

## 컴퓨터와 수학교육에서 환경의 설계

김 화 경\*

이 논문에서는 컴퓨터와 수학교육을 바라보는 두 가지 서로 다른 입장에 대하여 알아보고, 그 중 구성주의적(constructivist) 입장에서 컴퓨터 환경의 설계를 논의한다. 이를 위하여 컴퓨터와 수학교육을 위한 바람직한 기본 원리를 구성주의(constructivism) 시각으로 살펴본다. 나아가 이러한 바람직한 기본 원리를 '함수화'라는 용어를 사용하여 규정하고, 이를 통해 컴퓨터 환경 설계 방법을 모색해 본다.

이 연구를 통하여 우리는 설계 과정에 대한 논의가 부족한 상태로 학교 현장에 도입되는 기존 컴퓨터용 소프트웨어들에 대한 문제점을 인식하고, 새로운 환경의 설계 원칙과 과정에 대해 논의하고, 실제 구현한다. 그 예로 LOGO와 DGS를 새로운 대상의 도입을 통하여 통합하는 방안에 대하여 논의한다. 이 논문에서 구현한 새로운 대상은 LOGO와 DGS의 함수화 명령 체계를 모두 이용하여 만들 수 있고, 동시에 LOGO와 DGS의 함수화 명령 체계로 변화하고 조작할 수 있는 대상이다.

### 1. 들어가며

지식 기반 정보화 사회의 도래와 함께 컴퓨터 등의 기술공학적 도구와 수학교육과의 관계에 대한 여러 논의가 있어 왔다. 종이와 활자의 발명이 교육에 변화를 가져온 것처럼, 인터넷 네트워크를 통한 정보 전달 방법의 변화와 컴퓨터의 막강한 정보 처리 능력은 정보화 사회에 맞는 수학교육에 대한 연구를 요구한다.

컴퓨터와 수학교육에 관한 이전의 연구들은 크게 두 가지 흐름으로 나누어 볼 수 있다. 하나는 '컴퓨터를 매개로 하는 교수'를 강조하는 입장으로 Jonassen(1996)은 이러한 관점을 컴퓨터 보조 교수(computer-assisted instruction; CAI)이라고 설명한다. CAI에서 컴퓨터는 미리 지정

된 지식이나 기술을 학생들이 습득할 수 있도록 프로그램 되어 있다. 비록 시대별로 CAI의 형태는 바뀌었지만(drill and practice, tutorials, intelligent tutoring systems), 공통적으로 '컴퓨터로부터의 교수'의 관점을 취하고 있다. 다른 하나는 '컴퓨터와 함께 하는 학습'을 강조하는 것으로 Jonassen(1996)은 이러한 관점을 구성주의적(constructivist) 시각이라고 설명한다. 구성주의적 시각에서는 컴퓨터가 단순히 정보전달 수단이 아니라, 컴퓨터와 함께 학생들이 지식을 구성해 나갈 수 있다는 점을 강조한다.

두 가지 입장의 근본적인 차이점은 전자는 교수(teaching)를 그리고 후자는 학습(learning)을 강조한다는 점이다. Papert(1993)는 교수와 학습 중 교수를 보다 중시하는 관점을 교수주의(instructionism), 학습을 보다 중시하는 관점을 constructionism<sup>1)</sup>으로

\* 서울대학교 대학원, indices@korea.com

1) constructionism은 우리말로 구성주의로 번역될 수 있겠으나 constructivism과 구별하기 위하여 영어로 제시한다.

로 구별하여 부르고 있다. 교수주의에서는 더 나은 교육을 위해서는 교수법의 개선이 필수적이라고 여기고, 교수법의 개발을 중시한다. 이에 비하여, constructionism에서는 학습이 일어날 수 있는 교수 환경의 설계와 개발을 중시한다. 이러한 서로 다른 관점의 차이는 교육용 컴퓨터 환경에 대한 이해의 차이로 나타난다. 교수주의 관점에서는 컴퓨터 환경을 정보 전달을 위한 도구로 바라보아 소프트웨어(software)라는 용어를 주로 사용하며, 구성주의 관점에서는 컴퓨터 환경을 지식 구성을 위한 환경으로 여겨 마이크로월드(microworld)라는 용어를 사용한다. 전자는 효과적인 정보의 전달을 위한 소프트웨어의 다양한 기능을 중시하고, 후자는 의미 있는 지식의 구성을 위한 마이크로월드의 구조를 중시한다.

우리는 컴퓨터를 기존의 수학 정보를 전달하기 위한 도구로 보는 것을 넘어 수학적 지식을 구성하기 위한 환경이 되기를 기대한다. 이 글을 통해 우리는 constructionism의 관점에서 컴퓨터와 수학교육을 이론적으로 고찰하고, 컴퓨터와 수학교육에서의 바람직한 학습 원리를 찾아보고, 이를 이용한 컴퓨터 환경 설계에 대하여 알아보려고 한다. “배고픈 사람에게 물고기를 줄 수도 있지만, 그에게 낚싯대와 물고기를 잡을 수 있는 방법을 가르치는 것이 더 낫다”라는 아프리카 속담은 constructionism의 시각을 잘 설명하고 있다. 여기서 ‘낚싯대’는 의미 있는 지식의 구성이 가능한 좋은 컴퓨터 환경, 즉 마이크로월드이며, ‘물고기를 잡을 수 있는 방법’이란 마이크로월드를 통해 많은 지식을 이끌어내는 교수-학습 전략과 방법이다(Papert, 1993). Constructionism에서는 좋은 마이크로월드 환경에서 적절한 교수-학습 전략을 통한 풍성한 지식의 구성을 지향한다.

구체적으로 이 글은 기존의 컴퓨터 환경의

장점을 분석하여, 이를 극대화할 수 있는 원칙을 살펴보고, 컴퓨터 환경을 제시한다. 이 글의 의의는 컴퓨터 환경의 설계 원칙과 절차를 제시한다는 점이다. 기존의 컴퓨터 환경이 구성 원리나 절차를 알 수 없는 완성품으로 제시되는 데 비하여, 이 연구를 통해 우리는 컴퓨터 환경 설계의 원칙과 절차를 일부 밝혀 보고 실제로 구현하는 것을 목표로 한다.

## II. 이론적 배경

먼저 구성주의로 번역될 수 있는 두 가지 영어 단어인 *constructivism*과 *constructionism*의 유사점과 차이점을 알아보자. *Constructivism*은 Piaget 등에 의해 시작된 인지 이론으로, 지식은 학습자에 의하여 구성된다고 주장한다. 이 이론은 학습자는 읽거나 들어서 지식을 전달받게 된다는, 즉 지식이 학습자에게 전이된다는 고전적인 입장에 반대한다. Piaget(Ackerman, 2004, 재인용)는 지식이란 학습자가 복잡한 지식의 구조를 만들어가는 것이며, 학습자는 자신이 듣고 배운 것을 동화와 조절을 통해 자신에게 의미 있게 만들어 간다고 주장한다. *Constructivism*에서 놀이 활동을 통한 학습에 대해 많이 언급하고 있기는 하지만 Piaget 이후의 대부분의 *constructivism* 연구는 교수 환경에서의 학습에 집중되어 있다.

이에 비하여 *constructionism*은 *constructivism*의 이론을 바탕으로 보다 실천적인 방법을 함께 제시한다. Piaget와도 공동연구를 한 적이 있는 Papert(1980)는 학생들이 무언가를 만들 때 보다 깊은 학습이 일어난다는 점에 주목한다. Papert(1980)는 만드는 활동을 통하여 학습자는 새로운 문제에 직면하게 되고, 만들기 위하여 그 문제를 해결하려고 노력하며, 또한 이

러한 과정을 통해 배운다고 말하고 있다. 인공물(artifact)을 구성하는 물리적 구성주의를 거쳐서 정신적 구성주의를 강조한다. 예를 들어 주어진 컴퓨터 환경에서 ‘별’을 만들면서, 만들기 위해서 ‘별’에 내재하는 수학적 지식인 외가의 개념을 내면화하고 구성해 간다는 것이다. 이러한 인공물의 구성에 주목하는 Papert의 constructionism은 물리적 구성을 위한 환경을 강조한다는 면에서 이후 컴퓨터와 수학교육의 중요한 기본 이론이 되었다. Kafai, Resnick (1996)은 constructionism을 다음과 같이 설명하고 있다.

Constructionism은 학습이론이면서 또한 교육을 위한 전략이다. 기본적인 관점은 지식은 단순히 교사에게서 학습자로 전달되는 것이 아니라, 학습자의 마음속에서 활발하게 다시 구성된다는 Piaget식 구성주의(constructivist)이다. 즉, 아동들은 아이디어를 얻는 것이 아니라 만드는 것이며, 따라서 constructionism은 학습자들이 외부의 인공물들(로봇, 시, 모래성, 컴퓨터 프로그램)을 활발하게 만드는 여러 환경에 관심이 많다. 즉 constructionism은 개인적으로 의미 있는 인공물을 구성하며 또한 그러한 맥락 속에서 지식의 구성을 추구하는 두 종류의 구성과 관계된다. (Kafai, Resnick, 1996: 1)

Constructionism 즉 구성주의<sup>2)</sup>는 학습자 혹은 아동이 인공물을 직접 만들 수 있는 놀이 공간(playground)을 필요로 한다. 이러한 놀이 공간을 컴퓨터 세계 속에 구현한 것이 바로 마이크로월드이다. Papert(1980)는 인공지능(AI) 연구 과정에서 마이크로월드 LOGO를 만들어 냈는데, 이는 초기의 구성주의를 대표하는 마이크로월드이다. 이후 컴퓨터 환경의 급속한 발전

과 더불어 마이크로월드와 그의 특징에 대한 많은 논의가 있었다. 예를 들어, Edwards(1995)는 마이크로월드를 설계에 초점을 맞춘 ‘구조적인 면’에서의 특징과 학생들의 활동에 초점을 맞춘 ‘기능적인 면’에서의 특징으로 나누어 자세하게 설명하고 있으며, 또한 Dugdale (Edwards, 1995, 재인용)은 마이크로월드의 특징을 다음과 같이 제시하고 있다.

- ① 학생들에게 탐구하고 조작할 모델이 제시돼 있다.
- ② 수학은 그 모델에 내재되어 있다. 즉, 그 모델은 내재된 수학의 직접적 표현이다.
- ③ 피드백은 직접적이고, 적절하며, 진단적이며, 시각적이다. 그래서 학생들은 한번에 무엇이 에러이고 어떻게 올바른 답을 얻을지를 알 수 있다. 수학과 관련 없는 불필요한 군말이나 트릭은 없다.
- ④ 그 모델은 다양한 수학적 지식과 능력을 가진 학생들에게 풍부한 탐구 환경을 제시한다. 그러나 학생들이 수학을 더 많이 적용하면 적용할수록 그들이 할 수 있는 것은 더 많아진다. (Edwards, 1995: 141)

실제로 이용되는 대표적 마이크로월드의 예를 알아보면 기호적 마이크로월드인 LOGO (Papert, 1980), StarLogo(Resnick, 1994), Boxer (diSessa, 2000), JavaMAL(조한혁, 2003)나 마우스 끌기(dragging)를 통한 직접 조작 환경인 Geometer’s Sketchpad(Goldenberg, Cuoco, 1998; Serra, 1997), Cabri(Laborde, Mariotti, 2001), 도시 경제학을 위한 도시 시뮬레이션인 SimCity (Adams, 1998) 등이 있다. 이러한 마이크로월드의 일련의 예를 살펴보면 보다 구체적으로 그들의 공통된 특징을 보다 구체적으로 찾을 수 있다.

2) 이후 이 글에서는 constructionism을 구성주의라는 용어로 나타내겠다.

① 첫 번째 특징은 그 자체가 새로운 언어체계를 가지고 있다는 점이다. 언어적인 구조를 갖는 도구는 그 언어를 사용하는 사람에게 새롭고 복잡한 언어적 표현의 세계를 만들 수 있도록 한다. 기존의 교육용 소프트웨어가 가르칠 개념과 정보가 미리 담겨져 있는 그릇이었다면, 마이크로월드에는 대상과 조작이 있는 수학적 언어 체계로 학습자는 기존의 개념과 지식 등을 새롭게 해석하고 구성할 수도 있다. 예를 들어, LOGO의 거북 언어는 기존의 도형을 ‘가자’와 ‘돌자’라는 언어로 표현하고, 또한 이러한 표현을 통해 컴퓨터와 인간은 의사소통한다(Sutherland, 1995). 또한 이렇게 학습자가 마이크로월드에서 구성한 표현은 학습자의 사고와 반성을 위한 매우 중요한 도구가 된다(Brizuela, Carraher, Schliemann, 2000). 즉, 언어적 표현을 통한 학습자와 교사, 컴퓨터 사이의 의사소통내지 상호작용의 과정은 그 자체로 하나의 사고체계를 형성하며, 언어적 표현은 오류 분석 등을 통해 자신의 행동에 대한 반성적 사고를 할 수 있는 기회를 제공한다.

② 두 번째 마이크로월드의 특징은 ‘구성(construction)’과 ‘조작(manipulation)’의 강조이다(Eisenberg, 1995). 여기서 구성이란 명령을 통해 새로운 인공물을 만들어 나가는 과정을 말하고, 조작이란 마우스 끌기와 같이 만들어진 인공물에 변화를 가하는 것을 말한다. 예를 들어, LOGO 환경에서 ‘별’을 그리는 명령을 만들기, GSP나 Cabri와 같은 동적 기하 환경(dynamic geometry system; DGS)에서 점들의 종속성을 이용하여 ‘평행사변형’을 작도하는 것이 구성이며, 이미 만들어진 ‘별’이나 ‘평행사변형’에 동적인 변화를 주는 것이 조작이다. 이 때, LOGO에서는 언어적 명령이, DGS에서는 마우스 끌기가 조작의 수단이 된다. 이렇듯 마이크로

로월드는 인공물을 구성할 수 있는 환경인 동시에 만들어진 인공물을 조작할 수 있는 환경이다. 차이점은 LOGO를 비롯한 기호적 마이크로월드는 구성이 보다 강조되지만, DGS를 비롯한 동적 마이크로월드는 조작이 좀더 강조된다는 것이다.

Resnick(1995)은 좋은 컴퓨터 환경은 정보를 기존의 표현방식으로 제공하는 것을 넘어 이전에는 제공할 수 없었던 표현과 새로운 사고방법을 제공하는 것이라고 말한다. 기존의 수학 교육용 소프트웨어가 기존 수학의 복사판이라면 마이크로월드는 새로운 수학적 표현을 통해 다시 쓴 개정판이다. 예를 들어 LOGO의 거북 기하(turtle geometry; Abelson, diSessa, 1980)는 기하를 보는 새로운 방법을 제공하였고, DGS는 점과 점들 사이의 연결성을 통한 동적 기하(dynamic geometry)를 제공하여 새로운 형태의 기하 탐구를 가능하게 하였다. 또한 Resnick(1995)은 또한 컴퓨터를 이용한 교수-학습 활동과 컴퓨터 환경을 설계할 때는 학습자에 대한 이해, 학습 영역(수학)에 대한 이해, 컴퓨터 패러다임에 대한 이해가 필요하다고 지적하고 있다. 같은 맥락으로 장경운(1996)은 컴퓨터의 등장과 더불어 수학, 교수할 수학, 수학 교수 방법, 수학 학습의 각각의 변화에 주목하고 있다. 즉, 교수할 수학이 정해져 있고 이를 위한 컴퓨터 환경을 만드는 것이 아니라 교수할 수학도 컴퓨터 환경에 따라 달라질 수 있다는 것이다.

우리는 앞서 살펴본 배경과 연구들을 바탕으로 컴퓨터와 수학교육의 환경에서 일어나는 바람직한 학습에 대하여 알아보고 그 속에서 컴퓨터 환경 설계에 적용 가능한 원칙을 찾아보려 한다.

### III. 컴퓨터 환경의 설계

Marton & Booth(1997)은 구별하여 두드러지게 하는 행동에 초점을 두어 ‘구별하는 것’이 학습의 기본이라 주장 한다. 우리가 어떤 것을 구별해 낼 수 없다면 우리는 아무 것도 알 수 없다고 말한다. 예를 들어 도형을 전체 모양으로 보는 대신에 도형의 구성요소인 점과 선분을 구별하는 것이나 수를 한 자리 수, 두 자리 수와 같이 자리값에 따라 구별하는 것은, 도형이나 수를 다르게 보이게 하며, 이들을 통해 할 수 있는 것을 바꿔준다. Marton & Booth (1997)에 의하면 이렇게 학습을 구별을 만드는 것이라는 관점에서 접근하면 반드시 변화 차원(dimension-of-variation)을 고려해야 한다.

일반적 측면으로 특정한 상황을 경험하기 위해서는 우리는 일반적 측면을 경험해야 한다. 이러한 측면이 변화 차원들에 해당한다. 특정 상황에서 우리가 관찰하는 것은 우리는 그러한 차원에서의 값(values)으로 말없이 경험한다. 어떤 것을 경험하는 확실한 방법은 의식적으로 구별되고 동시에 주목하는 변화 차원에 의해서 이해하는 것이다. 또한 서로 다른 변화 차원들 사이의 관계로 이해하는 것이다. 어떤 것을 경험하는 다른 방법은 같은 것을 경험하는 다른 방법인듯이, 그것을 경험하는 방법에서 변화는 일련의 변화 차원으로 기술될 수 있다. (Marton, Booth, 1997: 108)

Mason & Johnston-Wilder(2004)는 이러한 변화 차원의 개념을 구체적으로 제시하고 있다. 이들은 변화 차원의 개념을 자세히 변화 가능 차원(dimension-of-possible-variation)과 대응하는 변화 가능 영역(range-of-permissible-change)으로 설명하고 있다. 여기서 ‘가능’이라는 용어를 사용한 이유는 사람마다 인식할 수 있는 차원과 그에 대응하는 영역이 다를 수 있기 때문이다. 예들어 식  $(x-1)(x+1) = x^2 - 1$ 에서  $x$ 가 가질 수 있는 값의 범위는 사람들마다 정수, 유리수, 실수, 복소수, 정사각행렬 등으로 다를 수 있으며 이것이  $x$ 라는 변화 차원에 대응하는 변화 가능 영역이다. Mason & Johnston-Wilder (2004)는 이러한 변화 가능 차원과 그에 대응하는 변화 가능 영역을 밝히는 것이 교육적으로 유용한 일이라고 지적하고 있다.<sup>3)</sup>

#### 1. 변화와 함수적 관점

학습을 어떤 것을 주목하여 구별하는 방법을 배우는 것이라는 관점을 취하면 학습에서 중요한 것은 ‘변화(variation)’이다. 이런 변화에 주목하는 것은 역사적으로 수학을 탐구하는 오래된 방법이다.<sup>4)</sup> ‘만약 이곳을 변화시키면 어떻게 될까?’라는 ‘What if?’라는 질문을 통해 탐구활동을 계속적으로 자극할 수 있으며 이는 또한 문제 제기의 기본 원리이다(Brown & Walter, 1993).

또한 변화에 관련하여 문제를 만들고 확인하고, 실험하는 것은 컴퓨터 환경의 특징과 상당

3) 이 글이 컴퓨터 환경 설계와 그 원칙에 대한 연구를 목적으로 한다. 이 글에서 자세히 살펴볼지 못한 학교수학과 관련한 변화 가능 차원과 그에 대응하는 영역에 대한 연구가 필요하다.

4) 원뿔곡선을 최초로 연구한 메나에크무스(B.C 375-325)는 원뿔을 모선에 수직인 평면으로 잘라내면서, 꼭지각의 크기에 따라 그 단면이 타원, 포물선, 쌍곡선이 나온다는 사실을 밝혔는데, 아폴로니우스는 하나의 원뿔에서 절단면의 기울기를 변화시킴으로써 세 종류의 원뿔곡선을 모두 만들어 내었다(남호영, 정춘희, 김세식, 원유미, 2001). 원뿔곡선을 얻기 위하여 메나에크무스는 세 개의 원뿔을 잘랐으며, 아폴로니우스는 하나의 원뿔에서 절단면의 기울기를 변화시켜 얻었다. 두 경우 모두 변화를 통한 수학적 탐구를 진행하였지만, 아폴로니우스의 변화 방법이 더 효율적이다.

한 부분에서 부합한다. Goldenberg & Cuoco (1998)는 이러한 변화의 관점으로 문제를 재해석하는 것은 마이크로월드를 통한 수학적 탐구 활동의 기본이 된다고 말하고 있다. 유사한 맥락에서 김화경, 조한혁(2004)은 DGS 환경에서의 함수적 정의에 대하여 논의하고 있다. 이와 같이 학교수학에서의 명제와 문제 등을 역동적인 함수적 관점에서 해석하고, 또한 함수적 시각으로 설계된 컴퓨터 환경에서 구성, 조작, 탐구될 때, 문제에 대한 시각적인 이해뿐만 아니라 바람직한 동적인 탐구는 비로소 가능해진다. 컴퓨터와 수학교육을 컴퓨터 환경에서의 수학교육과 더불어 컴퓨터 환경의 설계까지 영역을 확장시켜본다면, '변화'와 관련된 역동적인 함수적 관점은 시사하는 바가 크다. 즉, 변화와 관련된 역동적 교수-학습 활동이 일어날 수 있도록 컴퓨터 환경을 설계해야 한다.

이제 우리는 컴퓨터와 수학교육에서 고려되어야 할 일련의 '변화'와 관련된 원칙을 하나의 용어로 나타내려 한다. 우리는 함수적으로 설계된 마이크로월드 환경에서 기존 수학적 명제나 성질, 정의 등을 변화의 관점으로 접근하여 조작, 실험하는 것, 그리고 함수적 성질을 통하여 인공물을 구성하는 것 등의 컴퓨터와 수학교육에서의 '변화'와 관련된 일련의 활동을 '함수화(functionization)'라 부르도록 하겠다. 즉, 함수화는 컴퓨터와 수학교육의 함수적 특징을 정리한 용어이다. 여기서 함수화는 컴퓨터 환경에서 수학적 명제가 탐구 가능한 명제함수로 되어 간다는 진행의 의미이며, 또한 수학교육용 컴퓨터 환경의 설계를 함수적 관점으로 시도한다는 의미이다.

'함수화'라는 용어를 특별히 사용하는 이유는 우선 함수의 중요성 때문이다. 실제 컴퓨터와 수학교육 분야의 많은 연구들(Goldenberg, Lewis, & O'Keef, 1992; Laborde & Mariotti, 2001; Stroup, 2002)이 변화라는 함수의 특징에 주목하여 컴퓨

터 환경을 살펴보고 있다. 또 다른 이유는 수학적 표현, 특히 컴퓨터를 통한 표현이 만들어지는 과정 자체가 함수적이기 때문이다. 종이와 연필을 사용하여 수학을 표현하는 행위는 손의 움직임으로 표현이 생성되는 독립과 종속의 함수적 관계이며, 마찬가지로 컴퓨터 환경은 인간의 두뇌에서 시작하여 손을 움직여 입력 장치를 작동시키고 그 입력 장치의 전기적 신호는 중앙제어 장치를 거쳐 출력 장치가 작동하게 되는 일련의 과정은 각 단계 함수의 합성 과정이다.

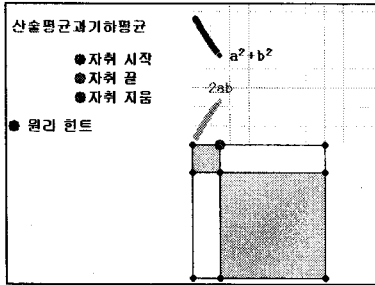
이제 함수화 관점을 다시 학습 상황 설계와 컴퓨터 환경의 설계의 측면으로 나누어 예를 통해 살펴보자.

## 2. 컴퓨터와 수학교육과 함수화

함수화는 명제를 변화 가능한 것으로 바라보는 것에서 시작한다. 예를 들어, 명제

'자연수  $n$  에 대해  $1+2+\dots+n=n(n+1)/2$ ' 는 변수를  $n$ 으로 갖는 명제함수  $p(n)$ 으로 재해석될 수 있으며, '삼각형의 내각의 합은 180도이다'라는 명제나 ' $a^2+b^2 \geq 2ab$ ' 라는 부등식도 모든 삼각형과 수에서 성립하는 넓은 의미의 명제함수이다. 학교수학에서 다루는 많은 문제나 정리들이 종이와 칠판 위에서 정적으로 표현되어 제시되지만, 컴퓨터 환경에서는 동적인 변수를 갖는 함수로 재해석되어 제시될 수 있다. 이러한 생각에서 Cuoco, Goldenberg, & Mark(1998)은 기하학적 탐구 명제를 도형공간에서 정의되는 함수로 이해하였다. 또한 Parzysz (1988)에 의하면 도형공간에서는 각각의 그림은 점으로 간주되며, 합동 등의 어떤 성질을 공유하는 점들은 동일시 될 수 있는 위상 공간이다. 이러한 관점에서 학교 대수의 수와 식 그리고 기하의 도형을 점으로 간주하여 마이크로월드 에 나타낼 수 있다. 예를 들어, 다음 [그림 III

-1]은 부등식 ' $a^2 + b^2 \geq 2ab$ '을 DGS 환경에서 대수와 기하 표현으로 구현한 것이다.

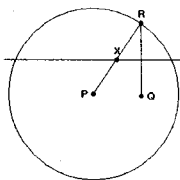


[그림 III-1]

학생들은 DGS에서 마우스로 빨간 큰 점을 끌면서  $a^2$ 에 대응되는 (작은)정사각형과  $b^2$ 에 대응되는 (큰)정사각형의 변화에 따른  $a^2 + b^2$ 과  $2ab$  값의 변화를 함수의 그래프로 관찰할 수 있다. 산술혁명을 넘어 20세기 초의 수학교과운동에서 페리와 무어는 각각 실험기하와 수학에서의 실험실법을 주장하였듯이, 컴퓨터 환경은 조건을 변화시켜 새로운 상황을 만들고 또한 그 상황에서 자신의 추측을 확인할 수 있는 함수적 활동이 용이한 환경이다. 다른 예를 살펴보자. 아래 문제는 이차곡선의 자취에 관한 것이다.

<문제>

한 점  $P$ 을 중심으로 하는 반지름이 일정한 원과 그 내부의 점  $Q$ 가 주어졌다고 가정하자. 이 때, 원 위의 동점  $R$ 에 대하여 선분  $QR$ 의 수직이등분선과 선분  $PR$ 과의 교점을  $X$ 라고 하자. 동점  $R$ 이 원 위를 움직일 때, 점  $X$ 의 자취가 무엇인지 구하고 그 이유를 설명하여라.



[그림 III-2]

이 <문제>는

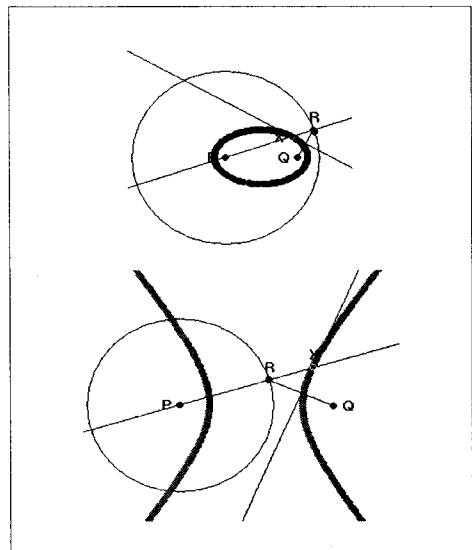
$$\overline{PX} + \overline{QX} = \overline{PX} + \overline{RX} = \overline{PR}$$

임을 이용하여 점  $X$ 는 두 점  $P, Q$ 로부터 거리의 합이 일정한 점이고, 따라서 자취가 타원이 됨을 알 수 있다.

이 정적인 문제를 움직이고 조작 가능한 함수화 관점에서 변화를 생각하자. 한 예로 주어진 <문제>에서 '원의 내부의 점  $Q$ '라는 조건을 변화시켜보자. 즉,

'만약 점  $Q$ 이 원 밖에 존재한다면'

어떻게 될지 생각해 보자. [그림 III-3]은 점  $Q$ 의 위치에 따른 점  $X$ 의 자취를 보여준다. 만약 [그림 III-3]이 지필환경에 존재한다면 두 개의 다른 문제에 대한 해답 그림이지만, DGS라는 컴퓨터 환경에서는 점  $Q$ 을 마우스 끌기를 통해 움직여 얻을 수 있는 연속선상의 '같은' 그림이다. 즉, 점  $Q$ 의 위치에 따른 각각의 함수값이 두 그림이다.



[그림 III-3]

나아가 두 그림에서 선분  $QR$ 의 수직이등분선은 각각 타원과 쌍곡선 위의 한 점에서의 접

선임을 발견하고 정당화 할 수 있다.<sup>5)</sup> 또한 이를 이용하면 주어진 타원과 쌍곡선에서 접선을 작도할 수 있다.

이 예를 통해 우리는 바람직한 학습 원리로 조작과 탐구 대상의 '변화'를 강조하는 함수화 원리 속에는 '합성'과 '역'에 대응하는 원리가 내포되어 있음을 확인할 수 있다. 여기서 '합성'이란 단편적인 지식을 전달하거나 하나의 탐구로 끝나는 에피소드를 넘어 이를 발판으로 새로운 인공물이나 새로운 명제함수를 만들 수 있어야 함을 의미한다. 예를 들어 [그림 III-2]의 정당화 방법은 쌍곡선인 경우로 확장될 수 있고 나아가 [그림 III-2]는 주어진 타원의 접선을 작도하는 방법에 관한 지식으로 확장될 수 있다. 또한 '역'의 측면에서 자취가 타원이나 쌍곡선이 되기 위한 점 Q의 위치를 실험하고 확인할 수 있는 공간이기도 하다.

이제 이러한 관점을 바탕으로 우리는 새로운 인공물을 구성하고 조작할 수 있는 컴퓨터 환경, 계속적으로 문제를 파생시킬 수 있는 수학적 컴퓨터 환경을 생각해 보자.

### 3. 컴퓨터 환경의 설계와 함수화

많은 마이크로월드 설계와 이를 통한 교육 경험을 바탕으로 Resnick, Silverman(2005)은 바람직한 설계 원칙을 제시하고 있다. 우리는 여기에 수학교육적 관점으로 함수화 설계라는 원칙을 다음 세 가지 측면에서 추가하려 한다.

첫째, 적절한 제약으로 함수적 상황이 다수 만들어질 수 있도록 설계되어야 한다. 교육용 컴퓨터 환경 설계에서 중요한 점은 너무나 자동화된 기능을 강조하게 되면 조작이나 탐구가 사라

지게 되고, 너무나 사용이 어려운 기능들은 그 기능으로 인하여 사고에 제약을 받게 된다. 적절한 조작 기능은 컴퓨터 환경 설계에 필수적이다.

둘째, 초기 구성이나 조작이 다음 단계의 구성이나 조작의 대상이 될 수 있도록 설계되어야 한다. 기본적 조작, 탐구를 이용하여 하나의 대상을 만들고 다시 그 대상을 이용하여 조작이나 탐구, 구성이 가능하도록 설계되어야 한다. 단절이 일어나지 않고 연속적인 학습의 전개를 위해 환경이 설계되어야 한다.

셋째, 시각적 표상을 만들어낸 명령을 확인할 수 있어야 한다. 즉, 입력 장치를 통하여 출력 장치에 전달되는 함수적 관계는 역으로 화면에 출력된 시각적, 동적인 함수값을 생성하는 명령을 확인할 수 있는 기능을 가지고 있어야 한다. 이러한 기능은 학습자에게 구성의 원리를 확인할 수 있게 해주며, 교사에게는 오류의 원인을 확인시켜 적절한 피드백에 제시할 수 있게 한다. 또한 학습자들 사이의 의사소통과 오류 수정을 가능하게 한다.

수학교육을 위한 바람직한 컴퓨터 마이크로월드로는 LOGO와 DGS가 대표적이다. 다음으로 우리는 이 두 환경의 특징을 함수적 관점에서 각각 찾아보고, 이 함수적 특징을 유지하면서 두 환경을 통합할 수 있는 환경을 함수화의 원리를 통해 설계하고 구현한다.

#### 가. LOGO와 DGS 환경

##### 1) LOGO 환경

Resnick(1995)은 LOGO는 기존의 기하를 보는 새로운 관점을 제공할 뿐만 아니라 절차적 사고방법을 제시하고 있는 전형적인 마이크로월드로 소개하고 있다. 일단 LOGO의 순서에

5) 타원인 경우만 생각해 보자. 만약 직선이 타원과 점 X 이외의 점 Y에서 만난다고 가정하면,  $\overline{PY} + \overline{YR} > \overline{PR}$ 이다. 그러나 점 Y가 타원 위의 점이므로  $\overline{PY} + \overline{YR} = \overline{PY} + \overline{YQ} = \overline{PX} + \overline{XR} = \overline{PR}$ 이다. 이는 모순이다. 즉, 직선은 타원과 단 한 점에서 만나므로 타원의 접선이다.

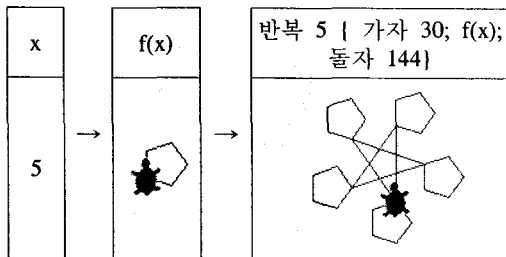


다른 절차적 명령은 거북이의 행동을 만들어내고 그 결과를 시각적으로 보여준다. 그러나 일단 구성된 인공물은 함수화 과정을 거쳐 보다 자신에게 의미 있고, 자신이 소유감을 가질 수 있는 인공물로 다시 태어난다. 예를 들어, [그림 III-4]은 별을 만들기 위한 구성을 보여준다. 여기서 ‘가자 30’의 값을 바꾼다면 그림은 어떻게 될까? ‘돌자 144’ 대신 ‘돌자 288’을 넣는다면 그림은 어떻게 될까? 라는 ‘What if?’ 질문 등이 가능하다.

명령	그림
반복 360 {가자 30; 돌자 144}	

[그림 III-4]

이러한 일련의 ‘What if?’ 질문은 LOGO 환경에서의 문제 상황에 대한 함수화를 유도한다. 더 나아가 ‘반복 5 {가자 30; f(x); 돌자 144}’와 같이 f(x)를 하나 더 쓰게 되면 어떻게 될까? 이는 x라는 변수에 따라 함수 f(x)가 정해지고, 다시 f(x)의 시각적 표현을 매개 변수로 다시 거북이의 행동을 만들어 내게 된다. 이는 함수의 합성 과정이다. 즉, [그림 III-5]는 합성함수  $x \rightarrow f(x) \rightarrow$  반복 5 {가자 30; f(x); 돌자 144}을 나타낸다.

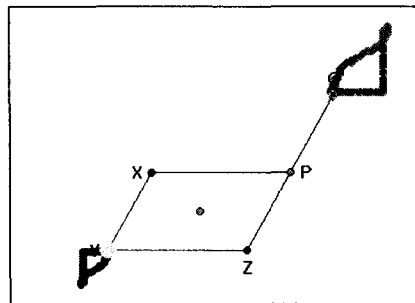


[그림 III-5]

LOGO 환경에서는 주로 언어적 명령의 구성과 조작을 통해 함수화가 일어난다는 점을 주목하자.

## 2) DGS 환경

기존의 학교기하가 완성된 그림과 움직이지 않는 정적인 그림을 통한 기하였다면, DGS 환경의 움직이는 기하는 마우스 끌기 등의 직접 조작을 통해 그림을 동적으로 구성하고 변화시키며 탐구할 수 있는 기하이다. 이는 LOGO와 마찬가지로 단지 기존 학교 기하의 새로운 표현 환경이 아니라 기하를 바라보는 새로운 동적 관점을 제공한다. 김민정, 김화경(2003)은 이러한 DGS가 가져야 할 특징과 ‘점’ 중심의 DGS 설계, 즉 점들의 연결성에 대하여 논의하고 있다. 특히 기하의 도형이 그려지는 과정을 중시하는 절차적 명령의 사용을 강조하고 있다. 특별히 ‘자취’ 기능을 이용하여 독립점의 변화에 따른 종속점의 변화, 즉 ‘이 점을 끌면 어떻게 될까’라는 문제 상황을 통한 함수화를 생각할 수 있다.



[그림 III-6]

[그림 III-6]의 점 X, Y, Z는 자유로운 끌기가 가능한 독립점이고, 점 P는 X, Y, Z와 함께 평행사변형을 이루는 점이며, 점 Q는 점 P에 대한 점 Z의 대칭점이다. 이 때, 독립점 Y를 특정한 방향으로 끌 때, 점 Q는 어느 쪽으로 얼마만큼 움직일지 생각해 보자. [그림 III-6]은

독립점 Y의 끝기의 자취와 이에 따른 점 Q의 자취를 나타낸다. 이 경우 점 Q의 움직임을 정당화하기 위해서는 점들 사이의 종속성과 합성을 인식해야 한다. DGS에서 어떤 점을 끝면 다른 점, 넓이, 성질이 어떻게 될까라는 질문은 자연스럽게 DGS는 이를 실험, 확인할 수 있는 환경이다. 주목할 것은 DGS 환경에서는 주로 끝기라는 직접 조작을 통하여 함수화가 일어난다는 점이다.

### 나. 통합 환경 설계

Eisenberg(1995)는 LOGO와 같은 프로그래밍 환경과 DGS와 같은 직접 조작 환경이 서로 대립되는 것이 아니라 상호 보완적이라는 점을 지적하고 있다. 또한 바람직한 교육용 컴퓨터 환경은 직접 조작 인터페이스와 쌍방향 프로그래밍 환경이 적절히 함께 존재하는 통합 환경이어야 한다는 점을 지적하고 있다. 이러한 통합 환경에 대한 연구로 Sherin(2002)은 프로그래밍 환경의 입장에서 DGS의 설계에 대하여 논하고 있다. 즉, DGS의 점이나 선분 등의 객체를 끝기나 메뉴라는 직접 조작 인터페이스와 더불어 프로그래밍 명령으로도 만들고 조작할 수 있도록 설계하였다. 같은 맥락에서 함수화 원리를 이용하여 새로운 대상을 도입하는 통합 환경을 설계해 보자.

#### 1) 타일환경

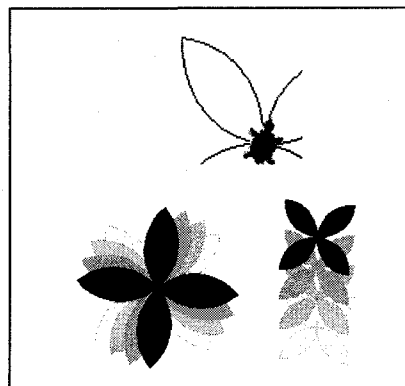
Cho, Kim, Han, Jin, Kim, & Song(2004)과 Cho, Han, Jin, Kim, & Song(2004)은 LOGO와 DGS의 통합 환경으로 자바말(JavaMAL) 마이크로월드를 설계하고 있다. LOGO와 DGS를 통합하기 위해 준동적(semi-dynamic) 대상인 타일(tile)을 설계하고 도입하고 있다. 타일은 LOGO의 언어적 명령으로도 만들 수 있고, 또한 DGS의 마우스 클릭 순서나 메뉴 등의 직접 조작 명

령으로도 만들 수 있으며, 또한 이렇게 만들어진 타일을 각각의 방식으로 조작할 수 있다. <표 III-1>은 타일의 생성, 조작 방법에 LOGO와 DGS의 명령이 공존하여 두 환경을 연결하는 중간 다리가 됨을 보여준다. 즉, 하나의 방법으로 구성하고 다른 방법으로 조작할 수 있다는 측면에서 상호 보완적 특징을 가지고 있다.

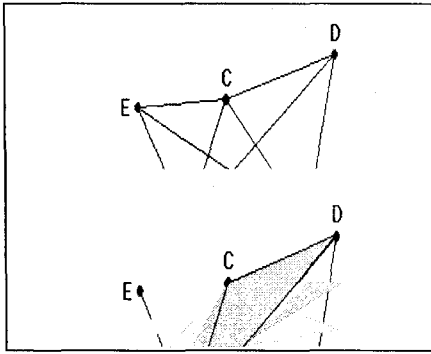
<표 III-1>

타일	구성 방법	조작 방법	유사 대상
LOGO 특성	언어적 명령	언어적 명령	거북이
DGS 특성	마우스 클릭 순서 메뉴	마우스 조작	점

대상(타일)을 구성하고 다시 그 대상(타일)을 조작하는 일련의 대상-과정(object-process)이라는 수학적 활동을 유도한다. 이는 LOGO로 만들어진 대상(별이나 다각형)을 가자와 돌자라는 언어로 움직이고 조작할 수 있으며, DGS로는 만들어진 동적인 그림의 순간을 포착하여 조작 가능한 타일로 만들어 조작할 수 있는 실험공간이다. 함수화의 입장에서 볼 때, 함수적 명령어로 만든 대상을 다시 함수적 명령어로 움직이는 함수의 합성 과정에 해당한다.



[그림 III-7]



[그림 III-8]

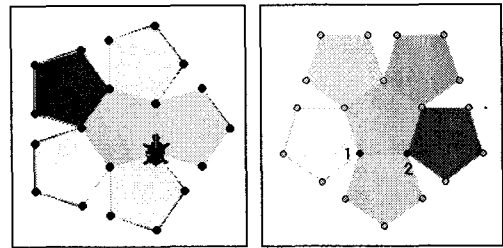
[그림 III-7]은 LOGO 환경에서 타일을 구성하고 회전, 평행이동이라는 조작을 나타낸다. 또한 [그림 III-8]은 DGS 환경에서 타일을 만들고 타일의 조작(회전이동)을 통하여 도형의 합동이나 닮음을 확인하는 과정을 나타낸다. 이렇듯 타일은 LOGO 그림은 움직이지 않는다는 정적인 측면에 동적인 움직임을 부여한 것인 동시에, DGS의 너무나 자유로운 동적인 측면에 정적인 요소를 부여하는 상호 보완적인 특징도 가지고 있다.

뿐만 아니라 LOGO와 DGS에서는 대상의 구성과 구성되는 대상 자체에 주목하도록 하지만, 타일은 만들어진 대상의 운동에 보다 주목하게 한다. 예를 들어 LOGO에서의 별이나 DGS의 평행사변형을 생각해 보자. 이들을 조작한다는 것은 별이나 평행사변형에 대한 조작이다. 이에 비하여 타일로 만든 행성타일을 타원 운동시키는 것이나, 피스톤 타일을 왕복 운동시키는 것은 대상의 운동에 대한 조작하는 것이다. 이러한 타일의 동적인 특성은 디자인을 통한 수학 학습(김남희, 2004; Kafai, 1995; Shaffer, 1997)을 가능하게 할 수 있으며, 보다 창의적 결과물 구성을 위한 환경으로 사용될 여지가 있다.

## 2) 전개도 환경

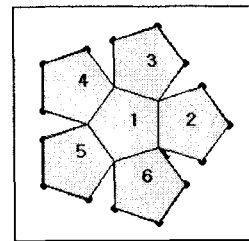
같은 방식으로 입체도형을 만들 수 있는 새

로운 대상을 도입하여 LOGO와 DGS의 통합 환경을 설계해 보자. 먼저 LOGO와 DGS 명령 방식으로 입체도형의 전개도를 만들고, 이를 컴퓨터 화면에서 접어 입체도형을 만드는 환경을 설계한다. 또한 이렇게 전개도를 접어 구성된 입체도형을 다시 LOGO와 DGS 명령으로 조작하여 움직임을 주도록 설계한다. 즉, 전개도를 만들고, 이를 접어 인공물을 구성하고, 다시 이것을 조작할 수 있는 환경이다. [그림 III-9]는 LOGO와 DGS 명령을 통하여 전개도를 만들고 그 전개도를 접어 입체도형을 만들고, 그 입체도형을 회전하는 조작을 단계별로 나타낸다.

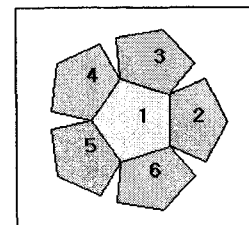


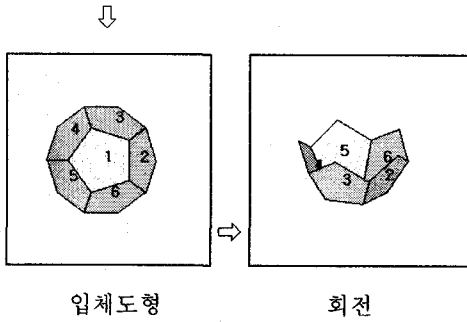
LOGO

DGS



전개도

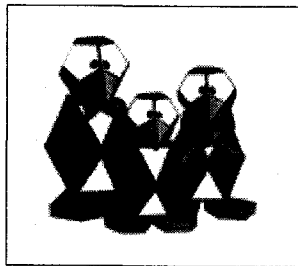




[그림 III-9]

LOGO나 DGS의 구성 방법을 통하여 만들어진 전개도는 면과 면사이의 인접관계(adjacency relation)를 고려하고, 면 사이의 각도를 지정해지면 컴퓨터 화면에서 입체도형으로 접히게 된다. 또한 접혀진 입체도형은 LOGO와 DGS 두 가지 방법으로 움직일 수 있는 대상이 된다. 직접 조작 인터페이스와 쌍방향 프로그래밍 명령을 통하여 입체도형을 재구성하고 조작하는 환경이다.

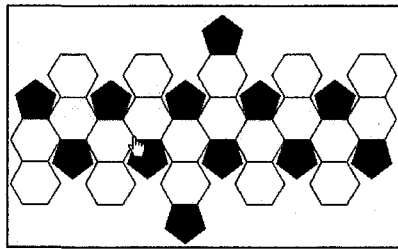
타일의 운동이 동적 디자인을 통하여 창의적 결과물을 위한 환경으로 이용될 수 있듯이, 전개도는 컴퓨터를 통해 입체도형을 접는 실험을 하고 다시 이를 프린트하여 종이로 접어 창작물을 만드는 데 이용할 수 있다. Eisenberg (2002)는 [그림 III-10]과 같이 전개도를 인쇄하여 입체도형을 만들고 또한 입체도형 끼리 붙이고 색을 칠하여 펭귄 인형을 만드는 창의적인 구성주의를 구현하고 있다.



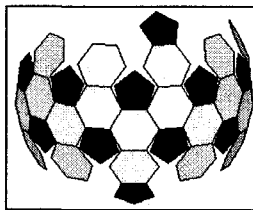
[그림 III-10]

타일 환경과 전개도 환경은 별도의 소프트웨어가 아니라 LOGO와 DGS의 통합 환경을 위하여 두 환경이 공존하는 환경 속에 존재한다. 또한 함수적 명령어를 사용하여 전개도 대상을 만들고, 입체도형을 만들어 함수적 명령으로 움직인다는 측면은 함수의 합성에 해당한다.

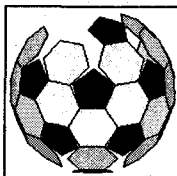
나아가 전개도가 입체도형으로 접히는 과정은 시간에 따른 함수이다. 예를 들어, 주어진 인접관계로 만들어진 입체도형에서 초기값을 '전개도(0)'으로 나타내며, 주어진 사이각으로 완전히 접힌 전개도는, '전개도(5000)'이라고 하자. 이 때, x의 값을 0부터 5000 까지 변화시키며 함수 '전개도(x)'를 변화시키면 전개도가 접혀지는 애니메이션이 일어나게 된다.



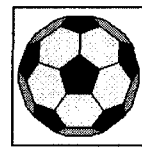
전개도(0)



전개도(2500)



전개도(3200)



전개도(5000)

[그림 III-11]

[그림 III-11]은 축구공 전개도를 그린 후, 전개도(0), 전개도(2500), 전개도(3200) 그리고 전개도(5000)의 값을 순서대로 나타낸 그림이다. 0부터 5000까지 순차적으로 실행하게 된다면 접히는 애니메이션이 일어나게 된다. 또한 전개도 환경에서 인접관계는 이산수학의 그래프(graph) 이론의 기본 관계이며, 사이각은 정사영이나 삼각함수와 같은 풍부한 학교 수학 내용과 연결되어 있다.

우리는 LOGO와 DGS의 함수적 특징을 알아 보았고, 이 특징을 유지하면서 새로운 대상의 도입을 통한 두 환경을 통합을 논의하였다. 특히, 함수적 명령으로 대상을 만들고, 다시 그 대상을 움직이는 일련의 과정은 함수의 합성 단계로 이해될 수 있으며, 특별히 전개도와 입체도형을 시간이라는 변수와 연계하여 설계하고 있다. 또한 각각의 명령을 확인할 수 있도록 설계하고 있다. 이는 함수의 역에 해당하는 것이다. 실제 만든 축구공의 명령은 이러한 언어적 명령으로 인터넷에 저장하여 컴퓨터와 학습자, 학습자 상호간의 의사소통이 가능하다.

#### IV. 마치며

이 논문에서는 바람직한 컴퓨터와 수학교육에서의 학습을 변화라는 측면을 통하여 바라보고자 하였다. 특히 기존의 컴퓨터용 소프트웨어들이 그 설계의 원칙이나 절차에 대한 소개 없이, 완성된 형태의 프로그램인 데 비하여 실제 프로그램의 설계 의도와 설계 과정을 논의하고자 하였다. 특별히 이 논문에서는 지식전달을 넘어 새로운 차원의 지식 구성을 위해 마이크로월드의 언어적인 특징과 구성 및 조작의 특징을 컴퓨터와 수학교육에서의 '함수화'로

이해하였으며, 또한 함수화 원칙에 따라 새로운 마이크로월드 설계가 가능할 수 있음을 살펴보고, '함수화'를 바탕으로 기존의 LOGO와 DGS의 통합 환경 설계를 제안하였다. LOGO와 DGS의 통합을 두 가지 프로그램 기능의 단순한 통합이 아닌, 새로운 함수적 대상인 타일과 전개도를 통한 두 가지 마이크로월드 기능의 상호 협력과 보완을 통한 통합을 제안하였다. 이 때 LOGO와 DGS의 함수적 특징이 사라지지 않도록 서로의 명령체계를 유지하면서 통합될 수 있도록 설계하였다. 이렇게 설계된 타일과 전개도라는 새로운 대상은 LOGO와 DGS의 함수적 명령으로 만들어지며, 움직이게 된다. 나아가 전개도는 시간을 변수로 하는 함수적 특징을 갖는다.

Noss, Hoyles(1996)는 컴퓨터가 수학적 의미를 보는 창이 되어야 한다는 점을 강조하고 있으며, Wilensky(1993)는 수학적 지식이 하나의 별도의 것으로 여겨지면 그 효용성이 떨어지며 개념이란 여러 가지 현상에서 발견되고 서로 연결될 때 개념의 효용성이 확인되므로 컴퓨터를 통한 물리적 현상과의 연결을 강조하고 있다. 수학적 의미를 보는 창과 수학과 물리적 현상의 연결성은 컴퓨터가 가질 수 있는 두 가지 큰 장점이다. 기존 지식에 의미를 붙여넣고, 서로 다른 지식들을 연결할 수 있도록 컴퓨터 환경을 설계하고, 수학교육에 컴퓨터를 이용하여야 할 것이다. 컴퓨터와 수학교육은 이러한 컴퓨터의 장점을 살리고, 수학교육의 본질을 보다 잘 구현할 수 있도록 연구하며 접근해야 한다.

#### 참고문헌

김남희(2004). 중등수학 탐구를 위한 예비수학 교사의 수학프로그램(GrafEq.) 활용 사례.

- 한국수학교육학회 시리즈 A <수학교육> 43(4), 405-417.
- 김민정 · 김화경(2003). DGS 동적 기하 환경에서 종속성에 의한 함수 개념 학습. *한국수학교육학회지 시리즈 E <수학교육 논문집>* 16, 67-80.
- 김화경 · 조한혁(2004). DGS 동적 기하에서의 새로운 함수적 관점의 정의. *한국수학교육학회 시리즈 A <수학교육>* 43(2), 177-186.
- 남호영 · 정춘희 · 김세식 · 원유미(2001). *원뿔에서 태어난 이차곡선*. 서울: 수학사랑.
- 장경윤(1996). 컴퓨터와 수학, 수학 교육. *수학교육학연구* 6(1), 33-44.
- 조한혁(2003). 컴퓨터와 수학교육. *한국수학교육학회지 시리즈 A <수학교육>* 42(2), 177-191.
- Abelson, H., & diSessa, A. (1980). *Turtle geometry*. Cambridge, MA: MIT Press.
- Ackerman, E. K. (2004). Constructing knowledge and transforming the world. In M. Tokoro & L. Steels (Eds.), *A learning zone of one's own: sharing representations and flow in collaborative learning environments*. Amsterdam: IOS Press.
- Adams, P. C. (1998). Teaching and learning with SimCity 2000. *Journal of Geography*, 97(2), 47-55.
- Brizuela, B., Carraher, D. & Schliemann, A. (2000). Mathematical notation to support and further reasoning. *Symposium paper, 2000 NCTM Research Pre-session Meeting*, 18.
- Brown, S. I., & Walter, M. I. (1993). *Problem posing: reflections and applications*. New Jersey: Lawrence Erlbaum Associates, Publishers.
- Cho, H., Han, H., Jin, M., Kim, H., & Song, M. (2004). Designing a microworld: activities and programs for gifted students and enhancing mathematical creativity. *Proceedings of ICME TSG 4*, 110-120.
- Cho, H., Kim, S., Han, H., Jin, M., Kim, H., & Song, M. (2004). Designing a microworld for mathematical creativity and gifted education. *The SNU Journal of Education Research* 13, 133-147.
- Couco, A. A., Goldenberg, E. P., & Mark, J. (1998). A role for geometry in general education. In R. Lehrer & D. Chazan (Eds.), *Designing learning environments for developing understanding of geometry and space*. London: Lawrence Erlbaum Associates, Inc.
- diSessa, A. (2000). *Changing minds*. Cambridge, MA: MIT Press.
- Eisenberg, M. (1995). Creating software application for children: some thoughts about design. In A. A. diSessa, C. Hoyles, R. Noss, & L. Edwards (Eds.), *Computers and exploratory learning*. Berlin: Springer.
- Eisenberg, M. (2002). Output devices, computation, and the future of mathematical crafts. *International Journal of Computers for Mathematical Learning* 7(1), 1-44.
- Edwards, L. D. (1995). Microworlds as representation. In A. A. diSessa, C. Hoyles, R. Noss, & L. Edwards (Eds.), *Computers and exploratory learning*. Berlin: Springer.
- Goldenberg, E. P., & Cuoco, A. A. (1998). What is dynamic geometry? In R. Lehrer & D. Chazan (Eds.), *Designing learning*

- environments for developing understanding of geometry and space*. London: Lawrence Erlbaum Associates, Inc.
- Goldenberg, P., Lewis, P., & O'Keefe, J. (1992). Dynamic representation and the development of a process understanding of function. In Ed Dubinsky & G. Harel (Eds.), *The concept of function: aspects of epistemology and pedagogy*(pp.235-260), The Mathematical Association of America.
- Jonassen, D. H. (1996). *Computers as mind-tools for schools: engaging critical thinking*. Prentice-Hall.
- Kafai, Y. (1995). *Minds in play*. New Jersey: Lawrence Erlbaum Associates, Publishers.
- Kafai, Y., & Resnick, M. (1996). *Constructionism in practice*. New Jersey: Lawrence Erlbaum Associates, Publishers.
- Laborde, C., & Mariotti, M. A. (2001). Grounding the notion of function and graph in DGS. *Cabri World 2001* - Montreal.
- Marton, F., & Booth, S. (1997). *Learning and awareness*. New Jersey: Lawrence Erlbaum Associates, Publishers.
- Mason, J., & Johnston-Wilder, S. (2004). *Fundamental constructs in mathematics education*. London: RoutledgeFalmer.
- Noss, R., & Hoyles, C. (1996). *Windows on mathematical meanings*. Dordrecht: Kluwer Academic Publishers.
- Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Cambridge, Massachusetts: Perseus Publishing.
- Papert, S. (1993). *The children's machine: rethinking school in the age of the computer*. New York: BasicBooks.
- Parzysz, B. (1988). "Knowing" vs. "seeing": problems of the plane representation of space geometry figures. *Educational Studies in Mathematics*, 19, 79-92.
- Resnick, M. (1995). New paradigms for computing, new paradigms for thinking. In A. A. diSessa, C. Hoyles, R. Noss & L. Edwards (Eds.), *Computers and exploratory learning*. Berlin: Springer.
- Resnick, M. (1994). *Turtles, termites, and traffic jams: explorations in massively parallel microworlds*. Cambridge: MIT Press.
- Resnick, M., & Silverman, B. (2005). Some reflections on designing construction kits for kids. *Proceeding of interaction design and children conference*. Boulder: CO.
- Serra, M. (1997). *Discovering geometry: an inductive approach*. Berkeley, CA: Key Curriculum Press.
- Shaffer, D. W. (1997). Learning mathematics through design: the anatomy of Escher's World. *Journal of Mathematical Behavior*, 16(2), 95-112.
- Sherin, B. (2002). Representing geometric constructions as programs: a brief exploration. *International Journal of Computers for Mathematical Learning* 7(1), 101-115.
- Stroup, W. M. (2002). Understanding qualitative calculus: a structural synthesis of learning research. *International Journal of Computers for Mathematical Learning* 7 (1), 167-215.
- Sutherland, R. (1995). Mediating mathematical

action. In R. Sutherland & J. Mason (Eds.), *Exploiting mental imagery with computers in mathematics education*. Berlin: Springer.

Wilensky, U. J. (1993). *Connected mathe-*

*matics: building concrete relationship with mathematical knowledge*. Unpublished doctoral dissertation, MIT, Boston, MA.

## Designing a Environment in Computers and Mathematics Education

Kim, Hwa Kyung (Graduate School, Seoul National University)

In this paper, we design a environment in computers and mathematics education. For this purpose, we study two different points of view about relations between computers and mathematics education. As theoretical background, we also study constructionism and microworld.

Next, we introduce functionization as a basic principle for computers and mathematics education. The concept of functionization focuses on the variation of mathematical objects, and it is a basic concept of both mathematics and computer science. We consider the concept of functionization as a paradigm for the research and practice of the

computers and mathematics education.

We also present the concept of functionization as a principle for designing a computer environment. Finally, we use the concept of functionization to integrate two famous microworlds, LOGO and DGS by introducing such objects as tiles and folding nets. Combining LOGO and DGS, we design a new microworld that can be used under the internet environment. We present tiles and folding nets to introduce how the concept of functionization is used to design a new microworld and to integrate two microworlds.

\* **Key words** : computers and mathematics education(컴퓨터와 수학교육), constructionism(구성주의), microworld(마이크로월드), functionization(함수화), JavaMAL(자바말), computer environment(컴퓨터 환경)

논문접수 : 2005. 9. 12

심사완료 : 2005. 11. 1