

# 문제해결력 향상을 위한 동료 에이전트 시스템의 교육적 효과

한건우<sup>†</sup>

## 요 약

프로그래밍 교육을 위한 다양한 연구가 진행되고 있다. 본 연구는 문제해결력 향상을 위해 교수 학습 모형을 기반으로 동료 에이전트를 개발하였다. 문제해결력에 대한 하위 요소를 학습내용 이해, 문제해결 전략, 자기조절 영역으로 구분하였고, 문제해결력 하위 요소에 대한 프로그래밍 교수 학습 방법을 제시하였으며, 교수학습 모형을 수행할 수 있도록 동료 에이전트를 개발하였다. 개발된 동료 에이전트 시스템의 교육적 효과를 살펴보기 위하여 프로그래밍 수업에 적용한 결과, 전통적 수업 방식이나 동료 프로그래밍 수업 방식에 비해 문제해결력 하위 요소에서 긍정적인 효과를 보였다. 이는 동료 에이전트가 학습자의 문제해결력을 향상시키는데 교육적 효과가 있음을 의미한다.

주제어 : 프로그래밍 교육, 문제해결력, 동료 에이전트

## The Educational Effects of Peer Agent System to Improve Problem-Solving Ability

Keun-Woo Han<sup>†</sup>

### ABSTRACT

There are several studies in order to learn about programming language. This paper develops a peer agent system based on teaching and learning strategies in order to improve problem-solving ability. Problem-solving ability involves three components - content understanding, domain-dependent problem-solving strategies, and self-regulation. This paper shows the teaching and learning strategies about components of problem-solving for learning to program and the peer agent system uses the teaching and learning strategies. This study intends to analyze the educational effects of the peer agent system. The results show that the system has superior effects on problem-solving ability compared to traditional programming courses or pair programming courses. It means that the peer agent system is the effective educational system in improving students' problem-solving ability.

Keywords : Programming Education, Problem-Solving Ability, Peer Agent

<sup>†</sup> 종신회원: 선부중학교 교사(교선적자)  
논문접수: 2009년 5월 15일, 심사완료: 2009년 7월 27일

## 1. 서 론

문제를 해결하기 위한 과정을 알고리즘이라고 말하며 일반적으로 알고리즘은 컴퓨터 프로그래밍 언어로 구현하여 실행한다. 프로그래밍 언어는 논리적인 과정으로 알고리즘을 표현하기 때문에 프로그래밍 언어를 배우는 학습자는 논리적인 사고력이나 추상적인 추론 능력 등의 인지적 능력이 필요하다[1]. 따라서 대부분의 학습자들은 프로그래밍 언어를 배우는 것을 어려워하고 두려워하며 중도탈락율도 높은 편이다.

프로그래밍 언어를 효과적으로 학습하기 위해서는 다양한 학습전략이 요구된다. 대표적으로는 프로젝트 기반 학습[2][3], 구성주의적 접근[4] 등 교수학습 전략을 통해 프로그래밍 언어에 대한 동기유발과 학업성취도를 높이고자 하였다. 다른 한편으로는 웹 기반 학습[5]이나 적응형 학습 시스템[6][7]을 도입하여 프로그래밍 효과를 높이고자 하였다.

프로그래밍을 배우는 학습자는 프로그래밍 언어에 대한 개념과 문법적 지식뿐만 아니라 문제 해결을 위한 프로그래밍 기법을 익혀야 한다. 그 기법은 프로그래밍을 학습하는데 있어서 효과적이어야 하며 문제해결력을 높이기 위한 학습 전략이 필요하다. 본 연구에서는 소프트웨어 공학적인 관점에서 등장한 동료 프로그래밍(Pair Programming)을 적용하고자 한다. 동료 프로그래밍은 두 명의 프로그래머가 하나의 컴퓨터에 같이 앉아 프로그램을 설계, 알고리즘, 코딩, 디버깅 등을 협업하는 형태를 말한다. 혼자서 프로그래밍을 하는 것보다 동료와 같이 프로그래밍 작업을 하는 것이 보다 효과적이며, 프로그래머 간의 쉬운 사회화가 가능하므로 부수적인 효과를 얻을 수 있다[8]. 또한 학업성취도나 학습동기에 긍정적인 영향을 주는 것으로 나타났다[9]. 선행 연구물들을 살펴보면 동료 프로그래밍은 프로그래밍 교육에 있어서 긍정적 효과가 있음을 말해주고 있다. 그러나 동료 프로그래밍은 동료의 인지적, 정의적 특성에 영향을 받으며, 책임감이 요구된다. 올바른 동료의 선정과 동료의 프로그래밍 수준에 따라 교육적 효과가 달라질 수 있다. 이러한 문제

점을 해결하고자 e-러닝 환경에서 동료 에이전트를 개발하여 프로그래밍 학습을 한 결과 학업성취도와 자아효능감 영역에서 긍정적인 효과를 보이기도 하였다[10].

본 연구는 프로그래밍 교육에서 필요한 문제해결력 향상에 중점을 두고자 한다. 동료 프로그래밍의 교육적 효과를 위해 동료 에이전트 시스템을 바탕으로 하고 있으며, 여기에 문제해결력을 향상시키기 위한 전략을 도입하였다. 동료 프로그래밍은 학습자에게 프로그래밍 개발의 효율성을 제공하지만 문제해결력을 향상시키기 위한 체계적인 모형을 포함하고 있지 않다. 따라서 동료 에이전트 기반 동료 프로그래밍 학습 시스템에 문제해결력을 위한 전략을 제공하여 학습자의 문제해결력에 영향을 주는 교육적 효과를 분석하였다.

## 2. 관련 문헌

### 2.1 동료 프로그래밍과 동료 에이전트

동료 프로그래밍은 두 사람이 하나의 컴퓨터를 사용하여 협력적 분업의 형태로 프로그래밍 하는 것으로 프로그래밍 개발 효율을 높여주며, 동료와의 상호작용을 통해 프로그래밍에 대한 학습 효과가 있는 것으로 나타났다. 프로그래밍에 대한 기초 개념과 지식을 서로 공유하게 되며 상호간의 대화로 지식이 구조화되고 지식 전이가 되는 등 학습에 긍정적인 영향을 주는 것으로 나타났다[11]. 동료 프로그래밍을 통해 교육적 효과를 보기 위해서는 동료의 역할이 매우 중요하다. Schneider와 Johnston[12]은 동료 프로그래머 간의 수준 차이 때문에 프로그래밍을 잘 하는 사람은 작업을 많이 하게 되고, 못하는 사람은 학습 효과가 없는 것으로 보고하였다. 동료가 잘 선정된 팀만이 효과가 나타난다는 문제점을 제시하였다. Srikanth 등[13]은 동료 프로그래밍을 적용하게 되면 1학기 동안 동료가 지정되는데 동료의 상태가 최적이지 아니므로 동료의 학습 효과를 위해 동료를 번갈아 가면서 학습하는 것이 동료 프로그래밍의 부작용을 줄이는데 의미가 있다고 말한다.

e-러닝 기반의 원격 교육은 학습을 도와주고

학습자 개개인에게 최적화된 환경을 제공하고 있으나 학습자에게 인지적 부담을 주거나 비인격적이고 동기 유발이 어렵다는 한계를 가진다[14]. 이러한 문제를 해결하고자 에이전트가 등장하였으며, 교수적 관점(Pedagogical Agent)이나 지능적 관점(Intelligent Agent)에서 연구가 진행되고 있다. 한건우 등[10]은 동료 프로그래밍의 문제점을 해결하기 위해 효과적인 프로그래밍 학습을 위한 동료 에이전트 기반 학습시스템을 개발하였다. 이들은 동료 프로그래밍을 지원하며 동료 교수법적 관점에서 동료와 같은 역할을 하는 에이전트를 동료 에이전트라고 하였다. 동료 에이전트는 Driver와 Navigator를 번갈아 가면서 동료 프로그래밍의 과정을 수행한다. 역할에 따른 동료 에이전트의 역할을 정리하면 <표 1>과 같으며, 이 과정은 강의자(Tutor)와 학생(Tutee)이라는 동료 교수법적인 관점으로 접근되었다.

<표 1> 동료 프로그래밍을 지원하는 동료 에이전트의 역할

역할	내용
Driver	<ul style="list-style-type: none"> <li>● 학습자에게 코드를 제공함.</li> <li>● 학습자는 동료 에이전트에게 문제해결전략을 순서도로 표현함.</li> </ul>
Navigator	<ul style="list-style-type: none"> <li>● 학습자에게 문제해결전략을 제공함.</li> <li>● 에이전트는 학습자에게 문제해결전략을 순서도로 제공함.</li> </ul>

동료 에이전트가 Driver의 역할을 수행하는 동안 에이전트는 학습자에게 학습을 받는 학생(Tutee)이 되며, 학습자는 가르침에 의한 학습(Learning by Teaching) 효과가 나타난다. 역할을 바꾸어 동료 에이전트가 Navigator의 역할을 수행하게 되면 에이전트는 학습자에게 학습을 하는 강의자(Tutor)가 되며, 실습에 의한 학습(Learning by Doing) 효과가 나타난다.

또한 동료 에이전트는 베이지안 네트워크 학습자 모델을 기반을 지능형 에이전트의 기능을 가지고 있다. 학습자가 부족한 학습의 영역에 대해 추가적인 보충 학습을 제공하도록 구현되어 있다.

동료 에이전트는 프로그래밍의 개념을 익히기 위한 동료 프로그래밍의 전략을 도입하여 교육적

효과를 보이고 있으나 문제해결력을 향상시키기 위한 구체적인 모형을 가지고 있지 않다. 에이전트는 학습자를 위한 개별화 학습을 위해 학습자의 수준을 판단하는 것도 중요하지만 학습자의 학습효과를 높이기 위한 교수-학습 모형을 가지고 있어야 한다. 동료 에이전트는 같이 공부하는 친구와 같은 역할을 하며 동시에 동료 프로그래밍과 같은 학습 전략을 제공해야 한다. 여기에 본 연구는 문제해결력을 위한 구체적인 모형을 추가로 도입하고자 한다.

## 2.2 문제해결력

지식기반 사회에서는 창의적이고 문제해결력이 높은 인재를 요구한다. 다시 말해 정보를 내면화된 지식으로 전이하여 문제 상황에 적용하고, 지식을 통해 문제를 해결하는 능력이 중요시 되고 있다. 문제해결력은 새로운 문제를 자신이 알고 있는 지식을 통해 해결하고 이러한 과정에서 새로운 지식을 배우는 데 사용한다. Bloom[15]은 경험된 정보와 기술을 바탕으로 분석하고 응용할 수 있는 능력을 지적 능력과 기법(intellectual ability and skills)이라고 정의하면서 Dewey의 반성적 사고나 문제해결력과 같은 의미라고 하였다. 문제해결력은 다양한 인지전략으로 구성되어 있어 일반화하고 이론화하는데 연구가 진행되고 있으며, 여러 변인들과의 관계에 대한 연구가 실시되고 있다.

Glaser[16]는 과학영역의 특징을 고려한 평가 설계를 인지적 관점으로 접근하였다. 학생들을 분석해보면 문제해결력을 위해 다음과 같이 5가지의 요소가 영향을 주는 것으로 보고하였다. 첫째, 구조화되고 통합된 지식(structured, integrated knowledge)이다. 우수한 학생은 독립된 사실로 접근하기 보다는 유기적인 정보로 문제를 해결하기 위해 정보를 통합하고 의미 있는 유형과 요소로 접근한다. 둘째, 효율적인 문제 표현(effective problem representation)이다. 우수한 학생은 문제의 본질을 파악하고 문제에 대한 추상화를 통해 사고의 모형이나 표현한다. 셋째, 절차화된 지식(proceduralized knowledge)이다. 문제해결력이 우수한 학생들은 무엇을 알고 언제 사용할 것인지

를 안다. 지식의 사용을 위한 절차와 활용 조건을 가지고 있다. 넷째, 자동성(automaticity)이다. 숙련된 방법으로 빠르게 일을 처리한다. 다섯째, 자기조절기법(self-regulatory skills)이다. 우수한 학생들은 자신에 대한 평가와 조절을 통해 자기조절을 개발한다. Sugrue[17]는 기존의 선행연구물들을 근간으로 여러 변인들을 분석하여 과학영역에서의 문제해결력에 대한 요인을 결정하였다. 그는 지식의 구조(knowledge structure), 인지기능(cognitive functions), 신념(beliefs)이라는 세 가지 영역을 선정하였다. 세 가지 영역은 다시 하위요소를 가지고 있으며, <그림 1>과 같다.



<그림 1> Sugrue의 문제해결력 모델

지식의 구조는 개념, 원리, 조건과 절차 요소를 지식의 요소로 제시하였으며, 인지기능은 계획, 모니터링을 하위 요소로 설정하였다. 신념은 자기 효능감, 과제 난이도, 과제 매력도를 하위 요소로 제시하였다.

기존의 연구물들에서 문제해결력을 위한 요소를 추출하여 정리해 보면 크게 세 가지 영역으로 구분해 볼 수 있다. 첫째는 개념지식이다. 문제해결을 위한 기초적인 지식을 가지고 있어야 하며, 체계적으로 관리되고 운영되어야 한다. 둘째는 문제해결의 과정이다. 문제해결을 위한 기술과 방법 등을 익히고 있어야 한다. 셋째는 자기조절력이다. 자신이 문제 해결하는 과정에 대한 초인지과정이 필요하다. 이와 같은 영역은 Herl 등[18]이 제시한 문제해결을 위한 요소와 유사하다. 이들은 연구보고서에서 문제해결력을 위한 세 가지 요소를 제시하였다. 첫째, 학습내용 이해(content understanding)로 문제에 대한 기초 개념과 지식이 필요하다. 둘째, 문제해결 전략(domain-dependent problem-solving strategies)으로 해결 과정을 표현하고 전략을 구성할 수 있는 것을 말한다. 셋째,

자기조절(self-regulation)로 초인지(metacognition)와 동기(motivation)를 하위 요소로 가지고 있다. 이를 도식화하면 <그림 2>와 같다.



<그림 2> Herl 등의 문제해결력 모델

제시한 자기조절력 중 초인지는 계획(Planning)과 자기점검(self-monitoring)으로 구성되며, 동기는 노력(effort)과 자기 효능감(self-efficacy)로 구성된다. 이 문제해결력 모델은 과학 실험 과정에 투입하여 검증하였으며, 그 효과를 살펴보고 있다. 본 논문에서는 프로그래밍 과정에 필요한 문제해결력 모델로 Herl 등이 제시한 모델을 선택하였다. 기존의 문제해결력의 요소를 충분히 반영하고 있으며, 프로그래밍 과정에서 요구되는 논리적 사고력 등에 부합되게 문제해결력의 요소를 갖추고 있다고 볼 수 있다. 학습내용 이해는 프로그래밍에 대한 기초적인 개념과 지식, 문제해결전략은 순서도 등을 이용한 알고리즘 표현, 자기조절은 에이전트와의 상호작용을 통한 초인지와 동기 부여 등이 될 수 있다.

### 3. 문제해결력 기반 동료 에이전트 개발

#### 3.1 문제해결력 기반 동료 에이전트 설계

본 연구는 프로그래밍 교육에서 문제해결력을 향상시키기 위해 동료 에이전트를 개발하는 것이다. 문제해결력의 하위 요소를 바탕으로 동료 프로그래밍을 재구성하여 교수학습 모형을 만들고 이를 실현하는 동료 에이전트를 개발하는 것이다. 선행연구물들을 분석하여 프로그래밍 교육에 적합한 문제해결력 모델로 Herl 등이 제시한 모델을 선택하였다. Herl 모델은 기존의 모델을 통합하여 최적의 요소만을 도출하여 제작된 정형화된

모델이다. 이 모델에서 제시하는 세 가지 문제해결 요소들(학습내용 이해, 문제해결 전략, 자기조절)은 프로그래밍 교수·학습 모형으로 활용하는데 적합하며, 이는 다음과 같다.

첫째, 학습내용 이해 영역은 특정 주제에 대한 학습 도메인의 이해를 의미한다. 학습자는 문제해결력을 향상시키기 위해 프로그래밍 언어에 대한 학습내용을 이해하고 있어야 하므로 프로그래밍에 대한 기초개념을 학습해야 한다. 동료 에이전트는 해당 프로그래밍 언어 학습 주제에 대한 기초개념을 제공하고 실습 환경을 제공한다.

둘째, 문제해결 전략 영역은 주어진 문제를 해결하는 논리적인 판단 과정을 의미한다. 프로그래밍 언어를 통해 문제를 해결하는 과정은 알고리즘이라고 말할 수 있다. 순서도를 통해 알고리즘을 표현하거나 문제해결의 전략을 이해한다. 동료 에이전트는 학습자에게 알고리즘을 표현할 수 있는 순서도를 제공하여 문제해결 전략을 제시한다.

셋째, 자기조절 영역은 학습이나 문제해결 과정에서 학습을 촉진시키는 메타인지와 동기에 대한 대화를 전달한다. 메타인지는 계획, 자기점검이라는 하위 요소로 구성되었으며, 동기는 노력, 자기효능감으로 구성되어 있다. 동료 에이전트는 학습자와의 지속적인 대화로 학습의 과정에서 자기조절력을 향상시켜 주어야 한다.

프로그래밍 학습의 관점에서 문제해결력 모델을 재구성하면 <그림 3>과 같다.



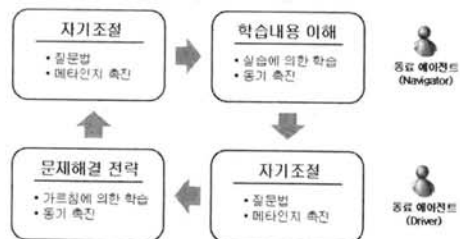
<그림 3> 문제해결력 기반의 프로그래밍 교육

프로그래밍 교육을 위해 동료 에이전트는 동료 프로그래밍의 교수·학습 전략을 사용한다. 동료 프로그래밍은 Driver와 Navigator의 역할로 구분되며 동료 에이전트는 두 가지 역할을 학습 주제마다 차례대로 수행한다.

먼저 동료 에이전트가 Navigator의 역할을 수행하고 학습자가 Driver의 역할을 수행한다. 에이전트는 가르치는 관점이며, 학습자는 배우는 관점이다. 학습자는 메타인지에 대한 자기점검과 계획의 과정을 동료 에이전트의 대화를 통해 접하게 된다. 에이전트의 안내를 받고 프로그래밍에 대한 학습을 진행하며, 주어진 문제에 대한 프로그래밍 실습을 한다. 실습의 과정에서 에이전트는 문제해결 전략을 구성하여 순서도로 제시해 주며 학습자가 작성한 코드에 대해 문법적인 오류를 해결할 수 있는 힌트를 제공한다. 아울러 학습자가 디버깅의 과정에 적극적으로 참여하고 극복할 수 있도록 노력과 자기 효능감의 대화를 제공한다.

다음으로는 동료 에이전트가 Driver의 역할을 수행하고 학습자가 Navigator의 역할을 수행한다. 에이전트는 문제 풀이의 과정을 배우는 관점이며, 학습자는 그 과정을 가르치는 관점이다. 학습자는 배우는 관점이며, 메타인지에 대한 자기점검과 계획의 과정을 동료 에이전트의 대화를 통해 접하게 된다. 에이전트의 안내를 받으며 프로그래밍에 대한 학습을 진행하며, 주어진 문제에 대해 학습자는 문제 해결의 전략을 알고리즘으로 구성하여 순서도로 표현한다. 에이전트는 순서도가 제대로 작성될 수 있도록 도와주며, 학습자가 문제해결 전략을 순서도로 표현할 수 있도록 노력과 자기 효능감의 대화를 제공한다.

동료 에이전트의 역할에 따른 프로그래밍 학습 절차를 문제해결력 요소 간의 관계로 제시하면 <그림 4>와 같다.



<그림 4> 프로그래밍 학습 절차

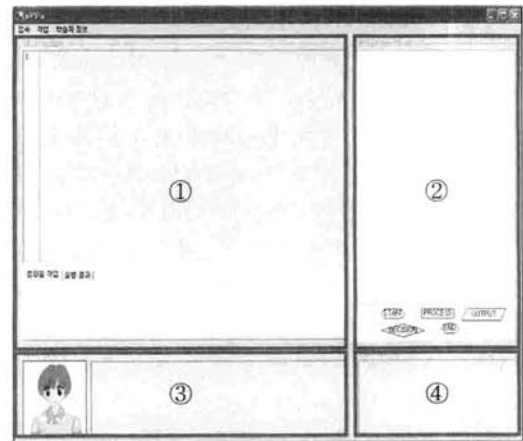
동료 에이전트가 Navigator의 역할인 경우 수행되는 학습 절차와 내용의 예시를 살펴보면 <표 2>와 같다.

<표 2> 학습 절차와 학습 예시

학습 절차	코드	순서도
1. for문 예제 학습	<pre>#include &lt;stdio.h&gt;  void main() {     int i;      for(i=1; i&lt;4; i++)         printf("Hello~\n", i); }</pre>	
2. 실습에 의한 학습 (동료 에이전트 - Navigator 학습자 - Driver)	<pre>#include &lt;stdio.h&gt;  void main() {     int i;      for(i=0; i&lt;10; i++)         printf("%d", i); }</pre>	
3. 평가	<pre>#include &lt;stdio.h&gt;  void main() {     int i;      for(i = 1 ; i &lt; 10 ; i++)         printf("5 * %d = %d\n", i, 5*i); }</pre>	

고 결과를 받을 수 있도록 하였다. 웹 서비스는 표준 기술(XML, HTTP, RPC 등)에 기초하여 이질적인 환경에 있는 프로그램 간에 통신할 수 있게 하는 기술이다. SOAP(Simple Object Access Protocol)을 이용하여 XML을 기반으로 동작하며 HTTP를 통해 다른 웹 서버에 연결해서 요구하고 결과를 받아올 수 있다.

동료 에이전트 학습 프로그램 화면은 <그림 5>와 같다. ① 영역은 학습자가 프로그래밍 코드를 작성하고 컴파일이나 실행 결과를 살펴볼 수 있는 곳이다. 이곳을 통해 학습자는 프로그래밍 학습 내용을 직접 구현하고 문제에 대한 해결 과제를 실습한다.



<그림 5> 동료 에이전트 학습 프로그램

### 3.2 문제해결력 기반 동료 에이전트 구현

에이전트 설계를 바탕으로 동료 에이전트 학습 시스템을 구현하였다. 먼저 학습자에게 프로그래밍 학습 내용을 직접 구현해보고 프로그래밍의 과정에서 문법적 오류와 논리적 오류를 해결하는 과정을 가질 수 있는 환경을 제시해야 한다. 동료 학습 시스템에서는 학습자의 컴퓨터에 컴파일러가 없더라도 제공된 시스템에서 학습자의 코드를 컴파일 할 수 있도록 구현하였다. .NET 프레임워크 환경에서 C# 언어로 구현되며 웹 서비스 기술을 적용하여 서버에서 학습자의 코드를 컴파일하

에이전트는 동료 프로그래밍의 역할에 따라 문제해결력 증진을 위한 기반으로 반응하도록 구성하였다. 학습 주제마다 <그림 5>의 ③ 영역에 설명을 제공하며, 관련 실습 코드를 ① 영역에 제공하여 학습자가 확인하고 실습할 수 있도록 하였다. 동료 에이전트가 Navigator의 역할을 수행하면 학습자가 문제를 해결할 수 있도록 알고리즘을 순서도로 제공한다. 순서도는 <그림 5>의 ② 영역이며, 순서도의 내용에 대해 ③ 영역을 통해 자세한 설명을 제공한다. 학습자는 제공되는 설명과 순서도를 바탕으로 문제 해결을 위한 코드를 작성하고 실행하여 결과를 분석한다. 학습 주제에

대한 과정이 끝나면 ④의 영역에 문제를 제시하고 학습자가 문제를 해결할 수 있도록 힌트를 제공한다. 다음으로 동료 에이전트가 Driver 역할을 수행하면 학습자에게 문제를 해결하기 위한 알고리즘으로 순서도로 작성해 보길 요구한다. 학습자는 ② 영역에 순서도를 작성하거나 제시된 순서도를 완성하여 문제해결 전략을 제공한다. 에이전트는 순서도가 완성되면 관련 코드를 ① 영역에 제공하고 학습자가 코드를 살펴보며 부족한 내용을 완성한다. 동료 에이전트와 학습을 하는 동안 ③ 영역을 통해 학습자에게 지속적으로 메타인지를 위한 학습 계획과 자기점검의 내용을 제시하며, 동기를 위한 노력과 자기 효능감을 촉진하기 위한 내용을 제시한다. 자기조절 촉진을 위한 대화의 내용은 <표 3>과 같다.

<표 3> 자기조절 촉진을 위한 에이전트의 대화

자기조절	대화내용
메타인지	<p>계획</p> <p>학습의 과정을 계획하도록 한다. 문제해결하기 전에 해결 전략을 구상하도록 안내한다. 학습자가 배우는 전체 과정이 무엇인지 인지하도록 안내한다. 학습자가 해야 할 일이 무엇인지 생각해보도록 한다.</p>
	<p>자기점검</p> <p>학습자가 문제를 잘 해결하고 있는지 생각해보도록 한다. 문제가 발생되면 문제점을 찾고 수정하도록 격려한다. 문제해결의 과정을 이해하고 있는지 질문한다. 문제를 정확하게 인지하도록 안내하여 확인하도록 한다.</p>
동기	<p>노력</p> <p>최선을 다해 문제를 해결하도록 대화한다. 많은 실수를 수행할 수 있도록 권고한다. 문제를 해결해 달라고 요청한다. 평가를 통해 지식을 점검하는 것이 유용하다고 대화한다.</p>
	<p>자기효능감</p> <p>학습자가 학습에 대해 자신감을 가질 수 있도록 대화한다. 평가를 잘 수행할 수 있을 것이라고 대화한다. 복잡한 문제도 해결할 수 있을 것이라고 대화한다. 이번 과정에서 높은 점수를 받을 수 있을 것이라고 대화한다.</p>

## 4. 연구 방법

### 4.1 연구 대상 및 검사 도구

연구를 위해 프로그래밍 수업을 실시하고 있는 전문계 고등학교 3개 반 110명을 대상으로 선정하였다. 대상 학생들은 정규 수업과정에서 C 프로그래밍 언어를 배우고 있으며, 같은 프로그래밍 담당 교사로부터 지도를 받았다. 수업 처치 후 실험에 끝까지 참여하고 설문에 성실히 응답한 학생 100명을 최종 연구 분석 대상으로 선정하였다. 본 연구는 학생들의 문제해결력 요소별 영향력을 바탕으로 학습 효과를 살펴보고자 한다. 검사도구는 학습내용 이해, 문제해결 전략, 자기조절 영역 별로 개발하였다. 학습내용 이해와 문제해결 전략은 본 연구자가 학습 차시에 맞추어 개발하였다. 학습내용 이해 영역의 사전 검사는 수업 진도에 따라 조건문 중심으로 프로그래밍 학습의 수준을 진단하였으며 사후 검사는 수업 처치 내용인 반복문에 대한 내용으로 구성하였다. 문제해결 전략 영역의 사전 검사는 조건문에 대한 문제해결 전략 수준을 진단하기 위해 순서도 작성으로 구성하였으며, 사후 검사는 수업 처치 내용인 반복문에 대해 순서도 작성 내용으로 구성하였다. 개발한 문항은 교과 담당 교사에게 문제 오류 및 난이도 등을 검토 받아 적용하였다. 자기조절 영역은 CRESST 보고서에서 제공하는 자기조절 설문지[18]를 번안하여 사전검사와 사후검사에 사용하였으며, 번안의 내용에 대해서는 교과 담당 교사와 영어 교육 전문가에게 검토를 받아 활용하였다.

### 4.2 연구 설계

본 연구는 동료 에이전트가 학습자의 문제해결력에 얼마나 효과적인지를 살펴보는 것이다. 이를 위해서는 기존의 전통적인 프로그래밍 수업 방식과 비교해 볼 필요가 있으며, 교수학습 모델이 되는 동료 프로그래밍 수업과도 비교해 볼 필요가 있다. 따라서 본 연구에서는 통제집단은 전통적인 실습시연 수업방식, 비교집단은 동료 프로그래밍

수업방식, 실험집단은 문제해결력 기반 동료 에이전트 수업방식으로 설계하였으며, 학교 현장의 수업 진도에 따라 반복문을 학습 주제로 선정하여 수업하였다. 본 연구를 위한 실험 설계는 <표 4>와 같다.

<표 4> 실험 설계

G <sub>1</sub>	O <sub>1</sub>	O <sub>2</sub>	O <sub>3</sub>		O <sub>4</sub>	O <sub>5</sub>	O <sub>6</sub>
G <sub>2</sub>	O <sub>7</sub>	O <sub>8</sub>	O <sub>9</sub>	X <sub>1</sub>	O <sub>10</sub>	O <sub>11</sub>	O <sub>12</sub>
G <sub>3</sub>	O <sub>13</sub>	O <sub>14</sub>	O <sub>15</sub>	X <sub>2</sub>	O <sub>16</sub>	O <sub>17</sub>	O <sub>18</sub>

- G<sub>1</sub> : 통제집단
- G<sub>2</sub> : 비교집단 - 동료프로그래밍
- G<sub>3</sub> : 실험집단 - 문제해결력 기반 동료에이전트
- X<sub>1</sub> : 동료 프로그래밍 수업
- X<sub>2</sub> : 문제해결력 기반 동료 에이전트 수업
- O<sub>1</sub> O<sub>7</sub> O<sub>13</sub> : 학습내용 이해 사전검사
- O<sub>2</sub> O<sub>8</sub> O<sub>14</sub> : 문제해결 전략 사전검사
- O<sub>3</sub> O<sub>9</sub> O<sub>15</sub> : 자기조절 사전검사
- O<sub>4</sub> O<sub>10</sub> O<sub>16</sub> : 학습내용 이해 사후검사
- O<sub>5</sub> O<sub>11</sub> O<sub>17</sub> : 문제해결 전략 사후검사
- O<sub>6</sub> O<sub>12</sub> O<sub>18</sub> : 자기조절 사후검사

학생들이 문제해결력 요소별 사전 검사를 실시하고 각 집단별로 해당 수업방식을 투입하였다. 주당 3시간씩 3주간 총 9차시 분량의 반복문 수업을 동일 교사가 실시하였다. 3개 집단에 대해 각각의 수업 처지를 한 후 10차시 수업 시 문제해결력 요소별 사후 검사를 실시하였다.

### 4.3 수업 처치

본 연구는 통제집단, 비교집단, 실험집단에 각기 다른 수업 처지를 한다. 통제집단은 학생들이 교사의 설명에 따라 교재를 활용하여 따라하는 일반적인 강의 중심의 실습시연 수업을 하였다. 비교집단은 교사의 임의대로 2인 1조 형태의 동료 프로그래밍 수업을 실시하였다. 실험집단은 교사의 안내에 따라 문제해결력 기반 동료 프로그래밍 수업을 개인별로 실시하였다. 학생들은 동료 에이전트의 안내에 따라 동료 프로그래밍 방식의 수업을 문제해결력 향상을 위한 교수학습

모형으로 진행하였다. 집단별 수업 처치 과정 및 내용을 살펴보면 <표 5>와 같다.

<표 5> 수업 처치 과정 및 내용

주	학습내용	통제집단	비교집단	실험집단
1	for문	실습시연 수업	동료 프로그래밍	동료 에이전트 (실습에 의한 학습, 자기조절)
2	for문	실습시연 수업	동료 프로그래밍	동료 에이전트 (가르침에 의한 학습, 자기조절)
3	for문	실습시연 수업	동료 프로그래밍	동료 에이전트 (실습에 의한 학습, 자기조절)
4	while문	실습시연 수업	동료 프로그래밍	동료 에이전트 (가르침에 의한 학습, 자기조절)
5	while문	실습시연 수업	동료 프로그래밍	동료 에이전트 (실습에 의한 학습, 자기조절)
6	while문	실습시연 수업	동료 프로그래밍	동료 에이전트 (가르침에 의한 학습, 자기조절)
7	do-while 문	실습시연 수업	동료 프로그래밍	동료 에이전트 (실습에 의한 학습, 자기조절)
8	do-while 문	실습시연 수업	동료 프로그래밍	동료 에이전트 (가르침에 의한 학습, 자기조절)
9	do-while 문	실습시연 수업	동료 프로그래밍	동료 에이전트 (실습에 의한 학습, 자기조절)

## 5. 연구 결과

### 5.1 학습내용 이해 변인

학생들의 프로그래밍 학습내용 이해에 대한 사전 수준이 동일한지 검증하기 위해 사전검사를 실시하였다. 학습내용 이해 변인에 대한 분산분석 결과는 <표 6>과 같다.

<표 6> 학습내용 이해 사전검사

구분	N	M	SD	F	p
통제집단	33	5.00	2.000	.110	.896
비교집단	33	5.15	2.526		
실험집단	34	4.88	2.483		

학습내용 이해 사전검사 결과 실험집단이 다소 낮은 평균을 보이고 있으나 통계적으로 유의미한



차이를 보이지 않았다( $p < .05$ ).

9차시의 수업 처지 후 각 집단별로 학습내용 이해 영역의 차이를 살펴보기 위해 사후검사를 실시하였으며, 분산분석 결과는 <표 7>과 같다.

<표 7> 학습내용 이해 사후검사

구분	N	M	SD	F	p
통제집단	33	5.09	2.097	3.438	.036
비교집단	33	5.52	2.108		
실험집단	34	6.50	2.538		

사후검사 결과 집단 간 유의미한 차이가 있는 것으로 나타났다( $p < .036$ ). 실험집단이 다른 집단에 비해 학습내용 이해 영역에서 높은 평균 점수를 보이고 있다. 따라서 동료 에이전트 기반 학습이 학습내용을 이해하는데 도움이 된다고 볼 수 있다. 동료 에이전트와 함께 개인별로 수업을 진행하고 보충학습을 실시하게 되므로 학습에 긍정적인 영향을 주는 것으로 보인다.

## 5.2 문제해결 전략 변인

학생들의 문제해결 전략에 대한 사전 수준이 동일하지 검증하기 위해 사전검사를 실시하였다. 각 집단별 차이를 살펴보기 위해 분산분석을 실시하였으며, 그 결과는 <표 8>과 같다.

<표 8> 문제해결 전략 사전검사

구분	N	M	SD	F	p
통제집단	33	5.30	2.325	.088	.916
비교집단	33	5.15	2.551		
실험집단	34	5.06	2.322		

문제해결 전략 사전검사 결과 집단 간에 유의미한 차이가 없는 것으로 나타났다( $p < .05$ ).

9차시의 수업 처지 후 각 집단별로 문제해결 전략 영역의 차이를 살펴보기 위해 사후검사를 실시하였으며, 분산분석 결과는 <표 9>와 같다.

<표 9> 문제해결 전략 사후검사

구분	N	M	SD	F	p
통제집단	33	3.97	3.067	3.190	.046
비교집단	33	4.24	2.398		
실험집단	34	5.50	2.440		

사후검사 결과 집단 간 유의미한 차이가 있는 것으로 나타났다( $p < .05$ ). 실험집단이 다른 집단에 비해 문제해결 전략 영역에서 높은 평균 점수를 보이고 있다. 또한 통제집단은 매우 낮은 점수와 함께 표준편차가 높게 나타났다. 따라서 동료 에이전트 기반 학습이 문제해결 전략을 향상시키는데 도움이 된다고 볼 수 있으며, 전통적인 수업 방식은 문제해결 전략을 수업하는데 학생별 수준 차이가 다른 집단에 비해 크게 나타났다. 동료 에이전트 학습은 동료 프로그래밍 수업 모형에 따라 전략에 대한 방안을 학생이 제시해야 하므로 전략에 대한 수업 효과가 있는 것으로 보인다.

## 5.3 자기조절 변인

학생들의 자기조절에 대한 사전 수준이 동일하지 검증하기 위해 사전검사를 실시하였다. 각 집단별 차이를 살펴보기 위해 분산분석을 실시하였으며, 그 결과는 <표 10>과 같다.

<표 10> 자기조절 사전검사

구분	N	M	SD	F	p
통제집단	33	80.52	13.424	1.491	.230
비교집단	33	75.64	15.463		
실험집단	34	74.41	16.652		

자기조절 사전검사 결과 통제집단 평균이 높게 나타나기는 했으나, 통계적으로 집단 간에 유의미한 차이가 없는 것으로 나타났다( $p < .05$ ).

9차시의 수업 처지 후 각 집단별로 자기조절 영역의 차이를 살펴보기 위해 사후검사를 실시하였으며, 분산분석 결과는 <표 11>과 같다.

<표 11> 자기조절 사후검사

구분	N	M	SD	F	p
통제집단	33	79.45	13.396	.124	.884
비교집단	33	80.94	15.350		
실험집단	34	79.50	12.626		

사후검사 결과 비슷한 평균을 보이고 있으며, 통계적으로 집단 간 유의미한 차이가 없는 것으로 나타났다(p<.05).

자기조절 영역의 사전검사와 사후검사는 동일한 설문지로 실시되었으며, 집단 내에서의 사전검사와 사후검사의 변화를 살펴보기 위해 t-test를 실시하였다. 그 결과는 <표 12>와 같다.

<표 12> 자기조절 사전사후검사 t-test

구분	N	사전		사후		t	p
		M	SD	M	SD		
통제집단	33	80.52	13.424	79.45	13.396	.423	.675
비교집단	33	75.64	15.463	80.94	15.350	-3.357	.002
실험집단	34	74.41	16.652	79.50	12.626	-2.525	.017

자기조절 영역에 대한 사전사후검사 결과 비교 집단과 실험집단은 유의미한 차이를 보이고 있다(p<.05). 비교집단과 실험집단은 사전검사보다 사후검사 점수가 높게 나타나 동료프로그래밍 수업 방식이나 자기조절 대화가 자기조절 영역에 긍정적인 영향을 준 것으로 나타났다.

## 6. 결 론

본 연구는 동료 에이전트에 문제해결력 향상을 위한 교수학습 모형을 적용하고 교육적 효과가 있는지를 살펴보았다. 동료 에이전트는 프로그래밍 학습을 위해 동료 프로그래밍 방식을 기본 모형으로 하고 있다. 여기에 문제 해결력 신장을 위한 교수학습 모형을 하위 요소별로 개발하였다. 학습내용 이해, 문제해결 전략, 자기조절이라는 세 가지 하위 요소에 대해 동료 에이전트의 역할을 설정하였다.

문제해결력에 대한 동료 에이전트의 학습 효과를 살펴보기 위해 전통적 수업 방식의 통제집단, 동료 프로그래밍 수업 방식의 비교집단, 문제해결력 기반 동료 에이전트 수업 방식의 실험집단을 구성하여 9차시의 수업 처지를 실시하였다.

각 하위 요소별로 분산분석을 실시한 결과 학습내용 이해와 문제해결 전략 영역에서 긍정적인 효과가 있는 것으로 나타났으며, 자기조절 영역 역시 비교집단과 실험집단이 향상되는 것을 알 수 있었다.

본 연구를 통해 동료 에이전트가 문제해결력을 향상시키는데 효과가 있음을 알 수 있다. 프로그래밍 교육에 있어서 문제해결력이라는 요소는 중요한 의미를 가지고 있다. 효과적인 프로그래밍 교육을 위해 문제해결력 향상을 위한 다양한 교수학습 모형과 학습 시스템에 대해 지속적인 연구가 필요하다.

## 참 고 문 헌

- [1] White, G. L. & Sivitanides, M. P. (2002). A Theory of the Relationships between Cognitive Requirements of Computer Programming Languages and Programmer's Cognitive Characteristics. *Journal of Information Systems Education*, 13(1), 59-66.
- [2] 김미량 (2002). 컴퓨터 프로그래밍 교육에 적용 가능한 효과적 교수방법의 탐색적 대안. *한국컴퓨터교육학회 논문지*, 5(3), 1-9.
- [3] Davenport, D. (2000). Experience Using a Project-Based Approach in an Introductory Programming Course. *IEEE Transactions on Education*, 43(4), 443-448.
- [4] Van Gorp, M. J. & Grissom, S. (2001). An Empirical Evaluation of Using Constructive Classroom Activities to Teach Introductory Programming. *Computer Science Education*, 11(3), 247-260.
- [5] 이충기·홍석원 (2004). 웹 기반 프로그래밍 과목의 효과적인 강의 전략. *한국컴퓨터교육학회 논문지*, 7(3), 1-14.

- [6] Brusilovsky, P. (2003). Adaptive navigation support in educational hypermedia: The role of student knowledge level and the case for meta-adaptation. *British Journal of Educational Technology*, 34(4), 487-497.
- [7] Butz, C. J., Hua, S. & Maguire, R. B. (2006). A Web-based Bayesian Intelligent Tutoring System for Computer Programming. *Web Intelligence and Agent Systems: An International Journal*, 4(1), 77-97.
- [8] Canfora, G., Cimitile, A. & Visaggio, C. (2004). Working in Pairs as a Means for Design Knowledge Building: an Empirical Study. *Proceedings of International Workshop on Program Comprehension*, Bari, Italy, 62-69.
- [9] 한건우 · 이은경 · 이영준 (2006). Pair Programming이 학업성취도와 학습동기전략에 미치는 영향. *한국컴퓨터교육학회 논문지*, 9(6), 19-28.
- [10] 한건우 · 이은경 · 이영준 (2006). 프로그래밍 교육에서 동료 에이전트가 학업성취도와 자기효능감에 미치는 영향. *한국컴퓨터교육학회 논문지*, 10(5), 43-51.
- [11] 한건우 (2007). 프로그래밍 교육에서 문제해결력 신장을 위한 동료 에이전트 시스템 개발. 박사학위 논문, 한국교원대학교.
- [12] Schneider, J. G. & Johnston, L. (2003). eXtreme Programming at universities: an educational perspective. *International Conference on Software Engineering*, Portland, Oregon, USA, 594-599.
- [13] Srikanth, H., Williams, J. G., Wiebe, E., Miller, C. & Balik, S. (2004). On Pair Rotation in the Computer Science Course. *Proceedings of the 17th Conference on Software Engineering Education and Training*, Norfolk, VA, USA, 144-149.
- [14] Smith, T., Affleck, G., Lees, B. & Branki, C. (1999). Implementing a Generic Framework for a Web-based Pedagogical Agent. *The 16th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education*, QUT, Brisbane, Queensland, Australia.
- [15] Bloom, B. S. (1954). *Taxonomy of Educational Objectives*(Hand Book 1: Cognitive Domain), N.Y.: Longman Inc.
- [16] Glaser, R. (1992). *Expert knowledge and processes of thinking*. In D.F. Halpern (Ed.), *Enhancing thinking skills in the sciences and mathematics*(63-75). Hillsdale, NJ: Erlbaum.
- [17] Sugrue, B. (1994). *Specifications for the Design of Problem-Solving Assessments in Science*. CSE Technical Report 387, CRESST/University of California.
- [18] Herl, H. E., O'Neil, Jr. H. F., Chung, Gregory, K.W.K. et al. (1999). *Final Report for Validation of Problem-Solving Measures*. CSE Technical Report 501, CRESST/University of California.



## 한 건 우

1994 한국교원대학교  
컴퓨터교육과(교육학학사)  
2004 한국교원대학교  
컴퓨터교육과(교육학석사)

2007 한국교원대학교 컴퓨터교육과(교육학박사)

2007~현재 선부중학교 컴퓨터교사

관심분야: 프로그래밍 교육, 문제해결력

E-Mail: flatfish@paran.com