

컴퓨터 보조수업을 위한
 저작 시스템설계에 관한 연구

고 대 곤 (대구교육대학)
 학박 상 희 (연세 대학교)

A STUDY ON DESIGN OF AUTHORING SYSTEM
 IN COMPUTER ASSISTED INSTRUCTION

* Kho Dee Ghon (Tae Gu Teachers College)
 ** Park Sang Hee(Yon Sei University)

ABSTRACT

In this paper a Korean authoring system is designed to write a CAI courseware in Hangul/English by an author who is a nonprogrammer.

It saves much time in authoring a courseware and maintains high level transplintanty among CAI systems. By interfacing an expert graphic utility, image information can be processed more easily and efficiently.

Programming control of the flow of CAI courseware can be ramification and individual learning possible, fitting various demands of learners and learning ability.

제 1 장 서론

학습자의 능력차를 고려하지 않은 교육의 양적 팽창과 이에 부응되는 질적 향상의 미비로 심화되어가는 학습길손을 보충하기 위하여 컴퓨터가 가지고 있는 다양한 교육공학적 잠재력을 이용하면, 개별학생의 인지적 구조나 정보처리 양식에 적합한 학습내용과 수업자극을 선별 제공할 수 있으며 상호작용이 손쉽게 이루어져 개별화 수업설계가 가능하다.

일반적으로 컴퓨터 시스템의 기능은 프로그래밍 언어로서 표현이 가능하지만 교육용 프로그램은 학습자의 주의를 모으고 학습동기를 유발함

학습효과를 위하여 특수한 기술과 난이도 높은 얼굴이즘의 구사가 요구되므로 이를 위한 저작 전문가 시스템이 필요하다

본 논문은 컴퓨터 보조수업 (CAI) 운영시 경제적 부담을 줄이고 기종간의 이식성을 높이며, 비 프로그래밍 방식으로 한글과 영문의 코스웨어를 저작하는 시스템의 설계에 대한 연구이다.

제 2 장 저작 시스템의 설계

본 연구에서는 저작 시스템을 학습관련 정보의 생성, 관리와 이의 효율적인 운영을 가능하게 하는 계를 의미하는 것으로 정의한다.

그러므로 본 연구의 범위는 문자, 영상(graphic), 음성정보의 생성 (논문참조 : 오디오 시스템의 퍼스날 컴퓨터에 의한 제어) 및 관리와 이에 대한 프로그래밍 제어를 가할 수 있는 제어언어의 개발등이다.

본 연구에서는 화면정보의 편집을 문자정보의 편집과 영상정보의 편집을 통칭하는 의미로 사용한다.

그림 2-1은 본 연구에서 제안한 저작 시스템의 구성에 대한 블록 선도이다.

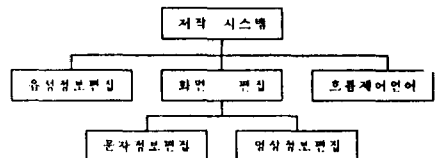


그림 2-1 저작 시스템의 블록선도
 Fig. 2-1 Block diagram of authoring system

2.2 화면정보의 편집

화면 편집기는 CAI 코스웨어를 저작하는 데 필요한 각종의 한글, 영문자 숫자 및 특수문자를 지원하여 텍스트 화일을 만들고 그리고 생성된 영상화일의 필요한 부분에 대한 정보를 가지는 역할도 한다.

CAI 코스웨어의 구성시 학습의 효과를 높이기 위하여 크기와 모양이 다양한 문자의 출력이 요구되어, 문자의 출력은 image oriented 방법에 의하였고, 영상정보는 영상을 2 자원의 비트 맵에 대응시켜 표현하는 방법을 취하였다.

2.2.1 화면 편집기의 한글 코드체계

화면 편집기(SDA)의 적절한 한글 코드 채택을 위하여 코드체계를 조사한 결과 기 사용 한글 코드 체계는 메이커 별로 약간씩 다른 코드 체계를 채택하고 있었으며, 같은 코드 체계내 에서도 작은 변형이 많이 있으나 대체로 7 bit, 8 bit 및 16 bit (2 byte) 체계등으로 구분 요약되었다.

7 bit 코드체계는 한글 시작코드(SO)와 끝 코드(SI)의 제어코드 문제가 있으며, 8 bit 코드체계는 한글 사용자 가변길이의 문제가 있어 영문 소프트웨어와의 호환을 기대하기가 어렵다.

완성형 16 bit 코드체계는 사용빈도수가 높은 음절만 골라 한글 한 음절마다 코드를 부여한 방식이므로 표현가능한 문자가 한정되어 있다는 단점 때문에 CAI 코스웨어의 다양성을 추구하기에 곤란하였다.

조합형 16 bit 코드체계는 초성, 중성 및 종성을 5 bit 씩 나누어 2 byte 에 할당하였으므로 이 코드체계에서는 조합 가능한 모든 한글의 출력이 가능하여 본 연구의 목적을 수행하기에 가장 적합한 체계로 간주되어 SDA 의 한글 구성은 이에 의한다.

2.2.2 문자의 표현

영문자는 ASCII코드체계를 따르므로 그 코드에 대응하는 이미지를 비트 맵으로 구성하여 화면에 표시할 수 있어 비교적 간단한 응용이 가능하지만 바람직한 CAI 코스웨어를 구축하기 위해서는 크기와 모양이 다른 여러벌의 문자체가 필요하여, 로마제는 8 dot * 8 line, 16 * 8, 16 * 16, 24 * 16 등 4

종류의 폰트를, 그리고 이체릭체는 16 dot * 16 line 및 16 * 24 등의 2 종류의 폰트를 준비하였다.

한글의 표시는 한글 구조의 복수성에서 기인되는 상당히 복잡한 방법이 요구되는데, 앞에서 기술한 잇점등으로 조합형 2 byte 체계를 채용하였으며, 폰트크기는 16 dot * 24 line과 16 * 16의 두 종류이다.

동일한 초성이라도 그 다음에 연결되는 중성과 종성에 따라 다른 모양의 자형이 요구되어 6벌식, 을 체택하였다.

2.2.3 특수문자

본 연구에서는 수식등이 많이 쓰이는 교과외 코스웨어 저작시 사용 빈도수가 많은 특수문자를 선정, 3 종류 크기, 총 213 개를 정의하여 사용하였다.

표 2-2, 2-3 및 2-4는 이들 각 경우의 특수문자의 예로서, 십진수 3 자리는 해당 특수문자의 코드값이다.

표 2-1 작은 특수문자 코드 (8 * 8)

Table 2-1 Special character code (8 * 8)

048 √	050 □	053 ↔	054 ↔
----------	----------	----------	----------

표 2-2 중간 특수문자 코드 (16 * 16)

Table 2-2 Special character code (16 * 16)

067 h	068 π (±)	069 r	070 y
082 γ	083 δ	084 φ	085 π (≠)
090 k	091 g	092 t	093 l

표 2-3 큰 특수문자 코드

Table 2-3 Special character code (16 * 24)

025 ∩	026 =	032 X	033 ÷
038 ≤	039 ≥	040 ≡	041 ≧
050 □	051 △	052 ∠	053 □

영상의 표시는 문자와 마찬가지로 그 영상의 바이트 맵을 구성하여 비디오 램에 옮겨 줌으로서 가능한데, 그러나 영상은 문자와는 달리 정정화면 패턴이 없고 임의의 배열로 구성되므로 필요한 영상은 그 때 마다 영상 스캐너를 통해 입력, 구성하여야 한다.

영상 화일은 전문 그래픽 패키지 (Dr Halo)를 저작 도구에 인터페이스시켜 생성하였으며, 영상을 축약하지 않고 그대로 저장, 재생하는 방법을 취하였다.

2.3 저작 시스템 제어 언어

CAI의 코스웨어는 문자, 영상, 그리고 음성정보로 구성된 다수 화일들의 집합이다.

이 때, 예상되는 학습자의 다양한 반응에 능동적으로 대처하기 위하여 코스웨어의 흐름을 제어할 프로그래밍 언어가 필요한데, 본 연구에서는 PASCAL 언어와 비슷한 구문을 갖는 간이언어를 개발하여 본 목적에 사용하였다.

2.3.1 문자, 예약어, 프로시저

저작 시스템 컴파일러가 인식할 수 있는 문자는 A에서 Z까지의 대문자 26 글자와 그리고 a에서 z까지의 소문자 26 글자 및 underline (-) 과 0에서 9까지의 숫자와 blank 이외의 8종류 특수문자이다.

화일의 생성, 관리등을 위하여 적절한 기능을 일으키도록 미리 그 의미를 정의한 예약어는 "BEGIN"외에 24 종으로 하였다.

다음은 get 명령의 프로시저에 대한 설명이다.

syntax는 명령어의 정확한 표기를 나타내며, **purpose**는 명령어의 사용목적, 그리고 **remarks**는 **purpose**의 보충설명으로 명령어의 사용상 주의점과 인수를 서술하며, **example**은 본 명령어의 사용 예를 보인 것이다.

i와 j는 정 정수(integer constant)를 의미하고 x\$ 및 y\$은 문자열(string)을 의미한다.

instruction : GET

syntax get (i1, j1, i2, j2, i)

purpose 영상화일에서 애니메이션 위치지점을

위한 버퍼와 각종 변수값을 설정한다.

remarks 지정된 영상화일에서 좌표점(i1, j1)과 점(i2, j2)를 대각선으로 하는 사각형을 애니메이션 범위로 설정, 이를 지정된 버퍼 i에 할당한다.

example see FREE

2.3.2 저작 시스템 제어언어의 구분

다음은 본 제어언어의 일부 **statement**에 대한 구문을 Backus-Naur Form 으로 나타낸 것이다.

```

<statement> ::= <process statement>
                | <structured statement>
                | <conditional statement>

<process statement> ::= <process declaration>
<structured statement> ::= <compound statement>
                            | <repetition statement>
<compound statement> ::= begin <statement> end
<repetition statement> ::= <repeat statement>
<repeat statement> ::= repeat <statement>
                        until <expression>
<conditional statement> ::= <select statement>
                            | <case statement>
    
```

제 3 장 결과 및 고찰

그림 3-1은 화면 편집기에서 표현 가능한 각 모드의 문자를 나타낸 화면이다.

제 1 줄은 영문 로마제의 4 가지 모드이고, 제 2 줄은 영문 이터릭제의 2 가지 모드이다. 그리고 제 3 줄은 한글 2 가지 모드 이고, 제 4 줄은 특수문자 3 가지 모드에 해당된다. 본 시스템의 성능을 평가하기 위하여 현행 중학교 2 학년 영어 한 단원 (제9과 : We are doing fine in Korea)에 대한 5 코스, 36모듈의 코스웨어를 작성하여 본 시스템의 성능을 확인하였다. 학습내용에 따라 적절한 모양과 크기의 문자모드로 진입, 비 프로그램 방식으로 교수 설계자에 의한 학습화면 설계가 가능하였다.

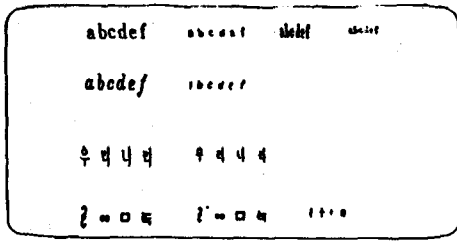


그림 3-1 여러 모드의 글자
Fig. 3-1 Characters in modes

그림 3-2는 본지학습 코스웨어의 흐름도 예이다. NO는 타이틀 화면으로 본 코스웨어의 제목을 나타내고, N은 메인 메뉴화면에 해당된다. 이 커리큘럼은 N1, N2 및 N3의 3 개의 학습 코스와 다시 N11 과 N12 및 N21, N31 및 N32의 부 학습코스로 이루어졌다. 그림의 진행 표시는 수평 및 수직 코스간의 학습의 진행을 나타낸다.

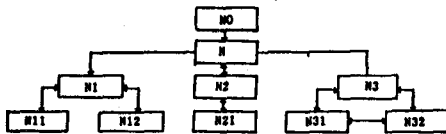


그림 3-2 네트워크방식의 코스웨어 흐름도
Fig. 3-2 Flow chart of network type courseware

부록 1의 프로그램은 그림 3-2의 네트워크 방식 코스웨어에 대한 제어 프로그램이다.

본 저작 시스템 제어언어의 프로그램은 동일 학습계열 내에서의 상 방향성 수직적, 수평적 코스웨어의 흐름을 일으키게 할 수 있을 뿐만 아니라 다른 학습 계열간의 이동이 자유로와 입체적인 제어가 가능하고, 수정이 손쉬워 학습자들의 당해 부분의 수정에 의하는 것이므로 학습자들의 인지적 특성을 고려한 다양한 개별학습이 이루어 졌다.

CAI의 코스웨어의 저작시 본 저작 시스템의 화면 편집기를 이용한 학습화면의 구성은 비 프로그램 방식에 의한 텍스트 화일의 작성이므로 프로그래밍 기술의 숙지가 필요없었으며 저작시간은 키이 엔트리의 속도에 따라 상당히 단축되었다.

특히 그래픽 패키지에 의한 영상화일의 생성은 정교하고 손쉬울 뿐만 아니라 단 시간내에 처리가

가능하여 타 언어의 사용시에 비하여 화일의 생성 및 관리시 여러가지의 장점이 있었고, 영상화일을 처리하기 위한 시스템의 하드웨어적 부담을 줄일 수 있었다.

그러나 본 저작 시스템에서는 코스웨어의 학습의 진행을 전적으로 학습자 조절 방식을 택함으로써 학습시간의 낭비의 소지가 우려되고, 시뮬레이션등 정교한 프로그램개발을 위한 유틸리티가 부족하며, CAI 코스웨어 설계시 설계 가이드를 제공해주는 매뉴얼링 방식의 병행과 더 많은 기능을 위한 명령어와 특수문자의 추가가 있어야 할 것으로 사료된다.

제 5 장 결 론

얻어진 결과는 다음과 같다.

1. 화면 편집기로 한글/영문 (숫자 및 특수문자 포함) CAI 코스웨어를 저작하였으며, 각 문자의 다양한 폰트는 프로그램 학습의 단조로움을 감소시켰다.
2. 비 프로그램 방식으로 CAI의 코스웨어를 저작하였으며, 저작시 프로그램 방식에 비하여 소요시간이 단축 되었다.
3. 이 방식은 프로그램 작성기술의 습득이 요구되지 않기 때문에 교수 설계자에 의한 CAI 코스웨어의 직접 저작이 가능하여 양질의 코스웨어의 확보에 기여할 수 있다.
4. 그래픽 패키지를 저작 시스템과 인터페이스 시킴으로써 영상정보의 처리를 위한 하드웨어 부담과 개발비용을 줄이고, 적은 시간내에 보다 정교한 영상 화일의 생성 및 갱신이 이루어 졌으며, 운용의 편의성이 있었다.
5. 저작 시스템 제어언어 프로그램은 네트워크식 CAI 스웨어의 흐름을 입체적으로 제어하였으며, 코스웨어의 내용 수정 및 흐름의 변경이 자유로와 다양한 학습자의 요구와 학습 능력에 따른 본지 및 개별학습이 가능하였다.
6. 본 저작 시스템은 system independant 타입으로 일정수준 이상의 하드웨어적 요건을 만족시키면 동일 레벨의 기종간에 충분한

이식성이 있었다.

6. 본 저작 시스템은 16 bit 퍼스날 컴퓨터를 메인 시스템으로 하는 저작 시스템이므로 마이크로 레벨 이상의 타 저작 시스템에 비하여 시스템의 구성 및 유지시 경제적, 인적 자원을 절감할 수 있었다.

본 논문의 결과는 기능과 환경이 개선되고 있는 퍼스날 컴퓨터의 DOS 하에서 기동되는 한국 저작 시스템을 설계, 비 프로그램 방식으로 코스웨어의 저작과 운용을 가능하게 하므로써, 정보화 사회를 대비한 개별학습에 이바지할 것이다.

참고문헌

1. Bunderson C. V., The computer and instruction -al design, In computer-assisted instruction, testing and Guidance, New York : Harper & Row, 1970, 7
2. Hall K. A., A research model for applying technology to interactive instructional process, J., computer-based instruction, vol.3, No.3, pp.68-75, 1977, 7
3. Okey j., & Majer K., individualized and small group learning with computer assisted instruction, An Communication Review, vol.24, No.1, p.19, 1976
4. 고대근, 컴퓨터 보조수업에서의 한글 저작도구에 관한 연구, 전국 교육대학 전산 교육연구, 제 1집, PP.25-44
5. 이육화, 개별학습과 CAI개발기법 : 국내 CBE R & D 및 실용화에 관한 세미나, KAIST 시스템 공학 센터, P.8, 1987, 7
- 6.정보교환용 부호, KSC 5601, 한국공업규격,1982,10 7.정보교환용 부호 확장법, KSC 5620, 한국공업규격, 1982, 6
8. 황종선, ICAI 모델 및 Authoring Tool에 관한 연구, 한국정보과학회 봄학술발표논문집, Vol.15, No.1, pp.81-84, 1987,5

부록 1. 마중 코스웨어 제어프로그램

```

PROCESS N32                                END PROCESS.
BEGIN                                       ;
      TEXT( 'N32' )                          PROCESS N21
SELECT                                      BEGIN
      CASE 1 : GOTO N32                       TEXT( 'N21' )
END SELECT                                  RETURN
END PROCESS                                END PROCESS
;                                           ;
PROCESS N31                                PROCESS N2
BEGIN                                       BEGIN
      TEXT( 'N31' )                          TEXT( 'N2 ' )
SELECT                                      SELECT
      CASE 1 : GOTO N31                       CASE 1 : RETURN
      CASE 2 : GOTO N32                       CASE 2 : GOSUB N21
END SELECT                                  END SELECT
END PROCESS                                END PROCESS
;                                           ;
PROCESS N3                                  ;
BEGIN                                       PROCESS N12
      TEXT( 'N3 ' )                          BEGIN
SELECT                                      TEXT( 'N12' )
      CASE 1 : GOTO N31                       RETURN
      CASE 2 : GOTO N32                       END PROCESS
END SELECT                                  ;
PROCESS N11                                END SELECT
BEGIN                                       END PROCESS
      TEXT( 'N11' )                          ;
      RETURN                                  PROCESS N0
END PROCESS                                BEGIN
;                                           TEXT( 'N0 ' )
PROCESS N1                                  SELECT
BEGIN                                       CASE 1 : RETURN
      TEXT( 'N1 ' )                          END SELECT
SELECT                                      END PROCESS
      CASE 1 : GOSUB N11                       ;
      CASE 2 : GOSUB N12                       ;MAIN ROUTINE
END SELECT                                  ;
END PROCESS                                BEGIN
;                                           CALL NO
PROCESS N                                    JUMP N
BEGIN                                       END.
      TEXT( 'N ' )
SELECT
      CASE 1 : GOTO N1
      CASE 2 : GOSUB N2
      CASE 3 : GOTO N3
    
```