

웹 환경에서 동적기하 프로그램의 비교 연구

김 부 윤 (부산대학교)

정 재 훈 (김해삼문고등학교)

DGS(dynamic Geometry System)와 WBI(Web Based Instruction)를 고찰해보고, 동적 기하 프로그램의 대표적인 프로그램인 GSP, Cabri, Cinderella를 이용하여 WBI를 제작해 보고 웹 환경 하에서 세 프로그램의 효율성을 비교·분석하였으며, 이들 세 프로그램의 장점을 정리하여 웹 환경에서 동적 기하 프로그램의 개선 방향을 제시하였다.

I. 서론

오늘날의 수학교육은 문제해결의 연장선에서 컴퓨터와 그래핑 계산기 등의 교육공학을 사용하는 시대, Standard 시대, 구성주의에 바탕을 둔 시대 등으로 회자되고 있다. 특히 Standard 시대는 미국 수학교사협회 (National Council of Teachers of Mathematics ; NCTM)에서 1989년 Curriculum and Evaluation Standards for School Mathematics가 나온 이래, 1991년의 Professional Standards for Teaching Mathematics, 1995년의 Assessment Standards for School Mathematics에 이어 2000년에는 Principles and Standards for School Mathematics(이후 Standard 2000으로 약칭)가 출간됨으로써 절정에 달하고 있으며([1]), 이후 많은 수학교육의 개혁방안을 Standards에서 찾고 있다. 특히, Standard 2000에서는 학교수학을 위한 원리를 여섯 가지를 제시하고 있는데 공평성(Equity), 교육과정(Curriculum), 교수(Teaching), 학습(Learning), 평가(Assessment) 그리고 테크놀러지(Technology)가 그것이다.

Standard 2000에서는 테크놀러지 원리를

테크놀러지는 수학의 교수와 학습에 중요하다. 그것을 가르쳐지는 수학에, 그리고 학생의 학습을 강화시키는 수학에 영향을 미친다.

Technology is essential in teaching and learning mathematics; it influences the mathematics that is taught and enhances students' learning.

로 서술하여 테크놀러지가 교수방법이나 학습방법에 국한되지 않고, 수학교육과정까지 변화시킬 수 있다고 언급하고 있다([13]). 또한 컴퓨터는 수학 내용을 변화시켰으며, 수학을 이해하는 방법과 교수·학습 방법을 변화시켰으며, 교수·학습이 일어나는 사회·문화적 환경과 교육제도를 변화시켰으

며, 학교 교육과 실생활과의 관계도 변화시켰다. 실제로 거의 모든 학교에서 컴퓨터의 영향을 깊이 받지 않은 교육적 측면을 찾기란 쉽지 않을 것이다([11]).

수학교육에서 활용되는 테크놀러지를 내용적으로 구분하면, 컴퓨터 대수 체계(Computer Algebra System : CAS)와 동적 기하 체계(Dynamic Geometry System : DGS)로, 교수 및 학습자에게 접근하는 방법으로 구분하면 전통적인 컴퓨터 활용 교육(Computer Aided Instruction : CAI), 웹 기반 교육(Web-Based Instruction : WBI), 휴대형 테크놀로지(Hand-held Technology)로 분류할 수 있다. 이 중 최근 학교수학에 많은 영향을 미친 테크놀러지는 여러 가지가 있으나, 내용적으로는 DGS, 접근방법으로는 WBI가 많은 교사들과 학생들에게 주목받고 있다. 교사들을 위한 각종 재교육 프로그램 - 예를 들면, 광역자치단체 규모의 교육청에서 실시하는 1급 정교사 자격연수과정, 수학교사를 위한 일반연수과정 등 - 에 DGS 프로그램의 활용이 많이 들어 있으며, 초고속 인터넷 보급률 세계 최고를 자랑하며 전 국민의 70%인 2천 7백만 명이 인터넷을 사용하고, 특히 중·고교생의 인터넷 이용률 96% 및 80%대에 달하는 초등학생의 인터넷 이용률¹⁾은 자연스럽게 WBI에 관심을 갖게 한다.

1985년 Geometric Supposer가 개발되면서 처음 등장하게 된 DGS는 1980년대 말 Geometer's Sketchpad(GSP), Cabri Géomètre가 개발되면서 획기적인 발전을 하게 되었다. 기존의 학교 기하교육은 학생들의 사고능력을 키워주기보다는 단순히 지식의 전달에 치중해 왔으며, 학생들의 수준에 비해 지나치게 엄밀한 추론을 요구함으로써 학생들이 매우 어렵게 기하를 공부해야만 하는 결과를 빚고 있다. 이런 단점은 활동을 통한 개념 획득방법으로 극복할 수 있는데, 그 도구로 적당한 것이 동적 기하 소프트웨어이다. 이러한 소프트웨어들은 변형되는 여러 도형으로부터 발견적 사고를 이끌어 낼 수 있다는 점에서, 그리고 학생들의 직접적인 경험이나 조작을 통해서 시각화가 가능하다는 점에서 기하학습의 어려움을 완화시켜 준다. 특히, 그래픽이나 애니메이션, 시뮬레이션을 통한 직관적 탐구활동은 수학의 역동적이고 발생적인 측면을 부각시킬 수 있다. 또한 동적 기하 소프트웨어들은 갖가지의 기하 작도가 가능하고, 거리, 각, 넓이 등을 바로 측정할 수 있을 뿐만 아니라, 측정이 동적으로 이루어지며, 기하 작도에서 어떤 변화도 그와 연결된 측정의 변화를 수반한다는 강점을 가지고 있다. 더욱이 작도 및 측정기능 외에도 여러 가지 변환기능(평행이동, 대칭이동, 닮음변환, 회전이동), 반복기능(매크로, 스크립트), 표현기능, 끌기기능²⁾ 등을 갖고 있다.

WBI는 1996년 이후 급격히 발전해 왔으며, 인터넷을 기반으로 한 웹을 매체로 활용하여 원거리에 있는 학습자를 교육시키는 방법으로, 멀티미디어·하이퍼미디어 기술을 이용해서 하이퍼텍스트의 형태로 교육 자료들을 제공한다. 또한, WBI는 특정한 그리고 여러 계획된 방법으로써 학습자의 지식이나 능력을 향상시키기 위해 웹을 통하여 의도적인 상호작용을 한다. 최근에는 인터넷을 이용하지 않더라도 프로그램이 운영되는 기본 플랫폼으로 웹 브라우저를 사용하여 컴퓨터의 하드웨어 구성이

1) 연합뉴스(www.yonhapnews.co.kr) 2004.4.22

2) Geometric Supposer가 개발될 당시에는 없었던 끌기기능(Dragging)은 그 뒤 GSP와 Cabri Géomètre에 채택되어 도형의 모양이나 크기를 변형시켜 관찰하면서, 일반적인 성질이나 법칙을 발견하는 귀납적 탐구활동을 가능하게 해는 기능이다.

나 운영체제에 관계없이 실행되는 프로그램이 많이 개발되고 있다. 이를 가능하게 해주는 프로그램 기술이 미국의 선 마이크로 시스템(Sun Microsystems)에서 개발한 자바(Java)이며, 자바 애플릿(Java Applet)을 이용하여 이전보다 더 동적이며 상호작용이 많이 보강된 WBI가 개발되고 있다.

최근 학교 현장에서 많이 논의되고 있는 웹 기반 교육은 어떠한 주제에서든지 학습자 주도형으로 학습자의 속도에 맞는 교수법을 제공할 수 있으며, 학습자 중심의 수업상황을 창출함으로써, 학습자는 수동적 역할에서 벗어나 자신의 학습에 대하여 계획하고 능동적으로 실행하는 역할을 수행할 수 있으며, 정형화된 교실 중심의 교육을 다양화 할 수 있다. 아울러 컴퓨터를 통한 교육은 그것을 적절히 사용함으로써 활기 찬 학습 환경을 조성하고, 여러 가지 멀티미디어 자료를 이용함으로써 학생들의 이해를 돕고, 지적 호기심을 유발할 수 있다고 하겠다.

우리나라는 세계적으로 초고속 인터넷을 포함한 정보·통신 인프라가 잘 구축되어 있어 모든 학교의 각 교실에 학습을 위한 컴퓨터와 인터넷 망이 이미 설치되어 있다. 아울러 이런 시설을 활용할 수 있는 많은 소프트웨어들도 개발되어 있으며, 심지어 이러닝산업발전법안이라는 특별법에서 e-learning (Electronic Learning)이라는 개념을 전자적 수단, 정보통신 및 전파·방송기술을 활용하여 이루어지는 학습이라고 정의하여 관련 산업을 적극적으로 장려하고 있다.

본 논문에서는 DGS와 WBI를 고찰해보고, 동적 기하체제 프로그램의 대표적인 프로그램인 Geometer's Sketchpad, Cabri Géomètre II 및 Cinderella의 사용을 통해 DGS를 활용하여 WBI를 제작하는 방법을 알아보고, WBI제작에 있어 이들 세 프로그램의 효율성을 비교·분석하였으며, 끝으로 이들 세 프로그램의 장점을 정리하여 웹 환경에서 DGS 프로그램의 개선방향을 제시하고자 한다.

II. 동적 기하 체계와 웹 기반 교육

1. 동적 기하 체계

1.1 동적 기하의 개념

요즘에는 보편화 된 용어인 동적 기하(Dynamic Geometry)라는 개념의 역사는 1945년 Syer가 '연속적인(continuous)' 기하학적 그림을 만들 수 있는 필름의 가능성을 언급한 것에서부터 찾을 수 있다. 그 후 1954년 Burnes가 바로 그 '동적(dynamic)'이라는 용어를 유용한 수학적 장치의 이점을 묘사하는데 사용했다. 종이와 연필로 대변되는 평면 기하에서는 학습자가 어떤 도형을 그릴 때, 전형적인 모양의 도형을 그린다. 예를 들어 이등변삼각형을 그릴 경우 학습자는 대부분 길이가 같은 두 변이 위로 향하게 하는데, 이것은 학습자가 주로 수업시간이나 일상생활에서 만나는 이등변삼각형이 대부분 그런 모양이기 때문이다([16]). 동적 기하 소프트웨어의 선구라 할 수 있는 Geometric Supposer의 목적은 주어진 다양한 크기와 여러 방향으로 기울어진 도형 - 위의 예에서는 이등변삼각형 - 을 무작위로 만들어내는 것이었다. 즉, 길이가 같은 두 변이 왼쪽이나 오른쪽, 또는 아래쪽

으로 향하게 하거나 심지어 한쪽으로 비스듬히 기울어지는 이등변삼각형 등도 만들어낸다. 더 나아가 오늘날 사용되는 동적 기하 소프트웨어는 학습자가 도형을 작도할 수 있고 도형을 변형할 수 있게 해 준다.

화면에 그려진 그림은 도형의 정의를 명확하게 하는 과정의 결과로, 그리고자 하는 도형에 내재된 기하학적 관계에 대한 명확한 이해를 나타낸다. 기하학적 관계에 의해 그려진 스크린의 도형은 도형의 가변적인 요소가 변화되었을 때, 의도한 특성들을 보존하면서 여러 가지 도형으로 변화된다. 그러므로 이들 소프트웨어를 활용한 작도는 필요한 도형을 그리는 것이라기보다는 도형에 내재된 기하학적 관계를 명확히 서술하는 것이다.

1.2 동적 기하 소프트웨어의 특성

사용자가 컴퓨터 스크린 상의 도형을 직접 조작하면서, 도형의 성질이나 관계들을 탐구할 수 있도록 1980년대 말에 고안되어진 동적 기하 소프트웨어는 동적인 기하환경(dynamic geometry environment)을 제공해 준다.

Finzer와 Jackiw는 동적인 환경의 특성으로 직접적인 조작(Manipulation is *direct*)³⁾, 연속적인 움직임(Motion is *continuous*), 몰입적인 환경(The environment is *immersive*)을 제안하였다([9]).

여기에서 직접적인 조작이란 사용자가 스크린 상의 한 대상을 선택하여 직접 움직여 볼 수 있음을 의미한다. 그렇게 함으로써 스크린 상에 보여지는 대상과 그 이면에 숨겨진 수학적 의미 사이의 인식적 거리는 좁혀질 수 있다. 즉 움직이고자 하는 대상을 스크린에서 직접 조작하고 있는 느낌을 갖게 된다. 두 번째 특성으로 연속적인 움직임이란 '드래그(drag)' 동안에 일어나는 변화와 관련된 것으로 '드래그' 동안에 스크린 상의 수학적 대상들이 항상 논리적인 관련성에 의해 모양을 유지하면서 움직여질 뿐만 아니라, 대상들이 변하는 중간상태를 사용자가 모두 볼 수 있음을 의미한다. 마지막으로 몰입적인 환경이란 사용자의 초점이 작도를 위한 기술적인 방법보다는 수학적 목표를 달성하는 방법에 맞추어질 수 있도록 하는 환경을 말한다.

이와 같은 동적인 환경을 제공하는 소프트웨어의 대표적인 것으로는 The Geometer's Sketchpad (이하 GSP), Cabri Géomètre II Plus(이하 Cabri), Cinderella, Euklides 등이 있다. 현재 국내에서 주로 쓰이고 있는 소프트웨어로는 Cabri와 GSP이다.

2. 웹 기반 교육

웹 기반 교육(Web Based Instruction ; WBI)은 웹의 등장과 함께 부각된 새로운 교수·학습 방법으로 기존의 CAI가 오프라인 형태의 컴퓨터 교수·학습 방법이라고 하면, WBI는 온라인 형태의 교수·학습 방법이라 볼 수 있다. 이러한 WBI는 다양하게 정의를 내릴 수 있는데, Khan은 그것을 다음과 같이 정의하고 있다([3],[4]).

3) 원문의 이탤릭체임.

WBI는 학습이 촉진되고 지원되는 의미 있는 학습 환경을 창출하기 위해, 월드 와이드 웹의 속성과 자원들을 이용하는 하이퍼미디어 기반의 교수 프로그램(hypermedia-based instructional program)이다.

또한, WBI는 특정한 그리고 미리 계획된 방법으로써 학습자의 지식이나 능력을 육성하기 위한 의도적인 상호작용을 웹을 통해 전달하는 활동이라고 정의 내릴 수 있으며, 웹을 매체로 활용하여 원거리에 있는 학습자를 교육시키는 혁신적인 접근 방식으로 볼 수도 있다. 그것은 Smith와 Ragan에 의하면, 교수는 학습자의 의도되고 특정한 학습 목표 달성을 촉진하는 정보와 활동을 전달하는 것이고, 그 교수매체는 그것에 의해 교수 메시지가 의사소통될 수 있는 수단이라고 말하고 있기 때문이다.

III. 동적 기하 소프트웨어의 비교(웹 환경을 중심으로)

기하 교육의 문제를 개선하기 위한 방안으로 동적 기하 소프트웨어의 활용에 관한 많은 자료가 개발되고 있으며, 그것에 관한 연구가 활발히 진행되고 있다. 그러나 개발된 자료들은 현행 교과서를 재구성한 것이어서 교사들은 개발된 자료를 활용하기 위해 학교교육과정을 재구성해야 하는 어려움을 갖고 있다. 또한 지필 환경에서의 활동이나 추론의 영역 내용들이 동적 기하 소프트웨어를 활용함으로써 확대될 수 있으며, 지필 환경에서 가능하지 않았던 문제들도 다룰 수 있게 되므로 기하 영역의 교육과정에서 교육과정의 구성 및 계열 자체에 대해 연구해야 하는 어려움도 가지고 있다.

동적 기하프로그램으로서 Geometer's Sketchpad와 함께 잘 알려진 프로그램은 Cabri Géomètre II 인데, 이 프로그램도 기능면에서 Geometer's Sketchpad와 유사한 부분이 많다. 그러나 이 두 프로그램과 유사한 역동적인 기하 프로그램인 Cinderella에 대해서는 상대적으로 국내에 많이 소개되어 있지 않아 보인다.

GSP와 Cabri Géomètre II는 서로 유사한 점과 차이점이 있는데, 이러한 이유는 이들의 개발자와 그 개발 배경을 살펴보면 도움이 될 것이다. Cabri Géomètre의 개발은 1985년 전산학자인 Jean-Marie Laborde에 의하여 자와 컴퍼스를 이용한 기하학적 도형의 탐구를 목적으로 기하 학습장의 개발을 제안하면서 시작되었다. 다른 전산학자 Philippe Cayet, Yves Baulac 그리고 교육학자 Franck Bellemain이 뒤이어 Cabri Géomètre 개발에 참여하였다. 이러한 이유로 인하여 Cabri Géomètre는 먼저 학교 현장에서 사용되기 시작하였으며, 다른 프로그램에 비하여 안전하다는 평가를 받았다. 거의 동시에 Geometer's Sketchpad는 수학과 교수들인 미국 펜실바니아주의 Swarthmore College의 Gene Klotz와 Moravian College의 Doris Schattschneider가 개발하였고, 프로그래밍은 학생이었던 Nicholas Jackiw에 의하여 이루어졌다. 초기의 Geometer's Sketchpad는 평행이동 기하에 대한 기능들을 강조하여 테셀레이션(Tessellation)의 작도에 편리하였다. Cabri Géomètre와 Geometer's Sketchpad의 개발자들은 서로의 정보를 교환하여 상대방의 좋은 기능들을 각자의 프로그램에 첨가하여 왔다. 따라서 근본적으로 Cabri Géomètre와 Geometer's Sketchpad는 사용방법만 다를 뿐 기능면에서는 거의 유사한 프로그램으로 변화였다.

한편 Cinderella의 개발은 1990년대 초에 Henri Crapo에 의하여 시작되었고, 자동 정리증명을 연구하는 Jürgen Richter-Geibert와 그의 학생인 Ulrich Kortenkamp가 개발에 참여하여 연구 목적으로 개발되었다. 이 연구의 결과는 Ulrich Kortenkamp의 박사학위 논문으로 발표되었다. 이러한 이유로 그의 논문에는 Cinderella의 기초가 되는 수학적 이론들이 상세하게 기술되어 있다([7],[2]).

여기서 대표적인 동적 기하 프로그램인 GSP, Cabri Géomètre II 및 Cinderella의 개발의 역사를 알아보고, 각 프로그램을 이용하여 웹에서 실행되는 동적 기하 즉, WBI에서 DGS의 구현에 대한 방법을 알아보고, 각 프로그램의 장·단점을 분석해본다.

1. The Geometer's Sketchpad(GSP)

1.1 GSP의 개발([15],[6])

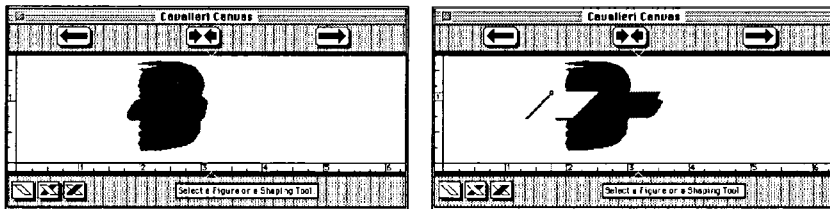
GSP(Geometer's Sketchpad)는 펜실바니아의 Morvian 대학의 Dr. Doris Schattchneider와 Swartmore 대학의 Dr. Eugene Klotz의 지도 아래 미국과학재단 기금 연구(National Science Foundation funded project)인 Visual Geometry Project의 한 부분으로 개발되었다. GSP 외에도 시각적 기하학 프로젝트에 의해 The Stella Octangula와 The Platonic Solids를 제작하였으며 Key Curriculum Press에 의해 출간된 비디오 테이프, 실용 교재 및 조작 교구 등이 있다. GSP의 개발자이자 프로그래머이며 VGP의 연구원인 Nicholas Jackiw은 프로그램의 초기 버전을 공개하고 많은 교사와 다른 이용자들이 실험해 보는 방법 등을 통하여 여러 사람들의 의견을 수렴하여 개발하였다.



Nicholas Jackiw

1980년대 중반 VGP는 3차원 기하에 초점을 맞춘 비디오 테이프를 제안했지만, Klotz는 비디오 테이프는 과도기적인 자료일 뿐이므로 상호작용이 뛰어난 프로그램을 제안하였다. 프로그래밍을 위해 Klotz는 대학 신입생 시절 도와주었던 Nicholas Jackiw에게 방향을 돌렸다. Jackiw는 아홉 살 때 프로그래밍을 시작했는데, 당시에는 영어와 컴퓨터에 관한 프로그램이지 수학에 관한 것이 아니었다. 게다가 그는 대학시절 수학을 완전히 포기한 상태였다.

VGP에서 Jackiw의 첫 작업은 Cavalieri라 불리는 프로그램이었다. 이 프로그램은 화면에 그림을 그린 후 적당한 도구 - virtual bulldozer라 불렀다 - 로 밀면 그림이 밀려가는 프로그램이었다.



Nicholas Jackiw가 처음으로 개발한 Cavalieri Canvas

원래 VGP의 목표는 3차원 기하 개념의 비디오 테이프를 컴퓨터 프로그램으로 만드는 것이었지만, 3차원 모델에 대한 프로그래밍의 어려움으로 Klotz와 Schattschneider는 평면기하에 초점을 맞춘 프로그램으로 선회하였다.

컴퓨터의 그래픽능력을 보여줄 수 있는 초기의 벡터 그래픽 프로그램으로 Ivan Sutherland의 'Sketchpad'가 있었다. 라이트 펜을 손에 쥐고 모니터에 나타난 점과 선분, 호를 조작할 수 있는 프로그램이었다. 이 Sutherland의 업적이 새로운 프로그램에 살아서 'The Geometer's Sketchpad' 즉, GSP가 된 것이다.

대상을 잡아끌어(drag) 점이나 선을 조작할 수 있는 것이 GSP나 Cabri Géomètre의 가장 큰 특징이다. 그러나 역사적인 관점에서 보면, 이 개념이 어느 날 갑자기 나온 것은 아니라 고대 그리스 시대의 수학자들도 이 개념을 사용하였다. 단지 적당한 도구들을 사용하였기 때문에 엄밀하게 수학적이 아니라고 생각하였다. 초기의 GSP 디자인에는 잡아끄는 기능(dragging component)이 없었다. Klotz는 학생들을 위해 유클리드 기하에서 나타나는 여러 가지 도형들을 아주 정밀하고 정확하게 그려주는 형태의 프로그램을 생각하였다. 그러나 이런 형태의 프로그램은 Macintosh 컴퓨터를 사용하는 Jackiw에게 의문을 가져다주었다. 그 당시에는 마우스가 그다지 많이 보급되지는 않았지만 Macintosh 컴퓨터의 입력장치는 마우스를 기본으로 하였는데, 평소 비디오 게임을 디자인하는데 관심과 경험이 있었던 Jackiw에게 새로운 아이디어를 떠올리게 했다. 마우스는 원래 어떤 대상을 정확하게 입력하는데 어려움은 있지만 대상을 잡아끌고 조작하는데는 아주 훌륭한 도구인 것이다. 그러나 Jackiw는 Sketchpad를 동적으로 만들기로 결정한 후 다른 것은 추가하지 않기로 했다. 심지어 점이나 선의 라벨도 눈에 거슬린다고 싫어할 정도였다.

Jackiw가 1990년에 Key Curriculum Press의 지원을 받아 교실에서 실험·탐구하고 있다는 소문이 나자 50개 이상의 장소로 퍼져 나갔고, 이로 인해 많은 사람들이 학교수학에서 GSP에 대한 필요성을 인식하게 되었다. GSP의 첫 해에 Key Curriculum Press는 그 프로그램이 학교에서 얼마나 효과적인가를 연구하기 시작하였다. 미국과학재단의 지원을 받은 이 연구는 1992년에 교수 지침과 교육과정 자료, 발전적인 변환과 프리젠테이션을 첨가한 버전 2인 GSP를 만들었다. 이 버전에서 특히 달라진 것은 프랙탈을 작도할 수 있는 재귀적인 스크립트가 포함된 것이다.

윈도우용인 보완된 버전이면서 측정에 의한 변환, 스크립트 도구, 개선된 수학적 기호, 호, 작도할 수 있는 자취 및 궤적, 해석적이면서 그래프를 작도할 수 있는 능력 등을 고루 갖춘 버전 3이 1993년 3월에 출시되었다. 이어 1995년 4월에 버전 3.1이 발표되었으며, 2001년 9월에 버전 4가 발표되었으며, 2004년 4월 현재 영문버전 4.06, 한글버전 4.03이 시판되고 있다.

이 GSP 프로그램은 기능상으로는 유클리드 기하의 도형을 완벽하게 구현할 수 있으며, 거기에서 더 나아가 애니메이션 Trace 기능 등을 통하여 실제 우리가 머리 속으로 상상하기 힘든 여러 가지 도형을 직접 시각적으로 보여줌으로써 기하의 여러 가지 성질을 발견하는데 아주 좋은 아이디어를 제공해 줄 수 있다. 뿐만 아니라 평행이동, 회전, 대칭이동, 벡터 등을 이용하면 기존의 유클리드 기

하의 정적인 도형에서 다양한 움직이는 그림을 그릴 수 있다. 또한 그 그림 자체만으로 하나의 작품이 되기 때문에 그것을 활용하면 수업시간을 한층 더 생동감 넘치는 수업이 가능하게 된다.

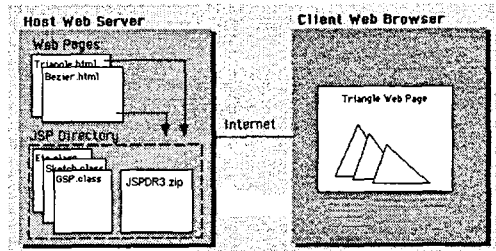
1.2 JavaSketchpad

GSP를 이용하여 만든 동적인 도형을 다른 사람들에게 소개하기 위해서 웹을 이용할 수 있다. 이 때, JavaSketchpad(JSP)라는 소프트웨어를 사용한다. JSP는 많은 .class 파일들로 이루어진 Java Applet이고 HTML로 바뀌어진 GSP그림을 웹에서 작동하게 하는 역할을 한다. 웹에서 작동하는 JSP로는 수학사랑(<http://www.mathlove.com>)의 「움직이는 기하」 사이트에 있는 모든 그림을 들 수 있다.

인터넷 상에서 GSP로 만들어진 움직이는 그림을 보려면 먼저 GSP로 만들어진 화일을 HTML 문서로 바꾸어야 한다. 이후 JSP라는 폴더와 같은 위치(폴더 속이 아닌)에서 변환된 *.html 문서를 놓고 웹 브라우저를 이용하여 *.html 문서를 불러오면 자바로 실행되는 움직이는 기하를 감상할 수 있다. 이것을 정리하면 다음과 같다:



① GSP에서 움직이는 그림을 작도하여 저장한 후, 다시 다른 이름으로 저장하기에서 HTML로 저장한다. 이 때, 저장된 HTML파일은 JSP폴더와 같은 위치에 있어야 한다. 익스플로러를 실행하고 위의 HTML 파일을 불러낸다. 실제로 많은 사람들에게 공개하려면 서버에 올려놓아야 하는데 그림을 참고하면 된다.

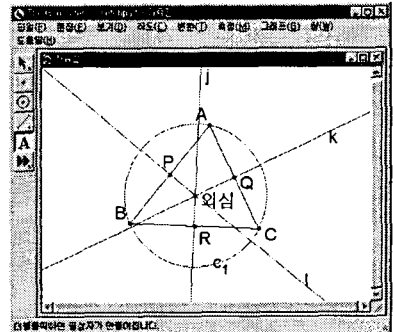
② 하드디스크에서 작동하는 것과 같은 원리로 웹에서 작동하게 되지만, 시간이 조금 걸리게 된다. 즉 JSP를 탑재한 서버에서 Class파일을 불러와서 자신이 실행하는 웹 브라우저에서 보여 주기 때문에 시간이 걸리게 되는 것이다. 이때 JSP폴더는 하나만 있으면 된다. 이 폴더 밑에 많은 GSP HTML 파일을 두어도 각 파일은 웹에서 잘 작동한다. 또 Java Sketchpad Construction Grammar을 이용하면 직접 손으로 HTML 파일을 만들 수 있다.



(1) GSP4 File 디자인


학습내용을 선택하고 GSP4를 이용하여 기본적인 수업 내용을 디자인한다. 여기에서는 삼각형의 외심과 외접원을 작도하였다.

- ①  점 도구를 이용하여 점 A, B, C를 작도한다.
- ②  선분 도구를 이용하여 선분 AB, 선분 BC, 선분 AC를 작도한다.
- ③ 각 선분의 중점(P, Q, R)을 작도한다. (선분을 선택한다. ⇨ 작도메뉴에서 ⇨ 중점을 선택한다.)



④ 선분과 중점을 선택해서 수선(j, k, l)을 작도한다. (수선과 중점을 선택한다. ⇨작도메뉴에서 ⇨ 수선을 선택한다.)

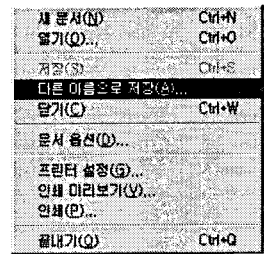
⑤ 수선의 교점(외심)을 작도한다. (두 개의 수선을 선택한다. ⇨ 작도메뉴에서 ⇨ 교점을 선택한다.)

⑥  원 도구를 이용하여 외접원 (C_1)을 작도한다. (첫 번째 선택점은 원의 중심, 두 번째 선택점은 원 위의 점이 된다.)


⑦ 완성이 되면 저장 - 여기서는 자료전.gsp 저장 - 한다.

(2) GSP4의 HTML 변환

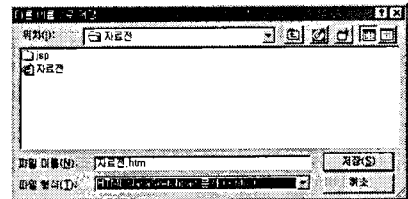
GSP3에서는 따로 gsp_html.exe를 구동시켜 HTML로 변환시켰지만 GSP4에서는 프로그램 자체에서 HTML 파일로 변환해서 저장시킬 수 있다. 물론 변환된 HTML 파일이 jsp 폴더와 같은 경로에 있어야 한다.




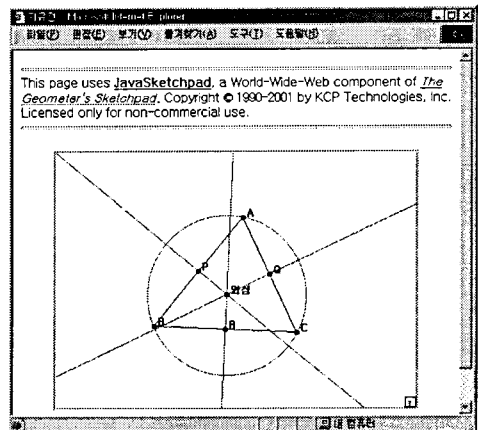
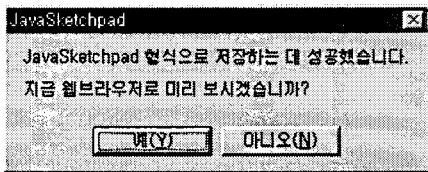
① 파일메뉴에서 “다른 이름으로 저장”을 실행한다.

② 파일 형식을 (*.htm)으로 지정하고  jsp 폴더와 같은 경로 상에 위치시킨다.

③ 저장이 끝나면 “JavaSketchpad 형식으로 저장하는데 성공했습니다. 지금 웹 브라우저로 미리 보시겠습니까?”라는 메시지가 나타난다. 하지만 아직까지 GSP4에서 작도한 내용 모두가 변환되는 것은 아니다.



④  을 클릭하면 Explorer에서 확인해 볼 수 있다.



```

1 <!--This file created by The Geometer's Sketchpad 4.03-->
2 <HTML>
3 <TITLE>자료전</TITLE></HEAD><BODY>
4 <H4 ALIGN=CENTER><APPLET CODE="GSP.class" ARCHIVE="jsp4.jar" CODEBASE="jsp" WIDTH=457 HEIGHT=324 ALIGN=
  CENTER><PARAM NAME=MeasureInDegrees VALUE=1><PARAM NAME=DirectedAngles VALUE=0>
5 <PARAM NAME=BackRed VALUE=255>
6 <PARAM NAME=BackGreen VALUE=255>
7 <PARAM NAME=BackBlue VALUE=255>
8 <PARAM NAME=Construction VALUE="
9 {1} Point(234,80)[label('A')];
10 {2} Point(123,215)[label('B')];
11 {3} Segment(2,1)[color(0,0,255)];
12 {4} Point(302,223)[label('C')];
13 {5} Segment(4,2)[color(0,0,255)];

```

(3) JavaSketchpad Construction Grammar([10])

http://www.keypress.com/sketchpad/java_gsp/dr_grammar.html에 있는 문법적 내용을 충분히 살펴보아야 한다. 문법적 내용에 있는 작도로만 GSP에서 작업한다면 (물론 작도 내용에서 많은 제한이 있지만) JSP로의 변환이 가능하다. Java Sketchpad Construction Grammar Developer's Release 4 (Rev. September 2003)의 기본적인 문법 내용들을 간단히 살펴보자.

- <APPLET></APPLET> Tag

- ▶ CODE Parameter는 제일 처음으로 가지고 올 Java.class 파일의 이름을 지정한다. 이 "GSP.class"의 이름을 바꾸면 안 된다.

- ▶ ARCHIVE Parameter는 JSP 폴더에 저장되어 있는 모든 .class 파일을 포함하고 있는 .JAR archive의 파일 이름을 지정한다. 웹 브라우저는 분리된 .class 파일의 집합으로 애플릿을 불러오지 않고 하나의 archive 파일로부터 애플릿을 불러오게 된다. 이는 원격의 웹 서버로부터 애플릿을 불러오는데 시간을 절약할 수 있다. 그러나 모든 웹 브라우저가 이 archive 포맷을 지원하는 것은 아니다. 그래서 JSP 폴더에는 하나의 .JAR archive 뿐만 아니라 개개의 .class 확일을 포함하고 있다. (모든 .class 파일은 JSP 폴더 밑에 있는 서브 폴더와 함께 오게 된다.) JSP에서 ARCHIVE="JSPDR4.JAR"로 설정되어 있다.

- ▶ CODEBASE Parameter는 <APPLET> 태그를 가지고 있는 웹 페이지의 URL과 관련하여, 웹 서버에 애플릿이 있는 폴더의 경로를 나타낸다. "JSP"라는 폴더의 이름을 바꾸면 안 되지만, JSP 애플릿을 포함하고 있는 폴더와 다른 위치에 웹 페이지가 있다면 이 애플릿이 있는 경로를 반드시 이곳에 써 주어야 한다. 흔히들 프로그램을 정확하게 작성하여 미리보기를 하면 프로그램이 실행되지 않는 경우들이 많다. 그 원인의 대부분이 위의 사실을 알지 못하기 때문이다.

- ▶ Construction Parameter

<PARAM NAME=Construction VALUE=" ~ ">

실제로 브라우저 상에 나타나는 움직이는 그림을 표현하는 부분이다. {1}, {2}, {3},... 은 <PARAM NAME=Construction VALUE=" 이후 몇 번 째 행을 나타내는지 표시하기 위해서 {1},{2},{3},... 를 나타낸다. 중괄호 { } 안에 포함된 어떤 글이라도 JSP에 의해 무시되게 된다. 이는 문법에 대한 설명을 넣는데 유용하게 쓰일 수 있다. 이렇게 { }으로 주석문을 나타내면 JSP를 수작업을 할 때 자신이나 타인들을 위해서 (나중에 수정하는 경우) JSP 소스의 이해를 크게 도울 수 있다.

• JSP 문법은 일련의 한 가지 이상의 개체 정의로 이루어진다. 각 개체의 정의는 다음과 같은 형태를 가지고 있다.

그림의 소스를 참고로 설명하면 다음과 같다.

```
[comment] objectName (objectArguments) [optionalFormats] ;
{주석} {도형의 이름} (대상도형) {도형의 모양}
(1) Point (234,80) [label('A')] ; 점 A
(2) Point (123,215) [label('B')] ; 점 B
(3) Segment (2,1) [color(0,0,255)] ; 선분 AB
```

그 밖의 문법은 위에서 언급하였듯이

http://www.keypress.com/sketchpad/java_gsp/dr_grammar.html

에 자세히 나와 있으며, 항상 개정된 최신 문법을 숙지하고 있어야 한다.

(4) GSP에서 JSP로 변환되지 않는 작도 내용([6])

① GSP의 그래프 메뉴에서 좌표로 점찍기에서 “좌표고정”으로 작도하면 JSP로 변환되지 않는다. (단, 이동가능에 체크된 상태에서는 JSP로 변환된다.)

② GSP에서 작도메뉴의 호에 대한 작도 - 원주 위의 호, 세 점을 지나는 호 - 가 JSP로 변환되지 않는다.

③ GSP에서 보기메뉴의 선모양에서 점선은 JSP에서 지원하지 않는다. (자취 부분도 동일하다.)

④ GSP에서 작도메뉴에서의 각의 이등분선은 JSP로 변환되지 않는다. (이런 경우는 실제로 작도하듯이 꼭지점에서 적당한 원을 그리고, 변과의 두 교점에서 2개의 원을 그려 작도한다.)

⑤ GSP에서 작도한 다각형 내부의 점은 JSP로 변환되지 않는다.

⑥ GSP에서 동작버튼 2개 이상을 선택한 연결실행은 JSP로 변환되지 않는다.

⑦ 원, 선분, 직선의 label은 만들어지지 않는다. (단, 점의 label만 JSP로 변환된다.)

⑧ GSP에서 수식이 JSP로 변환되지 않는다.

⑨ GSP에서 개체삽입은 JSP로 변환되지 않는다.

⑩ GSP에서 글상자로 입력된 내용은 JSP로 변환되지 않는다.

⑪ GSP에서 작도한 선분을 선택한해서 그래프메뉴의 “단위 길이로 좌표축 정의”로 만들어진 좌표축은 JSP로 변환되지 않는다.

⑫ 자취를 선택해서 작도메뉴의 “도형 위의 점”으로 작도된 점은 JSP로 변환되지 않는다.

(5) JSP에서만 가능한 작업([6])

GSP에서 작도한 내용은 JSP에서 HTML Tag를 직접 수정해도 되지만, 많은 시간과 노력이 필요하다. 노력과 시간을 절약하기 위해서는 최대한 GSP에서 작업하고, JSP에서는 GSP에서 작업할 수 없는 작업만 하면 된다. 그러기 위해서는 앞에서 나열한 JSP로 변환되지 않는 GSP에서의 작업내용을 피해서 작업해야 한다. 다음은 GSP에서 할 수 없는 JSP에서만 가능한 작업내용이다.

① RGB Code를 이용한 “배경 색” 지정 및 “선”, “원” 색 지정 (선이나 원 등은 GSP에서 단순 color로 지정할 수 있다. 여기서는 RGB color 등 여러 가지 색을 사용할 수 있음을 의미한다.)

② 글꼴 매개변수 정의를 사용한 글꼴 수정하는 작업

(Label Font, Action Font, Measure Font parameter)

③ 측정단위 매개변수 정의를 이용한 측정단위 수정하는 작업

(Angle Unit, DirectedAngles parameter)

④ 두 개 이상의 점을 동시에 한 점으로 이동시키는 작업

⑤ 측정값의 색상지정

⑥ GSP 문서에서 애니메이션 단추를 선택한 상태에서 저장하게 되면, 다음에 파일을 열었을 때 애니메이션이 자동 실행되지만, 이렇게 저장된 GSP문서를 JSP로 변환시켜 웹 브라우저로 열어보면 자동 실행이 되지 않는다. (optionalFormats) 부분에 수작업으로 auto를 추가하여야 한다.

⑦ GSP에서 고정된 점은 JSP로 변환되지 않는다. GSP에서 이동 가능한 점을 JSP로 변환시키면 Point (num1, num2)로 되는데, 이것을 FixedPoint (num1,num2)로 직접 수정하여야 한다.

(6) JSP의 단점([6])

이상에서 언급한 바와 같이, JSP의 가장 큰 문제는 GSP에서 작도한 모든 도형이 100% JSP로 잘 변환되지 않는다는 것이다. Java Applet의 경우 그 파일의 크기가 작아야 인터넷 상에서 전송속도가 빨라지므로 실제 GSP에서 작도가 가능하지만 JSP로는 변환되지 않는 것이 있다. 대표적으로 세 점을 지나는 호가 있는데, Java Sketchpad Construction Grammar Developer's Release 4 (Rev. September 2003)에서 나열되어 있는 명령어들 중에서 호에 관한 명령어 - 원주 위의 호, 세 점을 지나는 호 - 가 없기 때문에 GSP 문서의 호에 관한 작도내용은 JSP로 변환이 되지 않는다. 따라서 GSP 문서의 작도한 내용 중에서도 JavaSketchpad Construction Grammar에서 포함되어 있는 명령에 관한 작도내용만 JSP로 변환이 된다. JavaSketchpad Construction Grammar에만 포함되어 있고, GSP에서는 작도가 되지 않는 것도 있다.

FixedPoint라는 명령어가 있는데 GSP상에서는 선택할 수 없는 고정된 점을 작도할 수 없지만, 웹 상에서 나타날 때는 기준이 되는 고정점이 필요할 경우가 있다. 이런 경우에는 직접 HTML 코드상에서 JSP Applet의 코드를 직접 작성해야 한다.

2. Cabri Géomètre

2.1 Cabri Géomètre의 개발

1985년 프랑스의 Jean-Marie Laborde는 도형 및 그들 사이의 관계에 관한 특성을 탐구할 수 있는 Cabri Géomètre의 제작을 제안하였다. 당시 CNRS (National Centre for Scientific Research)와 Grenoble의 Joseph Fourier 대학의 연구원이던 Jean-Marie Laborde는 1980년대 초 미국의 Apple에

서 나온 Macintosh 컴퓨터의 마우스를 활용하는 그래픽 유저 인터페이스를 보고 Cabri Géomètre를 착안하였다. Cabri라는 이름은 'CAhier de BRouillon Informatique(IT Notepad)'에서 따왔다고 한다.

1988년 Cabri Géomètre의 첫 번째 버전이 Apple 컴퓨터에서 수여하는 교육용 프로그램 트로피를 받았고, 1989년 프랑스 교육부의 후원 아래 교육용 시장으로 진출한 후 많은 나라에서 사용되었으며, Jean-Marie Laborde뿐만 아니라 Philippe Cayet, Yves Baulac 및 교육학자인 Franck Bellemain이 Cabri Géomètre의 제작에 참여하였다. 그 후 1990년부터 Jean-Marie Laborde는 IMAG(Institute for Computing and Applied Mathematics - Grenoble)의 거대한 프로젝트로써 컴퓨터공학자, 수학자, 수학교육학자 인공지능전문가, 심리학자 및 교사들을 연구 그룹으로 묶어 Cabri Géomètre를 발전시켜 나갔다. 1995년 미국의 Texas Instruments사는 자사의 휴대용 그래핑 계산기인 TI-92에 Cabri Géomètre를 넣기로 결정하여 Cabri Géomètre의 개발을 지원하였으며, 그 결과 현재 사용되는 Cabri Géomètre II가 탄생되었다. 2000년 3월 Jean-Marie Laborde는 Cabri Géomètre의 지속적인 연구·개발과 보급을 위해 Cabrilog라는 회사를 설립하였으며, 전 직원의 75% 정도가 연구·개발을 담당하고 있다. 현재는 2003년 초에 개발된 Cabri Géomètre II Plus Ver. 1.2.4.8이 사용되고 있다.











도구 막대의 아이콘을 사용한 여러 가지 작도는 중학교에서 배우는 유클리드 기하(논증기하학)의 학습에 도움이 되며, 여기에 좌표축을 도입하면 고등학교의 「평면 도형의 방정식」(해석기하학)을 학습하는데도 유용하게 사용할 수 있다. 뿐만 아니라, Cabri Géomètre II는 Pull-down 메뉴와 Tool box 내의 Tool을 이용한다. 다른 동적 기하 프로그램과 마찬가지로 Cabri Géomètre II도 평면기하의 성질을 연속적이면서 역동적으로 관찰할 수 있는 초·중·고등학교에서 학생들이 기하 수업을 위해 사용할 수 있는 프로그램이다. Cabri Géomètre II의 화면에서는 칠판이나 종이 위에서와는 달리 마우스로 도형을 이동시키거나 변형하여 도형을 조작할 수 있다. 따라서 학생들은 움직이는 기하에 대해 직관적으로 학습할 수 있다. 즉, 교수 도구로서 Cabri Géomètre II는 기하학적 표현을 쉽고 명확하게 구현하며, 정적인 (책, 칠판 등) 표현에서는 설명할 수 없었던 부분들의 설명이 가능하다. 즉 도형들이 본질적인 관련성·엄밀성을 애니메이션 기능을 통해 동적인 표현으로 설명할 수 있다. 수업 도구로서 Cabri Géomètre II를 직접 사용하면, 수학적 규칙과 개념을 암기가 아닌 이해로 접근이 가능하며 학생들이 작도와 측정을 쉽고 정확하며 빠르게 할 수 있어 기하 학습에 대한 흥미를 갖게 된다.

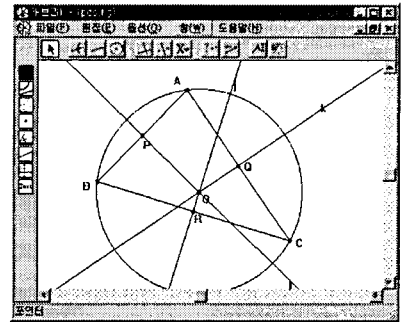
2.2 CabriJava

CabriJava는 Cabri Géomètre II로 작성된 파일을 웹에서 실행되도록 해주는 Java Applet이다. CabriJava는 GSP파일을 웹에서 실행시키는 JSP와는 달리, Cabri Géomètre II에서 작도된 파일을 그대로 사용한다. 즉, JSP처럼 별도의 언어로 재작성되는 것이 아니라, 웹 파일에서 Cabri Géomètre II의 Fig파일을 그대로 활용하는 것이다. 그러므로 JSP는 별도의 GSP파일이 필요하지 않은 반면, CabriJava는 반드시 Cabri Géomètre II의 Fig파일이 있어야 한다. 또한 이런 이유로 CabriJava를 이용

한 홈페이지에서는 Cabri Géomètre II에서 작성한 Fig파일을 인터넷 상에서 사용자의 컴퓨터로 다운로드 받아서 사용할 수 있다. 인터넷에서 다운로드 받은 Fig파일을 Cabri Géomètre II 프로그램에서 불러 재사용할 수 있지만, 제작자가 다운로드 받을 수 없게 설정할 경우 파일을 다운로드 받을 수 없다. 이것은 JavaSketchpad와 비교하여 큰 장점이나 단점이 될 수 있다.

(1) Cabri II File 디자인

- ①  점 도구를 이용하여 점 A, B, C를 작도한다.
- ②  선분 도구를 이용하여 선분 AB, 선분 BC, 선분 AC를 작도한다.
- ③  중점 도구를 이용하여 각 선분의 중점(P, Q, R)을 작도한다. ( 을 선택한 다음, 각 각의 선분을 클릭하면 중점이 작도된다.)
- ④  수선 도구를 이용하여 수선(j, k, l)을 작도한다. ( 수선 도구를 이용하여 수선이 결정되는 선과 점을 선택하면 작도된다.)
- ⑤  교점 도구를 이용하여 교점(외심)을 작도한다. ( 교점 도구를 이용하여 교점이 결정되는 두 개의 수선을 선택하면 작도된다.)
- ⑥  원 도구를 이용하여 외접원 (C1)을 작도한다. ( 원 도구를 이용하여 첫 번째 점(원의 중심)과 두 번째 점(원 위의 점)을 선택하면 작도된다.)
- ⑦ 완성이 되면 저장 - 여기서는 pds.fig로 저장 - 한다.



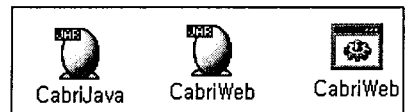
(2) Cabri II의 HTML 변환

먼저 HTML 변환에 필요한 파일을 다운받는다.

- ① CabriJava.jar 다운받기 (가장 최근의 것을 다운받는다)
<http://www.cabri.net/cabrijava>

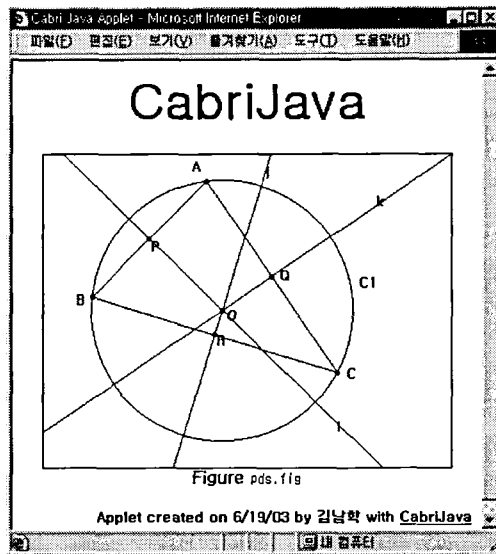
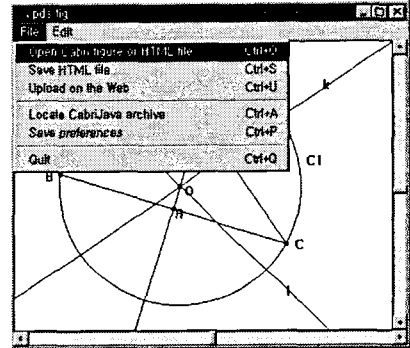
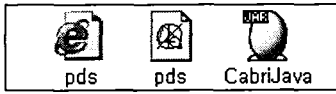
- ② CabriWeb 다운받기
<http://www.cabri.net/cabrijava/CabriWeb.html>

③ 압축파일을 풀면 2개의 파일 cabriweb.bat, cabriweb.jar이 나타난다. 이렇게 Cabri 파일을 변환시키기 위해서는 기본적으로 3개의 파일이 준비되어야 한다.



- ④ cabriweb.bat를 실행시킨다.
- ⑤ pds.fig 파일을 불러온다.
- ⑥ Save HTML file로 저장한다. 그러면 pds.htm이 만들어진다. 이때 반드시 pds.htm과 pds.fig, CabriJava.jar 파일 3개가 같은 경로상에 위치해야 한다.

Open Cabri figure or HTML file	Ctrl+O
Save HTML file	Ctrl+S
Upload on the Web	Ctrl+U
Locate CabriJava archive	Ctrl+A
Save preferences	Ctrl+P
Quit	Ctrl+Q



```

1 <HTML>
2 <HEAD>
3   <TITLE>Cabri Java Applet</TITLE>
4 </HEAD>
5 <BODY BGCOLOR="#FFFFFF">
6 <H2 ALIGN=center><FONT SIZE=+4>CabriJava</FONT></H2>
7 <P ALIGN=center><APPLET CODE="CabriJava.class" WIDTH="396" HEIGHT="304" ALIGN=bottom
  ARCHIVE="CabriJava.jar" >
8   <PARAM NAME=file VALUE="pds.fig">
9   <PARAM NAME=lang VALUE="en">
10  <PARAM NAME=xposition VALUE="-10">
11  <PARAM NAME=yposition VALUE="-20">
12 </APPLET><BR><B>Figure </B><TT>pds.fig</TT>
13 <P ALIGN=right><FONT SIZE=-1><B>Applet created on 6/19/03 by 김남학 with </B><A
  HREF="http://www.cabri.net/cabrijava"><B>CabriJava</B></A></FONT>
14 </BODY>
15 </HTML>
  
```

(3) CabriJava의 HTML 소스

기본적인 HTML 소스는 JSP와 같다. 그러나 JSP의 경우에는 작도의 내용이 명령어로 들어 있지만, CabriJava의 경우에는 작도의 내용 대신 Cabri Géomètre II로 작도한 Fig파일의 이름이 들어 있다.

- <APPLET></APPLET> Tag

- ▶ CODE Parameter는 제일 처음으로 가지고 올 Java.class 파일의 이름을 지정한다. 이 "CabriJava.class"의 이름을 바꾸면 안 된다.

- ▶ ARCHIVE Parameter는 CabriJava에서 ARCHIVE="CabriJava.jar"로 설정되어 있다.

- ▶ CODEBASE Parameter는 여기서는 없지만 <APPLET> 태그를 가지고 있는 웹페이지의 URL과 관련하여, 웹서버에 애플릿이 있는 폴더의 경로를 나타낸다. CabriJava 애플릿을 포함하고 있는 폴더와 다른 위치에 웹페이지가 있다면, 이 애플릿이 있는 경로를 반드시 이곳에 써 주어야 한다. JSP의 경우와 같이 웹 상에서 제대로 나타나지 않는 경우 이곳이 문제인 경우가 많다.

- ▶ Construction Parameter

<PARAM NAME=file VALUE="pds.fig">

실제로 브라우저 상에 나타나는 움직이는 그림을 표현하는 부분으로 Cabri Géomètre II에서 작도한 파일의 이름을 적는다. 주의할 점은 웹서버가 Unix기반으로 운영되는 경우 대소문자를 구별하는 경우가 많으므로, 파일명의 대소문자를 반드시 확인해야 한다. 또한 Fig파일을 별도의 폴더에 모아 둔 경우 경로를 같이 써 주어야 한다. 예를 들어 figure폴더 안에 파일이 있을 경우 "figure/파일명.fig"라고 적어주어야 한다.

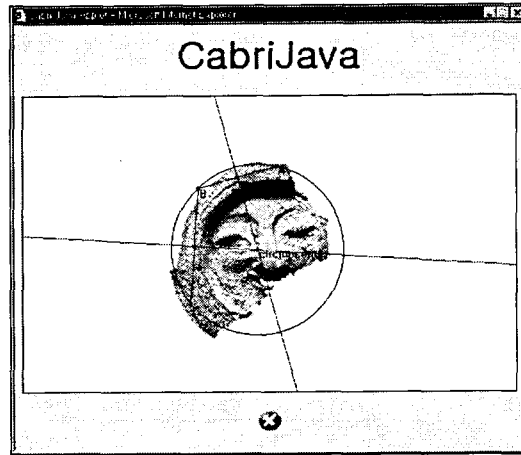
<PARAM NAME=lang VALUE="en">

Applet 상에서 표현되는 언어를 설정한다. en은 영어를 나타내며 프랑스어, 이태리어 등 주요 유럽권 언어 10가지 중 선택할 수 있고, 아직 한글이나 일본어 등 동양어는 지원되지 않는다.

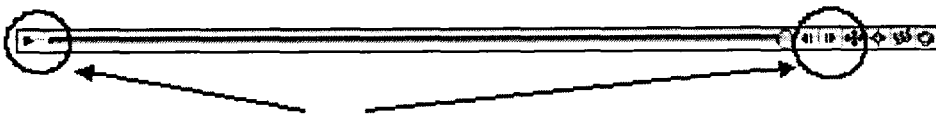
그 밖의 움직이는 흔적을 남기는 trace, 애니메이션이 실행되는 spring, 작도 단계를 순서대로 보여주는 step 등 여러 가지 매개변수들이 있으며, 자세한 문법적 설명은 <http://www.cabri.net/cabrijava>를 참고하면 된다.


(4) CabriJava의 특징

CabriJava의 가장 큰 특징은 도구막대(Toolbar)의 사용이라 할 수 있다. CabriJava의 Toolbar에는 작도과정을 차례대로 보기가 가능할 뿐 아니라, 애니메이션 스프링을 이용한 애니메이션의 지정, Cabri Géomètre II로 작도된 Fig파일 다운로드 등이 있다. 아래 예는 깨진 얼굴 모양 수막새의 중심을 삼각형의 외심을 이용해서 찾는 CabriJava를 이용하여 제작된 애플릿(Applet)이다.



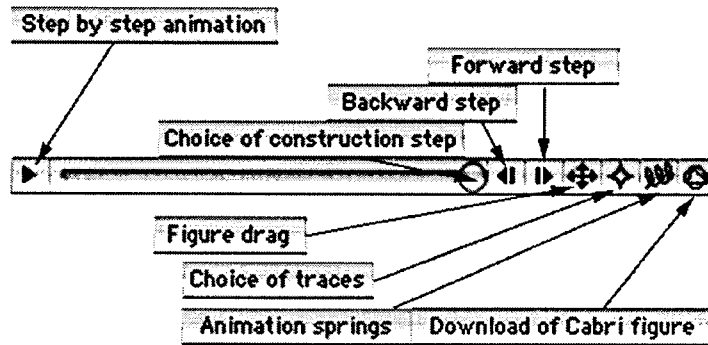
Applet 창을 더블클릭하면



Toolbar 가 나타나는데  을 클릭하면 단계별 작도 과정 등을 살펴볼 수 있다. 즉, 맨 왼쪽의 버튼을 누르면 다음 그림과 같이 작도과정이 처음부터 순서대로 반복되어 학습자가 작도과정을 다시 살펴볼 수 있다. 이때 오른쪽의 버튼을 누르면 한 단계씩 반복되므로 학습자의 학습속도에 맞게 조절할 수 있다. GSP로 작도된 도형을 JSP로 변환하여 웹에 올릴 경우, 이런 과정을 직접 버튼을 만들어주어야 하는데 CabriJava의 경우 기본적으로 이런 기능을 가지고 있다. 또한 애니메이션의 경우도 JSP는 처음 제작된 애니메이션을 학습자가 조절할 수 없는데, CabriJava의 경우 Toolbar의 스프링을 이용하여 애니메이션을 학습자가 제어할 수 있을 뿐 아니라, 제작된 애니메이션이 없어도 학습자가 직접 탐구하며 애니메이션을 가능하게 해 준다. 도형의 흔적을 남기는 trace기능 역시 CabriJava에서 직접 학습자가 선택할 수 있게 한다.

그러나 CabriJava의 경우 현재까지 JSP나 Cinderella와는 달리 Applet 상에서 한글이 지원되지 않으며, 별도의 동작을 제어할 수 있는 동작버튼이 없어 학습내용을 원하는 대로 제한하는 기능이 없다. JSP의 경우 애플릿을 제작할 때, 제작자가 학습목표에 맞게 제한해서 학습을 안내할 수 있는데 비해 CabriJava의 경우 학습자의 탐구기능에 의존하므로 주어진 학습목표에서 벗어날 개연성이 많다. 물론 이것은 역설적으로 학습자가 탐구할 수 있는 활동이 많아질 수도 있다는 것을 의미하고 있다.

아래 그림은 CabriJava의 홈페이지에 있는 것으로 Toolbar의 기능을 알려주고 있다.



3. Cinderella

3.1 Cinderella의 역사

Cinderella는 컴퓨터로 기하를 실행하는 소프트웨어이다. 이 소프트웨어는 1993년에서부터 1998년까지의 일련의 프로젝트에 의하여 탄생되었다. Cinderella는 사영기하학(Projective Geometry), Cayley-Klein Geometry, 복소해석학(Complex Analysis) 등 과거에서부터 현재까지의 여러 수학적 기반을 바탕으로 동적 기하를 가장 완벽하게 구현한 것으로 평가받고 있다([12]).

1992년 스웨덴의 Mittag-Leffler 연구소에서 열린 조합론에 관한 학회에 참석한 Henry Crapo와 Jürgen Richter-Gebert는 신데렐라(Cinderella)라는 배에서 서로 대화를 나누는 중에 새로운 형태의 기하 소프트웨어에 관한 생각을 하게 되었다. 당시 Jürgen Richter-Gebert는 몇 번의 마우스 클릭과 질문으로 기하학적 성질을 증명하게 하는 자동 정리증명을 연구하고 있었다. Henry와 Jürgen은 당시 가장 획기적인 NeXT사 NeXTSTEP이라는 운영체제를 바탕으로 개발에 착수하였다. NeXT사는 개인용 컴퓨터를 최초로 만들고 Apple 컴퓨터를 설립한 Steve Jobs가 세운 컴퓨터 회사이다. Jobs는 항상 컴퓨터의 이상적인 환경을 추구하였으나 대중적인 보급에는 거의 무신경하여 NeXTSTEP은 상업적으로는 성공하지 못한다. 그러나 당시 NeXT는 획기적인 프로그램 개발 도구, Objective-C, 화면과 인쇄에 PostScript를 채용하여 기술적으로는 가장 진보적인 운영체제였다. 이런 환경이 Cinderella의 개발환경으로 채용된 것으로 보여진다. 그러나 그 당시에는 너무 앞선 기술이었기 때문에 대중성을 확보하는데 실패하였으며, Cinderella 역시 1995년 여름 이후 개발이 더 이상 진전이 되지 않았다.

1996년 6월 Jürgen Richter-Gebert는 Mt. Holyoke에서 열린 Discrete and Computational Geometry에 관한 학회에서 Cinderella를 발표하기로 되어 있었는데, 당시 그곳에는 개발 초기의 NeXT만 몇 대 있었다. 발표시간 직전야 비로소 Cinderella가 돌아가는 NeXT를 구하여 Cinderella를 발표하였다. 당시 Jürgen Richter-Gebert은 많은 사람들이 사용하지 않는 컴퓨터 환경에서 다시는

발표하지 않겠다고 마음 먹었다. 그러나 어렵게 발표한 Cinderella가 크게 호평을 받아 다른 학회의 발표에 초청을 받았다. 시간이 촉박한 탓으로 일에도 능률이 올라 재개발하기 시작한 지 3주 만인 1996년 8월 Jürgen Richter-Gebert는 당시 베를린공업대학교 Günter M. Ziegler의 그룹에서 공부하던 Ulrich Kortenkamp와 더불어 실행속도는 NeXT보다 다소 느리지만 모든 컴퓨터 사양에서의 무리 없이 실행되는 독립 언어인 자바를 기초로 하는 새로운 프로젝트를 행하면서 완성이 되었다.


1999년 Jürgen Richter-Gebert와 Ulrich Kortenkamp는 Springer-Verlag 출판사를 통해 다른 소프트웨어와는 달리 패키지가 아닌 책의 부록으로 『The Interactive Geometry Software - Cinderella』를 출판하였으며, 같은 해 Ulrich Kortenkamp는 Cinderella의 개발 자체를 하나의 연구 목적으로 하여 그의 박사학위 논문인 The Foundation of Dynamic Geometry를 제출하였다. 이 논문에는 Cinderella의 기초가 되는 수학적 이론들이 상세하게 기술되어 있으며, GSP나 Cabri Géomètre와는 달리 Cinderella는 순수 수학을 연구하는 학자들이 중심이 되어 제작되었기 때문에, 초·중등학교의 교구라기보다는 대학과정에서의 기하 탐구도구로 인식되어져 왔다. Cinderella는 1998년 12월 이후 거의 기본적인 틀의 변화가 없이 현재까지 이어져 오고 있으며, Version 1.4(JP 13)이 나와 있다([12]).


Cinderella는 마우스 동작을 기초로 한 쌍방향 기하 프로그램이다. 몇 번의 마우스 클릭만으로 거의 모든 기하학적 도형을 그려낼 수 있다. 작도를 완성한 후에는 마우스로 기초가 되는 개체를 선택하여 드래그를 하면 마우스를 따라서 전체 도형이 연속적으로 움직인다. 이 움직임은 정확한 수학적 계산 하에 화면상에 나타낸다. 또 하나의 두드러진 특징으로 자동 증명 기능(automatic proving facility)을 들 수 있다. 사용자가 도형을 작도하는 동안 신데렐라는 일어날 수 있는 자명하지 않은 사실들을 증명하여 알려준다. 동시에 Cinderella는 다른 관점에서의 작도 - 예를 들면 유클리드 평면, 구면, 포앙카레 쌍곡면에서의 작도 - 를 보여준다. 또한 전체 프로그램이 자바로 구성되어 있기 때문에, 각각의 작도는 상호작용하는 웹 페이지로 export할 수 있다. 학생들을 위한 연습문제나 애니메이션도 자바 애플릿으로 export된다. 이 기능은 다른 소프트웨어처럼 단순한 애니메이션 프로그램을 웹에서 보여주는 것이 아니라, 학생들 스스로 주어진 문제에 대한 작도를 할 수 있고, 이에 대한 적절한 도움말까지 얻을 수 있다.



앞에서 이야기한 바와 같이, Cinderella는 웹에서 많은 장점을 가지고 있는 자바 언어를 이용하여 만들어졌으므로 WBI를 구현하는 데 있어서 기존의 프로그램에 비하여 탁월한 기능을 보유하고 있으며, 자동증명기법의 도입, 연속성 문제의 해결 등 교육적으로 중요한 기능적 우위를 보이고 있다. 무엇보다도 동적 기하를 구현하기 위한 내부 알고리즘의 엄밀성이 다른 소프트웨어에 비하여 우월하므로 이를 활용한 여러 가지 새로운 교육적 시도가 가능하도록 해준다.



3.2 Cinderella를 이용한 웹 파일 제작



(1) Cinderella File 디자인



- ①  점 도구를 이용하여 점 A, B, C를 작도한다.

②  선분 도구를 이용하여 선분 AB, 선분 BC, 선분 AC를 작도한다.

③  중점 도구를 이용하여 각 선분의 중점(D, E, F)을 작도한다. ( 을 선택한 다음, 각 선분의 두 점 사이를 드래그 하면 중점이 작도된다.)

④  수선 도구를 이용하여 수선(d, e, f)을 작도한다. ( 수선 도구를 이용하여 수선이 결정되는 선과 점을 선택하면 작도된다.)

⑤  교점 도구를 이용하여 교점(외심)을 작도한다. ( 교점 도구를 이용하여 교점이 결정되는 두 개의 수선을 선택하면 작도된다.)


⑥  원 도구를 이용하여 외접원(C1)을 작도한다. ( 원 도구를 이용하여 첫 번째 점(원의 중심)과 두 번째 점(원 위의 점)을 선택하면 작도된다.)

⑦ 완성이 되면 저장 - 여기서는 oc.cdy로 저장 - 한다.

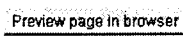

(2) Cinderella의 HTML 변환

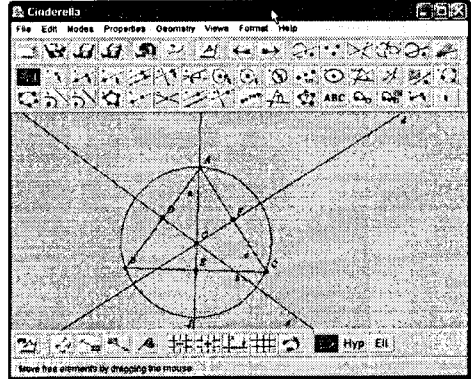
Cinderella에서는 프로그램 자체적으로 HTML 파일로 변환해서 저장시킬 수 있다. 물론 변환된 HTML 파일이 jsp 폴더와 같은 경로에 있어야 한다.

① 파일메뉴에서 “Export to HTML”을 실행한다.

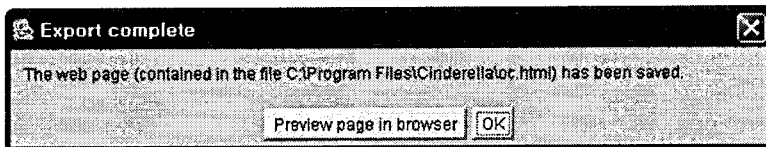
② “Save Webpage”라는 창이 뜨는데, 이 때  버튼을 누르면 파일명.html로 저장이 된다.

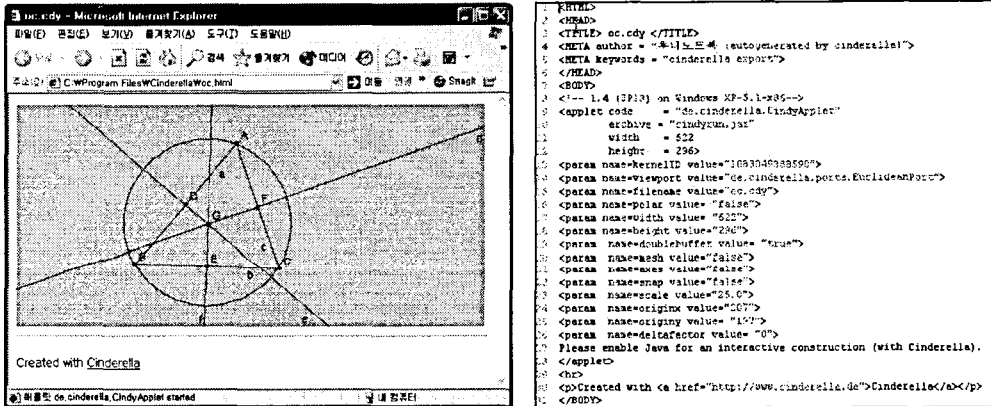
③ 저장이 완료되면 “The web page (contained in the file C:\경로\파일명.html) has been saved” 라는 메시지가 나타나는데, 이것으로 변환은 끝난다.

④  버튼을 클릭하면 즉시 웹 브라우저에서 확인할 수 있으며,  버튼을 누르면 확인 없이 그냥 파일만 저장된다.



New	Ctrl+N
Load	Ctrl+O
Save	Ctrl+S
Save as	Ctrl+Shift+S
Export to HTML	Ctrl+E
Open Exercise Editor	Ctrl+B
Print	Ctrl+P
Exit	Ctrl+Q





(3) Cinderella의 HTML 소스

Cinderella의 HTML 파일의 생성방식은 GSP4에서 JSP파일을 만드는 것과 같은 방식이며, HTML 파일 생성 후에는 Cabri Géomètre II를 이용한 작도 및 CabriJava를 이용한 HTML 파일과 마찬가지로 Cinderella로 작도한 cdy파일의 이름이 들어 있다. 즉, Cinderella 프로그램 자체에서 작도한 후 HTML 방식으로 Export한 후, 그 작도된 파일을 그대로 HTML 파일에서 불러 사용하는 방식이다.

• <APPLET></APPLET> Tag

▶ CODE Parameter는 제일 처음으로 가지고 올 Java.class 파일의 이름을 지정한다. 이 "de.cinderella.CindyApplet.class"의 이름을 바꾸면 안 된다.

▶ ARCHIVE Parameter는 Cinderella에서 ARCHIVE="cindyrun.jar"로 설정되어 있다.

▶ Construction Parameter

```
<PARAM NAME=filename VALUE="oc.cdy">
```

CabriJava에서 <PARAM NAME=file VALUE="파일명.fig">와 같이 실제로 브라우저 상에 나타나는 움직이는 그림을 표현하는 부분으로 Cinderella에서 작도한 파일의 이름을 적는다. 주의할 점은 웹서버가 Unix기반으로 운영되는 경우 대소문자를 구별하는 경우가 많으므로 파일명의 대소문자를 반드시 확인해야 한다. 또한 cdy파일을 별도의 폴더에 모아 둔 경우 경로를 같이 써 주어야 한다. 예를 들어 figure폴더 안에 파일이 있을 경우 "figure/파일명.cdy"라고 적어주어야 한다.

그 밖의 여러 <param name= > 등은 Cinderella 프로그램의 Appearance Editor에서 고칠 수 있으며 사용자는 가급적 고치지 않는 것이 좋다. Cinderella의 도움말에도 이 부분에 대한 자세한 설명이 없이 고치거나 지우지 않기를 당부하고 있다.

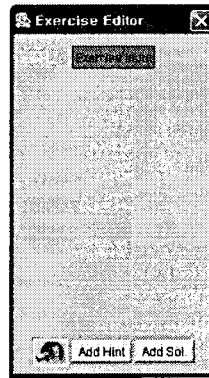
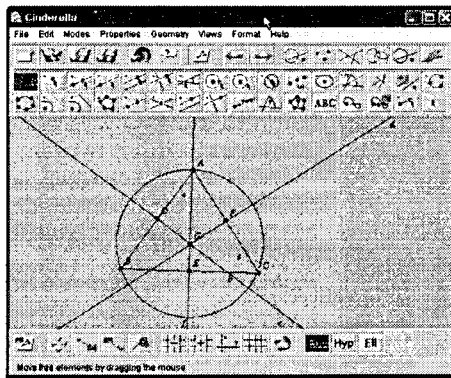
(4) Cinderella의 특징


Cinderella는 Java를 이용하여 개발하였으므로 일반적인 Windows 기반의 컴퓨터, Apple 컴퓨터의 Mac 환경은 물론, Linux 기반의 컴퓨터 등 컴퓨터의 환경에 상관없이 설치·사용할 수 있다. 이것은 Cinderella의 최초 버전이 폐쇄적인 NeXT 환경에서 개발되어 보급에 애를 먹었던 개발 초기의 역사에 기인한다. GSP나 Cabri Géomètre II가 그 작도 파일을 전부 웹 환경으로 옮기지 못하는 것과 달

리, Cinderella는 Java를 개발언어로 사용하였기 때문에 웹으로 전환시키는 데도 전혀 어려움이 없다. 이것은 Cinderella가 웹 환경에서 가지는 가장 큰 장점으로 Cinderella의 작도환경과 웹에서의 사용 환경이 완전히 같다는 것을 의미한다. 더욱이 Cinderella는 웹 환경에서도 작도환경에서 사용하는 각종 도구들을 사용할 수 있다. 즉, 학습자가 필요한 도구를 사용하여 직접 작도해 볼 수 있다는 것을 의미한다. 이를 활용하면 먼저 문제를 제시해 주고 학습자에게 도구를 사용하여 필요한 작도를 하게 할 수 있으며, Cinderella는 학습자가 작도한 내용을 정답과 비교하여 학습자에게 피드백 시킬 수도 있다. 물론 학습자가 해결과정에서 어떤 방법이나 순서로 작도하였는지는 중요하지 않으며, 시행착오를 겪어서 원하는 작도를 행한 경우에 그에 대응하여 알맞은 답을 한다. 아울러 학습과정을 설계하는 사람이 적당한 힌트를 줄 수도 있다.

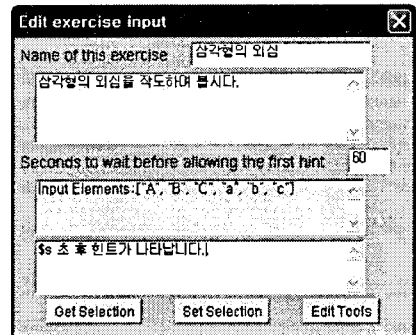
아래 과정은 Cinderella에서 삼각형의 외심을 도구를 이용하여 학습자에게 작도하게 하는 과정이다.

- ① 먼저 학습자가 작도해야 할 과제를 설계자가 완성한다. 여기서는 삼각형의 외심을 작도한다.



- ② 도구상자의  Exercise Editor를 누르면 그림과 같이 Exercise Editor가 나타난다.

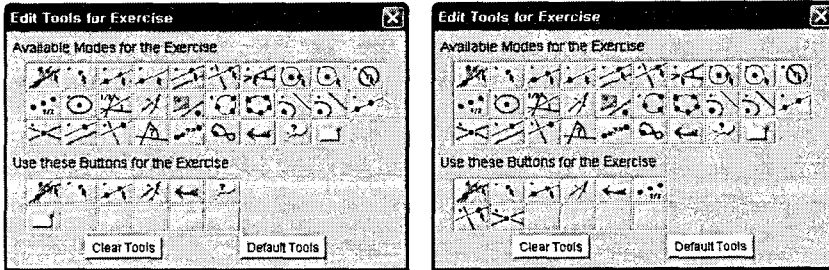
③ “Exercise Input”을 더블 클릭하면 그림과 같이 “Edit exercise input” 창이 나타난다. 여기에 과제의 이름, 학습자에게 제시될 작도 과제, 학습자에게 주어지는 기본 작도 내용(Input Elements) 및 힌트가 나타나는 시간을 설정한다. 여기서는 과제의 이름을 “삼각형의 외심”, 과제를 “삼각형의 외심을 작도하여 봅시다.”, 힌트가 나타나는 시간은 60으로 주었다.



④ **Get Selection** 버튼을 누르면 기본 작도 내용을 선택할 수 있다. 여러 개의 개체 - 즉, 점과 선 등 - 를 선택할 때는 shift를 누른 상태에서 선택하면 된다. 여기서는 기본 작도내용으로 삼각형의 세 점(A, B, C)과 세 변(a, b, c)을 주었다.

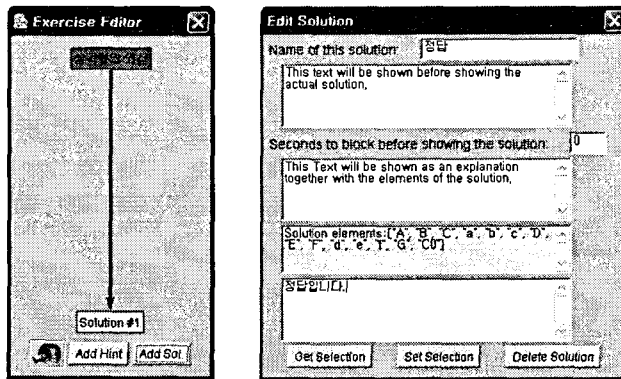
- ⑤ **Edit Tools** 버튼을 눌러 “Edit Tools for Exercise” 창을 열어 웹 상에서 학습자가 사용할 도구를


선택한다. Cinderella에서 사용할 수 있는 거의 모든 도구를 더블 클릭 하면 선택하여 사용할 수 있다. 아래 그림에서 왼쪽은 기본으로 설정된 도구이며, 오른쪽은 필요한 도구를 선택한 후이다.

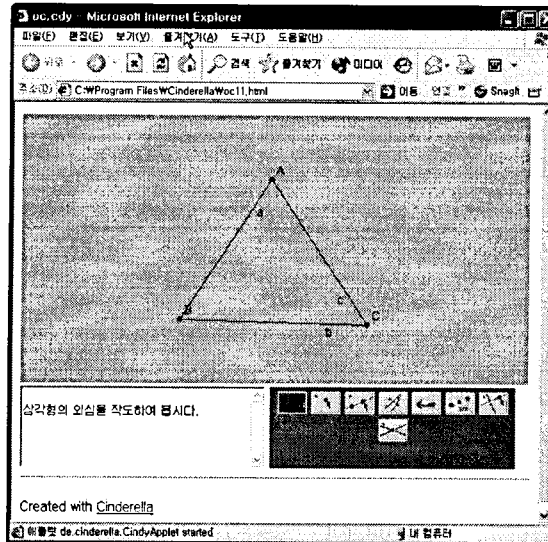


⑥ 도구를 선택하고 “Edit Tools for Exercise” 창을 닫은 후, 다시 “Exercise Editor” 창에서 **Add Hint** 버튼을 누르면 학습자에게 제공될 힌트를 작성할 수 있다.

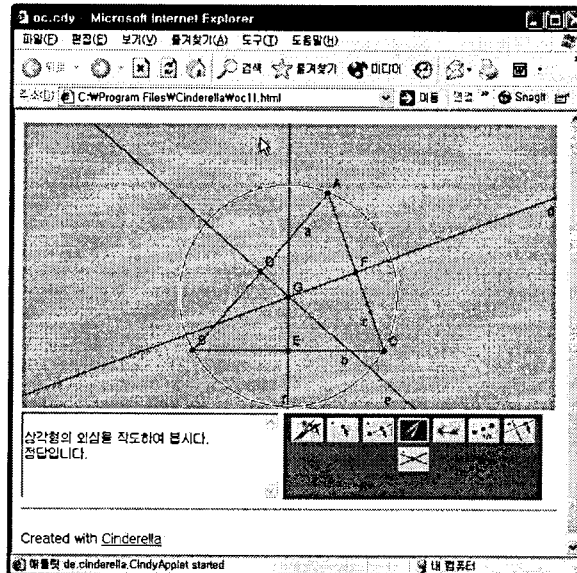
⑦ “Exercise Editor” 창에서 **Add Sol** 버튼을 눌러 학습자가 작도한 결과를 확인할 수 있는 정답을 작성한다. **Add Sol** 버튼을 누르면 “Solution #1”이 생기며, “Solution #1”을 더블 클릭 하면 “Edit Solution” 창이 나타난다. “Edit Solution” 창에서는 풀이(solution)의 이름과 풀이에 해당하는 작도내용, 풀이가 끝났을때 학습자에게 보여지는 내용을 편집할 수 있다. 만약 풀이가 여러 개일 경우 “Exercise Editor” 창에서 **Add Sol** 버튼을 눌러 풀이방법을 계속 추가할 수 있다.



⑧ 지금까지 제작한 내용을 웹으로 출판하기 위해서는 “Exercise Editor” 창에서  버튼을 누르면 된다. 여기서는 아래 그림과 같이 웹에 나타나는데 기본적으로 주어진 도형은 위의 과정을 거쳐 삼각형의 세 점과 세 변이 나타나며, 작도할 수 있는 공간 아래의 왼쪽에는 과제, 오른쪽에는 학습자가 사용할 수 있는 도구가 나타난다.



⑨ 학습자는 Cinderella에서 만들어진 동적 기하 기반의 웹 자료를 활용하여 학습할 수 있는데, 도형 아래 오른쪽에 주어진 도구를 이용하여 도형을 직접 작도할 수 있다. 학습자가 작도 도구를 이용하여 시행착오를 거쳐 여러 가지 작도를 완성하면 미리 입력된 문장이 나타난다. 물론 학습자는 어떤 과정이나 단계를 설계자의 의도와는 달리 생각할 수도 있으며, 여러 개의 풀이가 있을 경우, 모든 풀이를 다 확인한다. 여기서는 삼각형의 외심을 작도하는 과제를 제시하였는데, 과제를 완성하면 “정답입니다.”라는 문장이 나타난다.



이상에서 살펴본 바와 같이 Cinderella를 이용하여 웹 상에서 DGS를 구현하면, GSP를 활용한 JSP나 Cabri Géomètre II를 활용한 CabriJava와는 달리, 직접 도구를 이용하여 학습자에게 작도환경을 제공해 줄 수 있다는 매우 큰 이점이 있다. 그러나 애니메이션을 제어한다든지 작도된 도형에 라벨을 붙이는 등의 기능은 앞의 두 프로그램이 더 낫다.

IV. 결론 및 제언

1. 결론

지금까지 GSP를 이용한 JSP, Cabri Géomètre II를 이용한 CabriJava 및 Cinderella를 이용하여 웹 환경에서 DGS의 구현에 대해 살펴보았다. 본 논문에서는 전체적으로 어느 프로그램이 가장 낫다는 것보다는 각 프로그램이 가지는 장점을 중심으로 WBI 환경에서 DGS가 발전해야 할 방향을 제시하고자 한다.

첫째, 각 프로그램에서 작도를 한 후 웹 상의 HTML 파일로 바꾸는 과정은 GSP나 Cinderella에서 처럼 각각의 작도 프로그램의 메뉴에서 저장하는 방식을 선택하든지, 아니면 직접 HTML 형식으로 export 하는 방식이 가장 효과적이다. 그러나 CabriJava와 같이 별도의 변환 프로그램이 필요하지 않는 방식이 더욱 효과적이다.

둘째, HTML 파일 안의 자바 애플릿의 실행방식은 JSP의 경우처럼 애플릿 안에 별도의 작도 코드를 생성하는 방식과 CabriJava나 Cinderella의 경우처럼 각 프로그램에서 작도된 파일을 그대로 웹에서 불러 쓰는 방식 두 가지 모두를 지원해야 한다. 전자의 경우 고급 사용자가 코드를 이해하여 세밀한 조정을 할 수 있는 장점이 있으며, 후자의 경우 HTML 변환 후에도 HTML 파일의 변환 없이 원 작도파일의 수정만으로 웹 상의 자료를 수정할 수 있어 HTML 파일과 작도파일의 유지 보수를 따로 할 필요가 없으며, 초·중급자가 HTML 파일을 전혀 신경쓸 필요가 없다는 장점이 있다. 현재 Cabri Géomètre II Plus나 Cinderella의 경우 작도 과정을 텍스트 형태로 보여주는 메뉴가 프로그램에 이미 있으므로 이것을 활용하면 JSP처럼 코드를 생성하는 방식으로 제작할 수 있다.

셋째, 파일의 호환성을 고려해야 한다. 현재 널리 보급된 DGS프로그램 및 이미 제작된 학습파일의 대부분이 이 세 프로그램을 기반으로 제작되어 있다. 그러나 이들 프로그램 사이의 호환성이 전혀 없기 때문에 학습자가 웹 상에서 학습을 위한 DGS활용 파일을 구하더라도 파일을 제작한 프로그램이 없으면 전혀 사용할 수 없다. 따라서 프로그램 간의 파일 호환성을 높여 다른 프로그램에서 작성한 파일을 불러올 수 있는 기능이 필요하다. 아울러 웹에서 작도된 파일을 다운로드 할 수 있는 기능이 지원되어야 하는데, 현재 CabriJava의 경우에만 이 기능이 지원된다.

넷째, GSP의 이동, 애니메이션 및 보이기/감추기 버튼처럼 학습자에게 학습 단계를 안내하거나 작도하는 단계를 학습 설계자의 의도대로 안내할 수 있는 버튼을 사용할 수 있어야 한다. 현재

CabriJava의 경우 설계자의 의도와 관계없이 작도과정 하나하나를 마우스를 통해 일일이 눌러주어야 하거나, 아니면 작도과정 전체를 학습자가 처음부터 끝까지 지켜보는 방법밖에 없으며, Cinderella의 경우는 작도과정을 보여주는 기능을 지원하지 않는다.

다섯째, CabriJava의 경우처럼 학습자가 작도된 모든 개체를 애니메이션 시키거나 움직임의 흔적을 찾을 수 있는 기능을 지원하여야 한다. 학습자가 설계자의 의도와 계획대로 안내된 과정을 따라 하는 것 이상으로 학습자가 어떤 것이든 직접 탐구할 수 있는 기능이 필요하다. CabriJava의 경우 흔적남기기, 애니메이션 등을 작도된 모든 개체에 적용시킬 수 있으므로 웹 환경에서도 진정한 의미에서의 탐구활동이 일어날 수 있다.

여섯째, Cinderella의 경우와 같이 직접 작도도구를 학습자에게 제공해주어 웹 상에서 작도가 가능해야 한다. 단순히 설계자가 작도한 내용을 탐구하는 단계에서 더 나아가 학습자가 작도도구를 이용하여 직접 작도할 수 있는 기능이 필요하다. 아울러 학습자가 작도한 내용을 피드백 할 수 있는 기능도 필요하다. 현재 GSP나 Cabri Géomètre II Plus는 이 기능을 지원하지 않는다.

일곱째, 보다 나은 학습 환경을 조성하기 위해 작도된 개체 - 예를 들어 점 - 에 그림이나 음향 등을 연결할 수 있어야 한다. 점이나 선이 애니메이션 될 때 그림이 같이 움직이거나 움직임에 따른 음향이 지원되면 학습자에게 흥미를 불러일으킬 수 있다. 현재 GSP를 활용한 JSP의 경우는 그림을 연결하는 기능이 지원되지만 음향은 지원되지 않으며, Cabri Géomètre II Plus의 경우 작도는 가능하나 CabriJava에서는 이 기능조차도 지원하지 않는다. Cinderella의 경우도 마찬가지이다.

여덟째, 언어의 장벽을 극복해야 한다. 현재 JSP나 Cinderella의 경우 웹 환경에서 한글이 지원이 가능하지만 CabriJava에서는 한글이 지원되지 않는다. 그러나 Cabri Géomètre II Plus의 경우는 작도 환경에서 라벨 등에 한글이 지원된다.

아홉째, 좌표를 이용하는 해석기하의 지원이 더 강화되어야 할 것이다. 현재 각 프로그램은 동적 기하 환경을 웹 환경으로 변환시키는 기능에 중점을 두고 있기 때문에, 각 프로그램의 작도상태에서 지원되는 좌표나 그래프기능의 해석기하적 기능의 지원은 그다지 많지 않다.

위에서 언급한 내용들은 GSP를 활용한 JSP, Cabri Géomètre II를 활용한 CabriJava 및 Cinderella 라는 DGS프로그램을 사용하고 분석하면서 필요한 기능을 정리한 것이다. 앞으로 WBI환경에서 효과적인 DGS를 구축하기 위해서는 이런 기능들이 보강된 소프트웨어의 출현이 필수적이다. 아울러 많은 수학교사들이 개별 소프트웨어의 사용법 및 동적 기하에 대한 적극적인 마인드를 가지고 더 나은 학습 환경을 구축해야 할 것이다.

2. 제언

수학교육을 위한 우수한 테크놀러지가 있더라도 그 테크놀러지를 교수·학습에 사용하는 교사가 없으면 그것은 무용지물이며, 이미 주어진 테크놀러지라도 단순히 주어진 대로 사용하는 것은 매우 위험한 일이다. 어떤 테크놀러지나 도구가 주어지더라도 교사가 목적에 맞게 연구하고 사용하기 위

해서는 다음과 같은 것이 필요하다고 여겨진다.

첫째, 현대사회에서 테크놀러지의 발달은 하루가 다르게 빨라지고 있어 교사 개개인이 많은 것을 연구하고 사용할 수 없으므로 공동연구가 필요하다. WBI환경에서 DGS프로그램의 개선에 있어, 학교 현장에서 학생을 가르치는 교사는 프로그램 개발 능력이 부족하며, 프로그램 개발자는 학생을 가르친 경험의 부족으로 인하여 학습효과 및 사용자 편의성 등 프로그램의 발전 방향에 대한 피드백이 필요하므로 현장경험이 풍부한 교사, 수학교육연구자 및 기본적인 수학교육에 대한 마인드를 가진 개발자가 협력해야 할 것이다.

둘째, DGS나 WBI가 이미 많이 보급되어 학교에서 사용되고 있지만 그 효과에 관한 연구 및 DGS를 이용한 효과적인 WBI의 개발을 필요로 한다. 예를 들어, 학습 설계자가 안내한 방법대로 학습한 경우와 학습자에게 탐구활동의 대부분을 자율적으로 맡긴 경우의 비교를 통해 학습자의 개념형성에 대한 변화, 동적 기하 프로그램의 사용 환경이 학습자의 개념 형성에 미치는 영향 등 단순한 소프트웨어의 환경에 대한 비교나 분석에 관한 연구가 아니라, 학습자와의 상호작용 속에서 학습자에게 DGS가 미치는 영향 등에 대해 앞으로 더 많은 연구가 되어야 할 것이다.

참 고 문 헌

김부운 (2002). 수학교육의 흐름, 제 7 교육과정에서 그래핑 계산기(TI-92)의 수학과 교수·학습의 실제 2002년도 기초과정 연구교재, 부산 : 부산대학교 사범대학 부설 중등교육연수원.

김용구 (2003). 역동적인 기하 프로그램 Cinderella의 소개, 수학사랑 제 5회 Math Festival 프로시딩, 제5집 1권, pp.88-92.

나일주 (1999). 웹기반교육, 서울 : 교육과학사.

박홍선 (2000). 웹활용교육, <http://kmh.ync.ac.kr/html/whtml/wbi/>.

수학사랑 (2002). 한글판 GSP Version 4, 서울 : 수학사랑.

정재훈·손대원 외 (2002). 살아있는 기하(GSP에서 ISP까지), 창원 : 경상남도교육청.

Burgiel, H. (1999). Review of The Interactive Geometry Software Cinderella, *The American Mathematical Monthly*, <http://memeticdrift.net/tau/cinderella.htm>

Cabrilog (2003). *Cabri Geometry II Plus*, Cabrilog.

Finzer, W. & Jackiw, N. (1998). *Dynamic manipulation of mathematical objects* (<http://www.keypress.com/sketchpad>).

Jackiw, N. (2003). *JavaSketchpad Construction Grammar Developer's Release 4* (<http://www.keypress.com/sketchpad>).

Kaput, J. J. (1998). Mixing New Technologies, New Curricula and New Pedagogies to Obtain Extraordinary Performance from Ordinary People in the Next Century, *Proceedings of ICMI -*

EARCOME Vol. 1, pp.141-169, 박성선 (역), 다가올 21세기에 있어서 일반인을 탁월한 능력을 갖춘 사람으로 만들기 위한 새로운 테크놀로지, 새로운 교육과정, 새로운 교육방법의 통합적 운영.

Kortenkamp, U. (1999). *Foundation of Dynamic Geometry*, Doctoral Dissertation, Swiss Federal Institute of Technology Zurich.

NCTM(2000), *Principles and Standards for School Mathematics*, NCTM.

Richter-Gebert, J. & Kortenkamp, U. (1999). *The Interactive Geometry Software—Cinderella*, Springer-Verlag.

Scher, D. (2000). Lifting the Curtain : The Evolution of The Geometer's Sketchpad, *The Mathematics Educator* vol. 10 Number 2, pp.42-48.

Scher, D. (2002). *Students' Conceptions of Geometry in a Dynamic Geometry Software Environment*, Doctoral Dissertation, New York University.