

# 초보자들의 프로그래밍 학습을 위한 웹 기반 적응형 하이퍼미디어 시스템

정효숙<sup>†</sup> · 박성빈<sup>††</sup>

## 요 약

웹 기술의 빠른 성장과 더불어, 웹은 다양한 측면에서 우리의 삶뿐만 아니라 교육과 학습을 변화시켰다. 웹 기반 학습은 자기 주도적 학습과 창의적 학습을 언제 어디서나 지원해 줄 수 있게 되었지만 학습 내용들로 구성되는 하이퍼 공간의 구조가 복잡할 경우 학습자들은 방향 감각 상실이나 인지 과부하를 경험할 수도 있다. 본 논문에서는 인지적 부하 이론을 근거로 초보자들이 프로그래밍 학습을 할 때 경험할 수 있는 인지 부하를 줄일 수 있는 웹 기반 적응형 하이퍼미디어 시스템을 설계하고 구현하였다.

키워드 : 프로그래밍 학습, 적응형 하이퍼미디어, 인지적 부하 이론

## A Web-based adaptive hypermedia system for novices to learn programming

Hyosook Jung<sup>†</sup> · Seongbin Park<sup>††</sup>

## ABSTRACT

With the rapid growth of the Web technologies, the Web has changed teaching and learning as well as our life in various ways. Web-based instruction (WBI) supports self-directed and creative learning at anytime anywhere. However, learners may experience the problems of disorientation and cognitive overload when the hyperspace that consists of learning materials is complex. In this paper, we present a Web-based adaptive hypermedia system based on cognitive load theory which can reduce cognitive loads that novices may experience when they learn programming

Keywords : programming learning, adaptive hypermedia, cognitive load theory

## 1. 서 론

프로그래밍 학습은 동시에 여러 가지 지식과 기능을 필요로 하기 때문에 프로그래밍 경험이 없는 초보자들은 많은 어려움을 느낀다[1]. 프로

그램을 작성하기 위해 사용하는 프로그래밍 언어의 문법과 작성한 프로그램의 의미를 이해해야 하고, 주어진 문제를 풀기 위한 알고리즘을 설계할 수 있어야 하며, 그 알고리즘을 프로그램으로 만들 수 있어야 한다. 또한 이렇게 작성한 프로그램을 컴파일하고 발생한 에러를 수정하며 실행 결과를 확인해야 한다[2]. 이러한 복잡한 활동으로 인해 발생하는 인지적 과부하(cognitive

† 정 회 원: 고려대학교 컴퓨터교육학과 박사과정  
 †† 종 신 회 원: 고려대학교 컴퓨터교육과 교수(교신저자)  
 논문접수: 2004년 8월 23일, 심사완료: 2004년 9월 29일  
 \* 본 논문은 2003년 고려대학교 특별연구비에 의하여 수행되었음

overload)는 프로그래밍에 대한 지식과 기능의 습득을 방해하므로 학생들에게 발생하는 인지적 과부하를 알맞게 조정하여 효과적인 프로그래밍 학습을 수행할 수 있는 교수 설계가 필요하다.

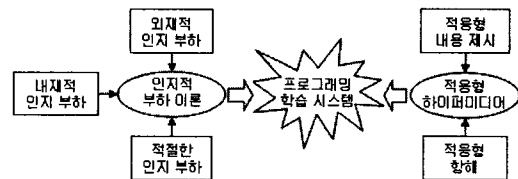
인지적 부하 이론(cognitive load theory)에서는 학생들의 학습 구조를 인간의 장기기억(long-term memory)과 단기기억(short-term memory) 또는 작동기억(working memory)에서 발생하는 정보 처리 과정으로써 설명하고 있다[3]. 장기기억은 반영구적이며 풍부한 스키마(schema)를 저장하고 있는 반면, 작동기억은 그 용량과 지속성이 매우 제한적이기 때문에 학습에 제약을 주게 된다. 작동기억은 내재적 인지 부하(intrinsic cognitive load), 외재적 인지 부하(extraneous cognitive load), 적절한 인지 부하(germane cognitive load)라는 세 가지 인지 부하에 의해 영향을 받는다. 내재적 인지 부하는 학습 내용 자체가 갖고 있는 고유한 인지 부하로서 고정적으로 작동기억의 용량을 차지하고 있다. 외재적 인지 부하는 불필요한 학습 활동 때문에 발생하는 인지 부하로서 스키마의 습득과 자동화를 방해하지만, 적절한 인지 부하는 스키마의 습득과 자동화를 위해 작동기억을 사용하는 효과적인 인지 부하이다. 따라서 복잡하고 어려운 학습에서 발생하는 인지 부하를 적절하게 조절하기 위해 교수 설계를 통해 외재적 인지 부하를 축소시키고 적절한 인지 부하를 증가시켜야 한다[3].

게다가 스키마 습득과 자동화는 학습자 개인의 배경 지식, 적성, 흥미, 학습 스타일, 지능 등에 따라 다르게 형성되기 때문에 한 교실에서 같은 내용을 배우더라도 학습 속도와 학습 능력은 학생들마다 다를 수밖에 없다. 따라서 학습에 대한 학습자의 인지적 특성을 고려하여 학습 내용을 제공하거나 설명해야 한다. 그러나 전통적인 교실 수업에서 많은 학생들에게 개별화된 학습을 지원하기는 매우 어렵다.

최근 웹 기술의 발달과 함께 웹 기반 교육(WBI : Web Based Instruction)은 언제 어디서나 학습할 수 있는 환경을 제공할 뿐만 아니라 학생 수가 많더라도 개인의 필요와 요구에 맞는 풍부한 콘텐츠를 제공할 수 있게 되었다. WBI는 웹의 속성과 자원을 이용함으로써 편리한 학습

환경을 제공하는 하이퍼미디어 기반의 교육적 프로그램이다[4]. 그러나 하이퍼미디어는 비선형적인 구조이기 때문에 하이퍼 공간이 복잡하게 구성되어 있는 경우 사용자의 방향 상실감(disorientation)을 초래하며, 동시에 여러 작업을 수행하면서 복잡하게 얽힌 내용을 이해해야 하기 때문에 추가적인 노력과 집중이 요구되므로 인지적 과부하가 발생하게 된다[5]. 적응형 하이퍼미디어(adaptive hypermedia)는 전통적인 하이퍼미디어와 달리 사용자의 배경 지식, 학습 수준, 흥미 등을 고려하여 적응형 내용 제시(adaptive presentation)와 적응형 항해(adaptive navigation)를 제공함으로써 방향 상실과 인지적 과부하라는 문제를 해결하고자 하였다[6][7]. 이러한 적응형 하이퍼미디어의 기능은 학생들의 불필요한 탐색 활동을 줄이고 자신의 능력이나 흥미에 따라 학습 내용을 효과적이고 자동적으로 살펴볼 수 있도록 돕기 때문에 외재적 인지적 부하를 줄일 수 있으며, 동시에 잉여 작동기억의 자원이 스키마 습득과 자동화에 사용되게 함으로써 적절한 인지 부하를 증가시키게 된다.

본 논문에서는 이러한 인지적 부하 이론에 기반을 두고 웹을 통한 프로그래밍 학습에서 발생하는 인지적 과부하를 알맞게 조절함으로써 초보자의 프로그래밍 지식과 기능의 습득을 돕는 웹 기반의 적응형 하이퍼미디어 시스템을 제안하고자 한다. [그림 1]은 프로그래밍 교육 시스템에 영향 미치는 요소들 간의 관계를 나타내고 있다.



[그림 1] 프로그래밍 교육 시스템에 영향 미치는 요소들 간의 관계

일반적으로 적응형 하이퍼미디어는 페이지 접근 여부에 따라 사용자 정보를 갱신하지만, 프로그래밍 학습에서는 텍스트로 된 프로그램에 대한 이해뿐만 아니라, 실제로 프로그램을 작성하고 저장한 후 컴파일 및 디버깅하여 실행해 보는 활동이 필수적이다. 따라서 본 논문에서 제안한 시

시스템은 학습자의 컴파일 이벤트를 감지하여 사용자 정보를 갱신하고 이를 바탕으로 학습자의 프로그래밍 지식수준을 판단하고자 한다.

본 논문은 다음과 같이 구성된다. 2장에서 관련 연구 내용들을 소개한 후, 3장에서 적응형 웹 기반 프로그래밍 학습 시스템의 설계에 대해 설명하고, 4장에서 본 논문에서 제안하는 시스템의 구현에 대해 설명하며, 마지막으로 5장에서 본 연구의 결론 및 향후 연구에 대하여 기술한다.

## 2. 관련 연구

적응형 하이퍼미디어는 각각의 사용자의 특성에 따라 내용 제시와 향해를 구성하여 제공하는 하이퍼미디어 시스템이다. 적응형 하이퍼미디어는 사용자 모델(user model), 영역 모델(domain model), 적응형 모델(adaptation model)로 구성되어 있다. 적응형 하이퍼미디어 시스템은 적응형 내용 제시와 적응형 향해를 통해 각각의 개인에게 적합한 개별화된 정보를 전달하고자 하였다[6][7].

Garner는 많은 학생들이 프로그래밍은 배우기 어렵다고 인식하고 있다고 보고 이와 같은 영역에 대한 교육 내용 및 방법을 설계하기 위해서 인지적 부하 이론을 반영해야 한다고 주장하고 있다[12].

Feinberg 등은 WBI를 설계하기 위해 인지적 부하 이론의 원리를 적용하였고, 이는 WBI 학습 경험에서 발생하는 외재적 인지 부하를 줄이는데 도움을 준다고 언급하고 있다. 인지적 부하 이론에 기반 하여 구축된 교육용 웹 사이트 상에서 게임을 통해 학생들이 웹 사이트의 콘텐츠를 이해하고 있는가를 테스트하고 있다[9].

Davidovic은 인지적 부하 이론에 기반 하여 예제의 구조를 통해 학습하는 SEATS(Structural Example-based Adaptive Tutoring System)라는 적응형 교수 시스템을 제안하였다. SEATS는 나란히 놓여 있는 두 예제를 비교하면서 프로그램의 구조적 구성 요소를 강조시킴으로써 프로그램의 중요 요소를 학습할 수 있도록 돕고 있다[10].

복잡한 내용의 학습에서 인지적 부하를 통제하기 위해, van Merriënboer 등은 비계 설정과 적

시에 정보를 제공하는 프레임워크를 제안하고 있다. 비계 설정이란 학생의 학습을 도와주는 모든 장비나 전략을 의미하는 것으로써 논문에서는 학습 내용을 단순한 것에서 복잡한 것으로 점차 확장시키고, 학습 예제를 통해 문제 해결 방식을 익히며, 생략된 부분을 완성해가는 활동을 통해 학생의 능력을 점차 신장시킬 것을 제안하였다. 적시에 정보를 제공하기 위해 실제 문제를 해결하기 전에 도움이 되는 정보를 제공하고, 문제 해결을 위한 절차적 정보도 제공할 것을 제안하였다[11].

Chang은 예제를 통한 학습 패러다임인 완성 전략(completion strategy)을 사용하여 초보자를 위한 프로그래밍 학습 시스템을 개발하였다. 완성 전략은 초보자에게 잘 설계된 프로그램 예제를 제공하는데, 이는 스키마 습득을 용이하게 하는 원리 중의 하나로 이용되고 있다. 완성 전략에 사용되는 예제는 프로그램 코드의 일부가 생략되어 있는 미완성 예제로서 학생들이 이를 완성하는 활동을 통해 주의 깊게 공부하도록 유도하는 방법이다. 또한, 미완성 예제를 완성하는 활동은 학생들이 학습에 적극 참여하도록 하고, 집중하여 학습하도록 함으로써 학습과 관련된 스키마를 형성하도록 돕는다[2].

Renkl 등은 페이딩 절차(즉, 완전한 예제에서 시작되고 있고, 점점 미완성된 예제를 제공하고 나중에는 문제만 제공하는 방법)에 관한 실험을 통해 이러한 방법은 학습에 매우 유용하며 학습을 촉진시킨다고 하였고, 잘 설계된 예제는 스키마 습득과 자동화를 돕는다고 언급하였다[12].

프로그래밍은 언어의 구문 및 의미, 논리, 알고리즘 등의 다양한 지식을 습득해야 할 뿐만 아니라, 컴파일 및 디버깅과 같은 작업도 함께 수행해야 때문에 인지적 부하가 높다. 특히 프로그래밍을 처음 접하게 되는 학습자들은 많은 부담과 어려움을 갖게 되며 기피하는 학습 영역이 되고 있다. 앞서 살펴본 시스템들이 공통적으로 시사하는 바는 인지적 부하 이론에서 제안하고 있는 교수-학습 방법을 시스템에 적용하여 학습자의 인지적 부하를 최적화시킴으로써 프로그래밍 학습의 문제점을 해결할 수 있다는 것이다. 본 논문에서는 인지적 부하를 최적화시키기 위해 학습

자의 인지적 상태에 알맞은 정보를 적용적으로 제공함으로써 인지적 부하 중에서 특히 학습과정에서 불필요하게 발생하는 외재적 인지 부하를 줄이고자 하였다. 또한 인지적 부하 이론에서 제시한 교수-학습 방법 중에서 프로그래밍 학습의 스키마 습득에 도움을 주는 방법으로서 학습 예제(worked example)와 완성 전략을 활용하였다. 이러한 교수-학습 방법의 원리를 적용형 규칙에 반영시킴으로써 초보자들의 프로그래밍 학습에 도움을 주고자 한다.

### 3. 적용형 웹 기반 프로그래밍 학습 시스템의 설계

본 논문에서는 초보자의 프로그래밍 학습에서 발생하는 인지적 과부하를 줄이기 위한 전략을 세워 이를 반영한 사용자 모델, 영역 모델, 적용형 모델을 다음과 같이 설계하였다.

#### 3.1. 인지적 과부하의 축소 전략

프로그래밍 학습은 초보자에게 높은 인지적 부하를 발생시키기 때문에[8], 이를 조절할 수 있는 교수-학습 설계와 시스템적 지원이 필요하다. 본 논문에서는 학습 예제, 완성 전략, 적용형 하이퍼미디어 시스템의 기능을 통해서 학습자의 인지적 부하, 특히 외재적 인지 부하를 줄이고자 하였다.

첫째, 학습 예제는 강의 후에 바로 문제를 해결하기 보다는 유사한 예제를 많이 연습하는 교수-학습 방법이다[3][11]. 초보자에게 다양한 프로그래밍 예제를 제시하여 이를 이해하고 연습하도록 하면, 좋은 프로그램의 구조와 특징을 익히고 프로그래밍 능력을 습득할 수 있으며 처음부터 프로그램을 작성해야 하는 어려움이나 부담도 줄일 수 있다.

둘째, 완성 전략은 미완성된 예제를 제시하고 학습자가 프로그램을 완성하도록 하는 방법이다[2][11]. 이 방법은 학습자가 프로그래밍 학습에 적극 참여하도록 유도하고 학습 예제를 주의 깊게 살펴보게 함으로써 프로그램을 이해 및 분석

하며 실제 프로그래밍을 할 수 있는 능력을 점차 향상시키는 효과가 있다[8]. 처음에는 완성된 프로그램을 제시하여 이를 탐구하도록 하고, 점차 미완성된 프로그램을 완성시키도록 하여 프로그래밍 학습의 인지적 과부하를 줄이도록 한다.

셋째, 적용형 하이퍼미디어 시스템은 사용자 모델에 따라 개별화된 화면과 링크를 제공한다[6][7]. 적용형 화면 제시에서는 주석을 통해 프로그래밍 작성을 지원하는데, 학습자의 프로그래밍 능력이 낮으면 자세한 주석을 제공하여 프로그래밍 작성을 돕고, 학습자의 수준이 높으면 주석을 간단하게 제공하여 학습자 스스로 프로그램을 작성할 수 있는 기회를 제공한다. 적용형 항해에서는 선수 학습을 수행하지 않았을 경우, 학습자가 선택한 주제로 링크되는 것을 막고 선수 학습의 내용이 담긴 페이지로 바로 연결될 수 있는 링크를 제공하여 학습자가 학습 콘텐츠를 찾기 위해 하이퍼 공간을 헤매지 않도록 돕는다.

#### 3.2. 사용자 모델(user model)

사용자 모델은 학습자의 ID, 패스워드, 이름, 이메일 주소와 같은 개인 정보와 최근 학습한 주제, 컴파일 된 예제에 대한 정보, 학습 점수와 같은 학습 정보로 구성되어 있다. 일반적인 적용형 하이퍼미디어에서는 사용자의 페이지 접근 여부에 따라 사용자의 정보를 갱신한다. 그러나 프로그래밍 학습에서는 학습자가 직접 프로그램을 작성하는 활동이 필수적이기 때문에 웹 페이지를 읽는 활동만으로 프로그램의 지식과 기능을 습득했다고 볼 수 없다.

본 논문에서는 학습이 진행되는 동안 학습자의 프로그램 컴파일 활동이 올바르게 이루어졌는지 시스템이 확인한 후, 사용자의 학습 정보를 자동으로 갱신하도록 하였다. 즉, 주어진 프로그램을 컴파일 하도록 한 후, 이를 성공적으로 수행했을 때 학습 수준을 증가시키며, 이러한 학습 정보의 변화에 따라 각 학습자에게 알맞은 화면과 항해를 제공한다. 컴파일 정보는 데이터베이스에 저장되며 각 주제에 속한 학습 예제를 성공적으로 컴파일 했을 때, 컴파일 한 예제의 개수와 학습

점수가 증가되고, 컴파일에 실패했거나 이미 컴파일 했던 예제를 다시 컴파일 한 경우에는 변화가 없다. 학습자가 컴파일 해야 하는 예제 개수는 고정되어 있는 것이 아니며, 교수자의 판단에 따라 변경이 가능하다.

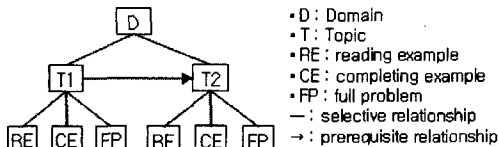
### 3.3. 영역 모델(domain model)

영역 모델은 학습 개념과 개념들 간의 관계로 구성되어 있다. 학습 영역은 여러 개의 주제들로 구성되어 있고, 각 주제는 여러 가지 학습 예제들을 포함하고 있으며 다음과 같이 세 가지 형태로 나뉜다. 첫째, 읽기형 예제는 프로그램을 읽고 그 구조를 이해하도록 하는 예제이다. 둘째, 완성형 예제는 프로그램의 일부가 제거되어 미완성된 예제를 완성하는 예제이다. 셋째, 독립형 문제는 프로그램을 학습자가 스스로 작성하는 예제이다. 본 논문에서는 자바 프로그래밍 학습을 주제로 학습 개념을 구성하였다. [표 1]은 자바 프로그래밍 언어의 기초 중에서 제어문에 대한 학습 개념의 구조를 나타내고 있다.

[표 1] 제어문 학습 개념 구조

영역	주제	학습 예제
Language Basics (control flow statements)	do/while	exam 1, 2, 3
	for	exam 1, 2, 3
	if/else	exam 1, 2, 3
	switch	exam 1, 2, 3
	branching	exam 1, 2, 3

개념들 간의 관계는 두 개념 간에 어떠한 관계를 갖고 있는지를 나타내는 것으로써 선택적 관계와 선행 조건적 관계로 나뉜다. 선택적 관계는 일반적인 하이퍼텍스트에서 나타나는 링크를 말하고, 선행 조건적 관계는 두 개념 중 하나가 다른 하나의 선수학습 요소일 경우, 이를 먼저 학습한 후에 다음 요소를 학습할 수 있음을 의미한다. [그림 2]는 영역 모델의 학습 개념 및 개념들 간의 관계의 계층적 구조를 나타내고 있다.



[그림 2] 영역 모델의 계층적 구조

### 3.4. 적응형 모델(adaptation model)

적응형 모델은 적응형 규칙들로 구성되어 있으며 이러한 규칙들은 적응형 기능을 어떻게 수행하는가를 나타내고 있다. 적응형 규칙은 사용자 모델에 저장된 각 학습자의 학습 정보를 기초로 하여 적응적으로 학습 내용을 제시하거나 항해 경로를 제공한다. 적응형 규칙은 인지적 과부하를 줄이기 위한 전략을 반영하고 있으며, 각 주제별로 제공되는 학습 예제에 대한 학습자의 컴파일 정보를 감지하여 사용자 모델은 자동으로 갱신된다.

학습자의 컴파일 활동은 학습자의 현재 프로그래밍 학습 수준을 판단하는 기준이 되는데, 학습자가 컴파일 해야 하는 예제 개수의 한계치는 고정되어 있는 것이 아니며, 수강한 학습자들의 특성에 따라 교수자가 변경할 수 있도록 하였다. 교수자는 학습이 시작되기 전에 사전 평가를 통해 현재 강의를 듣고 있는 학습자의 평균 수준을 파악하여, 평균 수준이 높으면 한계치를 높이고, 수준이 낮으면 한계치를 낮춘다. 이는 학습이 진행되는 중이라도 교수자가 변경할 수 있으며, 이러한 변화가 즉시 시스템에 반영될 수 있도록 하였다.

본 논문에서는 다음과 같은 적응형 규칙을 적용하였다. 첫째, 학습자의 학습 상태에 알맞은 학습 콘텐츠를 제시하기 위해 프로그래밍 코드의 내용을 학생의 지식수준에 따라 다르게 제시한다. 다음은 학습자의 학습 상태에 따라 프로그래밍 코드의 주석을 다르게 제공하는 적응형 규칙을 나타낸다.

```

if topic_tokens >= threshold
    hide.comment.match(//)
else
    show.comment.full
    
```

만일 한계치가 '3' 이라면, 어떤 주제의 학습 예제들 중에서 컴파일 된 예제의 개수가 3보다 클 때, 그 주제에 대한 학습이 완료된 것으로 간주하고, 프로그램 코드의 주석에서 일부를 숨긴다. 그렇지 않을 때는 프로그램 코드의 모든 주

석을 보여준다. 본 시스템은 프로그램을 작성하는데 도움을 주기 위한 절차적 정보로서 주석을 자세하게 제공한다. 그러나 과잉 콘텐츠는 오히려 외재적 인지 부하를 증가시키기 때문에 학생의 지식수준이 증가함에 따라 프로그래밍 코드의 주석은 간략하게 제시하고자 한다.

둘째, 학습 콘텐츠에 대한 접근 경로를 학습자의 학습 상태에 따라 다르게 제시한다. 학습하고자 하는 주제의 선수학습 요소를 완성하였다면 그 주제를 학습할 수 있으나, 그렇지 못한 경우에는 선수학습 요소를 담고 있는 주제로 이동할 수 있는 링크를 제공한다. 다음은 학습자의 학습 상태에 따라 다른 링크를 제공하는 적응형 규칙이다.

```

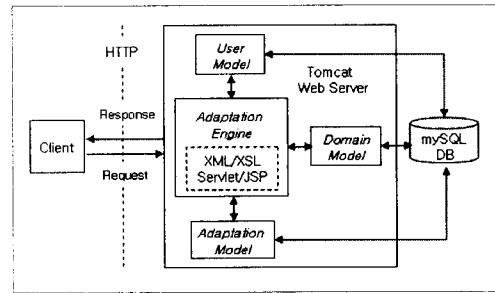
if prerequisite_topic_tokens >= threshold
    show.present_topic.exam
else
    show.prerequisite_topic.link
    
```

두 주제가 선행 조건적 관계일 때, 한계치가 '3'이라고 하자. 선행학습 주제의 컴파일 된 예제 개수가 3보다 클 때, 그 주제에 대한 학습이 완료된 것으로 간주하고, 현재 선택한 주제의 학습 예제들을 보여준다. 그렇지 않을 때는 선행 학습 주제로 이동할 수 있는 링크를 제공한다. 이는 불필요한 항해 활동을 제한함으로써 외재적 인지 부하를 줄이고 선행 지식과 새로운 정보를 통합 시킴으로써 스키마 습득을 촉진시키고자 한다.

#### 4. 적응형 웹 기반 프로그래밍 학습 시스템의 구현

본 시스템은 앞에서 설계한 사용자 모델, 영역 모델, 적응형 모델을 바탕으로 자바로 구현되었다. 각 모델의 정보는 데이터베이스에 저장되어 있으며, 적응형 기능을 생성하는 세 모델간의 상호작용은 적응형 엔진에서 이루어진다. 적응형 엔진은 자바 서블릿과 JSP로 구성되어 있으며, 사용자 모델에 기록된 학습 정보를 바탕으로 적응형 규칙을 적용하여 학습 내용 제시한다. 학습 내용은 XML 형식의 데이터에 XSL 스타일 시트를 적용하여 HTML 형태로 변환한 후 사용자에게 제공한다. 초보자가 자신이 사용하는 시스템

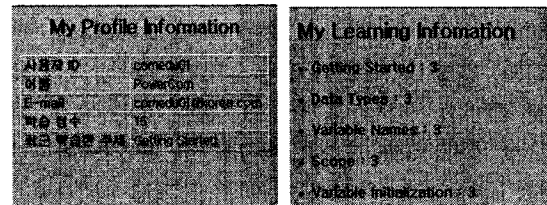
마다 JDK를 설치하고 패스와 클래스 패스 등 시스템 환경 변수를 변경하는 것은 어려운 일이기 때문에[1], 본 시스템은 서버에서 프로그램을 컴파일하고 실행한 후 그 결과를 학습자에게 전송한다. [그림 3]은 본 시스템의 전체적인 구조를 나타내고 있다.



[그림 3] 시스템의 구조

#### 4.1. 사용자 모델의 구현

사용자 모델은 사용자 모델은 학습자의 개인 정보, 학습 점수와 최근 학습한 주제에 대한 정보, 각 학습 주제별로 컴파일 한 예제에 대한 정보 등을 데이터베이스에 저장하고 있다. 학습자는 각 주제별로 얼마나 학습을 했는지를 확인할 수 있으며, 학습자 ID, 이메일 주소, 최근 학습한 주제, 학습 점수 등의 정보도 볼 수 있다. 학습자가 자신의 학습 상황을 확인해 볼 수 있는 기회를 제공함으로써 프로그래밍 학습에 대한 동기 및 도전감을 유발하도록 한다. [그림 4]는 주제별 학습 정보와 학습자의 개인 정보를 보여 주는 화면이다.



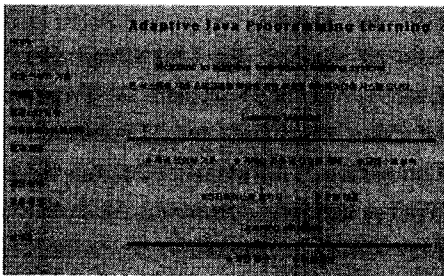
[그림 4] 학습 정보와 개인 정보

#### 4.2. 영역 모델의 구현

본 시스템의 학습 영역은 자바 언어의 기초, 객체, 클래스와 상속, 인터페이스와 패키지, 문제

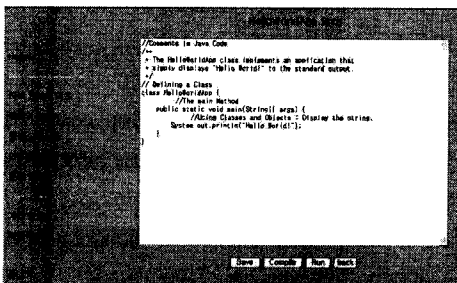
해결 등의 5개로 나뉘고, 각 영역은 여러 개의 하위 주제를 갖고 있으며, 각각의 주제는 상, 중, 하 3개의 학습 예제를 갖고 있다. ‘하’는 읽기형 예제이고 ‘중’은 완성형 예제이며, ‘상’은 독립형 예제이다. 컴파일 해야 하는 예제의 개수는 현재 가르치고 있는 학생의 수준에 따라 교수자가 변경할 수 있다. 본 시스템에서는 학습 예제의 난이도를 다르게 구성하였으나 이를 학습 점수에 반영하지는 못하였다. 향후 난이도에 따라 학습 점수를 부여하고, 컴파일 시도 회수 및 문제당 소요되는 시간도 고려하여 학습 수준을 파악하도록 할 것이다.

학습자는 학습 내용을 시스템이 제시한 메뉴 구성에 따라 선형적으로 학습할 수도 있지만, 학습자의 흥미나 관심에 따라 비선형적으로 학습할 수도 있다. 다만 현재 선택한 학습 주제의 선형 학습 요소에 대한 학습이 완료되었을 때 가능하다. [그림 5]는 학습자가 프로그래밍 학습 시스템에 로그인 했을 때 나타나는 첫 화면이다.



[그림 5] 프로그래밍 학습 첫 화면

학습자는 서버에서 제공하는 프로그램의 소스 파일을 수정할 수 있으며, 서버에 생성된 자신의 폴더에 파일을 저장하고 컴파일한 후 실행 결과를 확인할 수 있다. [그림 6]은 서버에서 제공하는 학습 예제의 소스 파일을 편집할 수 있는 페이지를 보여주고 있다.

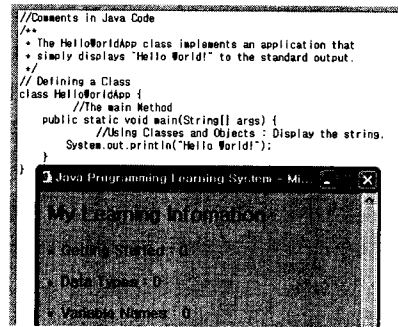


[그림 6] 소스 파일 편집 화면

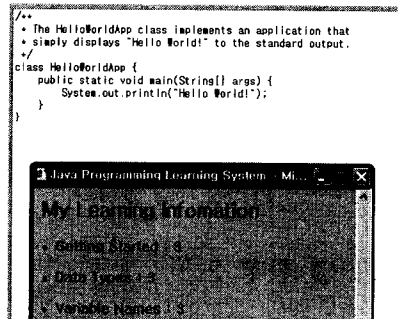
### 4.3. 적응형 모델의 구현

학습자의 학습 정보를 분석하여 적응형 화면 제시와 적응형 향해를 다음과 같은 방법으로 제공한다.

첫째, 학습자가 어떤 주제를 처음 학습하거나 아직 학습을 완료하지 못한 경우라면 자세한 주석을 제공하여 프로그래밍 작성을 돕고, 이미 완료한 학습을 다시 학습할 경우에는 주석을 간단하게 제공한다. [그림 7]과 [그림 8]은 학습 정보가 서로 다른 학습자가 “Getting Started”에 속한 예제를 볼 때, 프로그램의 소스 코드의 주석이 다르게 처리되는 것을 나타내고 있다.



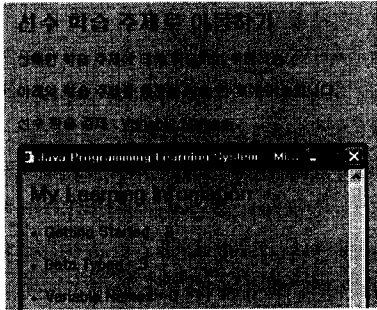
[그림 7] 적응형 화면 제시-A



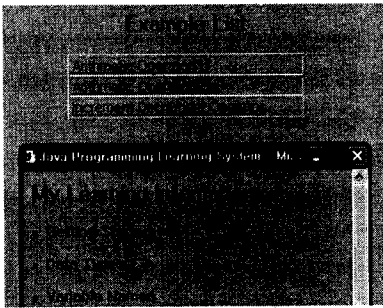
[그림 8] 적응형 화면 제시-B

둘째, 학습자가 어떤 주제를 학습하고자 할 때, 선수 학습을 이미 수행했다면 관련 내용으로 연결될 수 있는 링크를 제공하고, 그렇지 못한 경우 선수 학습을 제공하는 페이지로 연결될 수 있는 링크를 제공한다. [그림 9]와 [그림 10]은 학습 정보가 서로 다른 학습자가 같은 주제, “Arithmetic Operator”를 선택했을 때, 선수 학습

을 수행한 학생에게는 선택한 주제의 학습 링크를 제공하고 선수 학습을 아직 수행하지 못한 학생에게는 선수 학습 페이지로 이동하는 링크가 제시되는 것을 보여주고 있다.



[그림 9] 적응형 항해-A



[그림 10] 적응형 항해-B

본 시스템은 프로그래밍 학습의 인지적 부하를 줄이는데 도움을 줄 수 있도록 인지적 부하 이론의 교수-학습 방법 중 하나인 학습 예제와 완성 전략을 활용하였으며, 각각의 학습자마다 개별화된 학습화면 제시와 항해가 가능하도록 적응형 하이퍼미디어의 기능을 적용하였다.

## 5. 결론 및 향후 연구

웹 기반 프로그래밍 학습 시스템은 교수자의 부담을 줄이고 학생들의 프로그래밍 능력 향상에 도움을 줄 수 있으나 인지적 과부하와 방향 감각 상실을 초래할 수 있고[5], 더욱이 프로그래밍 학습은 초보자에게 높은 인지적 부하를 유발시키기 때문에[8], 이러한 가능성을 줄이기 위해서 인지적 부하 이론의 교수-학습 방법을 적응형 하이퍼미디어 시스템에 적용하였다. 이는 프로그래밍을 처음 접하는 학습자들의 인지적 부하, 특히 외재

적 인지 부하를 줄이는데 도움을 줄 것으로 기대하고 있으며, 현재 프로그래밍 학습에 시스템을 적용하는 실험 중에 있다.

본 시스템에서는 개별화된 학습 내용을 자동으로 생성할 수 있어야 하므로 학습자의 특성 및 학습 내용의 특성을 반영할 수 있는 사용자 모델과 적응형 규칙에 대한 연구를 통해 향후 보다 민감한 적응성을 수행할 수 있도록 사용자 모델을 개선하고 적응형 규칙도 보강할 것이다. 또한 프로그래밍 학습과 같이 많은 인지적 부하를 발생시키는 다른 학습 영역에도 본 논문에서 제안하는 시스템을 사용한다면 학습자들의 인지적 부하를 줄일 수 있을 것으로 기대한다.

## 참고 문헌

- [1] Tony Jenkins, On the Difficulty of Learning to Program, 3rd Annual LTSN-ICS Conference, Lough Borough University, 2002, pp. 53-58
- [2] Kuo-en Chang, Bea-Chu Chiao, Sei-Wang Chen, Rong-Shue Hsiao, A Programming Learning System for Beginners - A Completion Strategy Approach IEEE Transaction on Education, Vol. 43, No. 2, May 2000, pp. 211-220
- [3] Fred Paas, Alexander Renkl, John Seller, Cognitive Load Theory and Instructional Design : Recent Developments, EDUCATIONAL PSYCHOLOGIST, 2003, 38(1), pp. 1-4
- [4] Badrul H. Khan, Web-based instruction(WBI) : What is it and why is it?, Web-based instruction, pp. 5-18, Englewood Cliffs, NJ : Educational Technology Publications, 1997
- [5] Jeff Conklin, A Survey of Hypertext, ACM, Vol. 2 No. 2, Austin, Texas, 1987, pp. 69-109
- [6] Paul De Bra, Peter Brusilovsky, Geert-Jan Houben, Adaptive Hypermedia : From



Systems to Framework, ACM Computing Surveys, Symposium Edition, Volume 31, No. 4es, December 1999.

- [7] Hongjing Wu, Erik de Kort, Paul De Pra, Design Issues for General-Purpose Adaptive Hypermedia Systems, Proceedings of the ACM Conference on Hypertext and Hypermedia, Aarhus, Denmark, August 2001, pp. 141-150
- [8] Stuart Garner, Cognitive Load Reduction in Problem Solving Domains, International Conference in Computer Education, ICCE2001, 2001
- [9] Susan Feinberg, Margaret Murphy, Applying Cognitive Load Theory to the Design of Web-based Instruction, Proceedings of IEEE professional communication society international professional communication conference and Proceedings of the 18th annual ACM international conference on Computer documentation, 2000, pp. 353-360
- [10] Aleksandar Davidovic, Jim Warren, Elena Trichina, Learning Benefits of Structural Example-Based Adaptive Tutoring Systems, IEEE Transaction on Education, Vol. 46, no. 2, May 2003, pp. 241-251
- [11] Jeroen J. G. van Merriënboer, Paul A. Kirschner, and Liesbeth Kester, Taking the Load Off a Learner's Mind: Instructional Design for Complex Learning, EDUCATIONAL PSYCHOLOGIST, 38(1), pp. 5-13, 2003
- [12] Alexander Renkl, Rober K. Atkinson, Uwe H. Maier, From Studying Examples to Solving Problem : Fading Worked-out Solution Steps Helps Learning, Cognitive Science, CogSci2000, Philadelphia, Pennsylvania, USA, 2000

## 정 호 속



1998 서울교육대학교 교육학과  
(교육학학사)

2001 서울교육대학교교육대학원  
컴퓨터교육과(교육학석사)

2003~현재 고려대학교 컴퓨터교육학과 박사  
과정

관심분야: 컴퓨터교육, 적응형 하이퍼미디어

E-Mail: est0718@comedu.korea.ac.kr

## 박 성 빈



1990 고려대학교 전산과학과  
(이학사)

1993 University of Southern  
California (전산학 석사)

1999 University of Southern California  
(전산학 박사)

2003~현재 고려대학교 컴퓨터교육과 조교수

관심분야: 하이퍼텍스트, 컴퓨터교육, 알고리즘,  
계산이론

E-Mail: psb@comedu.korea.ac.kr