알고리즘 자동평가 시스템의 개발 및 적용 : 프로그래밍 학습 효과 분석

장원영† · 김성식††

요 약

학습자가 주어진 문제를 해결하는 알고리즘을 작성한 후 그것이 정확한지, 그리고 시간 효율적인지를 확인할 수 있는 알고리즘 자동평가 시스템에 대한 연구가 최근 들어 활발히 진행 중이다. 그러나 기존에 연구되었던 시스템은 대부분 프로그래밍 콘테스트를 위한 Online Judge 방식으로 본 연구에서는 교수·학습 기능을 강화한 클라이언트-서버 기반의 시스템을 개발하였다. 특히, 문제해결력 증진을 위한 교수 학습 설계 모델 CRESST을 토대로 학습자의 메타인지와 동기가 활성화되도록 설계하였으며, 알고리즘 자동평가 시스템의 구성요소인 문제, 채점데이터 세트, 자동평가 프로그램, 사용자서비스 환경 등의 전체 시스템을 구현하였다. 본 시스템의 프로그래밍 학습 효과를 분석하기 위해 초·중·고 학생 39명에 대해서 비동질 통제집단 사전사후측정 실험을 실시하였고, 사후검사에 대한 독립표본 T-검정 결과, 실험집단(18명)의 평균점수가 통제집단(21명)보다 유의미하게 높은 것으로 확인되었다. 이것은 본 시스템을 사용한 교수·학습 방법이 전통적인 교수·학습 방법에 비해 프로그래밍 학습에 더 효과적임을 의미한다.

주제어 : 알고리즘 자동평가, Online Judge, 프로그래밍 학습, CRESST

Development and application of algorithm judging system : analysis of effects on programming learning

Won-Young Chang† · Seong-Sik Kim††

ABSTRACT

Many studies on algorithm judging system which verifies the correctness and the time efficiency of your program have been underway recently, most of which are on an online judging system focused on programming contests. However this study is mainly about development and application of the judging system based on client-server. Especially, we designed to promote metacognition and motivation which are emphasized in CRESST model, and implemented the total system that consists of the problem, data set, validation program, and user service environments. We applied our system to elementary, middle, and high school students, and We noticed a significant difference of average score between the experimental and control group in posttest and concluded that the teaching method using our system gave the bigger positive effects on programming learning.

Keywords : Algorithm Judging, Online Judge, Programming Learning, CRESST

† 정 회 원: 한국교원대학교 대학원 컴퓨터교육전공 박사과정
†† 종신회원: 한국교원대학교 컴퓨터교육과 교수(교신저자)
논문접수: 2014년 6월 23일, 심사완료: 2014년 7월 21일, 게재확정: 2014년 7월 25일
* 이 논문은 2013년 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2013S1A5A2A01018796)

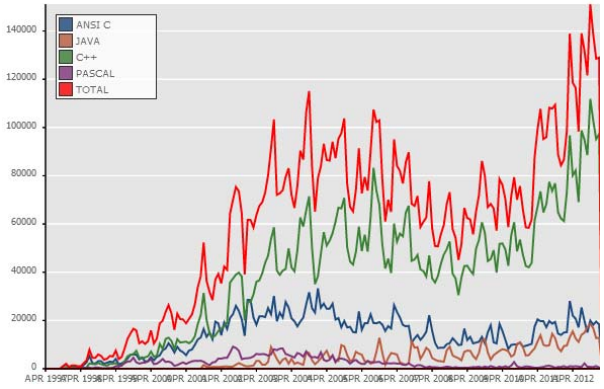
1. 서론

주어진 문제를 컴퓨터 프로그래밍을 통해 창의적이고 효율적으로 해결하는 능력은 지식 정보화 사회를 살아가는 구성원들이 갖추어야 할 필수 능력 중 하나다[1]. 이러한 '정보과학적 문제해결력'은 제시된 문제를 해결하기 위한 자료구조의 선정과 알고리즘의 구현이 선행되어야 한다. 이러한 능력들은 단기간의 교육을 통해 쉽게 얻을 수 없으며, 수년간의 경험과 지식이 쌓여 그 '임계값'을 초과했을 때 자연스럽게 발현되는 것이다. 따라서 이러한 창의적인 문제 해결력을 함양하기 위한 교육은 초등학교 때부터 체계적인 방법과 위계적인 내용으로 이루어져야 한다. 한편, 프로그래밍 수업 장면에서 학습자의 프로그래밍 코드를 평가하고 피드백을 부여하는 과정은 교수자에게 많은 부담으로 작용한다. 해당 수업에 참여하는 학습자들이 많고 학생들에게 제시한 프로그래밍 과제가 많다면 더욱 그러하다[2].

프로그래밍 언어의 문법을 학습한 후 다양한 문제를 창의적으로 해결해 보고, 주어진 문제에 대해 자신이 만든 프로그램이 올바른지, 그리고 얼마나 시간 효율적인지를 확인할 수 있는 자동평가 시스템이 있다면 교수자나 학습자에게 매우 유용하게 활용될 수 있을 것이다[3].

최근 주어진 문제에 대한 알고리즘을 채점해 주는 '자동평가 시스템'에 대한 연구가 활발히 진행되고 있다. 웹 서비스로 문제가 제시되면, 사용자는 이 문제를 정확하고 효율적으로 해결할 수 있는 프로그램을 작성한다. 이 프로그램의 소스를 복사한 후 웹 서비스 상에 제출하면 서버는 이 소스를 컴파일한 후 미리 만들어 놓은 채점 데이터와 비교하여 알고리즘의 정확성과 시간효율성을 평가한다. 이러한 방식의 알고리즘 자동평가 시스템을 일반적으로 'Online Judge 시스템'이라고 부른다[4][5].

<그림 1>의 UVa Online Judge 시스템은 ACM-ICPC와 같은 세계적인 프로그래밍 콘테스트 대회의 준비를 위한 사이트다. 스페인의 University of Valladolid에서 1997년부터 2013년까지 11,199,721의 자동 채점 횟수가 기록될 만큼 세계적으로 매우 인기 있는 시스템 중 하나다.



<그림 1> 국외의 대표적인 프로그래밍 자동평가 시스템 UVa Online Judge의 사용 통계

이러한 '알고리즘 자동평가 시스템'은 학습자가 웹 상에서 제출한 프로그래밍 소스 코드를 평가하고 즉각적인 피드백을 주어 학습자로 하여금 반성적 사고 능력과 고등 인지기술을 향상시킬 수 있는 기회를 부여한다. 또한 교수자에게는 학습자의 프로그래밍 코드를 평가함에 있어 객관성과 신뢰성을 부여한다[6][7].

그러나 Online Judge 방식의 시스템은 자신이 작성한 소스코드를 복사하여 웹페이지에 붙여넣는 방식이기 때문에 코딩과 채점의 전체 과정에서 불필요한 단계들이 포함되고, 채점데이터 세트가 공개되지 않기 때문에 오답처리의 원인을 확인할 수 없다[8].

본 연구에서는 학습자의 메타인지와 동기를 촉진하도록 설계된 클라이언트-서버 방식의 알고리즘 자동평가 시스템을 개발하고 이를 적용한 후 프로그래밍 학습 효과를 분석하였다.

2. 이론적 배경

2.1 알고리즘 자동평가 시스템

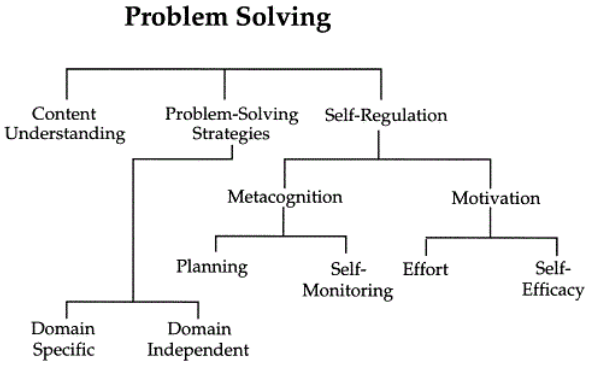
알고리즘 자동평가 시스템은 학습자가 프로그래밍 언어를 이용하여 주어진 문제에 대한 풀이를 작성하여 제출하면 제출된 소스 코드는 서버에서 파일로 저장되고 이를 컴파일 하여 실행파일로 만든다[7]. 실행프로그램은 문항마다 미리 준비된 입력, 출력 파일의 수에 따라 몇 차례 실행을 반복한다. 매 실행시마다 미리 준비된 데이터

가 입력되어 프로그램이 실행되고 그 결과 출력되는 값과 준비된 정답데이터의 값을 비교한다. 만약 두 데이터의 값이 같고, 시간제한, 메모리 제한을 위반하지 않았다면 해당 입력데이터에 대해서는 프로그램이 제대로 동작한 것으로 받아들여진다. 만약 프로그램이 실행되는 동안에 어떠한 오류가 발견되었다면 시스템은 Wrong Answer, Time Limit Exceed, Memory Limit Exceed, Runtime Error, Compile Error 중에서 해당되는 오류 메시지로 피드백 정보를 제공한다[6]. 그러나 지금까지 연구되었던 자동평가 시스템[6][7]은 Online Judge 방식이다. 이 방식은 대부분 프로그래밍 콘테스트 대비용으로 활용되며, 소스의 컴파일, 실행, 자동채점 등이 모두 Online 상에서 진행되기 때문에 사용자 편이성 등이 우수하지만, 채점데이터 세트가 공개되지 않아 사용자가 입력값의 특성과 정답 데이터를 확인하기 어렵다는 단점이 있다.

2.2 CRESST 모델

'문제 해결 능력'을 신장하기 위한 교수·학습 설계를 위해서는 그 하위 요소들을 확인할 수 있어야 한다. 문제해결 능력의 하위 요소에 대한 연구는 Herl 등(1999)의 CRESST(National Center for Research on Evaluation, Standards, and Student Testing) 모델이 대표적이다[8]. Herl 등은 CRESST의 연구를 토대로 문제해결력 모델을 <그림 2>와 같이 제시하고 있다. 문제 해결을 위해서는 3가지 구성요소가 필요한데, 첫 번째로 문제와 관련된 영역의 개념 지식을 습득하는 학습 내용의 이해(Content Understanding), 두 번째로 문제해결 과정을 표현하고 전략을 구성하는 문제해결전략(Problem-Solving Strategies), 마지막으로 메타인지(Metacognition)와 동기(Motivation)로 구성된 자기조절(Self-Regulation)이 그것이다. 여기서 메타인지는 계획(Planning)과 자기점검(Self-Monitoring)으로, 동기는 노력(Effort)과 자기 효능감(Self-Efficacy)으로 세분화된다[8][9].

이러한 CRESST 모델을 정보과학적 문제해결 과정에 적용해 볼 수 있다. 주어진 문제를 프로그래밍으로 해결하기 위해서는 해당 문제를 이해해야 한다(Content Understanding). 그리고 이 문제를 효율적으로 해결하기 위한 알고리즘을 설계해야 한다(Problem-Solving Strategies). 이때, 주어진 문제에 따라서 해당 문제만 해결할 수 있는 알고리즘을 적용할 수도 있으며(Domain Specific), 해당 문제뿐만 아니라 포괄적인 문제를 해결하는 범용적인 문제해결기법을 적용할 수도 있다(Domain Independent). 이러한 문제해결기법에는 분할정복법, 탐욕적인 방법, 동적 계획법, 백트래킹 등이 있다. 또한, 학습자는 프로그래밍을 통해 문제를 해결하는 과정에서 오류 확인 및 검증, 그리고 디버깅 단계를 거치면서 반성적 사고와 같은 고등인지기술을 향상 시킬 수 있고[10]. 메타인지를 활성화할 수 있다[11].



<그림 2> CRESST 모델

이렇듯 이 모델을 통해 창의적 문제해결 능력이 정보과학 분야에서 주어진 문제의 알고리즘을 고안하고 프로그래밍으로 구현하는 일련의 과정과 매우 유사함을 알 수 있으며, CRESST 모델을 통해 프로그래밍 교육에서 문제해결력을 함양하기 위해 필요한 하위 요소들을 확인할 수 있다.

3. 알고리즘 자동평가 시스템의 구성요소

'알고리즘 자동평가 시스템'은 일반적으로 <표 1>의 4가지 구성요소로 이루어져 있다.

3.1 문제와 채점데이터 세트

알고리즘 자동평가 시스템을 개발하기 위해서

는 먼저 알고리즘의 정확성과 시간효율성을 평가할 수 있는 문제를 제작하고 그에 상응하는 채점 데이터 세트를 만들어야 하며, '자동평가'를 위해서는 문제와 채점데이터 세트가 매우 형식적이고 명확해야 한다.

<표 1> 시스템 구성요소

구성요소	관련 설명
① 문제	학습자가 해결해야 할 알고리즘 문제
② 채점데이터 세트	해당 문제에 대한 알고리즘의 정확성과 시간효율성을 측정하기 위한 일련의 채점데이터 세트(입력데이터와 정답데이터의 쌍으로 구성)
③ 자동평가 프로그램	학습자가 제출한 소스코드를 컴파일한 후 생성된 실행파일에 채점데이터를 입력하고 출력값을 정답데이터와 비교하며, 실행파일의 소요시간을 측정하는 프로그램
④ 사용자 서비스 환경	문제의 제시, 자동 채점 결과, 랭킹 산출 등의 서비스를 제공하는 사용자 환경

3.1.1 문제

알고리즘 자동평가 시스템에서 사용하는 문제는 기본적으로 <표 2>의 구성요소로 이루어져 있으며, 각 구성요소들은 알고리즘의 기본조건들이 만족되도록 설계되어야 한다.

3.1.2 채점데이터 세트

채점데이터 세트는 입력데이터와 정답데이터의 쌍으로 구성된 10개에서 20개 정도의 데이터 세트로서 입력데이터와 정답데이터 사이의 종적인 구성을 통해 알고리즘의 정확성을 평가하며, 입력데이터의 범위가 점점 증가하는 횡적인 구성으로 알고리즘의 시간효율성을 평가한다. 그리고 이러한 평가요소는 <표 2>의 문제 구성요소에서 ② 핵심요구사항, ③제한시간, ④입력데이터의 범위와 밀접하게 관련되어 있다.

<표 2> 문제 구성요소

구성요소	알고리즘의 기본조건		
① 배경 설명			명확성
② 핵심요구사항	**정확성** (유효성)		
③ 제한시간	**시간 효율성**	유한성	
④ 입력데이터의 범위		입력	
④ 입력과 출력의 예		출력	

<그림 3>은 20개의 입력데이터와 정답데이터의 쌍으로 구성된 채점데이터 세트의 예를 보여준다. 여기서 정답데이터란 문제에서 제시한 범위 내의 입력데이터에 대해서 정확한 알고리즘으로 문제를 해결했을 때의 출력값을 의미한다. 시스템은 사용자가 제출한 프로그램에 입력데이터를 주

<그림 3> 채점데이터 세트의 구성

고, 출력값이 정답데이터와 일치하는지를 확인함으로써 알고리즘의 정확성을 평가할 수 있다. 따라서, 입력데이터를 제작할 때는 문제에서 제시한 입력 범위 내의 임의의 값에 대해서 사용자의 프로그램이 올바른 값을 출력하는지 확인할 수 있도록 입력 데이터의 특성, 즉, 극한값 등을 고려해야 한다. 또한 알고리즘의 시간효율성을 평가할 수 있도록 입력 데이터의 범위를 횡적으로 점점 증가시켜야 한다. 이때, 비효율적인 알고리즘의 경우 제한 시간 내에 값을 출력할 수 없을 만큼의 범위이어야 하므로 문제를 제작할 때 이 점을 고려하여 제한시간과 더불어 입력데이터의 최대 범위를 결정해야 한다[12].

문제와 채점데이터 세트를 어떻게 구성하느냐에 따라 알고리즘 자동평가 시스템의 성능이 좌우된다. 즉, 정확성을 측정하는 채점 데이터 세트의 종적 구성과 더불어 알고리즘의 시간효율성을 충분히 측정할 수 있도록 정교한 횡적 구성이 요구되며, 이 두 가지 평가 요소를 고려한 문제를 제작해야 하는 것이다.

3.2 자동평가 프로그램

자동평가 프로그램은 학습자가 제출한 소스코드를 토대로 생성한 실행파일을 실행시킨 후 채점데이터 세트의 입력데이터를 순서대로 투입하면서 실행파일이 출력한 값과 정답데이터의 일치여부를 확인한다. 이때, 입력데이터의 투입시간으로부터 실행파일이 결과값을 출력할 때까지의 소요시간이 문제에서 제시한 제한시간을 초과할 경우 실행파일의 프로세스를 강제 종료시킨 후 제한시간 초과 메시지를 표시한다(알고리즘의 시간효율성). 제한시간 이내에 결과값이 출력되는 경우에는 이 값과 정답데이터를 비교해서 일치여부를 확인하고 그 결과를 사용자에게 알려준다(알고리즘의 정확성). 자동평가 프로그램에 컴파일러를 내장시키면 학습자의 소스코드를 번역하는 역할까지 수행할 수 있다.

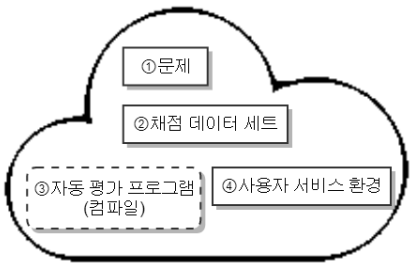
3.3 사용자 서비스 환경

사용자 서비스 환경은 학습자에게 알고리즘 문제를 제시하고 자동평가 프로그램의 채점결과와 더불어 학습자가 지금까지 제출한 소스코드를 누적하여 보여준다. 또한 각 문제별 채점현황과 더불어 채점결과를 토대로 산출한 점수비율을 제공함으로써 학습자는 문제의 난이도를 간접적으로 파악할 수 있다. 또한, 현재까지 자신이 채점한 문제들의 점수를 토대로 사용자 간의 실시간 랭킹을 산출함으로써 학습자는 많은 문제를 해결하고자 하는 동기를 갖게 된다. 사용자 서비스 환경은 대부분 웹으로 구현하는 것이 일반적이다.

4. 알고리즘 자동평가 시스템의 개발

4.1 클라이언트-서버 기반 시스템

Online Judge 방식은 <그림 4>와 같이 구성요소가 모두 온라인상에서 실행되는 방식으로 사용자가 제출한 소스코드가 서버에서 컴파일 되고 실행된다. 최근에는 HustoJ라는 오픈소스 기반의 도구가 공개되어 Online Judge 방식의 시스템을 쉽게 구축, 운영할 수 있다. HustoJ는 리눅스 상에서 아파치, MySQL과 연동하여 <표 1>에서 ③, ④의 구성요소를 제공한다[13].



<그림 4> Online Judge 방식

Online Judge 방식은 주로 프로그래밍 콘테스트 대비용으로 개발되어 활용되기 때문에 채점데이터 세트가 공개되지 않는다. 이러한 특징은 사용자가 채점결과에 오답이 있을 경우 그 원인을 찾기 어려운 문제점이 있다. 또한, 통합개발환경에서 작성한 소스를 복사해서 다시 웹에 붙여넣는 과정을 반복해야 하므로 상당히 번거로울 수

있다. 더불어 컴파일과 실행이 모두 서버 상에서 이루어지므로 다수의 사용자가 동시에 채점하면 서버에 상당한 부하를 줄 수 있다.

<그림 5> 클라이언트-서버 방식

본 연구에서는 이러한 문제점을 인식하고 클라이언트-서버 기반의 자동평가 시스템을 개발하였다. 이 방식의 특징은 <그림 5>와 같이 자동평가 프로그램이 클라이언트에서 실행되기 때문에 사용자의 통합개발환경과 연동되어 코딩과 함께 곧바로 자동평가를 수행할 수 있다는 점이다. 이때 자동평가 프로그램은 컴파일을 수행하지 않는다. 즉, 이미 사용자가 통합개발환경을 통해 실행파일을 생성했음을 가정하고 채점을 수행하므로 소스코드의 빌드 과정은 전적으로 사용자에게 위임한다. 또한 채점데이터 세트가 사용자의 컴퓨터에 다운로드된 후 자동평가가 이루어지기 때문에 오답으로 처리된 입력데이터와 정답데이터를 사용자가 직접 확인할 수 있다. 이렇게 함으로써 채점결과에 따른 반성적 사고를 활성화할 수 있으며, CRESST 모델에서 자기통제의 '메타인지'를 촉진할 수 있다.

4.2 문제와 채점데이터 세트

C언어를 학습한 후 쉽게 해결할 수 있는 기초문제 48개, 자료구조와 알고리즘에 관한 문제 17개, 그리고 각 문제에 대한 채점데이터 세트를 개발하였다. 또한 한국정보올림피아드의 지역본선 문제와 전국본선 문제도 함께 탑재하여 정보올림피아드 대회에 대비한 학습도 가능하도록 하였다. 다음은 기초문제의 예시다.

<표 3> 기초문제 예시

다이어트 운동을 시작한 정원이는 자주 체중계에 올라가서 자신의 몸무게를 잰다. 그러던 어느날 여느 때처럼 체중계에 오른 정원이는 자신의 몸무게를 확인하고는 깜짝 놀라고 말았다. 어제보다 무려 10Kg가 더 나가는 것이었다. 최근에 몸무게가 거의 변함이 없었고 다이어트 운동도 거르지 않고 계속했기 때문에 전혀 예상하지 못한 일이었다.

이상하다고 느낀 정원이는 문제의 체중계로 가족들의 몸무게를 재보았다. 그랬더니 모두 평상시보다 10kg이 더 무겁게 나온다는 사실을 확인하였고, 체중계가 고장이 났다는 결론을 내렸다.

정원이의 고장난 체중계로 잰 몸무게가 자연수로 주어질 때 몸무게를 정확히 다시 계산하는 프로그램을 작성하시오.

프로그램의 이름은 문제코드와 동일하고 실행시간은 0.5초를 넘을 수 없다. 부분 점수는 없다.

입력 형식
입력 파일의 이름은 input.txt로 한다. 입력 파일은 한 개의 줄로 이루어진다. 첫 번째 줄에 고장난 체중계로 잰 몸무게가 자연수로 주어진다. 몸무게의 범위는 30 ~ 150 이다.

출력 형식
출력 파일의 이름은 output.txt로 한다. 첫 번째 줄에 다시 계산한 정확한 몸무게를 출력한다.

입력과 출력의 예

입력(INPUT.TXT)

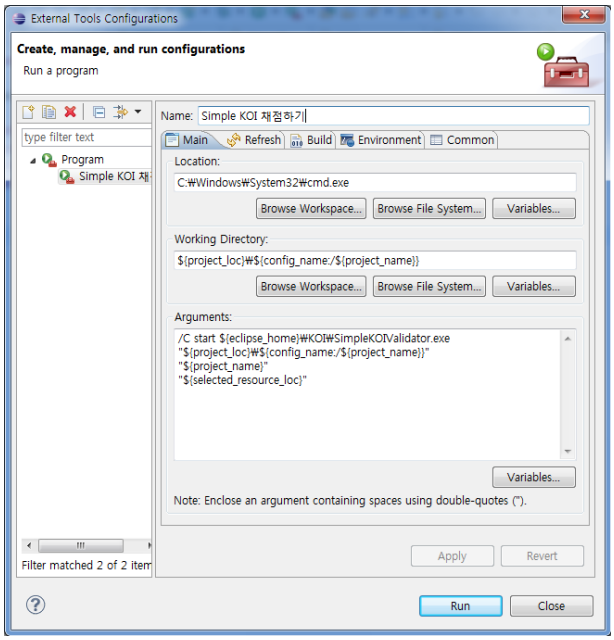
62

출력(OUTPUT.TXT)

52

4.3 자동평가 프로그램

자동평가 프로그램은 사용자의 컴퓨터에 설치되며, 통합개발환경의 외부도구로 등록할 수 있기 때문에 사용자가 통합개발환경에서 코딩을 하면서 곧바로 자신의 소스코드에 대해 채점을 수행할 수 있다. 마이크로소프트사의 Visual 계열, Eclipse, CodeBlocks, Dev-C++, CodeLite 등의 통합개발환경과 더불어 EditPlus, UltraEdit와 같은 텍스트 편집기와도 연동하여 사용할 수 있다. <그림 6>은 자동평가 프로그램을 Eclipse의 외부도구로 설정하는 화면이다.



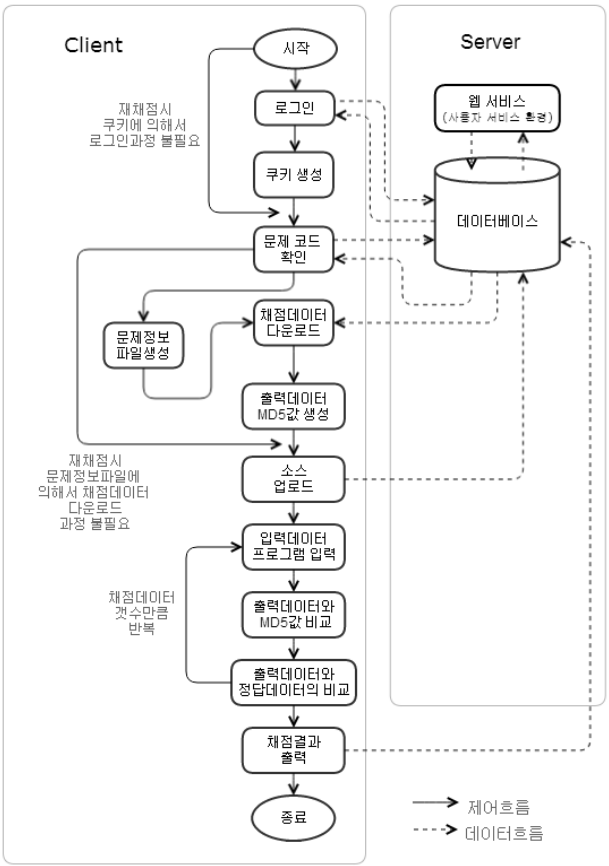
<그림 6> Eclipse의 외부도구 설정 화면

<그림 7>은 클라이언트-서버 방식으로 동작하는 자동평가 프로그램의 전체적인 실행 과정을 그림으로 나타낸 것이다. 자동평가 프로그램의 세부적인 실행 과정은 다음과 같다.

1) 아이디와 비밀번호를 이용하여 로그인을 시도하면 프로그램은 서버의 데이터베이스에서 로그인 정보를 확인한다.

2) 로그인 정보가 맞으면 로그인 쿠키를 생성한다. 이것은 다음에 다시 채점을 시도할 때 별도의 로그인 과정을 생략하기 위한 장치다.

3) 문제코드를 입력하면 채점데이터가 자동으로 사용자 컴퓨터에 다운로드된다. 이 과정에서 문제정보파일이 생성되는데 이것은 동일한 문제를 다시 채점할 때 이미 다운로드받은 데이터를 재사용하기 위한 것이다. 채점데이터 세트의 다운로드 위치는 <그림 6>과 같이 통합개발환경의 외부구설정에서 Working Directory(작업폴더)에 지정한 경로에 의해 결정되는데, 일반적으로 현재 작업중인 프로젝트에서 실행파일이 생성되는 상대경로를 나타내는 시스템 변수를 사용한다. 예를 들어, Elcipse의 경우 Working Directory에 ${project_loc}\${config_name:/${project_name}} 변수를 사용해서 상대경로를 지정할 수 있다.



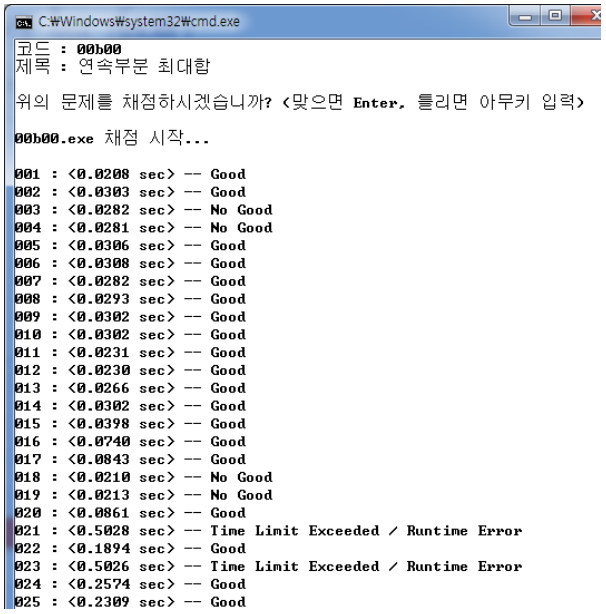
<그림 7> 자동평가 프로그램의 실행 과정

4) 한 세트의 채점데이터가 다운로드되면 개별 채점 데이터에 대해서 MD5 값을 생성한다. 이것은 채점데이터를 사용자가 수정하여 악의적인 채점을 시도하려는 것을 방지하기 위함이다.

5) 통합개발환경에서 작성한 소스의 내용을 서버의 데이터베이스에 업로드 한다.

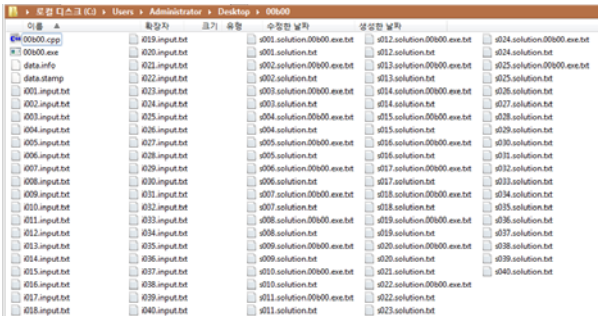
6) 먼저 한 세트 내의 첫 번째 입력데이터와 정답데이터를 4)단계의 MD5값과 비교한다. 만약 일치하지 않으면 채점데이터가 변경되었음을 메시지로 표시하고 자동 채점 프로그램은 종료한다. 일치하면 첫 번째 입력데이터를 사용자가 작성한 프로그램의 입력으로 리다이렉션하고, 사용자 프로그램은 입력받은 데이터를 기초로 출력데이터를 생성한다. 방금 생성한 출력데이터와 정답데이터를 비교한 후 해당 채점 데이터 케이스의 정답 여부를 출력한다. 일치하면 'Good', 하나라도 불일치하면 'No Good'이라고 출력한다.

7) 사용자 프로그램이 해당 문제의 제한 시간을 초과하는 경우에는 사용자 프로그램을 강제 종료한 후 'Time Limit Exceeded / Runtime Error'라고 출력한다. 사용자 프로그램이 실행 에러를 발생시키는 경우에도 동일한 메시지를 출력한다.



<그림 8> 채점 결과

8) 채점결과가 <그림 8>과 같이 No Good으로 판정되는 데이터가 있을 경우, <그림 9>에서 정답데이터와 자신의 프로그램이 만든 출력데이터를 비교함으로써 그 원인을 확인할 수 있다. 예를 들어, <그림 8>의 채점결과에서 003번째 입력데이터에 대해 No Good이 판정되었으므로 <그림 9>에서 입력데이터 i003.input.txt를 열어 입력값의 특성을 확인한 후 정답데이터 s003.solution.txt와 자신의 프로그램이 만든 출력데이터 s003.solution.00b00.exe.txt를 비교함으로써 오답으로 판정한 이유를 확인할 수 있다. <그림 9>는 자동으로 다운로드된 채점데이터 세트와 자신의 프로그램이 만든 출력데이터 목록을 나타낸 것으로 모두 txt 형식이기 때문에 메모장 등의 텍스트 편집기에서 열 수 있다.



<그림 9> 클라이언트에 다운로드된 채점데이터

9) 채점데이터 세트에 대해서 6), 7), 8)의 과정을 모두 수행한 후 평가 결과를 서버의 데이터베이스에 저장한다.

10) 데이터베이스에 저장된 평가 결과를 바탕으로 랭킹 산출, 각 문제별 평가 비율 등을 실시간으로 계산하여 사용자에게 웹 서비스 형식으로 제공한다.

4.4 사용자 서비스 환경

사용자 서비스 환경은 학습자가 <그림 10>의 알고리즘 문제와 더불어, <그림 11>에서처럼 클라이언트에서 수행한 자동 채점의 결과를 확인할 수 있도록 구현하였다.

<그림 10> 문제 정보

나의 채점 결과

문제	제목	개수	점수	소요시간	소스	확장자	IDE 연동	날짜	시간	시도
00b00	연속부분 최대합	0	0.0	0.000233	보기	c	✓	2014-02-23	20:17:04	30
00e00	입력값 출력하기	0	0.0	0.000251	보기	c	✓	2014-02-23	20:16:27	569
05ne2	곱셈	10	100.0	0.048106	보기	c	✓	2014-02-22	12:25:51	1
00e04	1부터 N까지의 합계 구하기	15	100.0	0.046865	보기	c	✓	2014-02-22	12:17:47	22
00e11	문자열 거꾸로 출력하기	15	100.0	0.008714	보기	c	✓	2014-02-16	17:35:46	8
13re1	그룹	0	0.0	0.000028	보기	cpp	✓	2014-02-07	14:12:13	1
00e47	팩토리얼 구하기	10	100.0	0.029729	보기	c	✓	2014-01-28	16:06:07	10
00b14	어디에 있나?	25	100.0	0.056654	보기	cpp	✓	2014-01-28	15:10:18	14
00b11	정렬하기	20	100.0	0.039622	보기	cpp	✓	2014-01-28	11:45:30	12
00e32	암호의 해독	30	100.0	0.029047	보기	c	✓	2014-01-28	09:59:58	1
00e15	수치 회문 판별	15	100.0	0.027974	보기	c	✓	2014-01-28	09:38:54	3
00e29	수학신동이 되는 마법증이	15	100.0	0.027504	보기	c	✓	2014-01-27	16:23:42	13
00e12	부분 문자열 출력하기	15	100.0	0.028845	보기	c	✓	2014-01-27	16:10:13	4

<그림 11> 채점 결과

또한 <그림 12>과 같이 전체 사용자 간의 랭킹 뿐만 아니라 <그림 13>에서처럼 자신이 소속된 학급 내의 랭킹을 확인할 수 있도록 구현했기 때문에 학습자는 경쟁을 통한 도전의식을 함양할 수 있다[14]. 이것은 CRESST 모델에서 자기통제의 '동기'를 고려한 기능이다.

전체 랭킹 1점 환산점수 합산 (각 문제에 대해 가장 최근에 채점한 결과가 반영됩니다.)

1 위 : ◆정보레인져 스위트◆ (209.90)
2 위 : Cloge (162.55)
3 위 : 가재 (158.09)
4 위 : SooraGenius (155.93)
5 위 : 장's (152.00)
6 위 : 도정혁 (144.06)
7 위 : 장수촌 (141.00)
8 위 : (/ㅡ_ㅡ)/~와마마 (125.45)
9 위 : ♥정보레인져 섹시♥ (118.58)
10 위 : 새우튀김 (117.31)
11 위 : BFer (117.18)
12 위 : 조창호 (116.60)

<그림 12> 전체 랭킹

2013교원심화연수 1점 환산점수 합산 (시도횟수 제한 : 9999)

1 위 : ★프동0호-대장★ (27,00) 땡땡 (27,00) 젊은태양 (27,00)
2 위 : 2030 (26,00) 정보 학생 (26,00)
3 위 : cjstk (25,85)
4 위 : ssomi88 (25,33)
5 위 : 애플k (24,33)
6 위 : 불줄이 (24,00) 오윤정 (24,00)
7 위 : anyppi (23,11)
8 위 : 박볼 (23,07)
9 위 : 윤승익 (22,95)
10 위 : bellsun (22,85)
11 위 : 동글사자 (22,20)
12 위 : 초록별 (18,83)
13 위 : 연습 (16,73)
14 위 : lex (15,00) 남미초 (15,00)
15 위 : 우리헌 (14,58)
16 위 : ultra (14,46)
17 위 : 코딩마스터 (14,00)
18 위 : 하늘그림 (11,00)
19 위 : 봄날개 (2,00)

베타 테스트 1점 환산점수 합산 (시도횟수 제한 : 1)

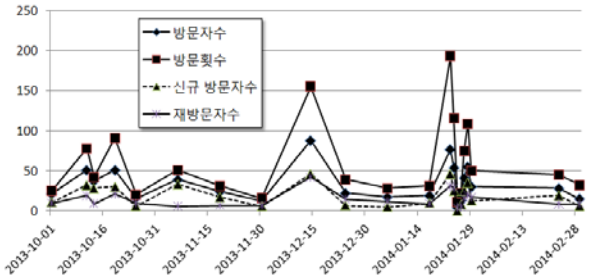
1 위 : Gravekper (2,00) STL_BackTrack (2,00) 정보 학생 (2,00)
2 위 : beginner (1,00)

<그림 13> 학급 랭킹

5. 알고리즘 자동평가 시스템의 적용

5.1 웹 서비스의 페이지 방문 분석

본 연구에서 개발한 자동평가 시스템을 충북의 초등학생, 중학생, 고등학생들에게 적용하였다. 시스템의 보급을 위해서 교사 대상의 프로그래밍 연수에서 시스템을 적극 활용하도록 권고하였고, 컴퓨터 영재교실의 강사들에게 시스템 사용법을 안내하였다. <그림 14>는 2013년 10월 1일~2014년 3월 1일까지 본 시스템의 사용자 서비스 환경에 대한 방문자수, 방문횟수, 신규 방문자수, 재방문자수를 그래프로 나타낸 것이다. <표 4>는 '전체랭킹'과 '학급랭킹' 페이지에 대해서 2013년 10월 1일~2014년 3월 1일까지의 페이지별 방문 횟수를 분석한 것이다. <표 4>에서 '전체 랭킹'의 방문횟수가 977회인 반면, '학급 랭킹'은 2193회로 약 2.3배 정도 더 많은 것을 확인할 수 있다. 더욱이 '전체 랭킹'은 로그인 과정 없이 곧바로 확인할 수 있는 반면, '학급 랭킹'은 로그인 과정이 필요한 것을 감안한다면, 사용자들은 '전체 랭킹'에 비해 '학급 랭킹'을 확인하고자 하는 욕구가 강함을 유추할 수 있다.



<그림 14> 페이지 방문 횟수

<표 4> 페이지별 방문 횟수 분석

순위	주요 페이지	방문횟수	비율
1	메인 페이지 1	41069	42.75%
2	메인 페이지 2	5415	5.64%
3	문제 검색	4638	4.83%
4	문제 정보	3789	3.94%
5	오늘의 랭킹	3005	3.13%
6	채점 결과	2494	2.60%
7	학급 랭킹	2193	2.28%
10	채점 현황	1818	1.89%
12	학급 문제	1456	1.52%
16	전체 랭킹	977	1.02%
18	강좌	845	0.88%
전체	전체	6471	100%

5.2 프로그래밍 학습 효과 분석

본 연구에서 개발한 알고리즘 자동평가 시스템의 프로그래밍 학습 효과를 확인하기 위해 다음과 같은 연구문제와 연구가설을 설정하였다.

연구문제 *: 프로그래밍 학습에서 자동평가 시스템을 적용한 교수·학습방법이 전통적인 교수·학습방법에 비해 더 효과적인가?*

H_1 *: 프로그래밍 학습에서 자동평가 시스템을 적용한 교수·학습방법이 전통적인 교수·학습방법에 비해 더 높은 성취도를 가져올 것이다.*

위의 가설을 검정하기 위해 참가학생 39명(초등학생 16명, 중학생 19명, 고등학생 4명)에 대해서 <그림 15>의 '비동질 통제집단 사전사후측정설계(nonequivalent control group design with pretest-posttest)' 실험을 실시하였다[15]. 실험집단은 C언어의 문법에 대한 설명과 함께 알고리즘 자동평가 시스템을 통해 실습하는 교수·학습 방법을 적용하였으며(X_1), 통제집단은 동일한 내용을 지도하면서 교재의 소스코드를 그대로 입력해 보는 방식의 전통적인 교수·학습 방법(X_2)을 적용하였다. 실험집단은 18명(초등학생 7명, 중학생 9명, 고등학생 2명), 통제집단은 21명(초등학생 9명, 중학생 10명, 고등학생 2명)으로 구성하였다.

O_0	X_1	O_1
O_2	X_2	O_3

<그림 15> 비동질 통제집단 사전사후측정설계

실험집단과 통제집단의 동질성을 확인하기 위해 O_0과 O_2에 대한 독립표본 T-검정을 실시하였으며, 독립변수의 효과를 확인하기 위해 O_1과 O_3에 대한 독립표본 T-검정을 실시하였다. 이때, 신뢰구간은 95%이며, 사전검사와 사후검사는 동일한 문항을 사용하였다.

5.2.1 검사 문항

프로그래밍 소스의 일부가 가려져 있고 이 부분에 적절한 코드를 채워넣는 방식의 문항 10개로 구성된 '프로그래밍 문제해결력 검사지'를 개발하였다. 이 검사지의 각 문항은 공란이 포함된 소스코드와 입·출력 예시가 명시되어 있는 것이 특징이다. 즉, 특정 범위 내의 입력값에 대해서 주어진 문제를 해결하는 컴퓨터 프로그램의 코드 일부를 작성하는 문항으로 제작되었다. 또한, C언어의 문법을 학습했다면 해결할 수 있는 기본 문항 8개와 문자열 처리 등에 관한 응용 문항 2개로 구성하였다. <표 5>는 기본 문항의 예시다.

검사지는 30점 만점으로 구성하였으며, 문항의 난이도에 따라 2점~5점의 배점을 부여하였다. 또한 공란에 컴파일과 실행에 문제가 없는 정확한 코드를 기재한 경우만 올바른 답으로 평가하였다.

<표 5> 기본문항 예시

5. 다음은 하나의 정수가 주어질 때 이 정수가 홀수인지 짝수인지 판별하는 프로그램입니다. 홀수는 1을, 짝수는 0을 출력합니다. [ ] 에 들어갈 코드는 무엇일까요? 단, 입력값의 범위는 10000 이하의 자연수입니다. (3점)

```
#include <stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    if( [                ] )
    {
        printf("0");
    }else
    {
        printf("1“);
    }
    return 0;
}
```

<입력 예시>	<출력 예시>
13	1

5.2.2 결과분석

1) 두 집단의 동질성 검정

<표 6> 사전검사 독립표본 t-검정

집단	N	평균	표준편차	t	df	p
실험	18	8.67	5.678	1.926	37	.062
통제	21	5.14	5.712			

* p<0.05

<표 6>에서 t값은 1.926, 자유도 37에서 양쪽검정의 유의확률이 .062이므로 두 집단 간 평균의 차이가 통계적으로 유의하지 않음을 의미한다. 즉, 사전검사에서 두 집단은 동질하다고 해석할 수 있다.

2) 독립변수의 효과 검정

<표 7> 사후검사 독립표본 t-검증

집단	N	평균	표준편차	t	df	p
실험	18	18.33	8.832	3.354	37	.002
통제	21	8.71	9.012			

* p<0.05

<표 7>에서 t값은 3.354, 자유도 37에서 양쪽검정의 유의확률이 .002이므로 두 집단 간 평균의 차이가 통계적으로 유의하며 실험집단의 평균점수(18.33)가 통제집단의 평균점수(8.71)보다 높다. 즉, 연구가설(H_1)은 채택되며, 전통적인 교수·학습 방법에 비해 알고리즘 자동평가 시스템을 적용한 교수·학습 방법이 프로그래밍 학습에 더 효과적이라고 해석할 수 있다. 다만, 사후검사에서 두 집단의 평균이 10점 이상 차이가 나는 이유는 알고리즘 자동평가 시스템의 효과와 더불어 통제집단의 전통적인 교수·학습 방법이 '컴퓨터를 이용한 문제해결의 자동화'라는 정보과학(Informatics)의 주요 내용을 학습자에게 각인시키지 못하는 문제에서 그 원인을 찾을 수 있을 것이다. 즉, 통제집단의 많은 학생들이 문항의 입출력 예시만 해결하는 프로그램을 작성한다는 것이다. 예를 들어 <표 5>의 문항에서 주어진 범위 내의 입력값에 대해서 올바른 결과를 출력하기 위해서는 네모칸에 변수를 사용한 수식 a%2==0 이 들어가야 하지만, 통제집단의 많은 학생들이 13%2==0 와 같은 형식으로 답안을 작성하는 경우가 많았다.

<표 8>과 <표 9>는 각 집단별 사전, 사후 검사의 평균에 대한 대응표본 t-검정을 실시한 결과다. 이 결과를 토대로 전통적인 교수·방법과 자동평가 시스템을 적용한 교수·방법이 모두 프로그래밍 학습에 효과가 있음을 확인할 수 있다.

<표 8> 실험집단 대응표본 t-검증

검사	N	평균	표준편차	t	df	p
사전	18	8.67	5.678	-6.328	17	0.000
사후	18	18.33	8.832			

* p<0.05

<표 9> 통제집단 대응표본 t-검증

검사	N	평균	표준편차	t	df	p
사전	21	5.14	5.712	-3.851	20	0.001
사후	21	8.71	9.012			

* p<0.05

6. 결론

6.1 연구 결과

본 연구에서는 CRESST 모델의 메타인지와 동기를 촉진하도록 설계한 클라이언트-서버 방식의 알고리즘 자동평가 시스템을 개발하였으며, 초·중·고 학생 39명을 대상으로 프로그래밍 학습 효과를 검정하기 위해 비동질 통제집단 사전사후 측정 실험을 수행하였다. 검정 결과, 사후 검사에서 실험 집단의 평균점수가 통제집단에 비해 유

의미하게 높은 것으로 나타났으며, 이를 통해 알고리즘 자동평가 시스템을 적용한 교수·학습방법이 전통적인 교수·학습방법에 비해 프로그래밍 학습에 더 효과적임을 확인할 수 있었다.

6.2 클라이언트-서버 방식의 장·단점

클라이언트-서버 방식의 알고리즘 자동평가 시스템은 학습자의 통합개발환경(IDE)과 연동되기 때문에 코딩 후에 자신의 소스코드를 곧바로 평가할 수 있으며, 채점이 사용자의 컴퓨터에서 수행되므로 다수의 학습자가 동시에 채점을 하더라도 속도가 매우 빠르다. 또한, 채점데이터 세트가 클라이언트에 다운로드 되어 학습자가 모든 입력값의 특성을 확인할 수 있기 때문에 문제해결력 증진을 위한 메타인지를 활성화하며, 학급생성 및 학급 내 랭킹산출 기능이 있어 관심 있는 대상과의 경쟁을 통해 학습동기가 강화된다. 그러나 Online Judge 방식에 비해 접근성과 편이성이 떨어지며 자동평가 프로그램을 사용자 컴퓨터에 설치해야 하므로 플랫폼 종속적이다. 또한 시간효율성에 대한 평가가 사용자 컴퓨터의 성능에 따라 다소 상이하게 산출될 수 있다는 점도 단점으로 작용한다.

6.3 활용 방안

알고리즘 자동평가 시스템은 초·중·고 단위학교에서 정보과학적 문제해결력 함양을 위한 교수·학습 도구로 활용할 수 있으며, 문제의 난이도를 다소 상향 조정함으로써 정보영재 교육에서 자료구조와 알고리즘에 대한 심화학습용으로 활용할 수 있을 것이다. 특히, 기초 프로그래밍 과정에서 본 시스템을 적용함으로써 문법에 대한 무미건조한 학습보다는 '문제'를 중심에 두고 그 문제를 해결하는 프로그래밍 학습으로 교육내용의 방향과 순서를 전환한다면, 학습자에게 프로그래밍 학습에 대한 강한 동기와 더불어 흥미를 부여할 수 있을 것이라고 강조하고 싶다. 더욱이 최근에 국가미래 창조경제를 위해 초·중·고 교육과정에서 '코딩 교육'과 'SW 원리교육'이 강조되는 만큼[16][17] 알고리즘 자동평가 시스템의 활용 범위는 더 넓어질 것으로 예상된다.

참 고 문 헌

[1] 한국교육과정평가원(2012). 미래 한국인의 핵심 역량 증진을 위한 창의적 문제 해결력 기반의 정보 교육 정책 방향 탐색, 연구보고 RRC 2012-7, 한국교육과정평가원.

[2] 김미혜(2007). 자동화된 프로그래밍 과제 평가 시스템의 설계 및 구현. 인터넷정보학회 논문지, Vol.8 No6., PP.75-85.

[3] 임형석, 김희철(2007). 경시대회를 통한 프로그래밍 교육 활성화 방안. 정보과학회지, Vol.25 No.7

[4] Yingwei Luo, Xiaolin Wang, Zhengyi Zhang(2008). *Programming grid : a computer-aided education system for programming courses based on online judge*. ACM SCE'08, Oct. 24-25

[5] Andy Kurnia.& Andrew Lim.& Brenda Cheang.(2001). *Online Judge*. Computers & Education 36. 299-315

[6] 정종광(2010). 과학고 학생을 위한 Online Judge 기반 프로그래밍 평가 시스템의 설계 및 구현. 석사학위 논문, 한국교원대학교.

[7] 송지혜(2011). 자기주도학습을 위한 자동채점 기반의 프로그래밍 교육 시스템. 박사학위논문, 숭실대학교

[8] Herl, H. E., O'Neil, Jr. H, F., Chung, Gregory, K.W.K. et al.(1999). *Final Report for Validation of Problem-Solving Measures*. Center of the Study of Evaluation Technical Report 501.

[9] 한건우(2007). 프로그래밍 교육에서 문제해결력 신장을 위한 동료 에이전트 시스템 개발. 박사학위 논문, 한국교원대학교.

[10] 유인환, 김태완(2006). MINDSTORMS을 이용한 프로그래밍 학습이 창의력에 미치는 효과, 한국컴퓨터교육학회 논문지, 9(1), pp. 49-59.

[11] 이봉주, 김원경(2003). 비주얼 베이식을 이용한 수학 문제해결 과정에서 고등학생의 메타인지적 능력 활성화. 한국수학교육학회지 시리즈 A, 42(5), pp.623-636.

[12] 임형식(2007). ACM-ICPC 문제의 출제 및 채점 과정. 한국정보과학회지, 제25권, 제7호, PP.52-55.
[13] *HUST ACM OnlineJudge*, Retrieved June 16, 2014. from https://code.google.com/p/hustoj/
[14] Miguel A. REVILLA, Shahriar MANZOOR, Rujia LIU(2008). *Competitive Learning in Informatics: The UVa Online Judge Experience*, Olympiads in Informatics, Vol. 2, 131 - 148
[15] 채서일(2005). 사회과학조사방법론(3판). 서울:비앤엠북스
[16] 강현우(2014), "SW로 창의인재 키우자 : 한국만 손 놓은 SW 조기 교육", 한국경제신문, A1면, 2014.03.31.
[17] 강현우(2014), "SW로 창의인재 키우자 : 초·중학교부터 SW 필수과목으로", 한국경제신문, A1면, 2014.04.04.

장 원 영

1998 충북대학교 회계학과(경영정보학 부전공, 경영학사)
2002 충북대학교 컴퓨터교육과(이학사)
2003 충북대학교 교육대학원 상업교육 전공(교육학 석사)
2013～현재 한국교원대학교 컴퓨터교육 전공 박사과정
2004～2013 충북과학고등학교 교사
2014～현재 충북교육정보원 파견교사
2011 교육과학기술부장관표창 수상(스승의 날 기념)
2014 국무총리표창 수상(정보문화 유공)
관심분야: Computational Thinking, EPL, 알고리즘 교육, 한국정보올림피아드, Java 언어 교육
E-Mail: won0c@naver.com

김 성 식

1977 고려대학교 경영학과(경영학사)
1988 오리곤 주립대학교 전산학과(이학석사)
1992 고려대학교 컴퓨터과학과(이학박사)
1992～현재 한국교원대학교 컴퓨터교육과 교수
관심분야: 컴퓨터교육, 알고리즘, 정보윤리교육
E-Mail: seongkim@knue.ac.kr