

# 사이버 교육을 위한 지능형 프로그래밍 강의 및 평가시스템

김 영 찬\*      최 종 명\*\*

## ◆ 목 차 ◆

- |                    |                           |
|--------------------|---------------------------|
| 1. 서 론             | 4. 복제 검색 시스템과 프로그램 유사도 평가 |
| 2. 연구 사례           | 5. 개발 환경 및 평가             |
| 3. 지능형 코스웨어 시스템 모델 | 6. 결 론                    |

## 1. 서 론

지능형 코스웨어는 대화식 학습 기능을 통해 기존의 코스웨어의 고정적이고 일방적인 교육 방식에서 벗어나 수강생들의 능력에 맞게 학습할 수 있으며, 학습에 필요한 정보를 온라인으로 검색할 수 있도록 제공하는 대화식 시스템이다. 이를 위해서는 온라인 상에서 일어나는 학생과 강사간의 정보교환이 동적으로 처리될 수 있는 시스템이 요구된다. 본 연구에서 구성되는 평가 도구는 학생이 프로그램 과제물을 온라인으로 제출하게 되면 이를 자동화된 도구를 통해 검사하고, 과제의 독창성을 확인하기 위하여 복제된 과제물에 대한 검색 및 색출을 한다. 이러한 기능은 학생들을 효율적으로 관리하는 데 큰 도움이 된다[Benford92].

코스웨어 시스템은 일반적으로 다음과 같이 크게 2가지 메뉴 즉, 학생을 위한 메뉴 및 강사 혹은 코스 설계자(관리자)를 위한 메뉴로 구분될 수 있다[Foxley93]. 학생 메뉴는 학생들을 위한 코스웨어 정보나 강의 내용 및 노트, 과제물에 관한 정보, 프로그램 코딩 및 컴파일, 디버깅에 관한 내용 등에 관련이 되어있다. 강사를 위한 메뉴에는 학생들이 과제물을 제출한 상황 및 날짜와 시간, 학생들에게 피드백 되어온 메일 검사기능, 강의에 필요한 정보 입력 및 과제 제출에 관한 사항 등에 관련된 내용으로 되어있다. 또한 강사를 위한

메뉴에는 코스웨어에 필요한 정보 입력, 필요하다면 프로그램 골격 제공 및 연관된 헤더 파일이나 모듈 및 평가에 이용되는 항목 등을 제시할 수가 있다.

많은 학생들을 대상으로 과제물을 정밀하게 정확하게 평가하기는 쉬운 일은 아니다. 그러나 복제에 의한 과제물 수행은 요즘 문제가 되고 있는 불법 복제에 의한 소프트웨어 사용의 싹이 될 수 있으며, 이는 소프트웨어 분야 산업의 붕괴를 가져올 수 있다는 점을 주지하고 이를 철저히 방지해야 한다. 프로그래밍 언어의 교육 단계에서부터 불법 복제의 관행을 막을 수 있는 철저한 평가만이 프로그램 개발자는 물론 사용자로서 불법 복제의 심각성을 인식하여 이러한 불행한 결과를 방지할 수 있는 유일한 방법일 것이다. 본 시스템은 이러한 불법복제의 방지를 위한 복제 검색 도구를 제공하여 프로그래밍 학습을 시작하는 과정에서부터 이러한 관행이 뿌리내리지 못하도록 하는데 기여하고자 한다.

본 코스웨어 시스템은 다음과 같은 특징을 가지고 있다. 첫째로 대화식 환경을 지원하는 시스템이므로 사이버 교육에 이용이 가능하다. 둘째로 컴파일러의 분석 방법을 이용하여 프로그램의 구조적 특성을 정량적인 방법으로 비교함으로써 소프트웨어 공학 도구로 활용할 수 있다. 셋째로 강의 도우미를 이용하여 처음 언어를 접하는 학생들의 진도 및 학습 방향을 바로 잡아줄 수 있다. 넷째로 과제물 복제 검색 모듈은 학생들이 프로그래밍을 학습하는 단계에서부터 소프트웨어 불법 복제 관행을 막을 수 있다. 다섯째로 일반 회사나 공공기관에서 사용될 경우 학습 비용 절

\* 명지전문대학 겸임교수

\*\* 숭실대학교 컴퓨터학과 박사

감 효과를 볼 수 있으며, 수강생 수와 상관없이 강의를 진행할 수 있다. 여섯째로 학생들의 학습 진행 과정을 검사할 수 있으므로 보다 효율적인 강의를 할 수 있다. 이외에도 학생이나 강사 모두에게 시간적으로 많은 절감 효과를 들 수가 있으며, 보다 객관적인 채점 및 평가를 할 수 있다.

본 논문은 다음과 같이 구성되어 있다. 제 2장에서 연구 사례에 대해서 언급하고, 제 3장에서는 지능형 코스웨어 시스템 모델에 대해서 언급한다. 제 4장에서는 복제 검색 시스템과 프로그램 유사도 평가에 대해서 언급하며, 제 5장에서는 개발 환경 및 평가를 기술하며, 마지막으로 제 6장에서는 결론을 밝힌다.

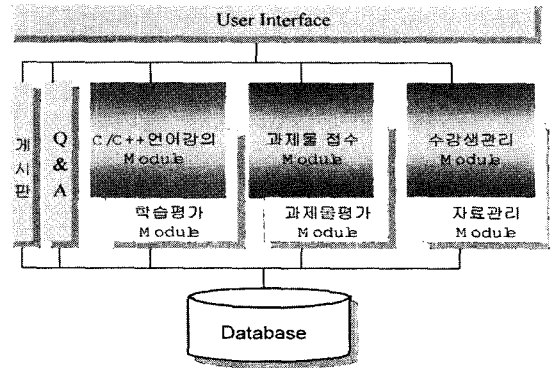
## 2. 연구 사례

사이버 교육을 위한 코스웨어 시스템은 국내외적으로 많은 발전을 거듭해 왔으나 아직은 그 수준이 미약하다. 현재 활성화되고 있는 사이버 교육 분야는 단순히 방송 매체나 인터넷을 통한 일방적인 지식 전달 체제로 이루어져 있다. 또한 국내에 많은 교육 사이트들은 단순히 강의자료 전달이나, 내용 전달, 과제물 제출 등과 같은 일반적 수준을 벗어나지 못하고 있으며, 학생의 학습 상황이나 학생과 강사의 피드백(feed-back) 등은 이루어지지 않고 있는 상태이다.

현재 코스웨어와 관련된 대표적인 시스템은 notingham 대학교의 ceillidh 시스템이다[1,2]. ceillidh 시스템의 주요 기능은 코스웨어 개설뿐만 아니라 과제물 출제 및 제출, 평가 등을 수행하는 통합 코스웨어 시스템이다. ceillidh 시스템의 개설 과목은 인공지능, C, 컴퓨터 이용 및 활용, 형식 언어, C++ 및 소프트웨어 도구 등 다양하게 제공하고 있다. 과제물은 유닉스 시스템의 "motd" 파일을 이용하여 출제하고 있으며, 늦게 제출한 것이나 복제 검사 등을 평가하고 있다. 또한 평가는 휴먼 검사와 자동 검사를 병행하여 평가하고 있으며, 자동 마킹(marking) 기능을 수행한다.

## 3. 지능형 코스웨어 시스템 모델

본 논문에서 제시하고 있는 지능형 코스웨어 시스템의 구성은 그림 1과 같다.



(그림 1) 지능형 코스웨어 시스템 모델

그림에서 나타나듯이 지능형 코스웨어 시스템은 학생들의 과제물이나 강의자료, 질문 및 평가 등을 온라인(웹)으로 제공하며, 제출된 과제물은 데이터베이스에 저장한 후, 평가를 수행하게 된다. 데이터베이스에는 수강생 관리 모듈과 연동되어 있어 검색 및 평가를 할 수가 있다. 데이터베이스에 저장된 과제물은 복제 과제물 검색 모듈에 의해서 검색되고 평가된다. 이 모든 과정이 수행된 후 자료들은 학습 평가 모듈을 통해서 평가되고 관리된다.

### 3.1 C/C++ 언어 강의 모듈

이 모듈에서는 C/C++ 언어 및 프로그래밍에 대한 강의 내용을 제공한다. 강의 내용은 HTML을 기본으로 작성되며 수강생들은 온라인으로 웹 브라우저를 통해서 검색한다. 강의 교재는 일반적인 강의 현장에서 사용되는 정보를 포함하며, 특히 컴퓨터를 매개로 한 온라인 강의라는 특성을 살려서 교재를 개발한다 [Robert96, Kelley84]. 또한 관련 프로그램의 설명에 대한 소스 프로그램을 제공하여 직접 실행해 볼 수 있도록 할 수 있으며, 관련 소프트웨어 정보 등의 부가 기능이 제공된다.

### 3.2 C/C++ 프로그래밍 도우미 / Q & A

실제 프로그래밍 시에 유용한 정보를 제공하는 서비스이다. 본 모듈에서는 관련 라이브러리 정보와 관련된 사이트를 검색할 수 있도록 하며, 프로그래밍 시에

자주 발생하는 문제나 언급되는 문제에 대한 질의/응답을 제공하여 참조 할 수 있도록 한다.

### 3.3 게시판 및 과제물 접수 모듈

강사와 수강자들 간의 의견교환을 위한 서비스로서 멀티미디어 웹으로 구성된다. 일반적인 웹 게시판으로 구성되며 특히 수강생들 간의 원활한 정보교류를 위해 수강생들의 메일링 리스트도 함께 운영된다.

과제물 접수 모듈은 수강생들의 리포트 제출 및 확인 서비스를 제공하는 모듈로서 수강생 관리 모듈과 연관되어서 운영된다. 수강생들은 웹 브라우저를 통해서 서버에 접속하여 온라인으로 리포트를 제출하게 된다. 이 경우, 수강생별 제출 디렉토리 관리와 함께 수강생 인증 서비스가 요구되며 최종본을 유지하기 위한 리포트 별 버전 관리 기능도 요구된다. 본 시스템에서는 핸드인(Hand-in) 모듈을 C 언어를 이용해서 구현하였으며, 수강생 정보관리 및 취합된 리포트의 저장은 데이터베이스와 연동해서 구현하였다.

### 3.4 복제 과제물 검색 모듈

수강생들이 제출한 과제물을 검사하여 복제 프로그램을 색출하는 모듈이다. 데이터 베이스에 저장된 학생들의 과제물은 파싱하여 생성된 AST를 이용하여 복제여부를 평가한다. 제출된 과제물은 제출한 시간(파일 생성시간) 순서대로 분석이 이루어지며, 이 정보는 별도로 관리되어 동일한 특성을 가지는 프로그램이 있는지를 비교하게 된다. 과제물간의 유사성이 높은 것으로 보고된 것들은 별도로 구분되어 강사가 직접 확인하여 다시 평가할 수 있도록 한다. 이때 사용되는 기술은 C 컴파일러에서 전반부 분석을 수행하는 부분과 실행 흐름을 추적할 수 있는 흐름 분석기가 결합되어 프로그램의 유사성을 비교하게 된다. 프로그램 복제와 평가 모듈은 제 4장과 5장에서 자세히 다루기로 하고 여기서는 생략한다.

### 3.5 리포트 및 수강생 관리 모듈

리포트 모듈에서는 강사가 강의진행 상황 및 수강

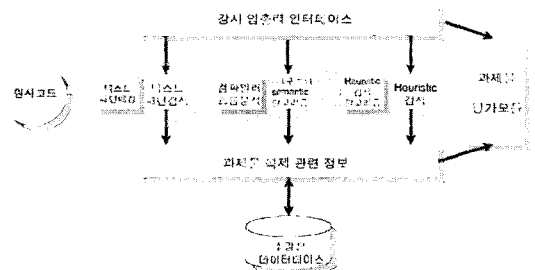
생들의 리포트 제출현황을 온라인으로 확인해 볼 수 있도록 구성되어 있다. 따라서 강사가 원하는 다양한 정보를 효율적으로 검토할 수 있도록 다양한 출력 형식을 지원한다. 이 모듈은 데이터베이스와 연동되어 있으므로 관련 정보들을 SQL문을 이용해서 접근, 검색하여 HTML 형태로 출력하게 된다. 기본적으로 강의일정 및 진행 상황, 수강생들의 출결현황 및 과제물 제출 현황 등의 보고서를 제공한다.

수강생 관리 모듈은 수강생관리, 학생 성적관리 등을 통해 모든 수강생들과 관련된 정보를 일관적으로 유지, 관리할 수 있도록 한다. 강의에 참여하는 수강생에 관련된 정보, 수강생의 성적 및 출석, 과제물 제출 여부 등을 관리한다.

## 4. 복제 검색 시스템과 프로그램 유사도 평가

### 4.1 복제 검색 시스템

복제 검색 시스템은 과제물로 제출된 프로그램이 다른 프로그램을 복제하여 제출된 것인지 여부를 판단하기 위해 사용된다. 과제물 접수 모듈에 의해 수집된 프로그램들은 복제 과제물 색출 모듈에 의해 검색되는데 그 결과는 평가모듈에서 참조하게 되고, 유사도에 따라 정리된 상태로 목록화되어 강사가 열람할 수 있도록 한다. 다음 그림 2는 복제 검색 시스템의 전체 흐름도를 보여준다.



(그림 2) 복제 검색 시스템의 데이터 흐름도

그림 2에서 알 수 있듯이 학생들이 제출한 과제물은 다음과 같은 세 가지 방식으로 유사도 측정 검사

를 하게 된다. 우선, 일반적인 텍스트 패턴매칭 방법에 의한 유사도를 측정한다. 그리고 복제되었다고 판단되지 않는 과제물들은 내장된 컴파일러에 의해 파싱되는데 이 과정에서 만들어지는 트리의 구조와 의미 분석 정보를 가지고 유사도를 측정한다. 이러한 검사과정에 의해서 역시 복제되었다고 판단되지 않는 과제물은 Heuristic한 검색에 의해 만들어진 정보에 의해 유사도를 측정하게 된다. 여기서, Heuristic한 검색이란 일반적으로 학생들이 사용하는 복제방법을 수집하고 분석한 정보들을 토대로 하여 그것에 대응하기 위해 만들어진 알고리즘에 의한 검색을 말한다. 복제하고자 하는 학생들은 복제된 과제물로 평가받지 않기 위하여 다양한 방식으로 복제를 시도할 것이므로 이러한 Heuristic한 검색에 쓰이는 알고리즘의 개발이 시스템의 정확한 평가에 도움이 될 것이다.

## 4.2 프로그램 유사도 평가

본 연구에서 제시한 프로그램 복제 유형 분석 즉, 프로그램 유사도 평가는 [김2003]에서 제시한 프로그램 유사도 평가 알고리즘을 이용하였다. [김2003]에서 제시한 프로그램 유사도 측정 알고리즘은 크게 2단계로 이루어져 유사도를 평가한다. 예를 들면, 다음과 같은 두개의 프로그램 노드 스트링이 존재한다고 가정하자(단, 다음의 숫자는 노드를 나타내는 숫자이다).

P1nodestring : 23 34 25 54 44 45 49 81 83 84 22 55 44 33 90 68

P2nodestring : 34 25 54 46 47 81 83 84 22 55 44 33 90 93 92 95 34 35

첫 번째 알고리즘에서는 두 노드 스트링의 일치된 부분을 검사한다. 즉, 예에서 { 34, 25, 54}, {81, 83, 84, 22, 55, 44, 33, 90} 을 찾아낸다. 또한 찾은 부분 스트링을 집합에 추가한 후, P1, P2 노드 스트링에서 이를 제거한다. 왜냐하면, 찾은 스트링은 다음에 일치하는 스트링과 다시 비교하는 것을 피하기 위함이다. 또한 두 번째 알고리즘에서는 찾은 스트링을 이용하여 유사도를 측정한다. 즉, 노드 스트링 P1, P2는 다음과 같은 유사도를 갖는다.

$$\begin{aligned} \text{sim}(P1, P2) &= \frac{2 * \text{MatchLength}}{\text{Length}(P1) + \text{Length}(P2)} \\ &= \frac{2 * (3 + 8)}{16 + 18} = 0.647 \text{ 이다.} \end{aligned}$$

[김2003]에서 제시한 프로그램 유사도 평가 알고리즘을 이용하면 다양한 방법으로 복제한 프로그램을 평가할 수 있다.

### 4.2.1 단순한 스타일 변화가 일어날 경우의 유사도 분석

일반적으로 프로그램을 복제한 제출지는 다양한 방법으로 원본을 편집하는 경향이 있다. 이 중 가장 쉬운 방법으로 다음과 같은 프로그램의 요소를 바꾼다.

- 1) 정확히 원본 프로그램을 복제하기
- 2) 단순 설명문 변경하기
- 3) 공백(White Space)이나 형식 변환하기
- 4) 변수나 함수 이름을 단순히 변환한 경우의 유사도

[김2003]의 프로그램 유사도 평가 알고리즘은 위에서 제시한 4가지의 복제 유형에 대해서 쉽게 판정할 수 있다. 즉, 원본 프로그램의 스타일이나 들여쓰기, 변수 이름 등이 변경되었을지라도 생성된 AST는 변경되지 않으므로 다음과 같이 유사도 1, 즉 “완전 복제”를 판정한다. 여기서 A, B는 해당되는 프로그램의 노드 스트링이다.

A : {N<sub>1</sub>, N<sub>2</sub>, N<sub>3</sub>, ..., N<sub>i</sub>, N<sub>i+1</sub>, ..., N<sub>n</sub>}

B : {N<sub>1</sub>, N<sub>2</sub>, N<sub>3</sub>, ..., N<sub>i</sub>, N<sub>i+1</sub>, ..., N<sub>n</sub>}

$$\text{Sim}(A, B) = \frac{2 * \text{Length}(A)}{\text{Length}(A) + \text{Length}(B)} = 1$$

### 4.2.2 불필요한 문장이나 변수를 삽입하였을 경우의 유사도 분석

이 경우의 유사도는 특정한 프로그램에 다음과 같이 노드가 삽입되어 나타난다. 따라서 해당되는 노드 스트링 A, B와 유사도는 다음과 같다. 여기서 A, B는 해당되는 프로그램의 노드 스트링이다.

A : {N<sub>1</sub>, N<sub>2</sub>, N<sub>3</sub>, ..., N<sub>i</sub>, N<sub>i+1</sub>, ..., N<sub>n</sub>}

B : {N<sub>1</sub>, N<sub>2</sub>, N<sub>3</sub>, ..., N<sub>i</sub>, N<sub>j</sub>, N<sub>j+1</sub>, ..., N<sub>j+m</sub>, N<sub>i+1</sub>, ..., N<sub>n</sub>}

$$\text{Sim}(A, B) = \frac{2 * \text{Length}(A)}{\text{Length}(A) + \text{Length}(B)}$$

#### 4.2.3 똑같은 구조를 갖는 제어 구조로 바꾸는 경우의 유사도 분석

다음과 같은 2가지 복제 유형 항목은 본 시스템에 서는 아주 강한 유사도 값을 반환한다. 즉, for, while, do while 등의 문장을 바꾸어 같은 효과를 나타낼 때 의 복제 유형과 프로그램 결과에 상관없는 타입을 변 환시켰을 경우이다. 만일 이러한 복제가 일어났을 경 우에는 다음과 같은 노드 스트링 A, B가 생성되어 유 사도 값이 평가된다.

A : {N<sub>1</sub>, N<sub>2</sub>, N<sub>3</sub>, ..., N<sub>i-1</sub>, N<sub>i</sub>, N<sub>i+1</sub>, ..., N<sub>i+m</sub>, N<sub>i+1</sub>, ..., N<sub>n</sub>}

B : {N<sub>1</sub>, N<sub>2</sub>, N<sub>3</sub>, ..., N<sub>i-1</sub>, N<sub>j</sub>, N<sub>j+1</sub>, ..., N<sub>i+m</sub>, N<sub>i+1</sub>, ..., N<sub>n</sub>}

$$\text{Sim}(A, B) = 2 \times \frac{\text{Length}(N_1, N_2, N_3, \dots, N_{i-1}) + \text{Length}(N_{i+m+1}, \dots, N_n)}{\text{Length}(A) + \text{Length}(B)}$$

#### 4.2.4 문장이나 함수, 변수 선언문 등과 같은 경우의 위치 바꾸기를 수행하였을 경우의 유사도 분석

이 경우는 다음과 같은 4가지 유형으로 프로그램을 복제하였을 경우에 해당된다.

- 1) 코드 블록 재배치하기
- 2) 코드 블록 내에서 문장들 재배치하기
- 3) 수식에서 오퍼랜드/오퍼레이터의 순서 바꾸기
- 4) 함수의 위치 바꾸기

이와 같은 복제가 일어났을 경우, 본 시스템에서는 다음과 같이 유사도를 평가한다.

A : {N<sub>1</sub>, N<sub>2</sub>, N<sub>3</sub>, ..., N<sub>i-1</sub>, N<sub>i</sub>, N<sub>i+1</sub>, ..., N<sub>i+j</sub>, N<sub>i+j+1</sub>, N<sub>i+j+2</sub>, ..., N<sub>i+j+m</sub>, N<sub>i+j+m+1</sub>, ..., N<sub>n</sub>}

B : {N<sub>1</sub>, N<sub>2</sub>, N<sub>3</sub>, ..., N<sub>i-1</sub>, N<sub>i+j+1</sub>, N<sub>i+j+2</sub>, ..., N<sub>i+j+m</sub>, N<sub>i</sub>, N<sub>i+1</sub>, ..., N<sub>i+j</sub>, N<sub>i+j+m</sub>, ..., N<sub>n</sub>}

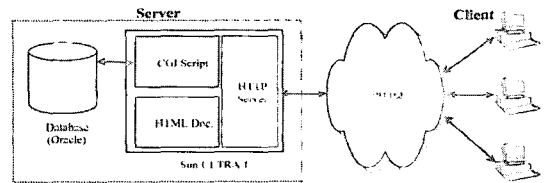
$$\text{Sim}(A, B) = 2 \times \frac{\text{Length}(N_1, N_2, N_3, \dots, N_{i-1}) + \text{Length}(N_{i+j+1}, \dots, N_{i+j+m}) + \text{Length}(N_i, \dots, N_{i+1}, N_n)}{\text{Length}(A) + \text{Length}(B)} = 1$$

즉, 이 수식의 의미는 두 프로그램이 구조적으로 똑같고, 함수 위치, 문장 순서가 뒤바뀌었을 경우에도 두 프로그램은 똑같음을 판명할 수 있다는 것을 의미 한다.

## 5. 개발 환경 및 평가

### 5.1 개발 환경

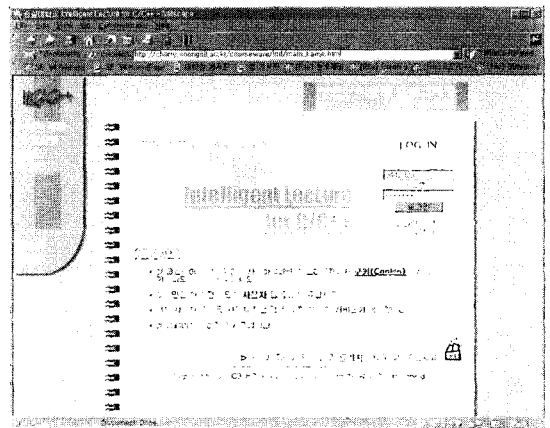
코스웨어 서버는 Sun ULTRAI을 이용하였으며, Java 언어로 구현되었다. 또한 프로그램 트리를 구성하기 위하여 구문 분석과 어휘 분석이 가능한 JLex[Berk, Lin98]와 Java CUP[Hudson] 유틸리티를 사용하였다. 본 시스템의 개발환경은 다음 그림 3과 같다.



(그림 3) 코스웨어 시스템의 개발환경

### 5.2 코스웨어 시스템의 구성 화면

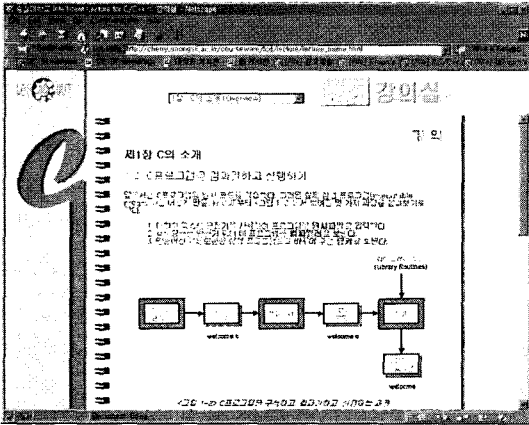
지능형 코스웨어 시스템의 초기화면은 다음 그림 4 와 같다.



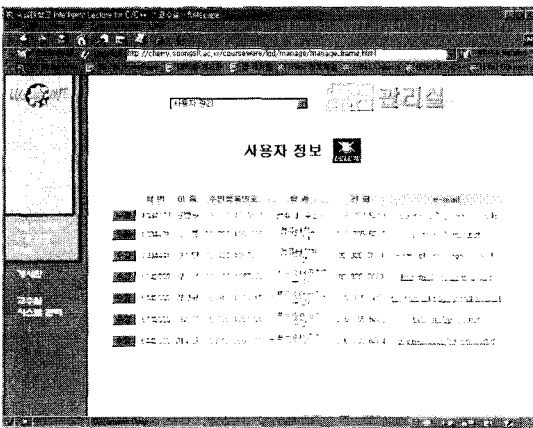
(그림 4) 코스웨어 시스템의 초기화면

과제물 내용을 보거나 제출하기 위해서는 수강생들은 먼저 로그인을 해야 한다. 제출된 과제물은 웹과 연동되어 있는 데이터베이스에 제출한 시간과 같이 입력된다.

다음 그림 5와 6은 각각 학생을 위한 강의실과 관리자를 위한 화면을 보여준다.



(그림 5) 코스웨어 시스템의 강의실



(그림 6) 코스웨어 시스템의 관리실

### 5.3 프로그램의 유사도 평가

실험을 위하여 다음 표 1, 2와 같은 최소 공배수를 구하는 두개의 프로그램이 있다고 가정하자. 표 2는 1에서 스타일이 변경되거나, 유사한 제어문으로 변경된 것이다.

(표 1) 원본 프로그램(source.C)

```
#include <stdio.h>
void main(void) {
    long result; int index=0;
    int n1, n2, m1, m2, divs[100], lnum, i, flag;
    printf("Input 2 numbers for calculating GCM...\n");
    scanf("%d %d", &n1, &n2); m1= n1;
    m2 = n2;
    while(1) {
        lnum = (m1>=m2?m1:m2); flag = 0;
        for(i=2; i<=lnum; i++) {
            if(m1%i==0 && m2%i==0) {
                m1/=i; m2/=i;
                divs[index++]=i; flag=1; break; }
        }
        if(flag==0) break;
    }
    result = m1 * m2;
    for(i=0; i<index; i++) result *= divs[i];
    printf("LCM of %d and %d is %d.\n", n1, n2, result);
}
```

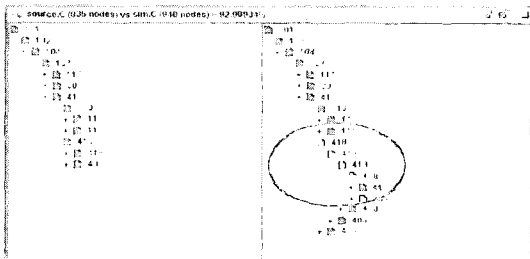
(표 2) 유사한 프로그램(sim.C)

```
#include <stdio.h>
void main(void) { /* 스타일 변화 */
    long result; int index=0; /* 문장 위치 바꿈 */
    int n1, n2, m1, m2, divs[100], lnum, i, flag;
    printf("Input 2 numbers for calculating GCM...\n");
    scanf("%d %d", &n1, &n2); m1= n1; m2 = n2;
    for(;;) { /* 제어 구조 변환 while ==> for */
        flag = 0; lnum = (m1>=m2?m1:m2); /* 문장 바꾸기 */
        for(i=2; i<=lnum; i++) { /* 스타일 바꾸기 */
            if(m1%i==0 && m2%i==0) {
                m1/=i; m2/=i; divs[index++]=i; flag=1; break; }
        }
        if(flag==0) break; }
    result = m2 * m1; /* 오퍼랜드 바꾸기 */
    result = m2 * m1; /* 중복 코드 추가 */
    result = m2 * m1; /* 중복 코드 추가 */
    result = m2 * m1; /* 중복 코드 추가 */
    result = m2 * m1; /* 중복 코드 추가 */
    for(i=0; i<index; i++) result *= divs[i];
    printf("LCM of %d and %d is %d.\n", n1, n2, result);
}
```

두 프로그램의 비교 결과는 다음 그림 7과 같다. 그림 7에서처럼 두 프로그램의 유사도 값은 92.98934



(그림 7) source.C와 sim.C 프로그램의 유사도 평가



(그림 8) source.C와 sim.C 프로그램 구문트리

값이다. 실험에서와 같이 같은 프로그램인데도 유사도 값이 적게 나온 이유는 위의 표 2에서처럼 중복 코드(dummy code)가 많이 들어갔기 때문이다.

그림 8은 위에서 비교한 source.C 파일과 sim.C파일을 구문트리틀을 보여준다. 그림에서 정수는 각각 구문트리의 노드를 나타낸다. 그림 8은 두 프로그램에서 다른 부분을 잘 보여주고 있다.

즉, 두 프로그램에서 일치되는 코드는 붉은(■)표시로 나타나며, 흰색(□) 표시는 두 프로그램에서 일치되지 않는 부분을 나타내고 있다. 따라서 본 시스템은 타 기법과 마찬가지로 중복 코드가 많이 들어가면 갈수록 유사도 값이 현저하게 떨어진다. 따라서 본 시스템은 중복 코드(dummy code)가 많이 있다면 본 시스템의 유사도는 현저하게 떨어질 수도 있다.

## 6. 결론

본 논문에서는 인터넷을 통해 C/C++ 언어 프로그

래밍 과정의 사이버강의를 지원하는 코스웨어와 학생들의 과제를 효율적으로 평가할 수 있는 평가도구로 이루어진 시스템 모델을 제시했다. 코스웨어 시스템은 사이버 강의뿐만 아니라 기존 코스웨어의 고정적이고, 일반적인 교육방식에서 벗어나 자신의 능력에 맞는 학습 진척 상황을 파악할 수가 있으며, 과제의 불법 복제에 관한 평가도 수행된다. 본 시스템에서 이용한 유사도 측정방법은 기존의 텍스트 기반 유사도 측정과는 달리 추상구문트리(AST)를 이용함으로써 보다 효율적이고, 정확한 측정을 할 수 있다는 것을 보여준다. 또한 추상 구문트리틀을 비교하기 위하여 [김2003]에서 제시한 프로그램 유사도 평가 알고리즘을 이용하였으며, 그 결과 다양한 복제 유형에 따른 복제 검사를 할 수 있다는 것을 보여주었다.

본 논문의 평가 부분에서는 실제로 사이버 교육에 필요한 프로그래밍 강의실 및 관리실을 보여주었으며, 두 프로그램 간의 유사도 평가 화면을 보여주었다. 본 시스템을 활용하면 가상대학(Cyber University), 주문형 강의(Lecture on Demand), 전자도서관 등에서 다양한 형태로 이용할 수 있으며, 컴퓨터 교육 분야에 많은 영향을 줄 것으로 기대 된다.

## 참고문헌

- [김2003] 김영철, 김성근, 염세훈, 최중명, 유재우, “구문트리 비교를 통한 프로그램 유형 복제 검사”, 한국 정보과학회 논문지, 제30권 8호, 2003.
- [Kelley84] A. Kelley, I. Pohl, *A Book on C: Introduction to programming in C*, The Benjamin/Cummings Publishing Company, Inc, 1984.
- [Berk] E. Berk, “Jlex:A Lexical Analyzer Generator for Java TM”, available at <http://www.cs.princeton.edu/~appel/modern/java/JLex/>.
- [Foxley93] E. Foxley, “Question/answer exercises in Ceilidh, LTR Report”, Computer Science Dept, Nottingham University. 1993.
- [Lin98] J. Lin, JLex Tutorial, available at <http://bmr.berkeley.edu/courseware/cs164/spring98/proj/jlex/tutorial.html>.
- [Robert96] W. Robert, Sebesta, *concepts of : Program*

*ming Languages*, Third Edition, Addison Wesley, 1996.

[Benford92] S. Benford, Edmund Burke, Eric Foxley, "Courseware to support the teaching of programming, TLTP Conference", University of Kent at Canterbury.

1992. pp. 158~166.

[Hudson] S. E. Hudson, "CUP Parser Generator for Java", available at <http://www.cs.princeton.edu/~appel/modern/java/CUP/>

## ● 저 자 소 개 ●



### 김 영 철

2003년 숭실대학교 컴퓨터학과 박사

2001년~현재 : (주)뉴스텍시스템즈 이사

2002년~현재 : 명지전문대학 겸임교수

관심분야 : 프로그래밍 언어, 컴파일러, 원격교육, 네트워크, 웹프로그래밍 언어

### 최 종 명

1992년 숭실대학교 전자계산학과 학사

1996년 숭실대학교 전자계산학과 석사

2003년 숭실대학교 컴퓨터학과 박사

관심분야 : 멀티패러다임 언어, XML, 컴포넌트, 유비쿼터스 컴퓨팅