

□ 기술예설 □

상호작용적 코스웨어를 위한 멀티미디어 저작도구의 설계 및 구현

시스템공학연구소 심부성* · 김상녕* · 이익현*

● 목

차 ●

- | | |
|--------------|---------------|
| 1. 서 론 | 3.4 명령어 라이브러리 |
| 2. 저작도구 현황 | 3.5 스크립트 해석기 |
| 2.1 저작도구 정의 | 4. MATIC의 구현 |
| 2.2 국내외 개발현황 | 4.1 시스템의 구현환경 |
| 3. 메틱의 설계 | 4.2 자료 구조 |
| 3.1 화면 제어기 | 4.3 주요 함수 |
| 3.2 객체 관리기 | 5. 향후 연구방향 |
| 3.3 메시지 관리기 | 6. 결 론 |

1. 서 론

컴퓨터활용교육/훈련법은 컴퓨터 하드웨어 및 소프트웨어 기술의 향상과 더불어 다양하고 풍부한 학습경험을 적시적소에 전달할 수 있는 정보화시대의 대표적인 교수-학습 방법으로 인식되고 있다. 그러나 컴퓨터활용교육/훈련을 시행하는데 필요한 교육/훈련용 소프트웨어 즉, 코스웨어의 개발은 여러분야 전문가들의 긴밀한 공동작업을 전제로 많은 시간과 노력을 요하는 어려운 분야이다. 이러한 개발의 어려움은 코스웨어 제작시간을 지연시키고, 제작비용을 올리며, 그 질을 떨어뜨리는 결과를 초래한다. 따라서 다량의 양질 코스웨어를 최소의 시간과 비용으로 개발하기 위해서는 코스웨어 제작에 필요한 전문가들의 기술을 집약하여 놓은 사용이 용이한 도구적 소프트웨어의 개발 및 보급

이 필수적이다.

저작도구란 코스웨어의 개발을 도와주는 소프트웨어이다.

기본적으로 저작도구로 개발할 수 있는 코스웨어는 C언어 등 일반목적의 고급언어를 사용해서도 개발할 수 있으나, 저작도구를 사용함으로써 쉽게 빠른 시간내에 개발 및 유지보수할 수 있다. 초기의 저작도구는 코스웨어 제작과정에 많은 시간과 기술을 요하는 프로그래밍 과정을 간략한 명령어사용으로 대체한 저작언어(Authoring Language)의 형태로 소개되었으나 오늘날의 저작도구는 What you see is what you get을 지향하는 저작시스템(Authoring System)으로 대부분 학습내용의 제시, 질의응답 및 Branching 등의 기능이 내장되어 있다. 또한 저작도구가 멀티미디어의 활용을 지원하게 됨에 따라 코스웨어에 보다 현장감 넘치는 학습정보를 실을 수 있게 되었다. 이러한 멀티미디어 저

작도구는 코스웨어 뿐만 아니라 안내시스템, 프리젠테이션 시스템, 광보/홍보물에서 전자출판물의 제작에 까지 손쉽게 활용할 수 있다. 따라서 배우고 사용하기 쉬운 저작도구의 개발 보급은 코스웨어를 비롯한 각종 타이틀의 개발 생산성 및 그 품질 향상과 직결되며, 국내 멀티미디어 타이틀 사업을 조기에 활성화 시킬 수 있는 관건이 될 것이다.

저작도구의 교육적인 잠재력, 활용의 융통성 및 경제 사회적인 기여도에 대한 인식이 높아짐에 따라 미국과 같은 선진국에서는 최근 10여년 사이에 200여개의 저작도구가 개발되었으며 그 종류도 포괄적인 저작기능을 제공하는 일반형에서 몇몇 기능 또는 특수기능을 강조한 것 까지 매우 다양하다. 최근 국내에서도 외국의 유명 저작도구를 수입하거나 자체 개발을 시도하고 있으나 저작기능의 다양성 및 확장성, 사용의 용이성, 그 가격 및 보급율이 만족스럽지 못한 상태이다.

본 논문에서는 시스템공학연구소에 의해서 순수 국내기술로 개발된 저작도구 메딕(MATIC: Multimedia Authoring Tool for Interactive Courseware)의 설계 및 구현내용을 소개하고자 한다. 먼저 저작도구의 기본개념을 알아보고, 국내외 개발물들을 비교 분석함으로써 메딕 개발의 정당성을 제시하였다. 메딕을 화면제어기, 객체관리기, 메시지 관리기, 명령어 라이브러리, 스크립트해석기의 5 모듈로 나누어 설명하고 그 구현방법에 대하여 기술하였다. 끝으로 메딕의 향후 보완방향과 국내외 저작도구의 개발방향에 대하여 논하였다.

2. 저작도구 현황

2.1 저작도구 정의

저작도구란 프로그래밍과 교수설계의 전문적인 지식없이도 교육/훈련용 소프트웨어를 손쉽게 개발할 수 있도록 지원하는 특수목적의 소프트웨어를 지칭한다. 저작도구의 목적은 코스웨어의 제작 및 유지보수의 효율성을 꾀함으로 궁극적으로 코스웨어의 질적 양적 향상을 도모하자는 데 있다. 저작도구로서 제공하여야 할 필요기능은 크게 교수설계 지원기능(코스웨

어의 대표적인 교수전략 구현 지원 및 학습데이터 관리기능), 화면개발 보조기능(다중매체 데이터의 활용/편집기능 및 기존 Utility Software와의 접속기능), 주변기기와의 접속기능, 학습자와의 상호작용 지원기능(다양한 질의/응답유형 제공, 융통성있는 판단 및 다양한 피이드백/Branching 기능) 등으로 대별되며 사용의 용이성은 저작도구로서의 필수조건이다. 저작도구는 각종 미디어를 통합 관리하는 기능을 지원한다는 측면에서 정보화시대의 필수적인 소프트웨어라고 말할 수 있다. 물론 각 미디어별로 상용화된 편집기들을 활용한다면 더욱 효율적으로 저작할 수 있을 것이다.

2.2 국내외 개발현황

국내외적으로 저작도구의 중요성이 인식됨에 따라 많은 저작도구가 개발되어 왔고 또한 개발중에 있다. 미국에서 현재 사용중인 저작도구의 수는 80여개에 이르고 국내에서도 미국의 유명 저작도구를 수입하거나 미약하나마 자체 개발물들을 내놓고 있는 실정이다. 저작도구는 포괄적인 저작기능을 제공하는 일반형에서 몇몇 기능 또는 특수기능을 강조한 전문형까지 매우 다양하나 본 논문에서는 대표적인 일반형 저작도구를 중심으로 그 특징을 비교 분석해보고 그 문제점을 지적해 봄으로써 메딕 개발의 정당성으로 제시하고자 한다.

2.2.1 국외 개발물 분석

국외에서 개발하여 상용화되어 있는 저작도구들 중에서 많이 이용되고 있는 저작도구들의 특성을 비교하여 그 장단점을 분석해 봄으로써 메딕을 설계하는데 참조하였다.

오씨웨어 프로페셔널은 미국 매크로미디어(MacroMedia)사에서 개발한 흐름도 형식의 저작도구이다. 흐름도 형식의 저작도구는 전산의 기본개념이 정립되어 있지 않은 사용자들에게는 어려운 개념(복잡한 논리를 구현하고자 할 때 더욱 어려움)이고 또한 요즈음 많이 활용되어지고 있는 하이퍼미디어 형태의 타이틀(예: 백과사전) 개발에는 적합하지 않다. 다양한 기능들을 간결하고 세련된 사용자 인터페이스로 표현하고 있으며, 또한 상호작용(Interaction)

과 의사결정(Decision) 아이콘을 사용하여 다양한 사용자와 컴퓨터 간의 상호작용 형태를 구현할 수 있으나 그 형태가 지나치게 세분화되어 있고, 세밀한 기능 구현시 저작이 어렵다. 11개의 아이콘을 이용하여 프로그램 언어의 개념없이 시나리오의 변경 및 시뮬레이션기능 구현이 가능하며, Map Icon 기능을 이용하여 사용자가 구현된 부분중 원하는 부분을 모듈화할 수 있다. 보급형으로 Authorware Star가 있다.

아이콘 오셔도 흐름도 방식의 프로그램 모델을 사용한 AimTech사 제품으로써, Authorware가 갖는 장단점을 거의 유사하게 가지고 있다. 55개의 도구 아이콘들을 제공하여 각 아이콘을 이해하기는 쉬우나 원하는 아이콘을 찾아서 흐름선에 배치하는 데는 시간이 걸리며 그 기능이 너무 세분화되어 있어 흐름도가 복잡해지는 경향이 있다.

멀티미디어툴북은 어서메트릭사 제품으로 하이퍼카드 개념에서 출발하여 페이지 상에서 아이콘기반으로 동작하는책방식의 이해하기 쉬운 인터페이스를 제공하고 있고, Event-driven방식으로 객체지향 기법을 도입하고 있으며 스크립트 언어를 사용하여 객체에 특성을 부여하여 복잡한 응용프로그램까지 유연하게 개발할 수 있다. 그러나 대부분의 흐름제어를 스크립트 언어에 의존하여야 함으로 프로그래밍 경험이 없는 일반 사용자가 미세한 제어를 구현하기는 어렵다.

2.2.2 국내 개발물 분석

현재 국내 멀티미디어 저작도구 제품으로는 현대전자의 애기보따리, (주)유니텍의 파이디온, 레이스시스템의 그레이트(Great), 국제전산의 HERO Author, 한국전산의 OKAUTH, 프로텍소프트웨어의 한울, 솔빛조선미디어의 대용량한글처리 전문 글터, 삼석컴퓨터의 배움자리 등이 있다. 이들 중 도스환경에서 운영되는 저작도구는 윈도우에서 제공하는 DLL, DDE 송/수신, OLE, 멀티미디어 확장지원 등의 기능이 없다.

금성사가 개발한 아트웨어는 MS-Windows에서 운영되는 흐름도 방식의 저작도구이다. 매크로미디어(MacroMedia)사의 오셔웨어 프로

페셔널을 모델로 하였고 미디어의 동시실행 아이콘 기능을 갖고 있어 비디오, 오디오, 애니메이션의 동시표현이 가능하다. 질의응답 및 피드백 구현방안이 미흡하나, 화면효과와 동시처리 기능이 우수하다.

2.2.3 메틱 개발의 정당성

메틱(MATIC: *Multimedia Authoring Tool for Interactive Courseware*)은 MS-Windows에서 운영되는 멀티미디어 저작도구로서 국내외 저작도구가 흐름도 또는 책방식 프로그래밍 모델 중 한가지만을 채택함으로써 범하는 제한점을 극복해보고자 하였다. 즉 이해하기 쉬운 책방식의 프로그래밍 모델을 채택하되 대부분의 제어흐름(예: 다양한 형태의 질의 응답 및 피드백 제공, 멀티미디어 링크)은 대화상자에 의해 개발할 수 있도록 하여 사용의 용이성을 극대화 하였고, 특수한 제어흐름의 표현 및 저작 기능 확장성을 고려하여 자체의 스크립트 언어를 제공하였다. 따라서 메틱을 사용하여 전자출판물과 같은 하이퍼미디어 형태의 타이틀 및 교육/훈련용 소프트웨어와 같은 시나리오 위주의 타이틀을 모두 쉽게 제작할 수 있게 되었다.

외국 유명 저작도구와 비교하여 볼때 아직은 다소 기능적으로 미약한 점도 있지만(많이 활용되지 않는 기능은 생략시키거나 축소하였음) 오히려 특화된 기능 즉, 멀티미디어 링크, 상호작용, 전자출판용 타이틀 저작에 필요한 하이퍼링크 기술, 개별 메체를 한번에 불러올 수 있는 동시 아이콘 기능 등을 추가하였고 또한 저렴한 가격에 보급될 계획이므로 고가의 외국산 수입 저작도구의 대체효과를 기대해 볼 만한 상태이다.

3. 메틱의 설계

상호작용적인 코스웨어를 저작할 수 있는 멀티미디어 지원 저작도구 메틱의 시스템 구성은(그림 1)과 같다.

메틱은 화면 제어기, 객체 관리기, 메시지 관리기, 명령어 라이브러리, 스크립트 해석기 등 크게 5가지 모듈로 구성된다. 화면제어기는 프레임 편집 기능을 포함하며, 외부의 파일을 임

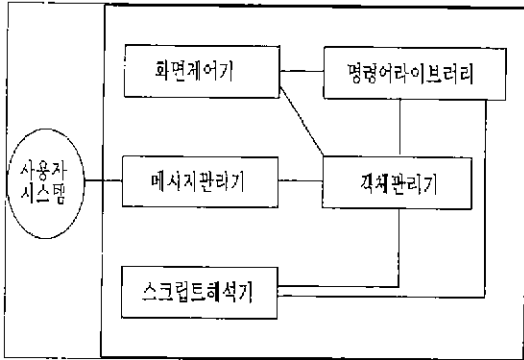


그림 1 메틱 시스템 구성도

포트(Import)하여 사용하게 하는 기능과 각종 객체 속성의 편집이나 프레임의 이동 기능, 실행모드의 화면에 발생하는 효과연출 기능 등을 담당한다. 주로 메뉴, 대화상자, 툴박스, 리본바, 상태바 등의 관리를 맡는다. 객체관리기는 저작도구에서 제공하는 모든 객체에 대한 관리를 담당한다. 객체에 대한 화일 입출력 함수, 메모리 관리 함수, 화면 입출력 함수, 객체 참조 함수 등을 포함한다. 저작도구 내의 실질적인 모든 관리를 담당하는 핵심 모듈이라 할 수 있다. 메시지 관리기는 객체관리기를 통해 메시지가 발생한 객체를 찾아서 그 객체에게 메시지를 보낸다. 즉 스크립트 해석기에 어느 객체인지와 메시지를 보내면 스크립트 해석기에서 메시지 처리기에 서술되어 있는 스크립트를 해석한다. 이 과정에서 스크립트의 특정 명령어나 문장의 처리를 위해 객체 관리기와 명령어 라이브러리를 호출하게 된다. 객체 간에도 메시지 전송이 필요하면 전송 메시지와 수신 객체를 메시지 관리기에 보낸다. 명령어 라이브러리는 객체 관리기에서 실행하고자 하는 기능이나 스크립트 명령어로 수행하고자 하는 각 기능을 각각 함수로 구현하여 호출할 수 있도록 한 것이다. 대부분의 메뉴 명령어는 스크립트 명령어로도 수행할 수 있도록 되어 있다. 스크립트 해석기는 사용자나 저작자 또는 시스템이 발생시킨 여러 가지 메시지와 제어가 메시지 관리기로부터 보내지면 해당 메시지의 처리를 찾아 해석하여 실행한다. 이 때 한 라인 단위로 스크립트를 해석하게 되며 제어문일 경우는 연산자의 우선순

위에 따라 계산하고 명령어일 경우는 명령어의 문법에 맞는지 확인하게 된다. 주어진 객체의 스크립트에 처리가 없으면 계층 구조 상의 상위 객체로 메시지를 보내며 메시지 관리기의 끝을 만나면 제어는 다시 시스템으로 돌아간다.

3.1 화면 제어기

화면 제어기는 프레임 편집기를 포함하며 저작모드나 실행모드에서 화면에 발생하는 효과를 담당하며, (그림 2)와 같이 저작도구의 툴박스, 메뉴, 대화상자, 메인 윈도우 등의 사용자 인터페이스를 담당한다.

사용자에게 프레임을 보여주는 윈도우로서 배경, 전경 등으로 구분되어 있으며 이에 속하는 모든 객체들(그룹 객체 포함)을 화면에 보여주고, 직접 객체를 그리거나 편집할 수도 있는 윈도우이다. 설정한 윈도우의 크기로 임의의 프레임을 보여주며 객체는 각각 계층번호를 갖고 있어 계층별로 화면에 놓이게 되며 툴박스(Toolbox)에 표시된 화살표 모양을 선택하고 객체를 선택하면 선택된 객체는 선택마커(Selection Marker)를 표시해 준다. 객체의 크기를 조정하거나 위치를 움직일 수도 있다.

(그림 5)와 같이 메인 윈도우는 크게 8가지(화일, 편집, 글자, 객체, 정렬, 다중매체, 창, 도움말 등)의 메뉴로 구성되어 있다.

1) '화일' 메뉴 : 메틱 화일을 처리하기 위한 메뉴로써 메틱 화일을 새로 생성해 주고, 기존의 화일을 열어 주며, 화일을 기존의 이름으로 저장하거나, 새로운 이름으로 저장하게 해주며, 원하는 프레임을 인쇄해 주고, 다양한 프린터를 지원하기 위해 프린터 종류에 따라 그 특성을 설정해주는 기능과 화일을 닫거나 윈도우로 실행권한을 넘기는 기능 등을 제공한다. (새 화일, 열기, 저장, 새이름으로, 인쇄, 프린터설정, 닫기, 종료 등 8가지 메뉴항목 제공)

2) '편집' 메뉴 : 선택된 객체를 화면 상에서 제거하여 버퍼에 넣어두는 '오려두기', 선택된 객체를 화면 상에서도 그대로 두고 버퍼에 복사해 두는 '복사', 버퍼에 저장된 객체를 마우스로 지정한 화면 상의 객체 위치에 복사하는 '붙이기', 선택한 객체를 화면에서 제거하는 '지우기' 등의 메뉴항목들로 구성되어 있다.

3) '글자' 메뉴: 필요한 글꼴(글꼴 종류, 크기 선택 등)을 선택하게 하는 '글꼴', 텍스트 객체의 문단을 정렬해 주는 문단, 핫워드를 보여주거나 가려주는 '핫워드 보이기/가리기'(토글 방식), 핫워드를 만들어주는 핫워드 만들기 등의 메뉴항목을 갖고 있다.

4) '객체' 메뉴: 객체의 속성을 보여주는 '객체', 현 화면에 위치하는 모든 객체의 목록을 보여주는 '객체 목록' 등의 메뉴항목으로 되어 있다. '정렬' 메뉴는 선택한 객체를 한 층 위에 위치시키는 '앞으로', 한 층 아래로 위치시키는 '뒤로', 가장 위에 위치시키는 '맨위로', 가장 아래로 위치시키는 '맨아래로' 등의 메뉴항목으로 되어 있다.

5) '다중매체' 메뉴: 텍스트, 그래픽(선, 박스, 타원, 다중선, 호 등), 정지화상(.BMP., PCX., GIF등), 동화상(. AVI., FLI., FLC, 비디오), 사운드(.WAV), Audio CD 등의 객체별 속성을 편집하는 기능과 보이지 않는 객체인 사운드나 Audio CD의 경우는 아이콘을 생성하는 기능과 이들의 동기화를 편집해 주는 동기화 편집 윈도우를 구동시켜주는 메뉴항목들로 구성되어 있다.

6) '창' 메뉴: 화면 상에 자를 제공해 주는 '눈금자 설정/제거'(토글 방식), 선의 굵기를 바꿀 수 있는 '선종류', 색상(외곽선, 내부채우기)을 바꿀 수 있는 '색상표', 배경 무늬를 제공하기 위해 패턴을 바꿀 수 있는 '패턴표' 등이 각각 토글 방식으로 지원된다.

7) '도움말' 메뉴: 메틱 시스템의 사용법, 도움말, 튜토리얼, 제작팀 소개 등을 제공하는 메뉴이다.

3.2 객체 관리기

객체 관리기에서는 각 객체별로 자료구조를 정의하고 있으며, 메모리 내에서 계층구조를 관리할 수 있도록 설계하였다. 메모리 내에서 객체를 생성하거나 삭제, 검색할 수 있는 함수들은 명령어 처리기 모듈에서 명령어를 구현하기 위해 사용된다. 이 외에도 메시지 처리기 모듈에서 메시지가 발생한 객체를 찾기 위해 검색 함수를 사용하며, 스크립트 해석기의 주 제어기에서 상위 객체로 메시지를 넘기고자 할 때나 스크립트 해석 중에 특정 객체를 참조하고자 할 때도 검색함수를 이용한다. 객체 관리기는 객체의 속성을 액세스하거나 변경할 수 있도록 함수를 제공함으로써 저작자가 객체를 생성하고 변경할 때 명령어 라이브러리를 호출하는데 해당 명령어는 이들 함수를 이용하여 구현된다. 그리고 스크립트 해석 중에도 특정 객체의 속성을 참조하거나 변경하고자 할 때 이들 함수들이 이용되며, 객체를 화면에 그리거나 프린터로 출력하기 위해 필요한 명령어들을 구현할 때도 이용된다.

3.3 메시지 관리기

사용자와 저작자가 발생시키는 모든 메시지, 시스템이 발생시키는 메시지, 객체가 스크립트를 통하여 발생시키는 모든 메시지는 메시지 관리기를 통해 제어가 분기된다. 메시지에 대해 스크립트를 통한 처리가 필요한 경우에 제어를 스크립트 해석기로 넘기면 된다. 물론 이 때 필요한 메시지, 메시지 수신 객체 등의 정보도 함께 넘겨야 한다. 이에 반해 스크립트 해석기의 주 제어기에서 주어진 객체에 메시지 처리기가 없으면 상위 객체로 메시지를 보내야 한다. 이때 메시지 관리기에서 해석기로부터 메시지를 받을 수 있는 함수를 제공하고 있다.

3.4 명령어 라이브러리

저작도구의 모든 기능들은 하나의 명령어로 대체된다. 즉, 메뉴와 툴박스를 통해 저작자가

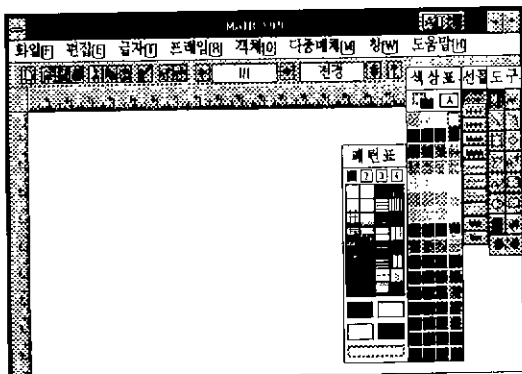


그림 2 메틱 메뉴 화면

코스웨어를 만드는 행위는 명령어 라이브러리의 함수가 실행되어 이루어지는 것이라 할 수 있다. 또한 이런 개별적인 기능들은 대부분 스크립트 명령어를 통해서도 가능하다.

1) Local, System : 시스템 변수 및 사용자 지정 변수를 선언하여 정보의 축적, 조건 분기 등에 이용할 수 있다.

2) Set : 객체의 속성값을 지정하거나 변수값을 지정한다.

변수는 실수, 문자열, 리스트를 저장할 수 있다.

3) Get : 객체의 속성값을 참조할 때 사용하며, Get의 결과는 시스템 변수 it에 저장된다.

4) Ask, AskPassword : Ask는 사용자에게 어떤 질문을 하고, 그 응답을 문자열로 받고자 할 때 사용한다. AskPassword는 문법과 결과는 Ask와 동일하나 응답편집영역에 쓴 문자열이 '*'로 표기되어 입력내용을 다른 사람이 볼 수 없도록 되어 있다.

5) Request : 사용자에게 질문을 하고 그 응답을 버튼으로 받고자 할 때 이용된다.

6) CreateFile, DeleteFile, OpenFile, CloseFile, ReadFile, WriteFile : 화일의 생성, 삭제, 열기, 종료, 읽기 등에 사용된다.

7) ChangeDirectory, MakeDirectory, RemoveDirectory : 디렉토리의 변경, 생성, 삭제에 이용된다.

8) Goto : 프레임 간에 링크를 부여하기 위해 사용된다.

9) MCIstring : 멀티미디어 장비를 제어하기 위해 MS-Windows에서 제공하는 MCI Driver와 API를 이용하여 문자열 형태로 제어를 담당하는 MCISTRING 명령어를 그대로 이용할 수 있게 한 것이다.

10) Beep : PC의 스피커로 Beep 음을 생성한다.

11) Wait : 주어진 시간 만큼 초단위로 기다린다. 화면효과를 일정 시간 동안 보여주기 위해 사용된다.

12) Active : 멀티미디어 링크 대화상자를 통해 프레임 상의 멀티미디어 객체들을 특정 객체에 링크시켜 반복횟수, 지연시간, 동기 조건 등의 환경을 설정해준 후 이 명령어를 실행하면 이 환경값대로 재생한다.

13) Close : 재생되고 있는 멀티미디어 객체들을 중지시킨다.

14) Move : 객체를 원하는 위치로(To) 또는 원하는 거리 만큼(By) 이동시킨다.

15) Follow : 객체를 지정해 놓은 Path에 따라 이동시킴으로써 2차원 애니메이션 효과를 볼 수 있다.

16) Show, Hide : 객체를 보이고, 숨긴다.

17) Resize : 객체를 By 이후에 지정된 크기 만큼 Bounds값을 변경한다.

18) Forward : 자신에게 온 메시지를 상위 객체로 보낸다.

19) Send : 특정 메시지를 특정 객체로 보낸다.

3.5 스크립트 해석기

스크립트 해석기는 주 제어기, 수식 연산기, 명령어 해석기, 변수/형변환 관리기, 어휘 분석기, 함수 라이브러리 등으로 구성되어 있다.

메시지와 함께 제어가 저작도구의 메시지 관리기로부터 스크립트 해석기의 주 제어기로 넘어오면 주 제어기는 해당 메시지의 처리기를 찾아 해석하여 실행한다. 주어진 객체의 스크립트에 처리기가 없으면 계층 구조 상의 상위 객체로 자신의 메시지를 보낸다. 메시지 관리기의 끝을 만나면 제어는 다시 시스템으로 돌아간다. 주 제어기는 한 라인 씩 스크립트를 해석하며, 제어문은 주 제어기에서 처리하고 제어문의 조건 심사를 위해 제어를 수식 연산자로 넘긴다. 명령어일 경우는 제어를 명령어 해석기로 넘긴다. 제어를 받은 수식 연산자는 연산자의 우선순위에 따라 피연산자를 참조하여 계산하며 그 결과를 반환하면서 제어를 호출 모듈로 넘긴다. 참조하는 피연산자에 따라 상수이거나 변수이면 변수/형변환 관리기로 제어를 넘기고, 시스템 제공 함수이면 해당 함수 라이브러리를 호출하게 된다. 연산자의 종류에 따라 피연산자의 자료형을 변환할 필요가 생기는데 이 경우 변수/형변환 관리기의 형변환 함수를 이용한다. 명령어 해석기는 스크립트가 명령어의 문법에 맞는지를 확인한다. 문법을 확인할 때 수식 연산기와 변수/형변환 관리기를 필요로 한다. 변수/형변환 관리기는 변수의 값을 반환하면서 제어를 호출한 모듈로 넘긴다. 어휘 분석기는

해석기 내의 모든 모듈에서 스크립트 해석을 위해 사용된다.

4. MATIC의 구현

4.1 시스템의 구현환경

본 시스템은 IBM PC-486 DX2/66에 SVGA 그래픽보드와 사운드 블라스터, 비디오 블라스터를 장착하였으며, 주메모리 8 MBytes, 하드디스크 500 MBytes로 H/W를 구성하였고, MS-Windows에 마이크로소프트사의 Visual C/C++의 소프트웨어 개발환경을 구현환경으로 하였다.

4.2 자료 구조

4.2.1 객체의 계층구조

메틱은 객체(Objects)가 상단으로부터 학습장(Book), 배경(Background), 프레임(Frame), 객체(Object) 등의 4단계 계층구조(4-Layered Architecture)로 구성되어 있다. 최하단의 객체는 다시 필드, 버튼, 선, 사각형, 둥근 사각형, 타원, 호, 곡선, 폐곡선, 그룹, 핫워드, 상호작용, 이미지, 비디오, 오디오, 오디오 CD, 애니메이션, 오버레이 등의 여러가지 객체로 분류할 수 있다.

한 학습장은 여러 배경과 여러 프레임을 가질 수 있으므로 자료구조상에 학습장 자체의 속성외에도 배경의 수와 각각의 배경을 액세스할 수 있는 포인터들의 리스트를 가리키는 포인터 및 다음 배경 Id를 갖고 있으며, 프레임의 수와 각각의 프레임을 액세스할 수 있는 포인터들의 리스트를 가리키는 포인터 및 다음 프레임 Id를 갖고 있어 현 배경이나 현 프레임들을 액세스할 수 있다.

또한 한 배경은 여러 프레임과 객체들을 가질 수 있으므로 자료구조 상에 배경 자체의 속성외에도 그 배경과 연관된 프레임 수, 객체들의 수와 각각의 객체를 액세스할 수 있는 포인터들의 리스트를 가리키는 포인터 및 다음 객체 Id를 갖고 있어 그 배경에 속한 현 객체들을 액세스할 수 있다. 마찬가지로 한 프레임은 여러 객체들을 가질 수 있으므로 자료구조 상에 프레임 자체의 속성외에도 그 프레임에 속한

객체들의 수와 각각의 객체를 액세스할 수 있는 포인터의 리스트를 가리키는 포인터 및 다음 객체 Id를 갖고 있어 그 프레임에 속한 현 객체들을 액세스할 수 있다.

각 객체들은 학습장으로부터 하위 객체에 이르기까지 모두 각각의 자료구조 내에 고유이름(Unique Name)을 부(sub) 자료구조로 갖고 있어 객체의 종류, 객체가 배경에 속한 것인지 프레임에 속한 것인지, 객체 Id(학습장, 배경, 프레임, 객체 각각의 오프셋 값)등을 통해 자기 개별적인 객체로서의 독립성을 유지한다. 또한 모든 객체들은 스크립트 명령어들과 제어구조로 구성된 스크립트 언어를 기술한 스크립트 포인터를 가질 수 있게 되어 있어 필요에 따라 객체 자체가 갖는 속성만으로 나타낼 수 없는 기능을 스크립트로 기술하여 자신의 속성을 확장하는 개념으로 사용할 수 있다. 메틱 시스템의 스크립트는 인터프리터 형식으로 되어 있어 실행 모드에서 발생하는 각종 메시지에 따라 해당 메시지 핸들러를 호출하여 실행하게 되는 데 현 객체에 해당 메시지 핸들러가 없으면 상위 객체로 메시지를 전달하는 형태로 되어 있다.

이 밖에도 학습장을 제외한 객체들은 각기 멀티미디어 객체들(텍스트, 그래픽, 정지화상, 동화상, 오디오, 오디오 CD 등)을 링크시킬 수 있도록 자료구조 내에 부 구조로 멀티미디어 링크(MMLINK)를 가질 수 있으며, 객체와 프레임은 각각 화면에 표시될 때 객체이동이나 그래픽 효과를 줄 수 있다.

객체 여러 개를 묶어 한 객체인 것 처럼 사용하기 위해 그룹(Group)이라는 특수 객체를 만들어 이 그룹 객체의 자료구조에는 포함된 객체의 수와 포함된 객체들을 액세스할 수 있는 포인터 리스트를 가리키는 포인터를 부 자료구조로 갖는다. 툴박스의 화살표를 누른 후 여러 객체를 선택한 경우에는 이들을 관리하기 위한 객체 선택 리스트 자료구조도 있어 선택된 객체의 수와 선택된 객체들을 액세스할 수 있는 포인터 리스트를 가리키는 포인터를 갖고 있다.

각 객체별 자료구조 외에도 여러 전역변수(Global Variables)를 두어 현재의 학습장, 현재의 배경, 현재의 프레임, 현재의 객체, 현재의 윈도우를 참조할 수 있게 하였으며, undo했을

때와 오려두기, 복사, 붙이기 등의 메뉴 명령어 실행시의 관리를 위하여 객체의 수, 객체의 포인터 리스트를 가리키는 포인터 등을 선언하였다. 메모리 비트맵(Memory Bitmap)을 관리하기 위한 핸들 HBITMAP과 폰트 정보를 관리하기 위한 LOGFONT를 두었다.

4.2.2 객체별 속성

학습장 객체는 표제 문장, 표제를 보일 것인지 가릴 것인지, 학습장 내용이 바뀌었는지, 학습장의 화면 크기, 학습장 Id, 화일 이름(Full Path 포함) 등을 그 속성으로 갖는다.

배경 객체의 속성은 배경에 채워질 색상, 배경의 Id와 이름, 그룹 수 등이며, 프레임 객체의 속성은 프레임의 Id와 이름, 그룹 수 등이다.

객체의 속성은 객체의 가장자리 두 좌표값(좌상단, 우하단), 팔레트에서 채울 색상과 선 색상 값, 선의 종류(두께 및 실선/점선 등), 객체의 Id와 이름, 선택되었는지 여부, 색상을 채우는 모드(투명/불투명), 객체를 현 위치에 고정시킬지 여부, 객체의 정점 수와 그 좌표값, 객체를 화면에 보일 것인지 숨길 것인지 여부, 그룹의 Id와 그 Level, 현재의 윈도우 등이다.

객체 자료구조중 pMultimedia 필드에 링크되어 사용할 이미지, 비디오, 애니메이션, 오디오, 오디오 CD, 버튼 객체의 자료구조를 살펴본다. 이미지 객체는 이미지 화일의 이름, 화일 종류(.BMP, .WMF, .DIB, .PCX, .GIF 등), 비트맵의 핸들, 팔레트 핸들, 이미지의 원래 크기의 좌표값(좌상단, 우하단)를 구조로 갖고 있으며, 비디오 객체는 비디오 화일 이름, 시작 프레임, 끝 프레임, 프리뷰 오디오, 프리뷰 비디오, 비트맵의 핸들과 팔레트의 핸들을 구조로 갖는다. 애니메이션 객체는 애니메이션 화일 이름, 시작 프레임, 끝 프레임, 비트맵의 핸들과 팔레트의 핸들을 구조로 갖는다. 오디오 객체는 화일 이름을 구조로 가지며, 오디오 CD는 시작 시간과 끝 시간을 구조로 갖는다. 버튼은 버튼에 쓰여지는 글씨(Label Name), 버튼 종류(None, Rectangle, Rounded, Shadowed, Checkbox, Radio, Pushbutton), 사용 폰트(LogFont), 연결프레임 번호(LinkTo), 색상, 확인 여부(Checked) 등을 자료구조로 갖는다.

멀티미디어 링크 객체는 객체 Id, 반복 횟수, 지연시간 등의 속성을 갖는다.

객체의 자료구조는 MMLINK라는 부 자료구조를 갖는데, 여기에는 링크시킬 멀티미디어 링크 객체의 수, 객체 Id의 포인터, 동기의 종류(동시에, 연속적으로), 링크시킬 전체 객체에 대한 반복횟수, 지연시간, 스크립트로 직접 제어할 것인지 동기화 대화상자로 제어할 것인지 여부, 메시지 번호 등의 내용을 선언한다.

이 밖에도 객체의 자료구조는 pInteraction 필드에 링크되어 사용할 상호작용 객체를 부구조로 갖으며(그림 3 참조), 상호작용의 종류(텍스트, 버튼, 이동형, 키입력형, 메뉴선택형), 객체의 영역, 정답, 정답 버튼 번호, 버튼의 갯수, 문제 풀이시 허용되는 최대시간, 최대 시도 횟수, 사용자가 사용할 수 있는 정답의 최대길이, 대소문자 구분 여부, 단어와 단어 사이의 공백(Space) 무시 여부, 판정 여부, 텍스트 상호작용의 포인터 리스트를 가리키는 포인터, 버튼 상호작용의 포인터 리스트를 가리키는 포인터, 이동형 상호작용의 포인터 리스트를 가리키는 포인터, 시도횟수, 시도기회 종료 여부, 사용 폰트와 그 색상 등이 선언되어 있다. 상호작용 객체는 그 종류에 따라 다른 속성을 가지므로 종류별 속성을 살펴보면, 텍스트 상호작용은 정답, 첫 시도시 틀린 경우의 메시지 내용, 정답 입력시 분기할 프레임 번호, 멀티미디어 링크, 입력에 대한 응답으로 메시지를 줄 것인지 여부, 프레임 분기를 할 것인지 여부, 멀티미디어로 응답할 것인지 여부 등을 속성으로 가지며, 버튼 상호작용은 버튼에 쓰여질 글씨와 그 색상, 버튼 모양, 버튼이 차지하는 영역, 선택 여부, 첫 시도시 메시지 내용, 첫 시도시 분기할 프레임 번호, 멀티미디어 링크, 입력에 대한 응답으로 메시지를 줄 것인지 여부, 프레임 분기를 할 것인지 여부, 멀티미디어로 응답할 것인지 여부 등을 속성으로 갖고, 이동형 상호작용은 버튼 텍스트, 버튼의 영역, 선택 여부, 목표 객체 번호, 첫 시도시 메시지 내용, 첫 시도시 분기할 프레임 번호, 멀티미디어 링크, 입력에 대한 응답으로 메시지를 줄 것인지 여부, 프레임 분기를 할 것인지 여부, 멀티미디어로 응답할 것인지 여부 등을 갖고 있다.

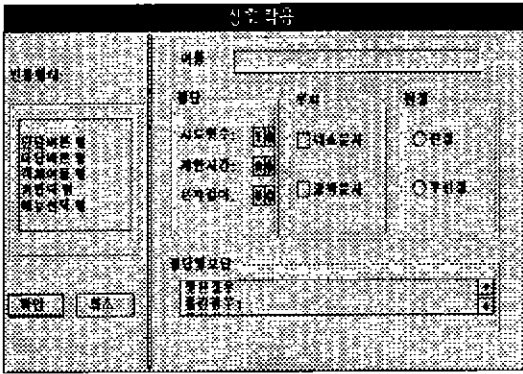


그림 3 상호작용 객체 화면

4.3 주요 함수

1) 메모리 내에서의 객체 관리 함수 : 객체의 생성, 삭제 함수 고유 이름 (Unique Name)에 의해 학습장을 찾는 함수나 오프셋에 의해 프레임을 찾는 함수, 커서 아래 있는 객체를 찾는 함수, 학습장의 생성 함수, 프레임의 생성 함수 등을 갖고 있다. 주어진 학습장 포인터 리스트의 각 학습장에 할당되어 있는 포인터 속성과 그 자신 (List pointer)을 소거하는 함수, 주어진 포인터에 해당하는 학습장의 여러 포인터 속성에 할당되어 있는 메모리를 소거하는 함수, 주어진 배경 포인터 리스트의 각 배경 포인터를 소거하고 그 자신도 소거하는 함수 등과 여러 배경, 특정 배경, 여러 프레임, 특정 프레임, 여러 객체, 특정 객체를 자기 소거해 주는 함수들이 만들어져 있으며, 학습장, 배경, 프레임, 객체를 생성하고, 그 포인터를 포인터 리스트에 추가하는 함수와 멀티미디어 속성, 고유이름, 구 객체 리스트 등을 생성하는 함수 등이 개발되었다.

2) 객체 속성 참조 및 변경 함수 : 스크립트 해석 중에 특정 객체의 속성을 참조하거나 변경할 때 이용할 수 있는 함수들이다. 주어진 고유이름에 해당하는 객체를 찾아 그 스크립트를 반환하는 함수, 주어진 학습장과 배경, 프레임 및 객체의 속성을 설정하거나 참조하는 함수들과 텍스트를 가져오거나 화면에 표시해 주는 함수, 핫위드를 생성, 삭제, 참조할 수 있는 함수들과 객체의 계층관리를 위해 객체를 화면의 가장 앞쪽으로, 가장 뒷쪽으로, 한 층 앞쪽으로,

한 층 뒷쪽으로 객체의 층 (Layer)을 옮기는 함수들도 개발되었다. 기타 객체를 관리하기 위한 함수로 주어진 객체, 프레임, 배경 등의 Parent 오프셋을 얻는 함수, 선택을 초기화하는 함수, 선택마크를 특정 객체나 모든 객체에 표시해주는 함수, 배경이나 프레임에서 선택한 객체들의 속성을 바꾸는 함수, 선택된 것의 리스트를 바꾸거나 추가 삭제하는 함수, 선택된 객체를 삭제하는 함수, 선택된 객체를 메모리에 복사해 두되 선택된 객체는 그대로 두거나 지우는 경우의 함수들, 메모리에 복사된 객체나 그룹 객체를 화면의 지정위치에 표시하는 함수, 그룹/언그룹 함수 등 외에도 많은 함수들이 개발되었는데 일반적인 데이터의 검색 프로그램에서 요구되는 기본 함수들인 찾기 함수가 Id, 고유이름, 이름 등에 의해 찾을 수 있도록 객체, 프레임, 배경, 학습장별로 각기 준비되어 있다. 현재 화면에 그리고 있는 객체들의 리스트 포인터를 찾거나 처음 객체, 다음 객체를 찾는 함수들과 현재 객체를 찾거나 목표 객체, 객체 자신의 고유이름을 얻는 함수들이 있으며, 배경, 프레임, 리스트 상의 객체들, 객체, 버튼, 종류별 버튼, 선, 화살표, 사각형, 타원, 호, 텍스트, WMF, 이미지, 비디오, 투명 비트맵 등을 표시하거나 오디오를 재생하는 함수 등이 객체 관리를 위한 기타 함수로 개발되었다. 보다 상세한 함수 내용은 생략하기로 한다.

3) 화면 제어 함수 : 객체를 화면에 그리거나 프린터로 출력하기 위해 필요한 명령어들을 구현할 때 이용되며, 프레임에 들어가고 나갈때 객체와 프레임의 화면 효과를 구현할 때 이용된다. 여러 종류의 툴바 (눈금자, 선택박스, 색상표, 패턴표, 툴박스 등)를 관리하는 함수 등도 포함하고 있다.

4) 화일 입출력 함수 : 생성된 코스웨어를 화일에 저장, 열기, 새이름으로 저장하기 등을 해주는 API이다. 학습장, 배경, 프레임, 객체 리스트, 객체, 스크립트, 멀티미디어, 상호작용, 텍스트 상호작용, 버튼 상호작용, 이동형 상호작용 등을 저장하거나 로드 (Load) 하는 함수와 학습장의 열기, 생성, 종료하는 함수들로 구성된다.

5) 하이퍼미디어 함수 : 특정 프레임으로 이동하기 위한 함수와 핫위드를 관리하기 위한

함수이다. 앞 프레임으로 이동하는 함수, 다음 프레임으로, 첫번째 프레임으로, 마지막 프레임으로, 오프셋이나 객체의 고유이름 등을 이용하여 프레임으로 이동하는 함수들과 특정 프레임의 오프셋을 구하는 함수 등이 개발되었다.

6) 멀티미디어 링크 관리 함수 : 각종 멀티미디어 객체들에 대해 이들을 특정 객체에 링크로 묶어 관리하는 함수로써 메텍의 프레임 편집기와 명령어 라이브러리에서 사용된다.

각 미디어 객체는 MMLINK 구조의 pObjectId를 시작점으로 하여 배열 형태로 연결되어 있어 이벤트가 발생하면 객체 관리자가 이들을 하나씩 활성화 한다. 멀티미디어 링크의 활성화는 CMD_ActiveMultiLink() 함수를 통해 시작되며, 동기화 방식이 동시 실행인 경우에는 CMD_ActiveMultiLinkConcur()를 연속 실행인 경우에는 CMD_ActiveMultiLinkSerial()을 호출한다. 호출된 함수는 멀티미디어 링크에 연결된 미디어 객체를 하나씩 읽어 들여 해당 수행함수를 부르게 된다.

7) 메텍 메시지 관리기 : 스크립트 해석기의 주 제어기에서 주어진 객체의 스크립트에 해당 메시지 핸들러가 없을 때 상위 객체로 메시지를 보내야 하는데 이에 필요한 메시지를 분배하거나 Forward, Send하는 함수들이 API로 준비되어 있다. 반환값으로는 메시지를 못받은 경우, 주어진 객체나 그 Parent가 발견되지 않은 경우, 메시지가 처리된 경우, 메시지가 메텍 시스템으로 보내진 경우 등의 상태를 표시하는 숫자가 된다.

8) 메텍 명령어 라이브러리 : 스크립트의 명령어는 명령어 해석기 모듈에서 그 해석이 성공적으로 종료했을 경우만 명령어 라이브러리의 제공 함수를 호출하게 되므로 제공 함수에 대해서만 명령어 해석 모듈을 작성하여 상기한 명령어들을 지원한다.

9) 스크립트 편집 관리 함수 : 각 객체의 속성 대화상자의 스크립트 버튼으로부터 구동되는 스크립트 편집기의 함수들, 편집 클래스 윈도우의 편집 버퍼를 설정하는 함수, 편집 버퍼에 있는 스크립트를 현재 객체의 스크립트로 갱신할지 안할지를 묻는 함수와 실제로 갱신하는 함수, 편집 버퍼를 목적 버퍼로 복사하는 함수, 활

성화되어 있는 윈도우 내에서 문장을 찾거나 비교, 변경하는 함수들, 스크립트 편집기 윈도우의 메시지 핸들러, 디폴트 편집 클래스 윈도우의 메시지를 가로채는 함수, 편집 클래스의 탭사이즈 변경함수, CD FindText(), CD ReplaceText() 프로시저어를 hook하는 함수, 여러가지 메시지를 출력하는 함수들이 있다. (그림 4 참조)

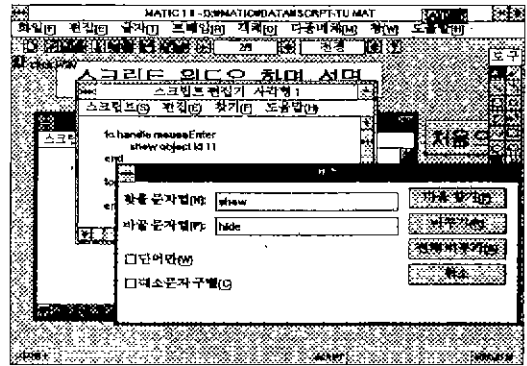


그림 4 스크립트 편집 에

10) 스크립트 해석기 : 스크립트 해석기를 세 부적으로 보면 (그림 5)와 같이 주 제어기, 수식 연산기, 명령어 해석기, 변수/형변환 관리기, 어휘 분석기 등으로 구성되어 있다.

- 주 제어기 : 메시지의 창구로 제어문을 관리한다.

- 수식연산기 : 수식을 계산하여 그 값을 반환한다

- 명령어 해석기 : 명령어 해석 및 라이브러리 함수를 호출한다.

- 변수/형변환 관리기 : 변수 관리 및 수식 연산시 형변환 관리를 수행한다.

- 어휘분석기 : 스크립트를 토큰단위로 구분한다.

어휘 분석기는 스크립트 문장을 토큰으로 나누어 관리하며 lookupAllEntry() 함수는 주어진 문자열을 토큰 테이블과 키워드 테이블에서 찾아 있으면 해당 토큰의 번호를, 없으면 TN_IDE-NTIFIER를 반환한다. lookupKeywordEntry()는 두번째 인자로 주어진 키워드 종류에서 주어진 문자열을 검색한다. 반환값은 있으면 해

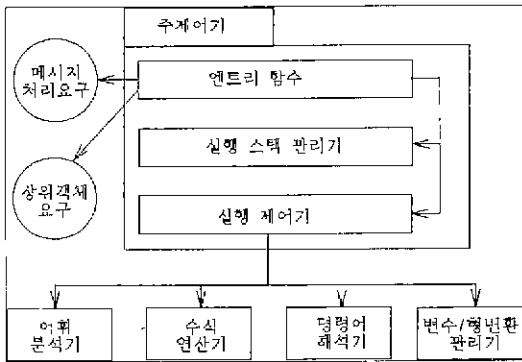


그림 5 스크립트 해석기의 내부 모듈

당 키워드 번호를, 없으면 TN_IDENTIFIER를 반환한다. 주 제어기는 메틱 모듈에서 제어를 넘겨 받는 스크립트 해석기의 창구가 되는 모듈로써 해석기의 전체 제어를 담당한다. 엔트리 함수, 실행시 스택 관리기, 실행 제어기로 구성된다. 명령어일 경우 제어를 명령어 해석기로 넘긴다. 엔트리 함수는 저작도구의 메시지 관리기로부터 객체, 메시지 및 메시지 인자를 받아서 객체의 스크립트에서 메시지 관리를 찾아 제어를 실행 제어기로 넘긴다. 해당 메시지 관리기가 주어진 객체에 없으면 객체 관리를 통해 계층구조 상의 상위 객체를 찾아서 다시 메시지 관리기로 제어를 넘긴다. 엔트리 함수가 실행 제어기로 제어를 넘기기 전에 실행 스택 관리를 통해 현재의 객체와 스크립트, 스크립트 내의 현재 실행 위치 등을 스택에 저장한다. 실행제어기는 메시지 처리기의 내부에 정의된 스크립트를 해석하여 그 종류에 따라 제어를 분기시킨다. 제어문이 아닌 경우는 명령어 해석기 모듈을 호출한다. 또한 제어문 해석 중 발생하는 여러 상태를 시스템 변수에 지정하고 참조하기 위해 변수/형 관리기를 이용한다. 수식 연산기는 제어문의 조건 검사나 set 명령어의 인자값을 계산할 때 호출되며, 그 엔트리 함수는 GetExpression()이다. 수식연산에 사용되는 연산자의 종류와 우선순위에 따라 차례로 호출되는데 마지막으로 호출되는 Primitive() 함수는 수식에서 더 이상 계산이 불필요한 경우 실수, 문자열, 변수, 상수 그리고 함수의 값을 반환한다. 변수/형변환 관리기는 다른 모듈

에서 변수를 등록할 수 있도록 하는 일과 변수의 이름으로 변수값을 참조할 수 있도록 하는 일 그리고 변수 간에 자료 형을 변환하는 일을 담당한다. 변수/형변환 관리기에서 제공하는 대표적인 API는 변수 테이블 관리를 위한 함수, 리스트 자료 관리를 위한 함수, 자료 형변환을 위한 함수, 시스템 변수 관리를 위한 함수 등이 있다. 스크립트 해석기의 주 제어기에서 분석한 문장이 제어문이 아닌 경우 명령어 해석기 모듈의 _Command.Proc() 함수를 호출하며, 명령어를 해석하기 위해서는 구문분석기(Parser)가 필요하여 본 연구에서는 명령어의 문법을 해석하기 위해서 UNIX의 YACC 중 일부 기능을 구현하였다. 명령어의 문법을 입력 받으면 해석할 수 있는 모듈을 생성한다. 생성된 모듈의 해석이 성공적으로 종료되었을 때 실행되는 부분을 명령어에 상응하는 라이브러리 함수로 대체하면 된다.

5. 향후 연구방향

메틱에 대한 향후 보완사항은 크게 다음과 같이 분류될 수 있다.

1) 메틱을 비롯하여 지금까지 개발된 저작도구의 교수설계 지원기능은 개발 보조기능에 비해 매우 빈약한 실정이다. 그러나 코스웨어의 질적향상을 위해서 교수설계 지원기능의 제공이 필수적이므로, 거시적 미시적 교수설계전략 구성요소들을 종합해 하나의 틀을 마련하고 더 나아가서는 교수설계전문가시스템을 개발하여 메틱에 접목시키는 방안을 모색해 갈 것이다.

2) 컴퓨터 그래픽을 비롯한 멀티미디어 지원기능을 보강하여 보다 현실감 넘치는 정보를 제공함으로써 학습전이 효과를 증진시키고, 시뮬레이션 개발 보조기능을 더욱 다양화하여 문제해결능력 및 사고력 증진과 같은 고차원적 학습능력을 신장시키는데 공헌할 수 있어야 한다. Edutainment 실현을 위하여 각종 게임기법을 지원하는 방안 또한 연구되어야 할 것이다.

3) 상호작용 개발지원 및 학습관리기능을 보강하여 더욱 의미있고 다양하며 개별화된 상호작용 및 정보제공의 기반을 제공할 수 있어야

할 것이다.

4) 사용자 인터페이스를 소프트웨어 및 하드웨어 측면에서 보강하여 코스웨어 저작자는 물론 학습자들에게 보다 친밀감을 부여할 수 있는 환경을 제공하여야 할 것이다. 또한 저작기능 확장에 필수적인 스크립트 언어사용의 어려움을 경감시키고자 스크립트 문법을 검사하는 기능과 실행중 발생하는 논리적 오류를 찾아주는 디버깅 기능을 제공하고, 스크립트를 작성시 스크립트 문법과 생성된 객체를 참조할 수 있는 객체 브라우저(Object Browser)를 개발 제공하여야겠다.

5) 통신망을 활용한 다중 사용자 및 다중 저작자의 공동작업 지원방안을 연구하여 저작도구에 통합시켜야 할 것이다. 더욱이 국가적인 차원에서 수행되고 있는 초고속통신망 구축 사업을 고려해 볼 때 저작도구의 통신망 활용 지원기능은 더욱 큰 의미를 지닐 수 있을 것이다.

위에 제시된 사항은 메텍 뿐만 아니라 기 개발된 타 저작도구 및 앞으로 새로운 저작도구 개발에 지침이 될 수 있을 것이다.

6. 결 론

저작도구 메텍은 MS-Windows상에서 객체지향기법을 기반으로 책 방식 모델을 적용하여 설계 개발되어 그 사용이 용이하고, 스크립트 언어를 제공하여 저작의 융통성 및 확장성을 극대화하였다. 사용자와 시스템간의 상호작용성을 강조하여 다양한 질문/응답 유형의 사용자 입력 및 피드백을 손쉽게 처리할 수 있고, 시스템 변수 및 함수를 사용하여 상호작용 관련 데이터를 관리할 수 있도록 하였다. 멀티미디어 데이터를 객체로 관리하며 이를 다시 다른 객체에 링크시킬 수 있는 기능과 매체별 동기화 기능을 제공하였고, 사용자 인터페이스 면에서도 가능한 한 메뉴, 아이콘, 대화상자를 활용할 수 있도록 하였다.

향후 메텍은 교수설계기능, 학습관리기능, 시뮬레이션 및 게임 개발지원 기능 및 통신망 활용지원 기능등을 보강하여 초고속정보화시대의 저작도구로 발전시켜 나갈 계획이다.

참고문헌

- [1] ACM, "Hypertext '91, Third ACM Conference on Hypertext Proceedings", ACM, 1989.
- [2] Hardman, L., Bulterman, D. C. A. and van Rossum G. "The Amsterdam Hypermedia model" Commun. of the ACM, 37, 2 (1994)
- [3] Conklin, J. "Hypertext : A survey and introduction" IEEE Computer, 20, 9 (1987)
- [4] John Waterworth, "Multimedia technology and applications", Ellis Horwood Limited, 1991.
- [5] Nigel Woodhead, "Hypertext and Hypermedia, Theory and Applications", Sigma Press, 1991.
- [6] Preece, Keller, "Human-Computer Interaction", Prentice-Hall, 1990.
- [7] Shneiderman, "Designing the user interface", Addison-Wesley, 1987.
- [8] MS Windows S/W Development Kit, Vol4 : Resources, Microsoft, p107-p115, 1992.

심 부 성



- 1981 한국항공대 항공전자공학과 학사
- 1989 중앙대 국제경영대학원 경영정보학과 석사
- 1981. 8~1986. 2 시스템공학연구소 연구원
- 1986. 3~1991. 2 교통개발연구원 책임연구원
- 1991. 3~현재 시스템공학연구소 선임연구원
- 관심분야 : 멀티미디어 저작도구, 모델링 및 시뮬레이션, 원격교육



김 상 녕

1985 아주대 전산과 학사
1987 동국대 전산과 석사
1991~1995 시스템공학연구소
연구원
관심분야: 멀티미디어 테이타베
이스, 저작도구, 시물
레이션



이 익 현

1984. 2 한국항공대 통신과 학사
1986. 2 한국항공대 전자과 대학
원 통신공학 석사
1989. 8~1995. 6 시스템 공학연
구소 선임연구원
관심분야 원적통신 교육
저작도구

● 정보과학회 영문 논문지 논문모집 ●

- 제출기한: 창간호 - 1995년 7월 31일(월)
- 제 출: 한국정보과학회 사무국 한영진
- 주 소: 서울특별시 서초구 방배3동 984-1(머리재빌딩 401호)
☎ 137-063 T. 02-588-9246 F. 02-521-1352