유가 없다. 그러나 ITS의 주제 영역에 따라 최적의 지식 표현 방법과 추론 방법이 서로 다르므로 공통성을 찾기가 힘들며, 따라서 지능형 저작 시스템을 위한 공신력 있는 패러다임이 없는 상태이다.

IAS가 ITS를 생성해 내는 형태로는 크게 두

\*종신회원

가지로 나누어 볼 수 있다. 첫째로 결과물이 프로그램 언어 형태로 생성되는 것이 있다. 대표적인 것으로 Begg와 Hogg의 BURROUGHS ICON 시스템[3,4]으로서 저작자가 데이터베이스와 레슨 저작 과정을 저작하면 C, LISP, PROLOG 등의 언어로 생성된다. 따라서 결과를 쉽게 수정할 수 있으며 유연성이 있다는 장점이 있으나 저작자가 인공 지능 지식과 프로그래밍 언어를 알고 있어야 한다는 단점을 가지고 있다. 두 번째로는 셸의 형태로써 저작자가 지식 습득기를 통해 지식을 구축하여 시스템을 생성해 내는 것이다. 이러한 지식으로는 가르칠 영역 지식뿐만 아니라 학습자의 학습 성취도 측정 및 예측에 관한 지식도 포함되어야 하므로 기존의 전문가 시스템 셸(Expert System Shell)의 이용은 불가능하다. 셸 형태의 시스템으로는 Bonor의 Bite Tutor[5], intelligent tutor에 대한 지식획득 도구를 설계한 Murray와 Woolf[18], 그리고 영역 독립 환경을 갖는 ITS Challenger를 제안한 Augusteijn[2] 등이 있다. 그러나 이들 연구의 문제점은 학습자의 학습 성취도의 세밀한 분석과 학습 능력을 예측하여 알맞은 교수를 할 수 없다는 데 있다.

## 2. ITS의 고찰

ITS는 자신이 가지고 있는 지식을 근거로 하여 학습자를 적응력 있게 가르칠 수 있는 것이다. ITS가 효율적인 교사가 되기 위하여 다음과 같은 지식을 가져야 한다. 첫째는 가르치는 주제에 관한 지식, 둘째는 학습자에 관한 지식, 셋째는 교수법에 관한 지식, 그리고 넷째는 인터페이스에 관한 지식 등이다[8,18]. 그림 1은 ITS의 일반적인 구성을 나타낸 것이다.

이러한 지식은 ITS가 학습자에게 가장 효율적인 교수법으로 교육 내용에 관한 결정을 할 수 있도록 하여야 한다. 위의 네 가지 지식 형태를 실제로 구현하고 처리하기 위한 모듈 형태를 전문가 모듈, 학습자 모듈, 교수 모듈, 인터페이스 모듈이라 한다.

### 2.1 전문가 모듈

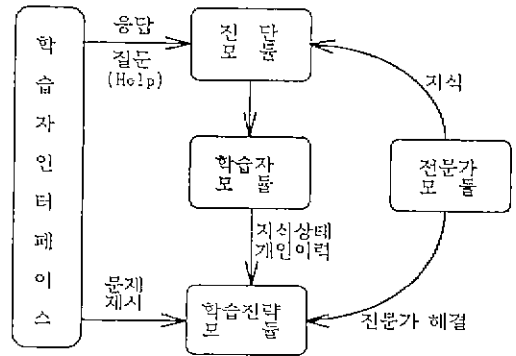


그림 1 ITS의 일반적 구성

ITS의 전문가 모듈은 학습자에게 가르치고자 하는 주제에 대한 개념 및 설명에 대한 지식으로 구성된다. 또한 기존의 지식으로부터 새로운 지식을 유추하여 학습하도록 하는 기능이 포함되어야 한다.

전문가 모듈은 학습자에게 가르치고자 하는 주제 영역에 관한 내용으로 주로 학습 내용에 관한 세부적 지식으로 구성된다. 학습 내용에 관한 세부적 지식에 속하는 것들은 학습해야 할 주요 개념, 학습해야 할 개념들 간의 상호 관련성, 절차적 지식 혹은 규칙, 개념들의 사용 방법 또는 적용과정, 전문가 경험을 통하여 획득한 공리나 탐구된 지식을 들 수 있다. 학습 내용에 관한 지식을 표현하는 방법은 다양하다. 전문가 모듈의 지식 형태는 그 방법에 따라 세 가지로 나눌 수 있다. 첫째로 블랙 박스 모델은 인간의 실제적 사고 과정 방법을 필요로 하지 않거나 어려운 영역에 대하여 수학적인 묘사를 하는 방법이다. 둘째로 글래스 박스 전문가 시스템 모델은 지식 공학적인 방법에 따라 만들어진 전문가 시스템과 같은 방식의 전문가 모듈을 사용한 시스템이다. 효과적인 ITS를 만들기 위해서는 전문가 모듈의 지식뿐만 아니라 그것을 사용하는 방법에도 주의 기울여야 한다는 점을 이해할 수 있다. 이러한 이유 때문에 인지적 접근 방법이 요구되는 것이다. 인지적 모델은 인간과 유사하게 의미 있는 지식 단위와 처리 방법으로 인간의 문제 풀이 과정을 시뮬레이션하는 것이다. 인지 모델에서는 학습 내용에 따라 세 가지 종류의 형태로 구별한다. 절차적 지식은 전문가 시스템의 영향

으로 대부분 규칙 기반 시스템으로 구성된다. 규칙 기반 시스템은 생성 규칙의 형태를 띠고 있으며, 인간과 같은 문제 풀이 방식을 묘사하는 좋은 모델이라고 알려져 있다.

IAS를 위한 전문가 모듈이 여러 영역 지식에 대한 ITS를 생성하기 위해서는 영역 지식의 표현 방법이 표현하고자 하는 지식의 종류에 영향을 받지 않아야 한다. 일반적으로 모든 지식을 효율적으로 표현할 수 있는 방법은 불가능하지만 많은 공통적인 부분을 표현하기 위해서 현재 가장 연구가 활발한 것으로는 객체 지향 지식 표현 방법론이 있다.

2.2 학습자 모듈

학습자 모듈은 학습자의 현재 지식 상태를 나타내는 학습자 모델과 학습자 모델을 추론하는 진단 과정으로 구성된다. 현재까지 연구된 학습자 모델링 기법은 크게 오버레이 모델, 차분 모델, 버기 모델, 버그 부분 목록 등으로 구분된다. 오버레이 모델은 학습자 모델링 기법 중 가장 많이 사용되는 것으로서, 학습자의 지식이 전문가 지식의 부분 집합으로서 표현되는 것이다. 따라서 전문가와 상이한 학습자의 지식이나 믿음을 다루는 항목을 가지고 있지 않다. 만일 학습자가 전문가가 가지고 있지 않은 항목에 대하여 더 잘 할 수 있는 다른 전략을 가지고 있다면 오버레이 모델은 그것을 인식하지 못할 것이고 교사가 학습자에게 교정시키도록 할 것이다. 그러나 오버레이 모델은 구현이 용이하여 표 1과같이 지금까지 연구되어 온 ITS 시스템의 학습자 모델링 기법으로서 가장 많이 사용되고 있다.

차분 모델은 동일한 상황하에서 전문가 행동과 학습자 행동의 차이에 의해 생성되어 진다. 차분 모델링은 학습자가 모르는 것을 학습자가 알아야 하는 것과 학습자가 알 것이라고 기대하지 않는 것의 부류로 나눈다. 차분 모델은 학습자 지식에 대하여 오버레이가 가지고 있는 가정의 문제는 피하지만 아직도 오버레이 모델의 문제점을 대부분 가지고 있다. 전문가 지식 영역의 밖에 있는 어떠한 학습자의 믿음도 표현할 수 없다.

버기 모델은 전문가 모델 영역 밖에 있는 학

표 1 기존의 ITS

시스템	교과 내용	학습자 모델	년도	개발자
SCHOLAR	남미의 지리학	오버레이	1970	CarlLancil
WHY	강우의 원인	오류개념확인	1982	Stevens
INTEGRATE	부정적분	오버레이	1982	Kimball
SOPHIE	전자회로의 고장	오버레이	1982	Brown
WEST	산수와 식	오버레이	1979	Burton, Brown
BUGGY	멜셀	오류개념확인	1978	Burton, Burton
MUSOR	논리의 관계	오버레이	1982	Goldstein
EXCHECK	논리 집합론	오버레이	1981	Suppes
BIP	BASIC 프로그래밍	오버레이	1976	Barr
SPADE	LOGO 프로그래밍	오버레이	1983	Miller
ALGEBRA	대수의 응용	오버레이	1983	Lantig
LMS	대수의 질차	규칙기반	1982	Sleeman
QUADRATIC	이차방정식	오버레이	1982	O'shady
GUIDON	컨셉맵	오버레이	1982	Clancey
PROLST	PASCAL 프로그래밍	오류개념확인	1983	Soleway
STRAWLER	배의 추진력	전자적 회로형	1981	Williams

표 2 학습자 모델과 진단 기법

대역폭	지식형태	평면질차식	계층질차식	선연적
사고상태			모델추적 Calculus ISP tutor Geometry Tutor	GUIDON
중간상태	주제추적 WEST MUSOR		제척인식 MACSYMA Advisor Spade Image	전문가 시스템 SCHOLAR WHY GUIDON
최종상태	경로경향 조건유도 LMS Pixio, ACM	컨셉맵 상호 진단 BUGGY	생성과 검사 DEBILGGY IDEELBGGY	생성과 검사 MEND PROLST

습자 믿음을 표현하기 위하여 긴밀한 연결을 유지하면서 학습자와 전문가 모델 사이에서 사용될 수 있다. 버기 모델은 전문가 지식 영역 외에 학습자가 잘못 이해한 지식의 집합을 버그라고 가정하고, 학습자의 지식을 진단할 때 모든 버그의 목록을 이용하는 것이다. 버그는 주로 교육학 분야의 연구에서 얻어지며, 학습자의 행동을 분석한 실험적 데이터를 분석하여 알아낼 수 있다. 또한 교육 내용에 대한 학습 이론이 존재하면 학습자가 행할 수 있는 버그들을 예측할 수 있다. 그러나 버그 라이브러리가 완벽하지 못할 경우 학습자가 잘못 이해한 개념(misconception)을 전혀 엉뚱하게 진단할 수 있다. 버그 부분 목록은 버그의 일부 목록만을 작성하여 이 목록으로부터 버그들을 유추하는 것이다[12,21].

학습자 모델의 진단 기법은 대역폭과 지식 형태에 따라 표 2와 같이 9가지로 나누어볼 수 있다.

학습자 모듈은 특정 학습 과정에 관한 학습자의 성향과 교수 영역에 관련된 학습자의 능력을 평가하기 위해서 여러 가지 용도로 사용되는 데 특히 진도의 결정, 조언의 제공, 문제의 생성,

설명의 채택 등에서 중요한 요인이 된다.

### 2.3 학습 전략 모듈

학습 전략 모듈은 학습 전략의 제어 방식에 따라 저작자 제어 방식, 내용 분석 제어 방식, 정보 분석 제어 방식으로 나눌 수 있다. 저작자 제어 방식은 저작자가 모든 학습 전략을 미리 설정하여 놓는 방식으로 주로 선형 프로그램 및 분기 프로그램에서 사용한다. 이러한 방식에서 시스템은 학습 내용을 단지 문자의 집합으로만 인식하므로 학습 내용에 대한 아무런 정보도 알 수 없으며 따라서 시스템이 자체적으로 학습 전략을 제어할 수 없다. 즉 저작자가 학습 전략을 정해 놓으면 프로그램 실행시에는 어떠한 경우에도 변경될 수 없는 정적 학습 전략만을 구축할 수 있다. 대표적인 시스템으로는 COURSE BUILDER, WISE, COURSE OF ACTION 등이 있다 [25,27].

내용 분석 제어 방식은 학습 내용을 구조적으로 분석한 다음에 가장 적절한 지식 표현 방법으로 지식 베이스에 내용을 입력한다. 다음 각 문제에 대한 학습자의 응답에 대하여 일반해 알고리즘(general solution algorithm)을 통하여 학습자의 오류에 대해 체계적인 설명을 하고, 학습자에게 알맞은 학습 전략을 구축하여 그 학습 전략에 따라 교육을 실시하는 방식이다. 이 방식은 학습자의 응답과 학습자의 지식 수준에 따라 다양하게 학습 전략이 변하는 동적 학습 전략을 구축할 수 있다. 그러나 내용 분석 제어 방식은 일반해 알고리즘의 개발과 학습 내용의 구조적 분석이 어렵고 지식 베이스에 저장할 수 있는 지식의 범위가 넓지 못하다는 단점이 있다. 현재 이 방식을 채택하고 있는 시스템으로는 tb-**t**EXPERT, MicroSEARCH, MicroProust 등이 있으며 전문가 시스템 및 지식 베이스와 연계되어 연구되고 있다[6,8].

정보 분석 제어 방식은 학습 전략의 구축에 필요한 정보를 학습 내용과 별도로 저장함으로써 시스템이 학습 내용을 저작자 직접 제어 방식처럼 문자의 집합으로만 인식한다. 학습 전략을 구축할 때는 내용 분석 제어 방식처럼 학습

내용의 정보를 접근하여 동적 학습 전략을 구축한 방식으로 SMITH, NP-TOAS 등의 시스템에서 채택하고 있다[25,27]. 이 방식은 저작자 제어 방식의 장점인 저작의 용이성과 저작자의 전산학에 대한 의존도를 배제하면서 동적 학습 전략 능력을 갖는다는 점에서 의미가 있다. 그러나 학습 내용에 대한 정보를 추가로 입력해야 하고, 시스템이 학습 내용에 대한 직접적 지식을 갖는 것이 불가능하기 때문에 동적 교수 능력이 제한되며, 전산학 전문가가 포함된 저작자가 양질의 ITS를 개발하려 할 때, 그 능력을 제한 시킨다는 단점이 있다.

## 3. IAS의 고찰

2장에서 살펴본 ITS의 여러 가지 구성 모듈들은 모두 IAS를 위해서도 필요한 모듈들이다. 특히, 기존의 ITS들은 한 가지 주제 영역을 목표로 하여 개발되어 그 시스템에 완전히 종속적으로 개발될 수 밖에 없었으나, IAS는 개발을 원하는 주제 영역을 미리 알 수 없으므로 가능한 모든 주제 영역에 대하여 저작이 가능한 일반적이고도 영역 독립적인 모듈이 되어야 한다. 이 장에서는 일반적인 교육용 소프트웨어 개발 도구에 관해 알아보고, IAS가 가져야 할 기능과 기존에 연구되었던 IAS에 대하여 고찰한다. 또한 현재 본 연구실에서 개발중인 IAS의 간단한 구조에 대해서도 고찰해 보도록 한다.

### 3.1 교육용 소프트웨어 개발 도구

본 절에서는 교육용 S/W 개발을 위한 도구를 4가지 범주로 분류하여 분석하고 장단점을 비교 분석한다. 4가지 범주는 전통적 프로그래밍 언어, 저작 언어, 저작 시스템, 저작 도움기(authoring aids) 등이다[10].

과거에 교사가 프로그램을 개발하는데 사용한 가장 일반적인 접근 방법은 약 200개의 가용한 프로그래밍 언어중 하나를 이용하는 것이었다. 프로그래밍 언어에서 저작의 모든 측면은 언어를 이용해 저작자의 직접적이고 완전한 제어를 하여야 한다는 것이다. 이러한 것은 저작자에게

가장 유연하고 강력하게 저작할 수 있게 해주나, 프로그래밍을 하는 일이 매우 복잡하고 성공적으로 완성하려면 많은 시간이 걸린다.

저작 언어는 특수한 목적의 컴퓨터 언어로서 교육용 S/W를 개발하는 도구로 설계되어 진다. 저작 언어 개발의 일반적 의도는 교육용 소프트웨어 개발을 쉽게 하자는 것이지, 개별적 명령어를 보다 단순하게 하자는 것은 아니다. 저작과 전통적 언어 사이의 또다른 차이점은 저작 명령어가 일반적으로 덜 추상적이고 영어와 유사하다는 것이다. 또한 저작 명령어는 대화적인 프로그램 작성을 단순하게 하도록 설계되어 있다. 그러나 저작 도구는 보다 단순해야 하고 전통적 프로그래밍 언어보다 빠르게 학습할 수 있어야 한다. 왜냐하면 저작 도구는 교육을 위해 특별히 제작된 것이기 때문이며, 상호 대화적으로 프로그램을 작성한다는 아이디어에서 출발했기 때문이다. 덧붙여, 그래픽이나 사운드를 위한 함축된 명령어와 학습자의 결과 레코드를 유지하기 위한 여러 규정을 포함하고 있다. 그러나 이것의 단점은 유연성과 강력함의 감소이다. 즉, 저작 언어는 단순하게 되어 있기 때문에, 전통적 언어에서 제시된 중요한 능력을 상실하게 된다. 또한 일부 저작 언어는 너무 많은 특징을 덧붙임으로서 전통적인 범용 프로그래밍 언어보다 배우기가 용이하지 않다. 저작 언어의 특수한 형태로 하이퍼텍스트 저작 언어가 있는데 이는 하이퍼텍스트 매체(하이퍼미디어)를 생산하는데 이용된다. 대표적인 것으로 매킨토시사의 하이퍼카드가 있으며, 이것은 하이퍼텍스트를 이용해 교육용 레슨을 작성할 수 있는 프로그램이다. 일부 연구자들은 이러한 것을 저작 언어 부류로 나누는 것을 반대하기도 한다[10]. 즉, 하이퍼텍스트 저작 언어를 비선형적인 데이터베이스 관리자라고 간주하고 있으며, 여기에 소프트웨어 저작 도구가 고려되어야 한다는 것을 덧붙이고 있다.

저작 시스템은 프로그래밍 명령어를 사용하지 않고 교육용 소프트웨어를 작성할 수 있도록 설계된 시스템이다. 교사가 레슨 프레임 저작할 때 단지 제시할 정보나 질문할 문제, 제공될 되돌림 등을 이어주는 형식이다. 저작 시스템은 저작 언어보다 더 구조적이다. 즉 저작 시스템은

필수적으로 레슨을 생성하여야 한다. 사용자는 컴퓨터 프롬프트에 대한 응답으로 제시될 자료 안을 채운다. 여기에 프로그래밍 명령어는 사용되지 않는다. 저작 시스템은 유연성이 부족하다. 즉, 제시 순서나 방법이 미리 정해짐으로써 다양한 교수 전략을 세울 수가 없다. 단지 몇몇 형태만이 가능한데, 다중선택 문제, 질문에 이은 텍스트의 블록, 게임 형태 등이다. 따라서, 저작 시스템은 저작자의 선택을 코드로 변환하는데 반해 전통적 언어나 저작 언어를 사용하는 사용자는 개인적으로 코드를 실제 생성해야 한다. 이러한 저작 시스템의 주요 장점은 단순성에 있다. 따라서 교육용 소프트웨어가 전통적인 컴퓨터 언어보다 저작 시스템을 이용하는 것이 10~15배까지 생산성이 향상된다는 발표가 있다[10]. 또한 언어를 배울 필요가 없으므로, 저작자는 레슨 개발 시간을 줄일 수 있다. 다른 장점으로 일부 저작 시스템에서 제공하는 최신의 특징들을 들 수 있는데, 이것은 대부분의 프로그래머가 전통적 언어로 구현하기 힘든 것들이다. 그런 특징은 answer-matching과 recordkeeping 기능이며, 일부 저작 시스템은 다른 하드웨어와 연결이 가능하다. 즉, 비디오텍 플레이어, 스케너의 다른 장비로서, 전통적 언어로 작성된 레슨에 포함시키기 힘든 것들이다. 저작 시스템은 비지능적 시스템과 지능적 시스템으로 구분할 수 있는데 비지능적 저작 시스템을 사용할 경우 성능은 보편적이나 생산성과 사용의 용이성은 대단히 높다고 할 수 있으며, 지능적 저작 시스템은 성능, 사용의 용이성, 생산성에서 가장 우수하다고 할 수 있다. 저작 시스템의 단점은 유연성의 부족이다. 왜냐하면 저작하는 과정을 단순하게 만들었기 때문에 가용한 학습전략의 수가 줄어든다. 이것은 저작자가 시스템내에 있는 교수 스타일이나 형태를 따라야 하기 때문이다. 따라서, 저작 시스템으로 작성된 모든 레슨은 정적 학습인 경향이 있다.

저작 도움기는 코드 모듈, 코드 생성 프로그램, 편집가능 소프트웨어등을 총칭한다. 처음 두 가지는 프로그램 작성을 돕는 프로그래머 도구라고 보는 것이 타당하다. 이것은 전통적 언어로 코스를 만들어 낼 때 시간을 절약하기 위해 설계된

것이다. 코드 모듈은 보다 대중적이고 널리 이용되고 있다. 이러한 유형의 저작 도움기는 다른 두 가지보다 더 많이 진보된 것일 것이다. 그들이 원하는 것을 찾을 수 있는 프로그래머는 그것을 이용할 것이다. 코드 생성 프로그램은 비교적 새로운 것이다. 이것은 유연성이 불가피한 손해를 감수할 정도로 강력하다는 것이 증명되고 있다. 편집가능 소프트웨어는 적어도 지금까지 논의된 모든 저작 도구 이상의 강력함과 유연성을 가지고 있다. 이런 소프트웨어는 보통 설계자에게 맡겨지는 일부 결정을 교육자가 할 수 있도록 한다. 그런 선택은 난이도, 속도, 피드백 내용 및 스타일의 선택, 분기, 결정 등에 관련한다.

### 3.2 IAS의 고찰

본 절에서는 지능형 저작 시스템이 갖추어야 할 일반적인 기능과 구성 요소들에 대해서 살펴보고, ITS를 구축하기 위한 기반으로 사용되었던 기존의 IAS에 대해 고찰한다.

IAS가 CAI의 개발에 사용된 저작 시스템과 다른 점은 크게 네 가지로 볼 수 있다. 첫째로 저작자는 구체적인 교수 순서가 아닌 지식망과 추론 규칙을 정의한다는 것이다. 둘째로 학습자 수행 모델은 동적으로 유지되고 교수 순서를 유도하기 위해서 사용된다. 셋째로 학습자들은 복잡한 질문이나 동적으로 생성된 문제들을 요구할 수 있다. 넷째로 교사는 코치의 특성이 학습자 오류에 대한 자세한 진단을 제공한다. 이러한 차이점이 내포하는 의미는 기존의 저작 도구는 내용과 논리가 서로 얽혀 있는 코스웨어를 사람이 만들도록 했으며 제어의 흐름은 미리 제시한 분기에 의해서 결정된다. 또한 교수 규칙과 오류 진단 규칙은 이 분기 교수에 포함되고 외적으로 나타날 수 없으며, 카운터나 버퍼로 하지 않고는 외적으로 보이는 학습자 모델을 만드는 방법이 없다. 그래서 IAS는 저작자에게 ITS의 요소를 접근할 수 있도록 해주어야 하고 다음과 같은 기능이 요구된다[19].

-사실과 개념 또는 절차와 기능으로 되어 있는 지식망 작성 및 수정- 지식망 내에 있는 정보에 연결되어 있는 에러 진단 규칙과

- 교수 규칙을 규명하고 수정
- 사용자 입력의 프로그램 출력 및 화면 디스플레이 형태를 규명하고 수정
- 지식망, 에러 진단 규칙, 교수 규칙에서의 불일치 불완전성을 식별

따라서, IAS를 설계하기 위해서는 다음과 같은 요건을 만족하여야 한다. 첫째, 임의의 영역 지식을 구축할 수 있는 구조적인 지식 표현 방법을 제공해야 한다. 둘째, 저작자로 하여금 영역 지식을 손쉽게 입력할 수 있는 효율적인 저작 인터페이스를 제공해야 한다. 셋째, 학습자의 지식 상태를 파악하고 오류에 대한 진단을 수행하며 이를 통해 학습자에게 적절한 새로운 교수 전략을 수립하는 동적인 학습자 및 학습 전략 모듈이 내재되어야 한다.

### 3.3 기존의 IAS

#### 3.3.1 BURROUGHS ICON

BURROUGHS ICON[3]은 프로그래밍 언어의 형태로 ITS를 생성하는 IAS 시스템으로 Begg와 Hogg가 개발하였다. 여기에서 개발된 시스템을 가지고 불어를 교육하는 ITS를 시범적으로 구축하였으며, 이 연구를 통해 IAS 개발이 가능하고 충분한 효율성이 있음이 입증되었다.

그림 2와 같이 이 시스템은 크게 사용자 인터페이스, 데이터베이스 편집기, 레슨 어셈블러, 코드 생성기로 구성되어 있으며 다음과 같은 구조로 동작하였다. 사용자 인터페이스는 코스웨어 저작자나 개발자가 주로 사용하며 세 가지의 주된 기능을 수행한다. 첫째로 사용자에게 시스템의 구성 요소에 규칙적인 접근을 제공한다. 둘째는 코스웨어 구성에 있어서 규칙적인 접근을 제공한다. 셋째는 ITS에 의해 다루어질 수 있는

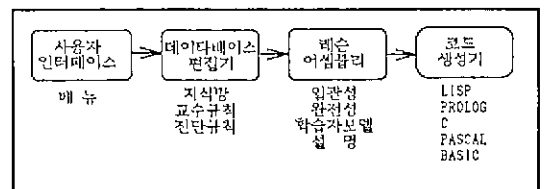


그림 2 BURROUGHS ICON의 구조

형태로 사용자가 문제나 교수 영역을 개념화하도록 하는 것이다. 이러한 기능을 완벽하게 갖춘 IAS가 개발된다면 현재 인공 지능에 전문가만이 개발할 수 있는 ITS가 훨씬 많은 사람들이 보다 다양한 영역의 주제를 가지고 구축할 수 있을 것이다. 사용자 인터페이스를 제공하는 세가지 주요 방법으로는 시스템 조사, 스토리보드, 메뉴 등이 있다. 이 중 가장 간단한 형태는 분석되고 있는 모든 교수 절차에 대하여 같은 질문을 준비하는 메뉴나 고정 프롭프팅 구조이다. 시스템이 처음 부팅되었을 때의 구조는 메뉴 형태가 되고, 최종 코스웨어는 여러 개의 레슨으로 구성되어질 수 있다. 이들 각 레슨은 대화식 에디터를 통하여 관계가 이루어 진다. 대화식 에디터는 아이콘 방식을 도입하며, 링크의 작동, 연결 규명, 이름 규명, 복사, 복구, 삭제 등의 의미있는 아이콘을 준비하고 있다.

데이터베이스 편집기를 통해 지식망, 교수 규칙, 진단 규칙의 세 가지 데이터베이스를 구축하게 된다. 편집기의 기능은 ADD, MODIFY, REMOVE, REPLACE, DISPLAY, FIND 등이며 융통성과 사용성을 극대화하기 위해 내부의 명령어 인터페이스나 메뉴 형태로 입력하게 함으로써 저작자는 정확한 문법을 알 필요가 없다. 데이터베이스는 11가지 종류의 항목으로 구성되는데 지식망은 주제 개념 사전에, 애러 규칙은 응답에, 교수 규칙은 질문 화면에, 학습자 모델은 오버레이 모델 출구 입구에, 인터페이스는 상태에 대응된다. 레슨 어셈블러 기능은 여러 가지 데이터베이스에 저장된 요소들을 조립하여 하나의 프로그램을 만드는 것이다. 이것은 학습자 모델과 필요한 추론 기법들을 포함한다. 또한 데이터베이스의 완전성과 일관성을 검사하기 위한 능력을 포함한다. 코드 생성기는 어셈블러의 부분으로 분류되는데, 조립된 수업으로부터 실행 가능한 코드를 생성하는 기능을 갖는다. 코드 생성기는 특정 출력 언어를 위하여 쓰여진 번역 규칙으로 되어있다. 이러한 규칙은 조립된 수업의 내적 표현을 코드로 변환한다. 이론상 번역 규칙은 여러 언어에 대해 쓰여질 수 있고, 저작자는 그의 ITS를 위해 요구되는 출력 언어를 규명할 수 있다.

BURROUGHS ICON은 지식망 데이터베이스를 구축하는데 필요한 명령어를 메뉴로 구성하여 저작자의 구문 오류를 줄이려는 시도를 하였다. 그러나 이러한 방식은 명령어 인터페이스의 한계를 극복하지 못하였다.

### 3.3.2 TEACHER'S APPRENTICE

TEACHER'S APPRENTICE[1]는 Anderson과 그의 동료들이 만든 고등학교 수학 대수를 주제 영역으로 하는 IAS이다. 이 시스템은 인간 교사가 갖는 많은 속성을 갖는 시스템을 만들어 내는 것이 목적이다. 이 시스템의 학습 이론 근거는 Anderson의 ACT 모델이며, 학습 이론에 따른 ITS의 8가지 설계 원칙을 제시하였다.

- 생성 규칙의 집합으로 학습자를 표현한다.
- 문제 해결에 기초가 되는 목표 구조를 전달한다.
- 문제 해결 문맥 안에 교육 내용을 제공한다.
- 문제 해결 지식의 추상적 이해를 증진한다.
- 작업 메모리를 최소화한다.
- 오류의 즉각적인 피드백을 한다.
- 학습의 단위를 조정한다.
- 목표 기술로의 연속적인 접근을 돕는다.

이러한 원칙을 수용할 수 있는 구성 요소는 버기 모델로 구성되는 학습자 모듈, 인터페이스, 교수 요소, 학습 교육 규칙을 위한 규칙 연역 요소 등이 있다. 학습자의 성적을 시뮬레이션하는 것은 교육하는데 모델 추적 방법을 사용하기 위해 필요하다. 학습자의 문제 해결 행위는 이상적 모델의 행위와 비교된다. 각 시점에서 학습자에게 제시되는 자료는 학습자의 내부 상태를 해석하는 것에 따른다. 이것은 학습자의 해결책을 버기 규칙에 의해 생성된 문제 상태에 매칭시킴으로서 추론된다. 교수 전략으로는 두 가지가 있는데, 하나는 질문과 대담으로 안내하는데 키보드 없이 마우스로서 클릭하게 하는 방법으로 오류의 종류가 제약적으로 발생한다. 또 다른 방법으로 학습자가 방정식을 직접 입력하게 하는 것이다. 교수 요소의 제어는 교수 규칙 제어기라 부르는 생성 시스템과 유사한 것이 있으며 이것의 기본 단위는 t-규칙이라 한다. 이것으로 시스템의 정확한 행위가 결정된다.

### 3.3.3 ITS CHALLENGER

셸 형태의 IAS로 Augusteijn이 제안한 ITS Challenger[2]가 있으며, 이는 영역 독립적인 ITS 셸이다. ITS Challenger는 그래픽 유저 인터페이스를 도입하여 저작자로 하여금 영역 지식과 문제 종류등을 쉽게 입력할 수 있도록 하였다. 또한 지식의 표현 기법도 객체 지향 지식 표현 기법을 사용하여 개발 과정에서의 일관성과 중복을 최소화하였다. 그러나 이 연구의 문제점은 학습자의 학습 성취도의 세밀한 분석을 위한 학습자 모듈이 저작자의 의도를 반영하여 시스템의 규칙을 변경할 수 있는 기능이 없어 완전한 IAS라고 보기는 어렵다.

### 3.3.4 COCA

COCA(Cooperative Classroom Assistant)[11]는 학습시킬 영역과 학습될 주제를 저작하기 위해 영국의 Nottingham 대학에서 개발된 ITS를 위한 셸로서, IBM호환 PC상에서 LPA Prolog로 구현되었다. COCA는 크게 학습될 주제의 표현을 위한 영역 계층(domain layer)와 학습전략을 표현하기 위한 전략 계층(strategic layer), 언제 어떠한 학습 전략을 사용할 것인지를 결정하기 위한 메타전략제어(meta-strategic control)로 구성된다. COCA는 고정된 수행 과정(cycle)을 따르고, 이 과정동안 여러 지점에서 메타 레벨을 호출한다. 이러한 지점들은 교사가 학습 과정을 체계화 시키기 위하여 만들 수 있는 결정들과 일치한다. 예로, COCA의 수행 과정이 인터럽트 되는 지점들 중 하나는 학생과의 다음 상호작용이 학습인지, 평가인지의 결정에 관여한다.

수행 과정이 이러한 지점들중 하나에 도달할 때, 제어는 특별한 결정을 내리고 이러한 결정을 객체 레벨에 되돌려 줄 권한을 가진 메타 레벨로 넘겨지게 된다. 객체 레벨은 메타 레벨에 의해 지지를 받음으로서 다시 재개되는데, 이러한 결정들을 내리기 위해 많은 경험적 데이터를 가지고 있다. 그러나 사용자들은 이것들을 재정의하여 학습전략을 변경할 수 있다. 사용자들은 주어진 많은 프리미티브들을 이용한 생성 규칙의 형태로 이러한 경험적 결정과 관련있는 지식들을 표현할 수 있으며, 따라서 하위 레벨의 프로그

래밍 언어에 의존하지 않고 사용된 전략적 지식을 변경할 수 있다.

COCA는 제어의 전략 레벨외에 또다른 제어 형태인 메타 전략 제어를 제공한다. 즉, COCA는 사용자가 시스템의 실행동안 전략 레벨에서 사용된 지식을 변경할 수 있게 하여, 학습동안에 사용된 학습 전략을 변경할 수 있도록 한다. 이러한 부가적인 유연성은 매번의 수행 과정이 끝난 후에 메타 전략 레벨에 제어를 넘겨 줌으로서 얻어질 수 있다. 메타 전략 레벨에서 시스템은 다음 과정동안 사용될 전략적 경험들을 변경시킬 것인지를 결정한다.

이러한 COCA는 다양한 분야에서 ITS를 개발하기 위해서 사용되었는데, 그 예로, 간단한 선형방정식, 과학에서 물의 상태, 지리학에서 물의 순환, 사학에서 미국 혁명등을 학습시키기 위한 ITS들이 있다.

COCA의 문제점은 4가지로 요약할 수 있다. 첫째, 프레임 기반 언어의 사용이 충분하지 못하다는 것이다. 둘째, 영역 지식의 다중 표현이 부족하다. 셋째, 다른 ITS와 비교해 볼때 상대적으로 학습자 모듈이 단순하다는 것이다. 넷째, COCA가 가지고 있는 교수 전략들이 다양하지만 저작자들이 학습 전략을 직접 저작할 수 있는 기능들이 포함되어야 할 것이다.

## 3.4 IAS의 개발 현황

본 연구실에서는 기존에 연구하였던 프로그래밍 언어 교육용 소프트웨어의 개발에 대한 경험을 바탕으로 하여 현재 IAS를 위한 각 모듈들의 설계를 완료하였고, 전체 시스템 개발을 설계 및 구현 중이다. 본 절에서는 현재 본 연구실에서 현재 개발중인 IAS에 대하여 고찰한다.

### 3.4.1 시스템의 구성

그림 3은 IAS의 전체 구성도를 나타낸 것이다 [23]. 선분은 데이터와 제어의 흐름을 나타내며, 번호는 그 순서를 나타낸다. 각 모듈 간의 정보 교환은 정보 인터페이스를 거친다.

### 3.4.2 지식망의 설계



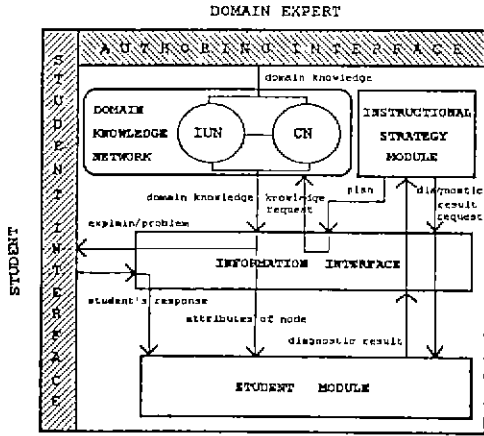


그림 3 IAS의 구성도

본 시스템의 영역 지식 표현을 위한 지식망은 크게 교수 단위망과 개념망의 두가지로 구성하였으며[20], 이들 지식 구조는 각각 클래스의 형태로 처리된다. 지식망을 객체지향 패러다임을 통해 구축함으로써 생기는 장점은 도메인 모델과 학습자 모델 그리고 학습 전략 모델을 기능별로 분리하는데 있어서 자연스러운 방법이 제시된다는 점이다. 또한 객체 지향 패러다임은 영역 모델을 지식 구조로 표현하기에 편리하다. 개념망과 교수 단위망은 각각 영역 모델 클래스의 하위 클래스로 이루어 진다. 영역모델 클래스는 영역 지식 모델의 최상위 클래스에 위치하면서 개념망과 교수 단위망을 하위 클래스로 가진다. 영역모델 클래스는 인스턴스인 그래픽 요소를 포함하고 있으며 이는 학습 내용에 대한 소개로서 초기에 학습자에게 제시된다. 또다른 인스턴스로서 개념 사전에 포함하고 있는데 이는 저작자의 개념과 실제 개념망에서의 표현을 매핑시켜 주는 메카니즘을 제공한다. 개념망은 학습하고자 하는 주제 영역에 포함되는 지식들을 서로 관련성 있게 구축해 놓은 지식망이다. 여기에는 주로 선언적인 지식 즉, ‘~는 ~이다’와 같은 형식으로 표현될 수 있는 모든 지식이 포함되며, 명사 형태의 자연어로 기술된다. 또한 개념 간의 링크는 관련성을 나타내며 역시 동사 형태의 자연어로 기술된다. 이러한 개념 간의 관련성을 이용해서 동적인 문제의 생성이 가능하며, 학습자의 오류 진단이 수행된다.

개념 클래스는 개념망의 노드를 정의한 클래스이다. 각각의 개념은 프레임으로 처리되며 제목, 정의, 속성, 관련된 개념, 상위 개념, 하위 개념의 필드로 구성된다. 제목은 개념망에서 유일함을 규정 짓는 특성이 되고 정의 필드는 개념에 대한 설명이 자연어 형태로 기술된다. 속성 필드는 개념이 갖는 속성을 관련된 개념과의 링크로 표현한다. 또한 상하위 개념 필드는 각 개념에 대한 추상적인 개념 혹은 구체적인 개념에 대한 링크를 가지고 있다.

교수 단위망은 학습 순서와 관련된 정보 및 학습자에게 제시될 이벤트를 포함하는 지식망이다. 개념망과 마찬가지로 교수 단위망은 계층적인 망으로 구성되며 각 노드는 하위 단위에 대한 링크를 가지고 있다. 교수 단위망에서 하위 단위라 하면 상위 단위의 선수 지식을 의미하게 된다. 다시 말해서 어떤 노드 즉 교수 단위를 학습하려면 하위 단위에 대한 학습이 선행되어야 한다. 결과적으로 이러한 계층적인 망 구성은 자연스럽게 학습 진도를 결정하는 수단을 제공한다. 교수 단위망 내에서 각각의 노드는 학습 단위를 나타내며 여기에는 단위에 대한 제목, 학습 목표, 학습 단위와 관련된 개념들에 대한 링크, 이벤트의 순서, 문제에 대한 링크 그리고 하위 단위에 대한 링크가 포함된다. 교수 단위망은 텍스트 및 그래픽 객체를 포함하고 있다. 텍스트는 학습시 학습자에게 제시될 스트링으로 저작자에 의해 입력된다. 텍스트는 학습을 위한 필수적인 요소로, 교수 단위망에 포함된 텍스트 필드는 실제 스트링이 저장된 주소에 대한 포인터를 가지고 있다. 텍스트는 교수 단위망 구축시 텍스트 에디터에 의해 생성되고 저장된다.

교수 단위망은 또한 문제 제공을 위한 필드를 포함하고 있다. 문제는 학습자가 어떤 개념이나 개념간의 관계에 대해서 확실하게 이해하고 있는지를 파악하기 위해서 제시되는 일종의 이벤트이다. 문제는 교수 단위망 내의 교수 단위 내에서 생성되고 각 교수 단위와 관련된 개념에 대한 리스트를 가지고 있어서 이들이 실제 문제 생성을 위한 기초가 된다. 저작자는 관련된 리스트에서 개념과 문제의 유형을 선택한 뒤 시스템에 의해 제공되는 템플릿을 결정함으로써

사전에 문제의 유형을 결정한다. 여기에서 템플릿은 테스트할 문제의 종류를 의미한다. 이러한 유형으로는 정의, 개념의 특성, 개념간의 관계 등에 대한 지식등이 될 수 있다. 각각의 문제 필드에는 그 문제에 대한 난이도 및 중요도가 입력된다. 학습자 모듈은 이 두가지 요소를 기준으로 학습자의 이해도를 측정하고, 그 결과는 새로운 학습 전략을 구축하는데 이용한다.

### 3.4.3 학습자 모듈의 설계

학습자 모듈의 입력으로는 학습자의 응답, 학습 전략 모듈로 부터의 진단 요구, 교수단위망으로 부터의 문제에 대한 속성이 있다. 학습자 모듈의 출력은 입력 데이터로 부터 진단한 결과이다. 이 진단 결과는 학습 전략 모듈에서 진도 결정, 조언 제공, 설명 선택등을 판단하는 추론 기준이 된다[9,14,22].

본 연구에서 설계한 학습자 모듈은 오버레이 모델을 사용한다. 현재까지 학습해 온 학습자의 지식은 학습자 모듈 내에서 학습자 데이터베이스로 저장되며, 학습 경로나 학습 성취도를 함께 표현할 수 있다. 학습자 데이터베이스 내의 학습자 지식망(Student Knowledge Network)은 학습이 진행됨에 따라 전문가의 지식베이스인 교수단위망과 같은 형태의 망 구조를 이루게 된다.

학습자 지식망 내의 노드는 학습자의 진단 결과, 제출된 문제의 종류와 그 보기가 저장되며 학습자 지식망의 형태는 학습 경로를 표현하게 된다.

학습자 응답의 진단과 지식의 추론을 위해서는 언어 변수를 사용한다. 퍼지 이론에서 사용되는 언어 변수는 저작자에게 보다 쉬운 진단과 추론 규칙의 입력을 위해 도입되었다. 사용되는 언어 변수는 표 3과 같다.

이러한 언어변수를 사용함으로써 영역 전문가는 모든 학습 요소의 어트리뷰트에 대하여 특정한 실수를 결정하여 입력하지 않고 언어를 이용할 수 있게 되어 보다 친숙한 환경에서 입력을 할 수 있게 된다. 언어변수는 애매한 인간의 표현을 컴퓨터에서 이용할 수 있도록 한 퍼지 이론의 핵심이다. 퍼지 이론에서 사용하는 응축(co-

표 3 사용 가능한 언어 변수의 종류

	교수단위	진단 결과
난이도	Difficult, Medium, Easy	Good, Medium, Bad
중요도	Important, Medium, Not Important	
수식어	[Very]*, [Little]*	

ncentration)과 팽창(dilation)을 이용하면 학습자의 지식 상태를 진단하는 함수 DV (Diagnostic Value)를 다음과 같이 정의할 수 있다.

$$DV(R) = \text{norm}(\sum (R_c/R_n)^k)$$

Norm : 정규화를 실행하는 함수.

$R_{1-n}$  : 학습자가 응답한 처음부터 n까지의 문제

$R_c$  : 언어변수에 따른 정답의 수

$R_n$  : 학습자에게 제시된 총 문제 수

$R_c/R_n$  는 총 문제수에 대한 맞은 문제의 비율이다.

k : 언어변수에 따른 역승의 수

\*Important or Difficult: 1/2

\*Medium : 1

\*Not important or Easy : 2

DV 함수에 의한 진단 값을 언어변수로 퍼지화하면 Good, Medium, Bad와 수식어 [Very]\*, [Little]\*로 이루어진 하나의 언어변수가 된다. 진단결과를 언어변수로 퍼지화하면 언어변수를 이용한 다른 추론 규칙과 함께 사용할 수 있게 된다. 추론 규칙은 테이블의 형태로 저장되며, 언어변수로 이루어져 있으므로 저작자가 쉽게 입력할 수 있게 된다. 또한 영역 전문가가 귀속 함수를 변경하면 같은 DV 함수 값에 대해서도 진단 결과는 바뀌게 되어 추론 규칙을 수정하지 않아도 된다. 저작자가 추론 규칙을 입력하여 사용하는 경우 규칙 사이에 충돌이 발생하면, 즉 같은 조건에 대하여 다른 결론을 입력할 때에는 저작자에게 조건부를 보여주고 입력된 결론 중 어느 것을 사용할지를 묻게 된다.

### 3.4.4 학습 전략 모듈

학습 전략 모듈은 학습자 모듈에 의해 진단된 학습자의 지식 상태 추론 결과를 토대로 하여 동적인 학습 전략의 수립이 가능하도록 해야 한

다[27]. 전체 학습 내용은 학습자에게 가르칠 학습 내용중 비교적 서로 독립적인 주 교수목표들의 망으로 구성되는데, 각각의 주 교수 목표는 자신을 루트 노드로 하는 독립적인 교수단위망으로 구성된다. 주 교수 목표는 선수지식을 다루는 부 교수목표들로 세분된다. 또한 부 교수 목표들은 다시 세분화되어 대상 모집단의 초기 학습에 적합한 교수 단위가 될 때까지 재귀적으로 확장되어 계층적 격자(hierarchical lattice)를 형성한다. 이와 같이 구성되는 교수단위망에서 자노드들은 그들의 부노드의 선수 학습 조건이 된다. 따라서 교과과정은 선수지식에 해당하는 노드들을 먼저 다루는 선형 리스트가 된다.

강의 계획은 교과과정 계획에 의해 결정된 교수단위의 교수자료와 평가문제를 학습자의 수준에 맞게 제시하기 위한 것이다. 학습 전략 모듈은 교수 단위에 대한 교수자료와 평가 문제를 강의 계획에 의해 결정된 방법으로 학습자에게 제공한다.

학습의 유형은 인지 수준에 따라 문제 해결 학습, 규칙/원리 학습, 개념 학습, 구별 학습, 자극과 반응 학습등 크게 5가지 유형으로 나눌 수 있으며, 학습 유형에 따라 교수 자료의 제시방법을 결정할 수 있다.

학습자가 계속해서 특정 난이도 이상의 교수 단위에 대한 학습에 실패한다면 그 학습자에게는 그 난이도 이상의 교수단위들에 대해서는 원래 교수목표의 달성을 강요하지 않는 것이 좋다. 이는 학습자의 수준에 적합한 단계적 교수목표를 설정함으로써 처리할 수 있다. 단계적 교수목표는 교수단위의 난이도와 이전까지의 학습자의 성적에 의해 예상되는 학습자의 수준을 고려하여 생성하도록 한다. 이전까지의 학습자의 성적을 예상하기 위하여 다음과 같은 식을 이용할 수 있다.

$$ESL = \frac{(a_1 * TSL + a_2 * LTL + a_3 * PKL)}{UDL}$$

단,  $a_1 + a_2 + a_3 = 1$

- ESL : 예상되는 학습자의 성적
- UDL : 학습할 교수단위의 난이도
- TSL : 이전까지 학습한 교수단위들에 대한

성적

LTL : 같은 학습유형의 교수단위들에 대한 성적

PKL : 직접적 선수지식에 대한 성적

학습자의 예상 성적이 결정되면 이를 저작시 저작자가 정해놓은 교수단위의 궁극적 교수목표와 비교하여 예상 성적이 궁극적 교수 목표보다 낮은 경우에는 단계적 교수목표를 예상 성적에 맞추어 교수목표를 하향 조정한다. 학습자가 하향 조정된 교수목표를 달성하는 경우에는 그 차이 만큼 단계적 교수목표를 상향 조정하여 궁극적 교수목표를 달성할 때까지 학습을 진행시킨다. 만일 학습자가 학습자 수준에 의거한 단계적 교수목표를 달성하지 못하는 경우에는 단계적 학습에서 보충교육 학습으로 넘어간다. 보충교육 후에도 계속해서 교수목표를 달성하지 못하면, 그 차이 만큼 교수단위의 난이도를 증가시키고 다시 단계적 교수목표를 생성한다. 이와 같이 단계적 학습과 보충교육 학습을 통해 학습자의 능력에 맞는 개별 교수를 제공할 수 있고, 학습자에게 성공에 대한 확신감을 심어줄 수 있으며, 궁극적으로 원래의 교수목표를 달성하도록 지도하는 일관성 있는 학습을 제공할 수 있다. 위의 모든 학습 과정을 거친 후에도 계속해서 학습자가 교수목표의 달성에 실패하는 경우에는 교수 자료나 평가 문제가 부적절하다고 판단하고 다음 교수단위를 학습하도록 한다. 저작자는 학습 전략 모듈이 결함이 있다고 판단하는 교수단위를 수정 할 수 있다.

보다 다양하고 정확하게 학습자의 지식을 진단하기 위해서는 풍부한 문제의 동적인 생성이 가능해야 한다. 학습 전략 모듈은 개념 학습과 학습에 대한 문제들을 개념망을 이용하여 동적으로 생성할 수 있다. 문제의 형태는 선다형과 단답형을 위한 템플릿에 따른다. 문제의 생성은 두 단계로 이루어진다. 먼저 학습 전략 모듈은 교수단위망의 개념 리스트로부터 개념과 문제의 유형을 선택한다. 그런 다음 템플릿에 해당 개념의 선언적 지식을 삽입함으로써 문제를 생성한다.

효율적인 학습을 위해서는 교과과정 뿐만 아

나라 문제의 제시 순서도 학습자의 수준에 맞게 동적으로 계획되어야 한다. 학습 전략 모듈은 제시할 문제의 난이도를 학습자의 예상 성적인 ESL로 초기화한다. 학습 전략 모듈은 이 난이도에 의해 문제를 제시하며, 학습자의 응답이 맞은 경우에는 확인 질문을 하고, 틀린 경우에는 보충교육 전략을 적용 시킨 후 다시 질문한다. 보충교육후에도 틀린 경우에는 문제의 난이도를 한단계 아래로 수정한다.

### 3.4.5 학습자 인터페이스 모듈 및 정보 인터페이스 모듈

학습자 인터페이스 모듈은 저작자에 의하여 교수 단위망에 저장된 설명을 위한 텍스트와 그래픽을 출력하고, 문제 템플릿에 의한 문제를 출제하는 기능을 가진다. 또한 문제에 대한 학습자의 응답을 입력받아서 정보 인터페이스로 전달하는 기능을 가진다. 교수 단위망의 텍스트 객체는 출력될 문자열의 색상, 속성, 크기, 출력 위치 등의 정보를 함께 저장하며, 그래픽 객체는 선분, 박스, 원 등을 출력하기 위한 명령어로 구성되어 있다.

정보 인터페이스는 그림 2에 나타난 바와 같이 4가지의 서로 다른 모듈간의 독립성을 유지하고, 효율적인 정보 전달을 위하여 필요한 모듈이다. 정보 인터페이스로 입/출력되는 데이터의 종류는 표 4와 같다.

이러한 데이터들은 모두 데이터의 흐름 제어를 위한 데이터 헤더를 가지고 있다. 데이터 헤더에는 데이터의 입력 모듈 이름과 데이터의 출력 모듈 이름을 포함하여 정보 인터페이스에 의하여 데이터를 정확히 전달할 수 있다.

표 4 정보 인터페이스의 입출력

입력 모듈	데이터	출력 모듈	비고
전문가 모듈	텍스트/그래픽 소스 확인	학습자 인터페이스	화면 출력
학습자 인터페이스	학습자의 입력	학습자 모듈	학습자 진단
학습자 모듈	학습자 진단 결과	학습 전략 모듈	학습 전략 수립
학습자 모듈	정답 및 문제 유형 요구	전문가 모듈	학습자 진단
학습 전략 모듈	진단 결과	학습자 모듈	학습 전략 수립
학습 전략 모듈	학습 전략	전문가 모듈	학습 유형/진도

## 4. 결론 및 IAS 발전 방향

현재 선진 외국에서는 교육용 소프트웨어와 관련해 많은 정부의 투자로 정규 학교에서의 활용 뿐만 아니라 비 정규 교육 기관에서도 상당히 활발하게 이용되고 있으며 그 결과가 상당히 고무적이다. 그러나 국내에서는 이론적으로는 논문이 발표되고 있으나 실제적으로는 단순히 교과서나 텍스트의 프리젠테이션 정도에 그치고 있으며, 멀티미디어 데이터의 활용으로 인하여 보기에는 좋은 소프트웨어가 많이 개발되었지만 교육용 프로그램의 특징을 살린 제품은 거의 전무한 실정이다.

현재 국내에서 개발된 지능형 저작 시스템은 일부 대학 연구실에서 발표된 논문이나 프로토타입을 제외하고는 응용이 거의 없는 실정이다. 국내에서 개발된 교육용 프로그램들은 교육개발원에서 배포하는 GREAT를 제외하고는 거의 프리젠테이션용 저작 도구나 범용 프로그래밍 언어로 제작되어서 교육용 소프트웨어의 가장 큰 특징인 학습자의 응답 진단 및 그에 따른 개별화 학습을 위한 특징적인 부분들이 미약한 실정이다. 따라서 제대로 개발된 ITS가 없는 실정에서 IAS를 개발하는 것은 불가능하다고 할 수 있다. 개발된 대부분의 교육용 프로그램은 단순히 코스웨어를 제시하고 간단한 문제를 풀어 그 정오 여부에 따라 진도를 계속할 것인가 아닌가를 결정하는 초보 단계에 있다. 그러나 사용자의 수준 향상과 컴퓨터 마인드의 확산으로 인한 소프트웨어의 발전에 대한 요구가 점점 증가함에 따라 그에 따른 소프트웨어의 발전이 기대된다. CAI가 아닌 ITS의 개발이 활성화되면 이러한 ITS를 제작하기 위한 IAS에 대한 요구도 증가하게 될 것이다.

또한 하드웨어 기술의 급속한 발전에 따른 멀티미디어 장비의 가격 하락과 일반인들의 관심 증대로 인하여 교육용 소프트웨어도 멀티미디어를 지원하지 않으면 시스템과 학습자 간의 전달 효과가 커지지 않게 되었다. 멀티미디어는 음성, 음향 효과의 처리와 정지화상 및 동화상과 같이 보다 다양한 매체를 인간과 컴퓨터의 상호 작용에 이용할 수 있게 하였으며, 따라서 학습자와

시스템간의 많은 상호작용을 필요로 하는 교육용 프로그램에 적절히 이용될 수 있다. 최근들어 멀티미디어 장비가 급속히 보급되고, 아직 초보 단계에 있지만 국내의 교육용 프로그램에의 이용이 점진적으로 확산됨에 따라 교육용 프로그램이 응용되는 분야가 점점 더 복잡하고 다양화되었다. 또한 인공 지능 분야의 발달로 인하여 학습자의 응답에 대한 반응을 보다 인간 교사와의 학습과 유사하도록 지능화시키는 것이 가능하게 되었다. 그러나 이러한 인공 지능 기법을 도입한 시스템은 외국의 여러 연구실에서 개발된 프로토타입 정도에 머물러 있고, 국내 시제품 중에서는 이러한 인공 지능 기법을 응용한 교육용 시스템만의 특징을 살린 제품은 극히 드물다고 할 수 있다. 따라서 지능형 저작 도구는 이러한 멀티미디어 데이터의 처리 및 보다 인간에 가까운 인공지능을 내장시키고, 또한 교실 수업의 대체를 위한 LAN 환경의 지원 기능도 포함시켜 완전한 ITS를 생성할 수 있는 IAS로 발전해야 할 것이다.

### 참고문헌

- [1] Anderson, J. R., Reiser, B. J., "The LISP Tutor," Byte, vol. 10, No., pp. 159~175, 1985.
- [2] Augusteijn, M. F., Broome, R. W., Kolbe, R. W., "ITS Challenger-A Domain-Independent Environment for the Development of Intelligent Training Systems," Journal of Artificial Intelligence in Education (1992) 3, pp. 183~205, 1992.
- [3] Begg I. M and Hogg, L., Authoring System for ICAI, In Kearsley, G. P. (editor), Artificial Intelligence & Instruction (Application and Methods), Addison-Wesley Publishing Co., Inc., Menlo Park, California, 1987.
- [4] Begg, I. M., "Intelligent Authoring System," IEEE. Proc. of COMPINT 85 Computer Aided Technologies, 1985.
- [5] Bonar, J., Cunningham, R., & Shultz, J., "An Object-oriented architecture for intelligent tutoring systems," Proceedings of OOPSLA-86, 1986.
- [6] Crew, P., "tbtEXPERT:A Case Study in Integrating Expert System with Computer Assisted Instruction," Proc. of the 3rd Int.Conf.on Data Engineering, IEEE, 1987, pp. 556~562.
- [7] Half, H. M., "Curriculum and Instruction in Automated Tutors," M. C. Polson & J. J. Richardson Eds., Intelligent Tutoring Systems, Lawrence Erlbaum Ass. Pub., 1988.
- [8] Kearsley, G. P., Artificial Intelligence & Instruction (Application and Methods), Addison-Wesley Publishing Co., Inc., 1987.
- [9] Lewis, M. W., Milson, R., Anderson, J. R., The TEACHER'S APPRENTICE: Design an Intelligent Authoring System for High School Mathematics, In Kearsley, G.P.(editor), Artificial Intelligence and Instruction, Addison-Wesley Publishing Co. Inc., 1987.
- [10] Maddux, C. D., "User-Developed Computer-Assisted Instruction", Educational Technology, 1992. 4.
- [11] Major, N. and Reichgelt, H., "COCA: A Shell for Intelligent Tutoring Systems", Intelligent Tutoring Systems, Second International Conf. ITS '92, 1992.
- [12] Martha C. Polson, J. Jeffrey Richaedon & Elliot Soloway, Foundations of Intelligent Tutoring Systems, Lawrence Erlbaum Associates, inc. Publishers, 1988, pp. 1~19.
- [13] Rich, E., ARTIFICIAL INTELLIGENCE, McGraw-Hill Book Co. 2nd, 1991.
- [14] Shim, L. S., STUDENT MODELING FOR AN INTELLIGENT TUTORING SYSTEM: BASED ON THE ANALYSIS OF HUMAN TUTORING SESSIONS, Ph.D. Dissertation, Illinois Institute of Technology, Chicago, Illinois, 1991.
- [15] Sleeman, D. H. and Brown, J. S., Introduction: Intelligent Tutoring System, In D. Sleeman and J. S. Brown (editors), Intelligent Tutoring System, Academic press, New York, 1982.
- [16] Waterman, D. A., A Guide to Expert Systems. Addison-Wesley Pub. Co., 1985.
- [17] Wenger, E., Artificial Intelligence and Tutoring System: Computational approaches to the communication of knowledge, Los Altos, CA: Morgan Kaufmann.
- [18] Woolf, B., & MacDonald, D. D., "Building a co-

mputer tutor: Design issues," *IEEE COMPUTER*, 1984. 10, pp. 61~73.

[19] 장운구, 이세훈, 윤경섭, 왕창중, "프로토타입 지능형 저작 시스템의 설계 및 구현," 추계전산 교육학술발표회 논문집, 1992. 11, pp. 71~91.

[20] 공희창, 이세훈, 진영배, 윤경섭, 왕창중, "지능형 저작 시스템을 위한 그래픽 사용자 인터페이스의 설계 및 구현," 한국정보 과학회 추계 학술발표 논문집, Vol. 20, 제 2호, 1993. 10, pp. 1293~1296.

[21] 백영태, 이세훈, 윤경섭, 왕창중, "ITS에서의 학습자 모듈 설계", 한국정보과학회 학술논문집, Vol. 19, No. 2, 1992.

[22] 서대우, 이세훈, 윤경섭, 왕창중, "지능형 저작 시스템에서 언어 변수를 이용한 학습자 모듈," 한국정보과학회 가을 학술발표논문집, Vol. 20, No. 2, 1993.

[23] 윤경섭, 왕창중, "Authoring System for Development ITS", *IEEE TENCON '94*, Singapore, Aug.. 1994, to be presented.

[24] 이세훈, 왕창중, "프로그래밍 언어교육용 S/W", 춘계전산교육심포지움발표논문집, 한국정보과학회 전산교육연구회, 1992.

[25] 인하대학교 부설 컴퓨터 과학 응용 연구소, "프로그래밍 언어 교육을 위한 지능적 교수 시스템에 관한 연구", 최종 연구 보고서, 1993.

[26] 조원희, 이세훈, 윤경섭, 왕창중, "지능적 프로그래밍 언어 교수 시스템에 관한 연구", 한국정보과학회 추계학술발표 논문집, Vol. 18, No. 2, 1991. 10.

[27] 탁진현, 손완기, 윤경섭, 왕창중, "지능형 저작 시스템을 위한 동적 교수 계획기의 설계," '93 가을학술발표논문집. 한국정보과학회, 20권 2호, 1993.

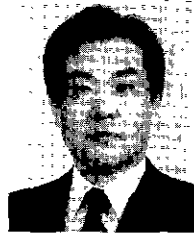
**윤 경 섭**



Intelligent Tutoring System

1982 인하대학교 전자계산학과 학사  
 1984 인하대학교 전자계산학과 석사  
 1990 ~ 현재 인하대학교 전자계산학과 박사과정  
 1987 ~ 현재 인하공업전문대학 전자계산과 부교수  
 동 대학 전자계산소장  
 관심 분야 : Software Engineering, Expert System,

**왕 창 중**



동 대학교 컴퓨터과학응용연구소장  
 관심 분야 : Software Engineering, CASE, Object-Oriented Methodology, Expert System, Intelligent Tutoring System

1964 고려대학교 물리학과 학사  
 1975 성균관대학교 경영학 석사  
 1981 ~ 1990 인하대학교 전자계산소장  
 1992 ~ 1993 한국정보과학회 부회장, 전산교육연구회 위원장  
 1979 ~ 현재 인하대학교 전자계산공학과 교수

● 데이터베이스 ●

**'94 총회·초청 세미나**

- 일 자 : 1994년 8월 19일
- 장 소 : 서강대학교
- 문 의 : 서강대 박 석 교수  
(02) 705-8487