

컴퓨터 교육 환경 변화에 따른 CAI 프로그램의 재사용 방안

○
백동근 · 전철부 · 박만곤
부경대학교 교육대학원 전산교육전공

A Study of CAI Program Reuse with Computer Education Environment Change

○
Dong-Geun Baik · Chul-Bu Jeon · Man-Gon Park
Dept. of computer science education, Pu-Kyong National University

- 요약 -

컴퓨터를 이용한 교육이 우리 나라에 도입된 것은 1989년 한국교육개발원에서 14편(초등학교의 경우)의 CAI 프로그램을 개발·보급하면서부터이다. 그 후 컴퓨터 기술의 급격한 발달로 컴퓨터 교육 환경에 많은 변화가 이루어졌다. 모노모니터에 도스를 사용하던 XT급 PC에서 이제는 윈도95를 운영체제로 한 멀티미디어 PC가 활용되고 있다. 그러나 초기에 개발·보급된 CAI 프로그램들은 그 당시의 컴퓨터 사양에 맞추어져 있어서 현재의 멀티미디어 컴퓨터에서는 동작하지 않는 것들이 대부분이다. 많은 비용과 노력을 투입하여 개발된 우수한 CAI 프로그램들이 컴퓨터 교육 환경의 변화로 인하여 폐기되거나 사장될 형편에 놓여 있다. 이러한 점에 착안하여 본 연구에서는 모노용으로 개발된 CAI 프로그램을 현재 환경에서 사용할 수 있도록 하는 방안을 찾아보았다.

본 연구를 통하여 Basic, C 언어로 개발된 CAI 프로그램은 소스코드의 수정으로 소프트웨어 재사용의 가능성을 확인하였으며, 저작도구 한울 2.0으로 개발된 CAI 프로그램은 별도의 코드 수정 없이도 윈도95의 도스 창에서 실행이 가능한 것으로 확인되었다. 이는 소프트웨어의 재사용을 통한 생산성 향상이라는 측면과 내용이 우수한 CAI 프로그램을 컴퓨터 교육 환경의 변화에 대응하여 계속 활용할 수 있게 하였다는 점에서 본 연구의 의의가 있다.

1. 서론

컴퓨터 하드웨어 기술의 발달과 함께 도스기반에서 윈도95기반으로의 운영체제 변화로 인해 도스용 모노 모니터에서 실행되던 CAI 프로그램의 거의 모두가 폐기되거나 사장되고 있다. 초기에 개발된 CAI 프로그램들은 나름대로의 특징을 가지고 있어서 매우 소중하게 사용되었던 프로그램으로서 교수-학습

기능이 우수한 것들이 많으며 학생들의 학습에 반드시 필요하다고 할 정도의 프로그램들이었다.

그러나 학교에 보급된 컴퓨터의 성능이 향상되면서 이전에 매우 유용하게 사용되었던 프로그램을 쓸 수 없게 되었다. VGA 모니터를 사용한 윈도95의 도스 모드에서도 실행되지 않으니 폐기할 수밖에 없는 프로그램이 된 것이다.

많은 인력과 비용의 투자로 개발된 프로그램의 코

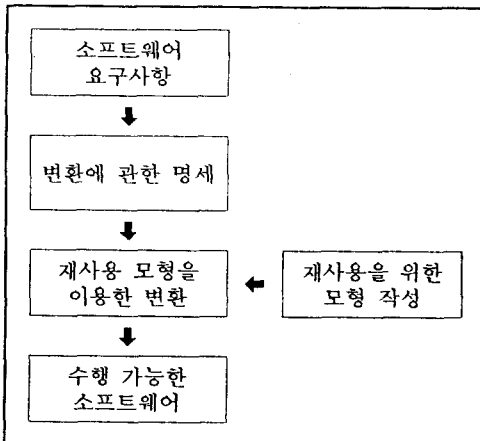
드 일부를 수정하여 재사용할 수 있게 한다면 이는 소프트웨어의 재사용에 따른 경제적 이익은 물론 교수-학습 기능이 뛰어난 CAI 프로그램을 이용한 학습이 가능해진다는 점에서도 매우 필요할 것으로 판단되어 윈도우95 환경에서 실행되지 않는 CAI 프로그램의 재사용 방안을 연구하고자 한다.

2. 소프트웨어 재사용을 위한 방법론

2-1. 소프트웨어의 재사용

소프트웨어 재사용이란 소프트웨어를 개발하거나 유지보수를 할 때 기존에 사용된 부품을 일부 수정하거나 완전히 다시 사용할 수 있는 개념을 의미한다. 이미 개발되어 그 기능이나 성능 및 품질을 인정받았던 소프트웨어의 전체 혹은 일부분을 다시 사용함으로써 소프트웨어의 개발 기간과 비용을 감소시키며 이미 개발된 모듈이나 프로그램 등을 동일한 업무 또는 서로 다른 업무에 재사용하거나, 프로그램의 일부를 수정하여 다시 사용 가능한 것으로 구성하기도 하는데 이렇게 소프트웨어를 재사용 함으로써 얻을 수 있는 장점은 소프트웨어 품질의 향상에 있다[1][2].

소프트웨어를 재사용하기 위해서는 기존의 소프트웨어에 수정을 가함으로써 소프트웨어 기능의 향상이나 새로운 시스템에 적용할 수 있도록 하는 작업이 필요한데, 이와 관련하여 소프트웨어 재사용을 위한 모형화 방안은 <그림1>과 같다[3][4].



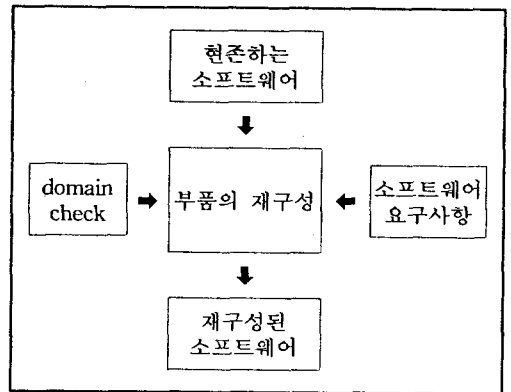
<그림1> 소프트웨어 재사용을 위한 모형화 방안

2-2. 소프트웨어 재구성의 방법

소프트웨어의 재구성 프로세스는 소프트웨어 개발의 전통적인 라이프사이클에 대해 소프트웨어 재공학 가능성을 부각시키는 것으로, 코드, 사양서, 설계 및 알고리즘의 라이브러리를 하는 것이다.

소프트웨어 재사용을 위한 부품의 수정은 첫째로 수정이 필요한 부품의 라이브러리를 작성하고, 둘째로 시스템의 요구사항을 기초로 하여 수정이 가능한 소프트웨어 부품의 새 코드를 설계하고, 셋째로는 수정되는 부품들이 적용되는 분야나 영역의 분석이 필요하다[5].

이에 관련된 소프트웨어 재구성에 관한 프로세스는 다음 <그림2>와 같다.



<그림2> 소프트웨어 재구성 프로세스

소프트웨어를 재구성하는 방법은 코드의 직접 재구성과 코딩 형태의 재구성이 있다. 코드의 직접 재구성 방법은 다른 수준의 정보에 의존함이 없이 코드를 직접 수정하는 코드 지향적 재구성 기법이고, 코딩 형태의 재구성 방법은 통제 구조나 데이터 구조를 변경하지 않고 원시코드를 더 쉽게 이해하기 위해서 이를 수정하는 방법이다[6].

본 연구에서 실제로 사용한 방법은 전체 전략 (whole approach)이라고도 하는 기존의 소프트웨어 수정 방안으로서 소스 코드를 직접 수정하여 실행 가능한 코드로 변환하는 것이다.

이러한 절차를 통해 컴퓨터 환경 변화에 따라 실행이 불가능하였던 코드는 새로운 환경에 맞게 수정되고 기존의 소프트웨어는 재사용이 가능하게 된다.

3. 개발·보급된 CAI 프로그램 분석

한국교육개발원에서는 1989년부터 1997년까지 550여 편의 CAI 프로그램을 공모전을 통하여 자체 개발하여 보급하였는데, 이 중 대부분의 CAI 프로그램이 도스용으로 개발된 것이며 그 중에서 4분의 1 정도가 모노모니터에서 실행되도록 개발된 것이다. 한국교육개발원에서 보급한 CAI 프로그램을 개발 언어, 사용 환경 등을 분석한 결과는 다음과 같다.

3-1. 개발·보급된 CAI 프로그램[7]

| 개발 년도 | 개발 언어 | 사용 환경 | | |
|----------|--------|-------|----|---------|
| | | 모노 | 컬러 | 모노 & 컬러 |
| '89 | Ebasic | 14 | . | . |
| '90 | Ebasic | 16 | . | .. |
| | C | 1 | . | 7 |
| | 기타 | . | . | 1 |
| '91 | Ebasic | 14 | . | . |
| | C | 2 | . | 15 |
| | 기타 | . | . | 4 |
| '92 | Ebasic | 1 | . | . |
| | C | 6 | . | 13 |
| | 한울 | 9 | . | . |
| | GREAT | . | . | 5 |
| | 기타 | 2 | . | 1 |
| '93 | Ebasic | 7 | . | 5 |
| | C | 1 | . | 13 |
| | 한울 | 6 | . | 14 |
| | GREAT | . | . | 22 |
| | 기타 | 1 | . | . |
| '94 | Ebasic | 2 | . | 2 |
| | C | 5 | 4 | 10 |
| | 한울 | . | 1 | 26 |
| | GREAT | 1 | . | 57 |
| | 기타 | 1 | . | 2 |
| '95 | 전체 | 24 | . | 95 |
| '96 | 전체 | 6 | . | 142 |
| 계 | | 119 | 5 | 434 |

3-2. 윈도우에서 실행되지 않는 CAI

286급 이하의 교육용 컴퓨터에서는 그래픽 처리를 위하여 40H 화면모드(Herc: 모노, 640×400, 화면 중형비 1.42:1)를 386 또는 486급의 교육용 컴퓨터에서는 70H 화면모드(CGA, MCGA) 및 60H 화면모드(VGA: 640×400, VRAM=A000, 컬러=16/256k)를 사

용하도록 되어 있었다[8]. 그러나 윈도우95 시스템은 40H, 60H, 70H 화면모드를 지원하지 않으므로 이전에 모노용으로 개발된 CAI 프로그램을 사용할 수 없게 되거나 또는 VGA 모니터에서 실행은 가능하지만 화면 해상도의 문제로 인하여 그래픽이 왜곡되는 현상이 발생하기도 하여 소프트웨어의 소스 코드를 수정하지 않으면 수업에 활용할 수 없게 된 CAI 프로그램이 많았다.

본 연구에서는 Ebasic, C, 한울로 개발된 CAI 프로그램을 중심으로 재사용을 위한 연구를 수행하고자 하며, 이에 따라 언어별 소스 코드 수정 가능성이 확인하기 위하여 분석한 결과는 다음과 같다.

| 개발 언어 | 오류 발생 요인 및 해결 방안 | |
|--------|------------------|--|
| Ebasic | 오류 발생 요인 | 화면모드에 따른 실행 불가 화면 해상도에 따른 그래픽의 왜곡 색상의 왜곡 |
| | 해결 방안 | 화면모드 변경 그래픽 관련 명령어 수정 |
| C | 오류 발생 요인 | 화면 해상도에 따른 그래픽의 왜곡 색상의 왜곡 |
| | 해결 방안 | 그래픽 관련 명령어 수정 |
| 한울 | 오류 발생 요인 | 코드별 옵션에 따른 색상 왜곡 |
| | 해결 방안 | 색상 관련 옵션 변경 |

4. 코드의 수정이 필요한 명령어, 함수

이처럼 컴퓨터 환경의 변화에 따라 사용할 수 없게 된 CAI 프로그램의 소스 코드를 수정함으로써 새로운 컴퓨터 환경에서 사용할 수 있게 할 필요가 있음에 따라 각각의 언어에서 소스 코드의 수정이 필요한 명령어를 분석한 결과는 다음과 같다.

4-1. Basic으로 개발된 CAI

| 수정이 필요한 Basic 명령어 |
|---|
| locate, screen, circle, draw, line, paint, preset, pset, get, put 등 |

4-2. C 언어로 개발된 CAI

| |
|--|
| 수정이 필요한 Turbo C 표준 라이브러리 함수 |
| arc, bar, bar3d, circle, drawpoly, ellipse, fillellipse, fillpoly, floodfill, gotoxy, line, lineto, outtextxy, pieslice, putimage, putpixel, puttext, rectangle, sector, setaspectratio, settextstyle, setviewport, textmode, window [9] |

4-3. 저작도구 한글로 개발된 CAI

| |
|--|
| 수정이 필요한 한글 명령어 |
| 한글 2.0으로 개발된 CAI 프로그램은 코드의 수정없이 윈도95의 도스창에서 실행할 수 있으며, 다만 화면 출력에서 색상을 추가하고자 할 경우에만 관련 명령어를 수정할 필요가 있다. |

5. 재사용을 위한 소스 코드의 수정

Ebasic으로 개발된 CAI 프로그램 'BASIC 학습마당'을 다음과 같은 절차에 따라 소스 코드를 수정하여 윈도95 시스템에서 실행되도록 하였다.

5-1. 소프트웨어 요구사항

교육용 컴퓨터 사양의 변화에 따라 사용할 수 없게 된 CAI 프로그램의 재사용을 위하여 필요한 소프트웨어의 요구사항은 다음과 같다.

- (1) 기존의 사용자 인터페이스를 유지해야 한다.
- (2) 운영체제 윈도95 시스템에서 수행되어야 한다.

5-2. 프로그램 분석 및 대책

CAI 프로그램 'BASIC 학습마당'의 프로그램을 분석한 결과는 다음과 같다.

- (1) 소스코드 라인 수 : 2,923 LOC
- (2) 수정이 필요한 명령어 : screen, circle, locate, line, paint, put
- (3) 수정이 필요한 라인 수 : 1,804 LOC
- (4) 수정이 필요한 이미지 데이터 : 기능 키 모양

(5) Ebasic으로 개발된 BASIC 프로그램을 그대로 VGA용으로 수정하기보다는 한글 Quick Basic으로 변환하여 수정하면 보다 많은 호환성을 가질 수 있을 것으로 보고 수정된 코드는 한글 Quick Basic으로 컴파일 하였다.

5-3. 그래픽 모드의 변경

모노용과 VGA용 CAI 프로그램에서 시각적으로 확인되는 차이의 가장 큰 요소는 스크린 모드에 따른 화면 출력 가능 여부와 해상도에 따른 그래픽의 왜곡, 그리고 색상으로 구별될 수 있다. 따라서 프로그램 재구성의 첫 단계로 화면모드를 변경하였다.

(1) Ebasic에서의 화면모드 및 해상도

Ebasic에서 표현 가능한 화면모드는 다음과 같다.

| 화면모드 | 해상도 | 색상 수 |
|----------|-----------|-----------------------|
| SCREEN 0 | 640 × 400 | text 모드 |
| SCREEN 1 | 320 × 200 | graphic 16색, 한글 출력 불가 |
| SCREEN 2 | 640 × 400 | graphic 2색, 한글 출력 불가 |
| SCREEN 3 | 640 × 400 | graphic 2색, 한글 출력 가능 |

(2) 한글 Quick Basic에서의 화면모드 및 해상도

한글 Quick Basic에서 표현 가능한 해상도 및 색상 수는 다음과 같다[10].

| 화면모드 | 해상도 | 색상 수 |
|-----------|-----------|---------------------------------------|
| SCREEN 11 | 640 × 480 | graphic 2색, 문자 색 = 청색 text 30 line 출력 |
| SCREEN 12 | 640 × 480 | graphic 256색 text 30 line 출력 |
| SCREEN 71 | 640 × 400 | graphic 2색 |

(3) Ebasic의 모노용에서 화면모드로 사용된 명령어 'SCREEN 3'을 한글 Quick Basic에서 그래픽 모드 'SCREEN 12'로 변경하였다.

한글 Quick Basic에서 'SCREEN 71'을 사용할 수는 있으나 한글DOS에서만 실행이 가능하므로 이를 선택하지 아니하였다.

5-4. 화면 출력 관련 명령의 좌표 수정

화면모드의 변경으로 프로그램의 내용들이 화면에 출력된다. 그러나 이전의 모노모드에서 640×400의 해상도에 맞추어 코딩된 그래픽이 현재의 VGA 640×480모드에서는 화면의 위에서부터 세로로 400픽셀 만큼만 화면의 내용이 출력되어 아래쪽에 세로로 80픽셀만큼의 빈 공간이 생기고, 화면에 출력되는 그래픽이 모니터 상에서 왜곡된다. 이를 해결하기 위하여 화면 출력 좌표와 관련된 명령어들의 세로 좌표를 변경해야 한다. 따라서 다음과 같은 방법으로 이를 해결하였다.

(1) 해상도의 차 480:400의 비율에 따라 각 그래픽 관련 명령어의 세로 좌표에 1.2를 곱함으로써 해결하였다. 이 때 숫자 상수를 직접 곱하지 않고 변수 Vertical(=1.2)을 사용하여 그래픽 관련 명령어가 수정되었는지를 구별할 수 있게 하였다.

(2) Basic에서 locate명령으로 좌표를 정하게 되는 print문의 경우 세로 좌표에 Vertical 변수를 곱한 결과 값이 소수가 될 경우 출력되는 좌표는 정수로 처리하므로 약간의 위치가 달라지게 된다. 이러한 문제의 해결은 화면을 직접 확인하면서 그래픽 또는 print문의 좌표를 조정하였다.

5-5. 색상 수정

모노용으로 개발된 CAI는 최초의 스토리보드 설계에서부터 색상을 고려하지 않았으며, 프로그램의 내용도 다양한 색상을 요구하지 않는 내용으로 구성되어 있기 때문에 소스 코드를 수정하면서까지 색상을 표현할 필요는 없다고 본다. 그것은 색상을 표현하는 과정에 투입되는 인력과 비용, 컬러로 표현됨으로써 향상될 수 있는 교수-학습 기능과 비교할 때 생산적이지 못하다고 판단되기 때문이다.

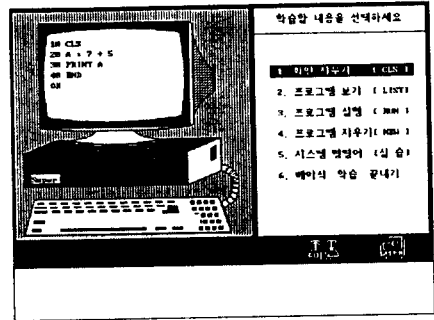
다만 CAI 프로그램 설계 시 색상을 고려하여 개발된 소프트웨어는 화면모드, 그래픽 좌표 수정 후 색상과 관련된 부분을 점검할 필요가 있다고 본다.

5-6. 수정된 프로그램의 화면

이상과 같은 방법으로 재구성·수정한 CAI 프로그램 'BASIC 학습마당'의 화면은 다음과 같다.

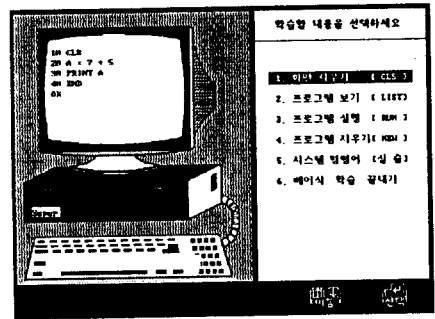
(1) Screen 명령만을 수정한 상태에서 VGA 모니터에서 실행한 결과 화면은 다음 <그림3>과 같으며

화면 아래의 80픽셀이 빈칸으로 나타나고 있음을 발견할 수 있다.



<그림3> 화면모드 변경 후의 CAI 화면

(2) VGA 모니터에서 그래픽이 정상 출력되도록 화면모드를 변경한 다음, 그래픽 좌표와 관련된 명령어를 수정하여 학습 내용이 전체 화면에 출력되도록 조정한 후의 화면은 다음 <그림4>와 같다.



<그림4> 세로 좌표를 조정한 후의 화면

6. 결론 및 향후 연구 과제

6-1. 결론

CAI 프로그램의 재사용을 위하여 본 연구에서 수행한 소스 코드의 수정을 통하여 도스, 모노 모니터 환경에서만 실행되었던 CAI 프로그램이 윈도우의 도스창에서도 실행되게 함으로써 도스용으로 개발되었던 100여 편의 CAI 프로그램의 재사용 가능성을 확인하였다.

또한 많은 비용으로 개발되었던 과거의 CAI 프로그램을 적은 비용으로 수정하여 현재의 윈도우 환경

에서 실행 가능하게 함으로써, 폐기할 수밖에 없었던 CAI 프로그램을 다시 사용할 수 있게 하였다.

이는 초기에 개발되었던 우수한 CAI 프로그램을 학생들의 교육 활동에 다시 투입할 수 있게 되었다는 점에서 본 연구의 의의가 있다.

6-2. 향후 연구과제

소스 코드의 수정을 일일이 수 작업으로 한다면 많은 인력과 시간의 투입이 필요하므로 그만큼 CAI 프로그램 재사용에 따른 생산성은 낮아질 수밖에 없다. 따라서 CAI 프로그램 재사용 생산성을 높이기 위하여 개발 언어에 따라 소스 코드를 수정할 수 있는 자동화 도구의 개발이 필요하다고 판단된다.

< 참고 문헌 >

- [1] Arnold S.P., S.I. Stepoway, "The Reuse System : Cataloging and Retrieval of Reusable Software", Proceeding of COMPCONS'87, 1987.
- [2] Biggerstaff T., C.Richter, "Reusability Framework, Assessment, and Direction", IEEE Software, Mar. 1987.
- [3] Wasserman A.I., 'Tutorial : Software Development Environments', Washington DC : IEEE Computer Society, 1981.
- [4] 이주현, '실용소프트웨어 생산공학론(중)', 법영사, 1994.
- [5] Zelkowitz M.V., R.T.Yeh, R.G.Hamlet, J.P.Gannon and V.R.Basili, "Software engineering Practices in the U.S. and Japan", Computer, vol.17, no.6, 1984.
- [6] McClure C.L., 'Managing Software Development and Maintenance', Van Nostrand Reinhold, 1981.
- [7] 한국교육개발원, '교육용 소프트웨어 편람(초등학교용)', 서보인쇄주식회사, 1994.
- [8] 한국교육개발원, '94 교육용 소프트웨어 연구 개발', 덕수종합상사, 1994.
- [9] 황희용, 'Turbo C 2.0 라이브러리매뉴얼', 교학사, 1994.
- [10] 이희춘, A·C·T, '처음사용자를 위한 Quick Basic 4.0', 도서출판 세운, 1989.