論 文
39~8~11

CAI를 위한 저작 시스템의 설계

A Design of Authoring System for CAI

高 大 坤*·朴 相 曦**
(Dae-Ghon Kho · Sang-Hee Park)

요 약

CAI 코스웨어의 대부분은 교수 설계자가 아닌 프로그래머에 의하여 이루어져 왔으며, 그리고 저작 시스템은 각기 고유의 저작도구를 사용하므로 시스템간의 이식성이 없어 양질의 코스웨어의 확보가 어렵다. 본 논문에서는 이러한 제한성을 극복할 목적으로 CAI를 위한 소규모의 저작 시스템을 설계, 구현하였다. 본 시스템의 특성은 실험을 통하여 확인하였으며, 다른 시스템에 비하여 기대되는 결과를 얻을 수 있었다.

Abstract- It is hard to keep good quality of coursewares for Computer-Assisted Instruction (CAI), since most CAI coursewares have been developed by programmers rather than by instruction designers. Also, there is no trasplantations among authoring system. This paper describes the design and implementation of an authoring system for CAI to overcome the problems mentioned above. The performance of the system is assured experimentally and then good results are shown compared with other systems.

1. 서 론

1920년대에 Presley는 교수기계(teaching machine)의 가능성에 대한 이론을 제시하였는데, 이것은 1950년대에 Skinner에 의하여 operant condition이라는 행동과학적 교육 이론으로 정립되었고, Skinner의 이 이론은 후일 CAI개발의 이론적 근거가 된다.

산업 사회에서는 능력차가 심한 다인수 학급에서 1인 교사에 의한 획일적 수업이 이루어 지므로 개인별로 교정, 심화 및 보충하기가 어려우나 [20], 다가오는 정보화 사회에서는 개개인의 개성과 창의력을 중시하는 사회가 될 것인 바, 교육은 각 개인에게 최고의 선을 실현할 수 있도록 하는 학습형태가 바람직하다[18] [22].

*正 會 員 : 大邱敎育大 科學科 助敎授 · 工博
**正 會 員 : 延世大 工大 電氣工學科 敎授 · 工博
接 受 日 字 : 1989年 2月 28日
1 次 修 正 : 1989年 5月 2日
2 次 修 正 : 1990年 1月 9日
3 次 修 正 : 1990年 7月 23日

개인차에 대한 수업설계의 적정화 방안은 교수 설계 및 교수 심리학 연구의 중심이 되어 왔지만, 이를 가능하게 하는 공학적 매체가 없었으나 오늘날에는 컴퓨터가 가지고 있는 다양한 교육 공학적 잠재력을 이용하면 개별학생의 인지적 구조나 정보처리 양식에 적합한 학습 내용과 수업자극을 선별 제공할 수 있고 상호작용이 손쉽게 이루어질 수 있어 개별화 수업설계가 가능하다[8] [12].

CAI코스웨어의 저작은 초창기에는 전문 프로그래머들에 의하여 기계어나 어쎔블리어등을 이용하여 작성되었는데, 이 경우 1시간분의 학습물을 준비하는데 약 500~1000시간[20]이 요구되어 개발에 따른 엄청난 경제적 부담과 학습의 과정에서 교수 개발자에 의하여 이루어져야 하는 많은 의사결정이 프로그래머에 의해 결정됨으로써 양질의 코스웨어 확보에 어려움이 따랐다.

컴퓨터의 발달 제2세대에는 BASIC등 범용 프로그래밍 언어가 개발되어 이들을 이용한 코스웨어의 개발[2]이 활발하였는데, 이 경우 기계어등을 사용할 때에 비하여 저작의 편의성은 다소 증가하였지만, 여전히 프로그래머 중심의 코스웨어의 작성, 과중한 저작시간의 소요와 경제적 부담, 제한된 저작 기능으로 인한 단조로운 학습에 대한 개선은 이루어 지지 못했다[16].

데이타 처리용 고급 프로그래밍 언어에 의한 코스웨어 저작의 불편을 해소하기 위하여 CAI 코스웨어를 전문적으로 개발하는 저작 시스템이 출현하였는데, 이러한 저작 시스템은 코스웨어의 저작시 프로그래밍 언어 사용에 비하여 사용의 편의성, 저작의 다양성등을 추구할 수 있을 뿐만 아니라 저작시간도 상당히 단축이 가능하여, 1시간분의 학습물을 위한 저작 소요시간은 약 300~400시간 [3, 10]정도로 단축되었다.

그러나 CAI에서 요구되는 각종의 다양한 학습효과를 구사하기 위해 사용되는 명령어가 400여개 이상이나 되어 프로그래머가 암기하여 손쉽게 사용하는 정도를 초과하였으며 여전히 프로그램 형식으로 저작이 이루어 지므로써 현장 수업전문가인 교사에 의한 코스웨어 저작은 곤란하다.

1972에 개발된 TICCIT(Time Shared Interative Computer Controlled Instructional Television)은 NOVA 미니레벨 시스템을 갖는 메뉴 운영방식의 저작 시스템이다[1, 15, 21].

이러한 메뉴 운영방식의 저작시스템은 정규 alphanumeric키 이외에 별도의 기능 키를 부착시켜 학습자로 하여금 이 키를 이용하여 스스로의 학습속도와 학습의 양식에 맞추어 수업계열을 조정해 나갈 수 있도록 함으로써, 수업자료 설계자는 개별학생에 대한 복잡한 수업계열을 정할 필요가 없다.

메뉴 운영방식 저작 시스템은 1시간분의 CAI 코스웨어 저작시 소요시간을 약 50~150 시간[9, 14]으로 단축시켰고, 코스웨어 설계를 위한 학습설계 가이드를 제공하는 기능을 구비함으로써 학습전략 선정시 시간과 노력이 절약되며, 그리고 CAI 인적 구성원간의 커뮤니케이션 과정에서 일어나는 이해부족등의 문제해소가 어느정도 가능하다 [7, 11, 13].

메뉴 운영방식은 사용의 용이성이라는 관점에서 보면 생산성에 크게 이바지 하는 등 많은 유익한 점은 있으나 이 방식은 정형화된 학습모델을 전제로 메뉴를 발생시키므로 코스웨어의 저작시 저자의 의도를 제한하게 될 것이며, 잘못 제공되는 메뉴는 저작시 장애를 발생시키고, 숙달자에게는 여러 단계를 거쳐서 원하는 메뉴 지점에 이르게 됨으로 오히려 번거로운 단점이 될 수가 있다.

메뉴 운영방식 저작 시스템은 호스트 컴퓨터의 레벨을 대형에서 중형정도로 낮추어 시스템의 경제적 부담을 감소시켰으나, 그러나 중형 시스템의 유지는 여전히 CAI의 저변확대에 지장을 준다[17].

현재 개발되어 쓰이고 있는 CAI 코스웨어 저작 시스템들은 고급 프로그램 언어들과 상당히 상이한 구조적 특성을 가진 또 하나의 프로그램 언어에 불과하여 이의 숙달에 많은 시간과 노력이 요하며, 이들은 일반 프로그래밍 언어들이 갖는 화일 관리의 다양한 잇점을 갖지 못하여 CAI 전용 저작도구들의 실제 사용은 저조하다[4, 5, 19].

본 논문은 CAI운영의 경제적 부담을 줄이고 기종간의 이식성을 높이며 프로그램 작성의 기술 습득없이 비 프로그램 방식으로 코스웨어를 저작하기 위하여 16bit 퍼스날 컴퓨터를 이용한 저작 시스템을 설계하고 그 특성의 평가에 대하여 연구한다.

학습화면의 설계는 학습화면 편집기(Screen Design Adapter : SDA)에 의하며, 음성정보의 처리는 음성관리 매체인 카세트 데크의 여러가지 기능을 컴퓨터에 의해 제어함으로써 CAI에서 시청각 학습을 실현한다. 영상정보의 생성 및 관리는 그래픽 패키지를 저작도구에 인터페이스시켜 시스템의 부담을 감소시키고, CAI 코스웨어의 각종 화일의 관리와 학습능력과 학습자의 예상되는 반응에 대한 분기학습이 이루어지도록 하는 망구조식 코스웨어의 흐름을 제어하기 위하여 PASCAL 언어와 유사한 구문체계를 갖는 저작 시스템 제어

언어를 개발한다.

2. 저작 시스템의 설계

2.1 저작 시스템의 구성

일반적으로 컴퓨터 시스템의 기능은 프로그래밍 언어로서 표현이 가능하지만 교육용 프로그램은 학습자의 주의를 모으고 학습동기를 유발하기 위한 다양한 기법이 요구된다.

고급 프로그래밍 언어에 의한 이러한 기법의 처리는 어렵고, 때로는 구사할 수 없는 경우가 있을 뿐만 아니라 언어 자체를 활용할 수 있도록 이해하는데 상당한 학습기간과 노력이 요구되어 효과를 기대할 수 있는 학습 프로그램의 개발은 현재의 여건하에서는 어렵다.

교육용 프로그램은 특수한 기술과 난이도 높은 알고리즘의 구사가 요구되므로 그 복잡성은 전문 프로그래머들도 곤란을 느끼는 실정이므로 CAI 코스웨어를 전문적으로 저작하는 시스템이 요구된다.

그림 2-1은 본 연구에서 제안한 CAI저작 시스템의 구성에 대한 블럭 선도이다.

2.2 화면정보의 편집

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

본 시스템은 CAI 코스웨어 화면의 편집을 문자정보의 편집과 영상 정보의 편집으로 구분하였다.

문자정보의 편집은 하드웨어적 요소에 의존하는 code oriented 방법과 그리고 직접 문자의 형태를 표현하는 image oriented방법이 있는데, 전자의 경우에는 표현가능한 문자의 종류가 한정되어 있지만 후자의 경우에는 원하는 모든 형태의 문자의 출력이 가능하다.

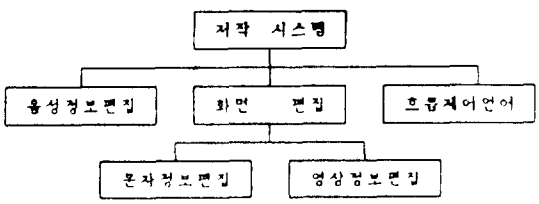


그림 2-1 저작 시스템의 블럭선도
Fig. 2-1 Block diagram of authoring system

CAI코스웨어 구성시 학습의 효과를 높이기 위하여 크기와 모양이 다양한 문자의 출력이 요구되므로 이러한 특성을 만족하며 영상 정보와의 호환성을 위하여 image oriented 방법에 의거 화면에 문자의 출력이 이루어 지도록 하였다.

영상정보의 표현은 영상을 2차원의 비트 매프에 대응시켜 표현하는 비트 매프 방식과 화면을 가상의 2차원 평면으로 가정하고 이 가상의 평면상에 기하학적인 평면도형 요소를 표현하는 벡터 방법이 있는데, 본 연구에서는 비트 매프방식에 의하였다.

그리고 코스웨어를 구성하는 문자정보는 코드로써 저장할 수는 없고 다만 영상만을 저장 할 수 있도록 하였다.

2.2.1 화면 편집기의 한글 코드체계

화면 편집기의 한글 코스웨어 저작을 위한 적절한 한글 코드 채택을 이하여 한글 코드의 여러 경우를 고찰하였다.

행정 전산망을 필두로 5대 기간 전산망이 구축되어짐에 따라 각 기종간의 정보 교환이나 자료의 공유가 매우 중대한 요소로 부각되고 있는 바, 기사용 한글의 코드 체계는 메이커 별로 다른 코드 체계를 채택하고 있으며, 같은 코드 체계내 에서도 작은 변형이 많이 있으나 이들은 7bit, 8bit및 16bit (2 byte)체계등으로 구분 요약된다.

7bit 코드체계는 기본적으로 영문 코드의 체계를 그대로 채용한 것으로써 한글 시작코드(SO)와 한글 끝코드(SI)를 지정하여 사용하므로 이들 제어코드에 의하여 영문전용의 하드웨어나 소프트웨어를 그대로 사용할 수가 없으며, 또한 한자를 수용할 코드영역의 확보에 어려움이 있어 CAI 용으로 채택하기가 곤란하였고, 8bit 코드체계는 7bit코드 체계가 가지는 한글 시작 및 한글 끝 코드에 대한 문제점을 해결하였지만, 한글 사용시의 가변길이의 문제가 그대로 남아 있어 하드웨어 및 소프트웨어의 특성에 맞게 개발하여야 하는 점과 여전히 영문 CAI소프트 웨어와의 호환을 기대하기는 어려웠다. 16bit 코드 체계는 초성, 중성, 종성의 자소마다 각각 5bit씩 지정하여 각자소를 코드 체계와 초성, 중성, 종성을 구별할 수 있는 조합형 16 bit 코드 체계와, 초성, 중성, 종성을 구별하지 않고 사용빈도수가 많은 한글 음절을 골라 각 음절에 대하여 연속적이고 순차적인 코드값을 배열하는 완성형 16bit 코드체계가 있다.

16bit 코드체계는 7혹은 8bit 코드체계가 가지는 문제점을 근본적으로 해결하였으며 한글 한 음절이 가지는 기억장소의 길이가 일정(2byte)하여 화면에 표시되는 음절의 길이와 일치하기 때문에 소프트웨어의 개발이 매우 편리하였으나 종래의 영문 주변기기등을 그대로 사용할 수가 없어 이에 대한 별도의 주변기기가 개발되어져야 한다.

완성형 16bit코드체계는 "가나다……"순으로 정렬하되 사용빈도수가 높은 음절만 골라 한글 한 음절마다 코드를 부여한 방식이므로 한글 처리시 모아쓰기 방식의 하드웨어가 사용될 수 없어 각 코드의 문자를 표시하는 문자발생기의 크기가 커지게 된다.

이는 관련 주변기기의 별도의 개발을 요할 뿐만 아니라 기기의 용량을 많이 필요하게 되어 경제적 부담을 높이며, 한정된 글자만을 사용하여야 하는 단점 때문에 CAI코스웨어의 다양성을 추구할 수가 없어 적합하지 아니하나, ISO의 국제 표준규격을 따를 수가 있으므로 국제적 통신상의 잇점이 있다.

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

이 방식은 하드웨어 설계 및 개발이 완성형에 비하여 덜 복잡하고, 문자 발생기의 크기를 작게 할 수 있으나 국제 표준 규격을 따르기가 어렵다는 단점이 있다.

2.2.2 문자의 표현

문자의 표현은 영문자(숫자 포함)와 한글과 특수기호로 나누어 그 표현방법을 다음과 같이 하였다.

영문자는 ASCII코드체계를 따르므로 그 코드에 대응하는 이미지를 비트 매프로 구성하여 화면에 표시할 수 있어 비교적 간단한 응용이 가능하지만 교육 프로그램의 환경으로 구축하기 위해서는 여러벌의 문자체가 필요하며 이를 위해 로마체의 경우에는 8dot*8line, 8dot*16line, 16dot*16line,

표 2-1 조합형 2byte 한글의 구성
Table 2-1 Ogaition of assembling 2 bytes Hangul

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	*	초성표시영역					중성표시영역					종성표시영역				

(*한글 표시시 MSB는 1로 setting 됨)

표 2-2 초성자형(ㄱ)의 기본형
Table 2-2 Basic models of Hangul initial consonant"ㄱ"

1형	가
2형	각
3형	과
4형	곽
5형	고
6형	곡

표 2-3 중성자형의 기본형
Table 2-3 Basic models of Hangul vowel

1형	가	고	받침없음
2형	각	곡	받침없음

16dot*24line등 4종류의 폰트를, 그리고 이태릭체의 경우에는 16dot*16line및 16dot*24line등 2종류의 폰트를 준비하였다.

이처럼 크기와 모양이 상이한 영문자의 표시를 위해서는 별도의 처리절차가 필요하다.

한글의 표시는 한글 구조의 특수성 때문에 상당히 복잡한 방법이 요구되었는데, 앞에서 기술한 잇점을 고려하여 조합형 2byte체계를 채용하였다.

표 2-1은 2byte의 각 bit에 한글 한 글자의 초성, 중성, 종성자소의 대응을 나타낸 것이다.

동일한 초성이라도 그 다음에 연결되는 중성과 종성에 따라 다른 모양의 자형이 요구되는 바, 표 2-2는 한글 "기억자"에 대한 6가지의 경우의 쓰임새 예를 보여준다.

중성은 다음에 종성(받침)이 있을 때와 없을 때의 두가지 경우의 자형이 필요하다. 표 2-3은 그 예를 나타낸 것이다.

종성은 한가지만 있으면 된다. 이러한 자형은 한글의 코드에 준하여 조합하면 요구되는 모든 한글의 표기가 가능하다.

표 2-4는 본 연구에서 채택한 화면 편집기의 2 byte 조합형 한글의 초, 중, 종성에 해당하는 코드값을 나타낸 것이다.

2.2.3 특수문자

화면편집기의 수식표기시 사용 빈도수가 높은 총 213종의 특수문자 코드를 표 2-5, 표 2-6 및 표 2-7과 같이 정의하여 코스웨어 작성시 요구되는 각종 기호를 출력하도록 하였다.

특수문자를 출력시키려면 특수문자 루틴으로 진입한 후에 그 문자에 해당하는 코드값을 부여하면 되는 데, 각 특수문자의 코드값은 십진수 3자리로

표 2-4 2바이트 조합형 한글 코드표
Table 2-4 2Bytes Assembling Hangul codes

HEX	초성	중성	종성
00	사용안함		
01	(FILL)		(FILL)
02	ㄱ	(FILL)	ㄱ
03	ㄲ	ㅏ	ㄲ
04	ㄴ	ㅐ	ㄳ
05	ㄷ	ㅑ	ㄴ
06	ㄸ	ㅒ	ㄵ
07	ㄹ	ㅓ	ㄶ
08	ㅁ		ㄷ
09	ㅂ		ㄹ
0A	ㅃ	ㅔ	ㄺ
0B	ㅅ	ㅕ	ㄻ
0C	ㅆ	ㅖ	ㄼ
0D	ㅇ	ㅗ	ㄽ
0E	ㅈ	ㅗㅏ	ㄾ
0F	ㅉ	ㅗㅐ	ㄿ
10	ㅊ		ㅀ
11	ㅋ		ㅁ
12	ㅌ	ㅗㅣ	ㅁㄱ
13	ㅍ	ㅛ	ㅂ
14	ㅎ	ㅜ	ㅂㅅ
15	사용안함	ㅜㅓ	ㅅ
16	사용안함	ㅜㅔ	ㅆ
17	사용안함	ㅜㅣ	ㅇ
18	사용안함		ㅈ
19	사용안함		ㅊ
1A	사용안함	ㅠ	ㅋ
1B	사용안함	ㅡ	ㅌ
1C	사용안함	ㅝ	ㅍ
1D	사용안함	ㅣ	ㅎ
1E	사용안함	사용안함	사용안함
1F		사용안함	

정의하였으며, 이때 같은 모양의 기호이라도 크기에 따라 그 코드 값은 달라지며, 때로는 하나의 특수문자를 위한 코드가 2개 인것도 있다.

표 2-5는 본 연구에서 정의한 폰트 크기 8dot*8line인 특수문자 코드표이다(총 19종).

표 2-6은 본 연구에서 정의한 폰트 크기 16dot*16line인 특수문자(총 117종)의 일부에 대한 코드표이다.

표 2-5. 작은 특수문자(8*8)
Table 2-5. Special Character Code(8*8)

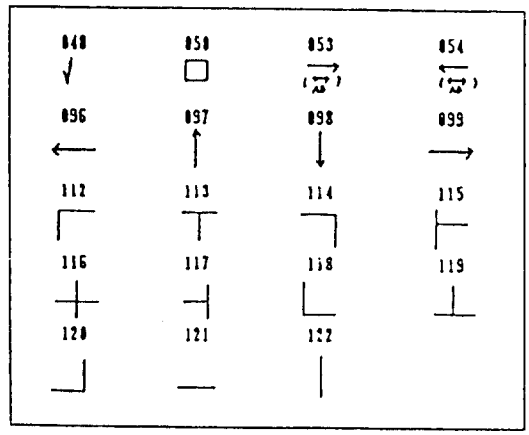


표 2-7은 본 연구에서 정의한 폰트 크기 16dot*24line인 특수문자(총 77종)의 일부에 대한 코드표이다.

영상의 표시는 문자와 마찬가지로 그 영상의 비트 매프를 구성하여 비데오 램에 옮겨 줌으로써 가능하다.

그러나 영상은 문자와는 달리 정형화된 패턴이 없고 임의의 배열로 구성되므로 필요한 영상은 그때 마다 영상 스캐너를 통해 입력, 구성하여야 한다.

영상정보는 처리시 많은 메모리 용량이 요구되고 표시 방법도 다양한데 본 연구에서는 처리의 편의를 위하여 영상을 축약하지 않고 그대로 저장, 재생하는 방법을 택하였다.

이 방법은 한 화면(Hercules Graphic Adapter의 720×350 해상도 mode)을 위하여는 약 32KB의 저장용량을 필요로 한다.

영상 화일의 생성을 위하여는 별도의 그래픽 팩키지를 사용하였고, 기억 장소의 절약을 위하여 "Dr Halo"의 영상 compress method를 이용하였으며, 필요한 그림의 영역을 지정하여 화면으로 전사 하는 방법을 취하였다.

2.3 저작 시스템 제어 언어

CAI의 코스웨어는 문자(character), 영상(image) 그리고 음성(voice) 정보로 구성된 다수 화일들의 일정한 순서에 의한 집합이다.

이때, 순서는 예상되는 학습자의 다양한 반응을 전제로 하여 미리 각 경우의 반응에 맞게 코스웨어의 흐름을 프로그램 형식으로 정의한 것인 바,

표 2-6. 중간특수문자 코드(16*16)

Table 2-6. Special Character Code(16*16)

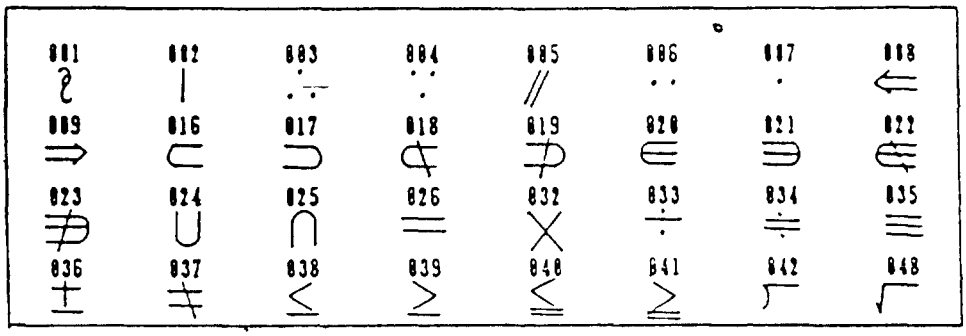
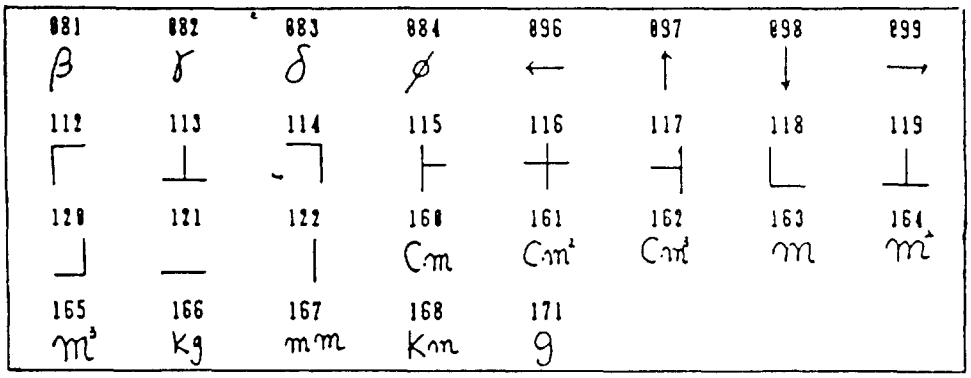


표 2-7. 큰 특수문자 코드

Table 2-7. Special Character Code(16*24)



이를 위한 언어가 필요하다.

본 논문에서는 CAI 코스웨어의 흐름을 제어할 목적으로 PASCAL언어와 비슷한 문법적 구조를 갖는 간이 컴퓨터 프로그래밍 언어를 고안, 한글 저작시스템의 구성에 사용하였는 바, 편의상 이를 저작 시스템 제어 언어(Flow Control Language: FCL)라고 명명 하였다.

2.3.1 문 자

저작 시스템 제어용 언어의 컴파일러가 인식할 수 있는 문자는 A에서 Z까지의 대문자 26글자와 그리고 a에서 z까지의 소문자 26글자 및 underline(_)등이다.

그리고 숫자는 0부터 9까지의 10개이며 특수문자의 종류는 다음과 같다.

blank	:	
greater than	:	>
less than	:	<
comma	:	,
period (decimal point)	:	.
colon (comment)	:	:
equal	:	=
semi colon	:	;
apostrophe	:	'

2.3.2 예약어

저작 시스템의 운영과 CAI 코스 웨어를 구성하는 각종 화일의 생성과 관리(save, load, update 등)등을 위하여 요구되는 기능을 일으키도록 그 뜻을 미리 정의한 예약어는 다음과 같다.

BEGIN	FREE	REPEAT
CALL	GET	RETURN
CASE	GOSUB	SELECT
MOVE	NOSOUND	PICTURE
CLS	JUMP	SOUND
DELAY	PICTUREDIR	TEXT

END PROCESS TEXTDIR
ENDPROCEDURE PROGRAM TEXTDIR
END SELECT

2.3.3 저작 시스템 제어언어 프로시져

저작 시스템 제어언어 일부 명령어의 syntax, purpose, remarks및 example에 대한 정의는 다음과 같다.

여기에서 "syntax"는 명령어의 정확한 표기를 나타내며, "purpose"의 보충설명으로 명령어가 어떻게 쓰이는 지와 각종 파라메타의 정의를 나타내며, "example"에서는 본 명령어의 실제 사용 예

⟨program⟩ : : =⟨program heading⟩ ⟨block⟩
⟨block⟩ : : =begin ⟨statement⟩ end.
⟨program begin⟩ : : =program ⟨identifier⟩
⟨statement⟩ : : =⟨process statement⟩

를 보인 것이다.

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

1. DELAY

syntax delay(i)
purpose i만큼 현재의 동작을 지연한다.
remarks i의 길이는 millisecond에 해당한다.
example
```
begin
    delay(1000)
end
```

2. FREE

syntax free(i1, i2, i3)
purpose graphic buffer를 dispose하고 관련 변수를 clear한다.
remarks 저작시스템제어언어의 graphic buffer는 1에서 10까지 준비되어 있는데, 이 때 필요없는 buffer를 지운다.
example
```
begin
    get (10, 10, 200, 200, 1)
    move(10, 20, 1)
    free(10, 20, 1)
end
```

3. GET

syntax get(i1, j1, i2, j2, i)
purpose image file중에서 animation을 행할 부분을 위한 graphic buffer와 각종 변수값을 설정한다.
remarks 지정된 image file에서 좌표점(i1, j1)과 점(i2, j2)를 대각선으로 하는 사각형 부분을 animation area로 설정, 이를 지정된 graphic buffer i에 할당한다.
example see 2. FREE

2.3.4 저작 시스템 제어언어의 구문

저작 시스템 제어언어의 identifier, expression, statement, process declaration 및 program 등의 일부에 대한 구조를 BNF (Backus-Naur Form)으로 나타내면 다음과 같다.

```
                : ⟨structured statement⟩
                : ⟨conditional statement⟩
⟨process statement⟩ : : =⟨process declaration⟩
⟨structured statement⟩ : : =⟨compouend statement⟩
                           : ⟨repeatition statement⟩
⟨compound statement⟩ : : =begin⟨statement⟩end
⟨repeation statement⟩ : : =⟨repeat statement⟩
⟨repeat statement⟩ : : =repeat⟨statement⟩
                        until⟨expression⟩
⟨conditional statement⟩ : : =⟨select statement⟩
                                : ⟨case statement⟩
⟨select statement⟩ : : =⟨select statement⟩
                          end select
⟨case statement⟩ : : =case⟨unsigned integer⟩
                         : ⟨case branch⟩
⟨case branch⟩ : : =goto⟨identifier⟩
                  : gosub⟨indentifier⟩
                  : return
⟨process declatation⟩ : : =⟨process heading⟩
                            begin⟨statement⟩
                            end process
⟨process heading⟩ : : =process⟨identifier⟩
⟨expression⟩ : : =⟨count⟩=signed integer number
count : : =⟨digit⟩
signed integer number : : =⟨sign⟩ ⟨unsigned integer⟩
unsigned integer : : =⟨digit sequence⟩
digit sequence : : =⟨digit⟩
sign : : =+ : −
digit : : =1 : 2 : 3 : 4 : 5 :
          6 : 7 : 8 : 9 : 0
identifier : : =⟨letter⟩ {⟨letter⟩ : ⟨digit⟩
                 : ⟨underscore⟩}……
underscore : : =−
letter : : =A : B : C : D : E : …… : Z :
           a : b : c : d : e : …… : z
```

2.3.5 화면 편집기의 기능 키

텍스트 화면의 편집기는 저자가 원하는 형태의 CAI 코스웨어를 저작하도록 하기 위하여 여러 모드로의 진입이 가능하고, 자유로워야 하며, 이러한 작동이 손쉽게 일어나도록 설계되어져야 한다.

화면 편집기에 의한 텍스트 화일의 구성에 쓰이는 한글은 2종류의 폰트를, 그리고 영문자의 경우에는 로마체와 이태릭체 각각 4종류및 2종류의 폰트를, 그리고 특수문자는 3종류의 폰트를 설계하였다.

본 시스템의 화면 편집기는 학습화면 구성시 각 문자 모드로의 진입을 위하여 이에 대한 루틴을 작성하고, 그 기능을 표 2-8및 표 2-9와 같이 기능 키에 부여하였다.

2.3.6 한글 저작 시스템의 하드웨어 사양

16bit퍼스날 컴퓨터의 DOS하에서 한글과 영문의 CAI코스웨어 자작을 가능하도록 설계되어진 본 시스템은 저작과 효율적인 운용을 위하여 일정 수준 이상의 하드웨어적 환경이 구축되어져야 하는 데, 이를 위한 최소한의 요구는 다음과 같다.

(1) IBM PC/XT/AT or it's compatable

(2) 640 KB RAM or more

(3) 2 FDD (20 MB HDD)

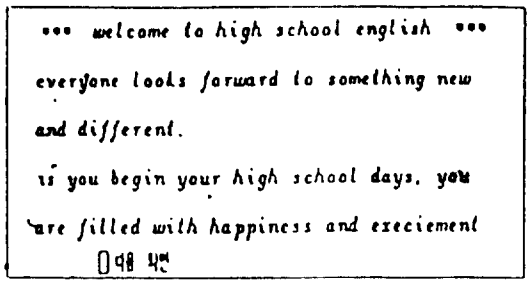
(4) DOS VERSION 2.10 or LATER

(5) HERCULES GRAPHIC BOARD & IT'S MONITOR(MONO)

(6) EGA COLOR GRAPHIC BOARD & IT'S MONITOR(COLOR)

3. 결과 및 고찰

CAI의 코스웨어 구성시 학습자에게 중요한 내용등을 환기시킬 목적으로 화면 편집기에서 크기



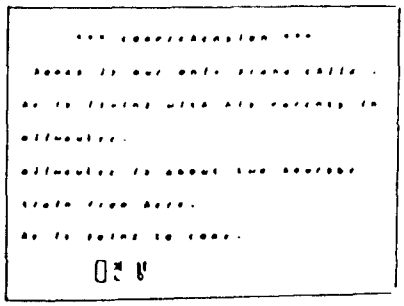
화면 3-1 이태릭체 큰 글자(16dot*24line)
Scrn. 3-1 italic character(16dot*24line)

표 2-8 화면 편집기의 기능 키이
Table 2-8 Function key of screen design adapter

function key	mode	font size	remark
F1	한글	16*24, 16*16	UNLOCK
F2 F3	영문로마체 영문로마체	16*24, 16*16 8*16, 8*8	LOCK은 대문자 위와 같음
F4	영문이태릭	16*24, 16*16	위와 같은
F5	그래프	8*8	F5+SHIFT+CCK
F6	특수문자	16*24, 16*16, 8*8	F6+코드값
F9	mono		화면의 상태(흑백)
F10	color		화면의 상태(칼라)

표 2-9 ALT+기능 키
Table 2-9 ALT+FUNCTION key

Key	function	font	remarks
F1	upper line	16*24, 16*16	ALT+F1에 의해 upper=on
F2	underline	8*8, 8*16	ALT+F2에 의해 under=on
F8	redisplay		화면의 재정리
F9	clear		화면의 소거
F10	black/color		화면의 color를 바꿈



화면 3-2 이태릭체 작은글자(16dot＊16line)
Scrn. 3-2 Italic character(16dot＊16line)

와 모양이 다양한 글자체를 구사하도록 하였는데, 화면(Screen : Scrn) 3-1은 폰트 크기 16dot＊24line인 영문 이태릭체의 예이다. 이 경우 한 화면의 표현 가능한 글자수는 26＊16이다.

화면 3-2는 폰트 크기 16dot＊16line인 영문 이태릭체의 예이다. 이 경우 한 화면의 표현 가능한 글자수는 40＊25이다.

화면 3-1과 3-2에 의하여 영문 이태릭체의 코스웨어 작성이 가능하였다. 학습화면의 설계시 글자의 크기와 모양의 적절한 선택은 학습자에게 학습에 대한 호감을 줄 수 있을 뿐만 아니라 종래의 프로그램 학습이 범하는 단조로움을 다소 제거하였다.

본 논문에서는 총 104개의 로마체 폰트를 설계하였다. 화면 3-3은 폰트 사이즈 16dot＊24line인 영문 로마체의 예이다. 이 경우 한 화면의 표현 가능한 글자수는 26＊16이다.

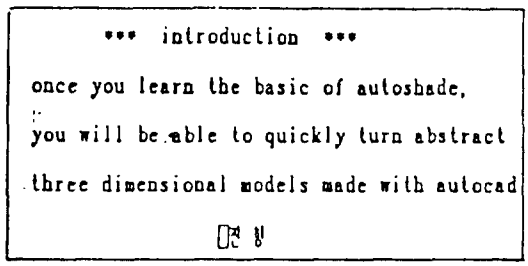
화면 3-4는 폰트 사이즈 16dot＊16line인 영문 로마체의 예이다. 이 경우 한 화면의 표현 가능한 글자수는 40＊25이다.

화면 3-5은 폰트 사이즈 8dot＊16line인 영문 로마체의 예이다. 이 경우 한 화면의 표현 가능한 글자수는 80＊25이다.

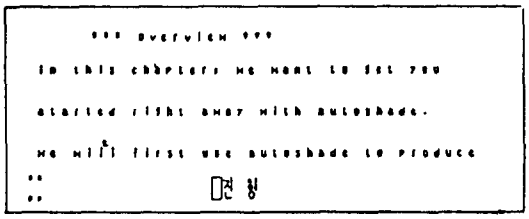
화면 3-6은 폰트 사이즈 8dot＊8line인 영문 로마체의 예이다. 이 경우 한 화면의 표현 가능한 글자수는 80＊80이다.

화면 3-3에서 3-6까지는 로마체의 다양한 글자의 크기에 대한 예로써 코스웨어의 본문, 요점정리, 주석 및 각 종 첨자의 표기가 가능하며 미려하고 균형이 잡힌 학습화면을 얻었다.

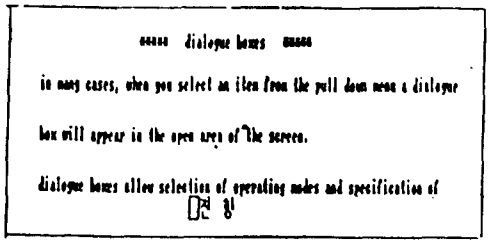
본 논문에서는 총 208개의 영문 로마체 폰트를 설계하였다. 화면 3-7은 본 논문에서 정의한 폰트



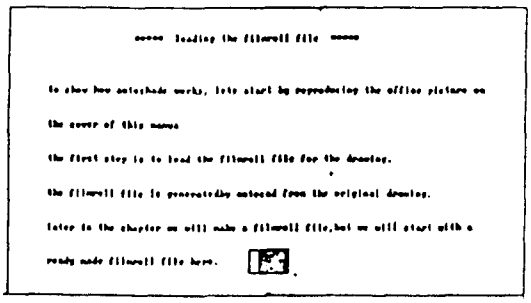
화면 3-3 로마체 큰 글자(16dot＊24line)
Scrn. 3-3 Romanic character(16dot＊24line)



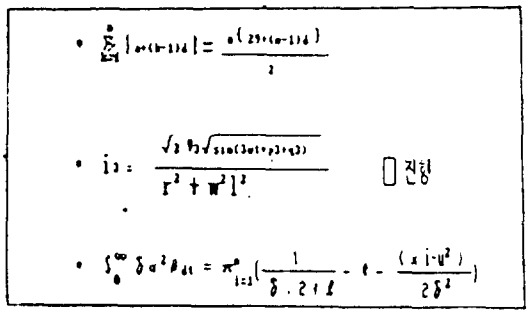
화면 3-4 로마체 중간 글자(16dot＊16line)
Scrn. 3-4 Romanic character(16dot＊16line)



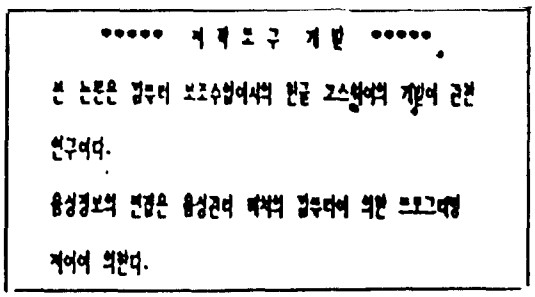
화면 3-5 로마체 작은 글자(8dot＊16line)
Scrn. 3-5 Romanic character(8dot＊16line)



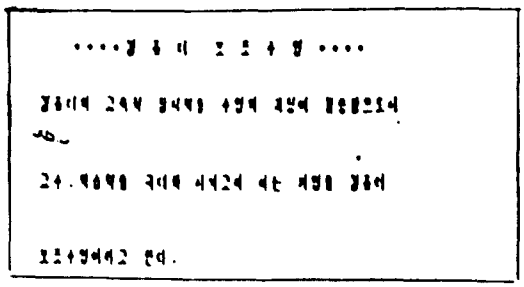
화면 3-6 로마체 제일 작은 글자(8dot＊8line)
Scrn. 3-6 Romanic character(8dot＊8line)



화면 3-7 특수문자
Scrn. 3-7 special character



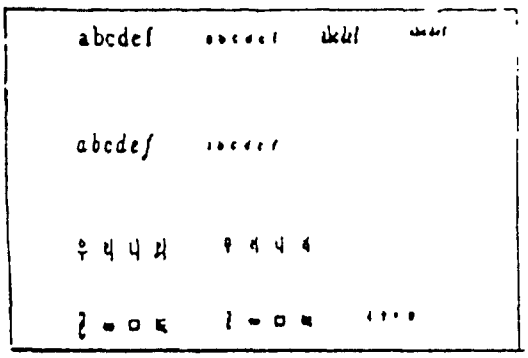
화면 3-8 한글 큰 글자(16dot*24line)
Scrn. 3-8 Hangul(16dot*24line)



화면 3-9 한글 작은 글자(16*16)
Scrn. 3-9 Hangul(16*16)

사이즈 8dot*8line, 16dot*16line및 16dot*24line의 총 213개중의 일부 특수문자를 사용하여 수식을 나타낸 것이다.

일반적으로 컴퓨터 시스템에서 정의하여 사용하고 있는 특수문자는 상당히 제한되어 있어서 CAI의 코스웨어의 학습화면의 디자인시 곤란이 따르는데, 본 시스템의 화면 편집기를 사용한 결과 다른 도구로써 표기할 수 없는 상당 부분에 대하여 만족할 만한 결과를 얻었다.



화면 3-10 여러 모드의 글자
Scrn. 3-10 characters in different modes

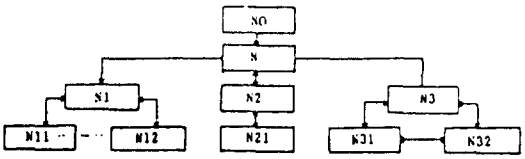


그림 3-11 네트워크방식의 코스웨어 흐름도
Fig. 3-11 flow chart of network type courseware

그러나 nested ROOT나 번분수의 표기시에 세련미가 결여되었고, 준비된 특수문자의 대부분은 수식등의 표기에는 적합하였으나 음악이나 미술 및 체육분야를 위한 특수문자의 추가가 있어야 한다.

화면 3-8은 화면 편집기로 작성한 폰트 사이즈 16dot*24line인 한글의 예이다.

이 경우 한 화면에 표기 가능한 한글의 수는 26*16이다. CAI코스웨어에서 이러한 글자체는 제목, 정답, 학습의 주안점등의 표기에 적절히 혼용해 씀으로써 학습의 효과를 증진시켰다.

화면 3-9는 폰트 사이즈 16dot*16line인 한글의 예이다. 이 경우 한 화면의 표현 가능한 글자수는 40*25로써 일반 문서의 표기시 가장 많이 사용하는 크기의 글자이다.

이것은 비교적 눈에 피로를 적게주면서 화면에 표기할 수 있는 글자수가 많기 때문에 고학년용 CAI학습화면의 구성에 자주 쓰이는 한글 모드이다.

본 연구용으로 준비한 한글 자소 폰트는 초성용 288개, 중성용 97개 그리고 종성용으로 48개 이다.

화면 3-10은 본 화면 편집기에서 표현 가능한

각 모드의 문자를 비교하기 위하여 한 화면에 나타낸 것이다. 제1줄은 영문 로마체의 4가지 모드이고, 제2줄은 영문 이태릭체의 2가지 모드이다. 그리고 제3줄은 한글 2가지 모드 이고, 제4줄은 특수문자 3가지 모드에 해당된다.

그림 3-11은 가상의 분지학습코스에 대한 흐름도이다. "NO"는 타이틀 화면으로 본 전체 코스웨어의 커리큐럼을 나타낸다. 그리고 "N"은 메인 메뉴를 나타내주는 화면이다.

이 커리큐럼은 "N1", "N2"및 "N3"등의 3개의 학습 코스로 이루어져 있으며 다시 "N1"은 "N11"과 "N12"의 부 코스로 구성되며, "N2"와 "N3"의 경우도 각각 "N21"및 "N31", "N32"등의 부 코스로 이루어져 있다.

"N1"과 "N2"및 N3의 쌍 방향 표시는 동일 계열 상하위 학습코스간의 자유로운 진행을 나타내준다. 특히 "N3"와 그의 서부 코스인 "N31"및 "N32"의 관계는 학습의 수평적 진행을 나타낸다.

아래 프로그램은 그림 3-11의 네트워크방식의 코스웨어 흐름을 일으키게하는 저작 시스템 제어언어 프로그램이다. 저작 시스템 제어언어의 프로그램은 동일 학습계열 내에서의 쌍 방향성 수직적, 수평적 코스웨어의 흐름을 일으키게 할 수 있을 뿐만 아니라 다른 학습 계열간의 이동이 극히 자유롭게 일어 났으며, 코스웨어의 흐름 변경과 수정은 프로그램 당해 부분의 수정에 의하는 것이므로 학습자들의 인지적 특성을 고려한 다양한 개별학습이 가능하였다.

```
PPRCESS N32
BEGIN
    TEXT('N32')
    SELECT
        CASE1 : GOTO N32
    END SELECT
END PROCESS
;
PROCESS N31
BEGIN
    TEXT('N31')
    SELECT
        CASE 1 : GOTO N31
    END SELECT
END PROCESS
;
PROCESS N3
BEGIN
    TEXT('N3')
    SELECT
        CASE 1 : GOTO N31
        CASE 2 : GOTO N32
    END SELECT
PROCESS N11
BEGIN
    TEXT('N11')
    RETURN
END PROCESS
;
PROCESS N1
BEGIN
    TEXT('N1')
    SELECT
        CASE 1 : GOSUB N11
        CASE 2 : GOSUB N12
END POROCESS
;
PROCESS N21
BEGIN
    TEXT('N21')
    RETURN
END PROCESS
;
PROCESS N2
BEGIN
    TEXT('N2')
    SELECT
        CASE 1 : RETURN
        CASE 2 : GOSUB N21
    END SELECT
END PROCESS
;
;
BEGIN
    TEXT('N12')
    RETURN
END PROCESS
;
    END SELECT
END PROCESS
;
PROCESS NO
```

```
BEGIN
    TEXT('NO')
    SELECT
        CASE 1 : RETURN
    END SELECT
END PROCESS
;
; MAIN ROUTINE
    END SELECT
END PROCESS
;
PROCESS N
BEGIN
    TEXT('N')
    SELECT
        CASE 1 : GOTO N1
        CASE 2 : GOSUB N2
        CASE 3 : GOTO N3
;
BEGIN
    CALL NO
    JUMP N
END
```

본 시스템의 화면 편집기를 이용하여 CAI코스웨어의 작성시 이를 위한 별도의 프로그래밍 언어의 숙지가 요하지 않으므로 교수 설계자에 의한 CAI코스웨어의 저작이 가능하였고, 저작시 고급 프로그래밍 언어의 사용에 비하여 상당히 소요시간을 단축시켰다.

특히 그래픽 패키지에 의한 영상화일의 생성은 정교하고 손쉬울 뿐만 아니라 단 시간내에 처리가 이루어 지기 때문에 프로그래밍 언어의 사용에 비하여 영상화일의 생성과 다양성의 추구 및 관리시 편리하였고, 영상화일 처리를 위한 시스템의 하드웨어적 부담을 줄일 수 있었다.

그러나 본 저작 시스템은 CAI코스웨어 학습의 진행을 전적으로 학습자 조절 방식을 택함으로써 학습시간 낭비의 소지가 있었고, 시뮬레이션이나 애니메이션등 정교한 프로그램개발을 위한 유틸리티나 패키지가 요구되었고, 코스웨어 설계시 가이드를 제공해주는 매뉴운영 방식의 명령어의 추가가 필요하다.

4. 결 론

얻어진 결과는 다음과 같다.

1. 화면 편집기로 한글/영문(숫자 및 특수문자 포함) CAI코스웨어를 저작하였으며, 각 문자의 다양한 폰트에 의해 미려하고, 균형이 잡힌 학습 화면 설계로 프로그램 학습의 단조로움을 감소시켰다 :
2. 화면 편집기에 의한 저작은 비 프로그램 방식이므로 프로그램 작성기술 습득이 요하지 않고, 교수 설계자에 의한 직접 저작이 가능했으며, 작성시 소요시간이 단축 되었다.
3. 그래픽 패키지를 저작 시스템과 인터페이스 시킴으로써 영상정보의 처리를 위한 하드웨어 부담과 개발비용을 줄일 수 있었으며, 프로그램 방식에 비하여 적은 시간내에 보다 정교한 영상 화일의 생성 및 갱신이 이루어 졌으며, 운용의 편의성이 있었다.
4. 저작 시스템 제어언어 프로그램은 네트워크식 CAI 코스웨어의 흐름을 입체적으로 제어하였으며, 코스웨어의 내용 수정 및 흐름의 변경이 자유로와 다양한 학습자의 요구와 학습 능력에 따른 분기 및 개별학습이 가능하였다.
5. 본 저작 시스템은 system independent타입으로 일정수준 이상의 하드웨어적 요건을 만족시키면 동일 레벨의 기종간에 충분한 이식성이 있었다.
6. 본 저작 시스템은 16bit퍼스날 컴퓨터를 메인 시스템으로 하는 저작 시스템이므로 마이크로 레벨 이상의 다른 시스템에 비하여 시스템의 구성 및 유지시 경제적, 인적 자원을 절감할 수 있었다.

본 연구의 결과는 기능과 환경이 개선되고 있는 퍼스날 컴퓨터의 DOS하에서 기동되는 한글저작 시스템으로써, 비 프로그램 방식으로 코스웨어의 저작과 운용을 가능하게 함으로써, 정보화 사회를 대비한 개별학습에 이바지 할 것이다.

참 고 문 헌

[1] Alderman D. L., Evalution of the TICCIT computer assisted ins truction system in the community college, Education Testing Service, Princeton, New Jersey, 1978.

2] Anderew G. Gibbons, Development technology for CBT : Theory and Practice, Seminar KSSVE, pp. c1-c12, 1986.

[3] Barkers, P.G., Singh, R., A praction, Part3 :

MICROTEXT,. British J., Educational Tech., No. 2, Vol. 15, pp. 82~105, May, 1984.
[4] Chambers, Jack A. and Jerry W. Sprencher, Computer Assisted instruction : current trends and critical issues, ACM, Vol. 13, No. 6, pp. 332 ; 342, 1985.
[5] Creg Kearsley, Artificial Intelligence and Insdustrial on Application and Methods, pp. 1 ; 45, 1971.
[6] Cupertino Union School District, Cupertino school district develops computer literacy curricculum, The computer Teacher, Vol. 9, No. 1, 1981.
[7] F.E. Fairweather, A.F. O'Neal, The impact of advanced authoring systems oin CAI productivity, J., Computer Based Instruction, Summer 1984.
[8] Hasll K. A., A research model for applying technology to interctive instructional process, J., computer-based inatruction, Vol. No. 3, pp. 68~75, 1977, 7
[9] M.J, Hilleisohm, Benchmarking authoring system, J. Computer, Based Instruction, Summer 1984.
[10] Mahidon J. M., Issue overview : Trends in computer-assisted instruction J., Edecation Technogy, Vol.18 No. 4, pp. 5~8, 1988.
[11] Mathews J.L., Micro cpuiter vs: Min 11 pp. pp1~pp. 22, 1178
[12] Okey j., & MAJER K., individualized and small group learning with computer instruction, AN Communication Reviw, Vol. 24, No. 1, p. 19, 1976.
[13] P.G. Fairweather, a.F.O' Neal, The Impact of Advances Authoring Systems on CAI Productivity, J., Computer-Based Instrucytion, vol. 11, No. 3, pp. 90~94, Summer, 1984.
[14] Park O.C., Designing a CBI course, A conference on the instructional use of computers : Department of Program Development and Evalution, State University of New York at Albany, Mar. 1983.
[15] Smith, P. L., Boyce B. A., Instructional design considerations in the develop(ment of computer-assisted instruction. J., Educational Technolopgy, pp. 5 ; 11, Jul. 1984.
[16] Stansfield D., The computer and education : The computer education, Engle Wppd Cliffs N. K., Edicatopma Tdlchnology. Vol. 10, No, 1, . 21, 1970.
[17] 고대곤외, 컴퓨터 보조수업에서의 한글 저작 도구에 관한 연구, 전국교육대학 전산교육연구, 제1집, pp. 1~24, 1989, 1.
[18] 국제전산(주), Vol.11, No.2, pp.81~100, 1984, 5 No.2 pp.81~100. 1984, 5
[19] 김영택외, 한글 프로그래밍 언어 설계에 관한 연구, 한국정보과학회 논문지, Vol. 11 2, 81~100, 5 5
[20] 이옥화, 컴퓨터의 교육적 이용과 학습, 학교공학연구, 제1권, 제2호, 171~209~1985.
[21] 장옥배, ICAI 모델 프로그램에 관한 연구, 한국정보과학회, Vol. 15, No, 1, pp. 105~108, 1987.
[22] 허운나, CAL의 의 학습효과에 관한 연구, 한국 교육항 연구 실재권, 제 1호 pp. 61~85 1985, 12.