

논문-98-3-1-02

스크립트 해석기를 기반으로 하는 멀티미디어 저작도구의 개발

차 현 성*, 한 광 록*

Development of a Multimedia Authoring Tool Based on Script Interpreter

Hyun-Sung Cha* and Kwang-Rok Han*

요 약

멀티미디어 저작도구의 목적은 기존의 프로그래밍 도구보다 빠른 시간과 적은 비용으로 응용 프로그램을 쉽게 개발하는 것이다. 따라서 본 논문에서는 사용자가 멀티미디어 데이터들을 쉽고 정확하게 조작하여 고품질의 CD 타이틀이나 프리젠테이션물을 저작할 수 있도록 하는 스크립트 기능을 갖는 이벤트 구동 방식의 저작도구 개발에 관하여 논한다.

본 논문의 저작도구는 다양한 속성을 지닌 객체를 포함하고 있으며, 여기서 가장 중요시 되는 속성은 스크립트 속성이다. 스크립트는 객체에 대한 보다 복잡한 행동을 정의 할 수 있고 또한 사용자가 의도하는 행위를 객체에 지정 할 수 있다.

Abstract

The multimedia authoring tool is used to develop an application program easily within faster time and lower expenses than conventional program languages. In this paper, we describe the development of event-driven multimedia authoring tool based on script function in order that user manipulates the multimedia data easily and concisely and authors high quality CD title and presentation.

The authoring tool mentioned in this paper includes many objects that have various attributes and the most significant attribute of them is the script property. User can define the various actions of an object and specify some intended ones of them by the script.

I. 서 론

1990년대 이후 하드웨어 기술의 급속한 발전으로 대용량 저장 매체들이 등장하였기 때문에 멀티미디어 데이터의 실시간 처리도 가능하게 되었다. 이러한 영향으로 멀티미디어 데이터들을 프리젠테이션하거나 CD 타이틀 제작을 위한 편집 작업을 할 수 있는 저작도구에 대한 연구

및 개발이 활발해지면서 많은 저작도구들이 등장하게 되었다. 멀티미디어 저작도구의 특징은 기존의 프로그래밍 도구보다 빠른 시간과 적은 비용으로 응용 프로그램을 개발할 수 있다는 것이다. 또한 특별한 전문 지식 없이도 손쉽게 프로그래밍 할 수 있기 때문에 점진적으로 그 시장을 넓혀가고 있는 실정이다.^[1]

초기 멀티미디어 저작물의 형태는 단순히 멀티미디어 자원을 배열하는 수준 정도였다. 그러나 현재 멀티미디어 저작물이 교육, 의료, 사업 등 그 이용 분야가 확대되고 있지만 이러한 단순한 저작 형태만으로는 많은 한계를 드러내게 되었다. 따라서 본 논문에서는 이러한 문제를 해결하고 동적인 멀티미디어 저작을 위하여 다양한 멀티미디어

* 호서대학교 컴퓨터공학과
Dept. of Computer Engineering, Hoseo University

※ 본 논문은 97년도 호서대학교 교내연구비 지원하에 수행되었음.

어 객체들을 설계하고, 각 객체마다 고유한 속성을 부여하여 객체가 다양한 행동 변화를 할 수 있게 하였다. 또한 객체의 속성에 스크립트 속성을 갖도록 하여 스크립트로서 기본적인 이벤트에 대한 정의뿐만 아니라 객체에 대한 특별한 행동을 구현하고 저작 시스템 전체를 동적으로 변화시킬 수 있도록 하였다. 이를 위하여 스크립트 언어를 프로그래밍 언어 수준으로 개발하여 이 스크립트로서 객체 지향적인 멀티미디어 관리를 하고 각 객체가 갖는 속성 및 동작을 기술하고, 이벤트에 반응하여 멀티미디어 저작을 실행할 수 있는 저작도구를 설계하고 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 저작 도구들을 분류 및 그 기능에 대하여 논하고 3장에서는 본 논문에서 구현하려고 하는 저작도구에 대한 전체적인 구조 및 객체들을 설계와 구현에 대하여 기술한다. 4장에서는 저작을 위한 스크립트 해석기를 설계하고 저작물의 실행과정을 기술하고, 마지막으로 5장에서는 결론 및 향후 계획에 대하여 서술한다.

II. 멀티미디어 저작도구의 분류 및 기능

1. 멀티미디어 저작도구의 분류

기존의 저작도구들은 그 저작 방법 및 프로그래밍 방법에 따라서 흐름도(flowchart), 책(book & page), 시간선(time-line)방식 등으로 구분된다.¹¹²⁾ 또한 멀티미디어 저작도구는 저작 기반 도구 사용에 따라 페이지 혹은 스크립트 저작도구와 아이콘 기반 저작도구, 시간 기반 저작도구로 구분하거나 여기에 카드 기반 저작도구를 추가하기도 한다. 또한, 프로그래밍 방법에 따라 선형 구조, 계층 구조, 네트워크 구조 및 복합 구조 등 네 가지 형태로 세분

화된다.¹²⁾ (그림 1)은 저작도구들을 분류한 것이다.

2. 멀티미디어 저작도구의 기능

멀티미디어 저작도구에서 요구되는 기능을 분류하면 다음과 같다.¹¹¹¹²⁾

- ① 각 미디어 자체(intramedia)나 미디어 장치간(intermedia) 연결 기능
- ② 사용자 입력에 관한 미디어 제어기능
- ③ 사용자 입력에 관한 미디어 재생기능
- ④ 미디어 파일의 동시 실행 기능

사용자 입력에 관한 미디어 제어 및 재생 기능은 사용자가 임의로 정의한 스크립트에 의한 처리와 저작도구에 정의되어 있는 모든 이벤트에 해당되는 기본적인 행위에 의하여 처리된다.¹⁶⁾ 전자의 경우 사용자가 스크립트 에디터를 호출함으로써 해당 미디어에 관한 기능을 정의할 수 있고, 후자의 경우 저작도구에 기본적으로 정의된 기능을 수행하는 것이다. 미디어 파일의 동시 실행 기능은 저작도구에서 가장 중요시되어야 할 기능이라 할 수 있다. 동시 실행 중에 동기화가 어긋나게 되면 음성과 모양이 일치하지 않는 어색한 상황이 발생되게 된다. 따라서 저작도구를 구현하기 위해서는 위와 같은 미디어에 관한 기본적인 네 가지 기능을 통합한 멀티미디어의 다양한 기능이 요구된다.

3. 기존 저작도구에 대한 연구 및 비교

본 저작도구의 저작구조는 다수의 객체가 모여 하나의 페이지를 형성하고 다수의 페이지가 모여 하나의 책을 형성하는 구조의 저작형태와 이벤트 구동형식으로 실행되는 형식을 가지고 있다. 이러한 구조는 대표적으로 기존의 저

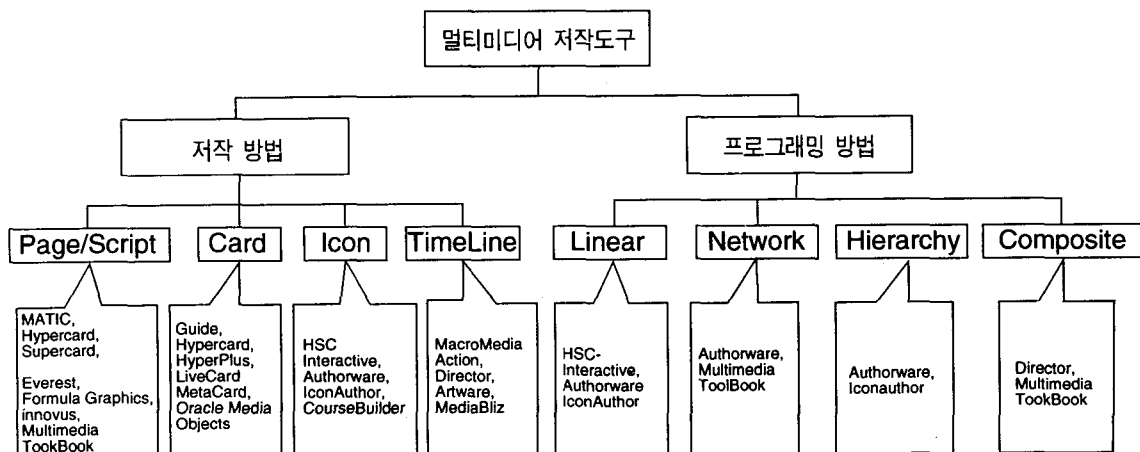


그림 1. 저작도구의 분류
Fig. 1. Hierarchy of authoring tools

작도구인 어시메트릭스 사의 멀티미디어 툴북(Multimedia ToolBook, Asymetrix Co.)에서 찾아볼 수 있다. 본 절에서는 기존에 국내외에서 개발된 대표적인 저작도구를 사용 환경, 저작 방식 및 기타 객체의 특징에 의하여 각 기능별로 분류한다. 다음 (표 1)은 기존의 저작도구와 본 논문의 저작도구를 기능별로 분석하여 비교한 것이다.

본 논문에서 구현하는 저작도구는 윈도우즈용 멀티미디어 저작도구로서 기존의 저작도구에서 지원되는 여러 기능들을 최대한 수용하는 것을 목표로 하고 있다. 고급 수준의 사용자를 고려하여 구조적 스크립트 프로그램을 가능하게 하였고 초보적 수준의 사용자를 위하여 기본적인 이벤트에 대한 처리를 GUI형태로 제공하여 보다 쉽게 저작을 할 수 있는 환경을 구현하였다.

III. 개발 시스템 구조

1. 시스템 개요

본 논문에서 구현하는 저작도구의 형태는 책(book & page)구조의 스크립트 저작도구로서 각 객체들의 실행은 사용자 및 시스템에 의한 메시지를 기반으로 한다. 일반적인 책 구조 저작도구의 특징은 객체 지향적 프로그래밍을 제공하는 데에 있다.^[12] 실행을 위하여 책 내부의 각 객체들은 이벤트를 필요로 하며, 각 객체는 스크립트 및 기본적인 행동을 속성으로 가지고 있고 발생한 이벤트와 비교하여 실행시킨다. 이러한 구조의 저작도구는 주로 교육용 타이틀, 종합 안내 시스템인 키오스크(KIOSK), 게임, 데이터베이스 전위시스템, 오락, 그리고 교육과 오락을 결합한 에듀테인먼트(edutainment)등의 응용 프로그램 개발이 용이하다.

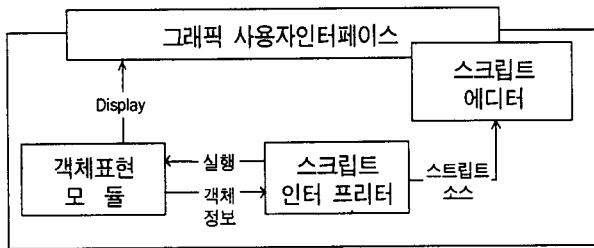


그림 2 저작 시스템
Fig. 2 Authoring system

본 저작시스템은 (그림 2)과 같이 세 부분으로 구분되어진다. 그래픽 사용자 인터페이스(GUI)는 저작 단계에서 객체를 배치하고 실행시 저작물을 보여주거나 작성된 저작물을 수정한다. 스크립트 해석기는 사용자에게 의하여 입력된 스크립트 소스(script source)를 분석하여 테이블을 생성하고 실행을 위해 해당 객체에 분석된 테이블을 저장

한다. 객체 표현 모듈은 객체에 대한 정보를 클래스형태로 저장하고 있으며 각 멤버 변수는 각 객체의 속성에 해당한다. 또한 스크립트에서 정의한 명령을 수행하고 각 객체를 GUI에 출력하는 역할을 한다.

2 객체 설계

저작도구에서 사용되는 텍스트, 그래픽, 애니메이션, 오디오 및 동영상 등의 모노미디어(monomedia)들은 상호작용(interactive)기능과 대량의 데이터를 보관, 전송, 표현기능, 실시간 처리 등을 기반으로 하여 데이터에 대한 통합성, 공존성, 독립성, 상호작용성의 특징을 갖도록 설계하고, 특히 북과 페이지의 계층구조 형태를 도입하였다.^{[6][3][12]} 본 시스템에서 설계한 객체들은 (그림 3)과 같은 계층구조를 갖는다.

1) 북

북(book)은 본 시스템의 가장 최상위 객체로서 개발하려고 하는 멀티미디어 타이틀의 기본 성질이나 속성을 포함하고 있으며 다수의 페이지들을 소유하고 있다. 즉 북은 모든 페이지의 가장 상위 계층으로 그 대표적인 기능으로는 관리 기능이라고 할 수 있으며 구체적으로 다음과 같은 기능을 하도록 설계하였다.^[6]

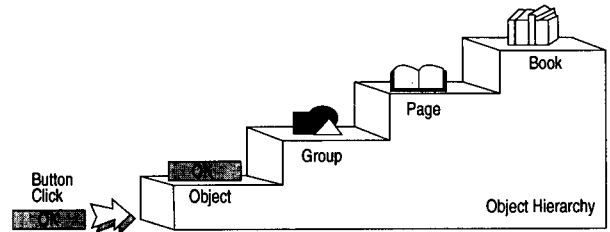


그림 3 객체 구성도
Fig. 3. The object construction diagram

① 이벤트 처리 기능

사용자 이벤트 및 시스템 자체에서 발생하는 이벤트를 감지하여 해당 페이지로 메시지를 맵핑한다. 또한 운영체제에서 발생한 이벤트들을 번역하여 본 시스템의 내부 메시지로 전환한다.

② 페이지 또는 객체의 생성 및 제거

사용자가 특정 페이지로의 네비게이션(navigation)을 원할 경우 북은 요구되는 페이지를 검색하여 메모리에 적재시키고 페이지의 제거에 관련된 이벤트가 발생할 경우 메모리에서 제거시켜 주는 기능을 한다. 또한 현재 페이지에 특정 객체를 생성하거나 제거하기 위하여 북은 현재 페이지에 이것을 요청을 하고, 페이지는 이 요청에 따라 반응하게 된다.

③ 객체의 존재 시간

북은 저작도구에서 사용되는 객체들을 자동적으로

표 1. 저작도구의 비교
Table 1. Comparison of authoring tools

	사용 환경	저장방식	디스플레이 (텍스트 및 그래픽)	오디오 및 비디오 처리	애니메이션	데이터 처리
Authorware	윈도우즈, Mac	아이콘 흐름도	<ul style="list-style-type: none"> 기본적인 워드 프로세서 기능 선, 원, 사각형 등 그림물 제공 BMP, PCX, DIB 등 그래픽 파일 지원 	<ul style="list-style-type: none"> 다양한 속도와 디지털 오디오 재생기능 SoundWave 편집기 제공 AIFF, PCM 오디오 지원 정지 및 동화상 비디오 지원 전체 / 일부 화면 출력 화면 번호에 의해 속도 조절 가능 하나의 채널에 2개의 오디오 연결 가능 	<ul style="list-style-type: none"> 목적지, 경로 등 정해진 방향으로 이동하거나 경로, 속도를 조절 셀과 비트맵 애니메이션 생성 	<ul style="list-style-type: none"> 다양한 로직 구사 가능 응용 프로그램 호출, 복귀 기능 CMJ 기능으로 사용자의 학습 수준과 결과 판단 가능
ToolBook	윈도우즈	책 / 페이지	<ul style="list-style-type: none"> HotWord기능으로 HyperText의 제작이 쉬움 선, 원, 사각형 등 틀 팔레트 지원 OpenScript 제공 BMP, DIB, WMF, 파일 지원 	<ul style="list-style-type: none"> WAV 오디오 카드 지원 비디오 오버레이 카드 지원 	<ul style="list-style-type: none"> 스크립트 언어를 통한 애니메이션 제어 Authorware, IconAuthor 보다 풍부한 편 	<ul style="list-style-type: none"> dBase의 dbf파일과 호환 OpenScript제공 free runtime패키지 제공으로 산출물의 복사, 배부 가 용이 Password로 보안 가능 제공
본 논문의 저작도구	윈도우즈	책 / 페이지	<ul style="list-style-type: none"> 텍스트 색상, 글꼴 및 다양한 효과 지원 벡터 그림(선, 원, 사각형 등) 지원 BMP 파일 지원 	<ul style="list-style-type: none"> 윈도우즈의 사운드 카드 드라이버 이용 WAV, MIDI 등의 MCI지원 AVI 비디오 포맷 지원 	<ul style="list-style-type: none"> FLC, FLI 애니메이션 파일 지원 스크립트에 의한 애니메이션(비트맵, 객체, 텍스트 등) 지원 	<ul style="list-style-type: none"> 스크립트 언어 지원 : 자체 인터프리터 개발 다양한 개발 유형 지원(반복 연습형, 개인 교수형, 개인형, 시뮬레이션형 등) 다양한 CMJ기능 구현 가능

생성 또는 제거하기 위하여 생성된 객체들의 존재 시간을 관리하는 기능을 가지고 있다. 복은 자체 타이머를 사용하여 객체에 지정된 시간마다 객체를 생성하고 제거한다. (그림 4)는 타이머에 따른 객체의 자동 생성 및 제거 과정을 나타낸다.

④ 스크립트 실행
복에서는 각 이벤트에 해당하는 기본적인 스크립트들을 정의하여 저장하고 복이 열릴 때마다 번역을 실행하여 실행 테이블 형태로 보관하게 된다. 이러한 기본 스크립트들은 이벤트가 발생된 해당 객체에 이벤트와 일치되는 스크립트가 없을 경우 복에서 사

전에 정의된 기본 스크립트들이 실행된다. 복에 포함되는 스크립트의 내용을 편집함으로써 복이 수행할 기본적인 행동을 정의 할 수 있다.

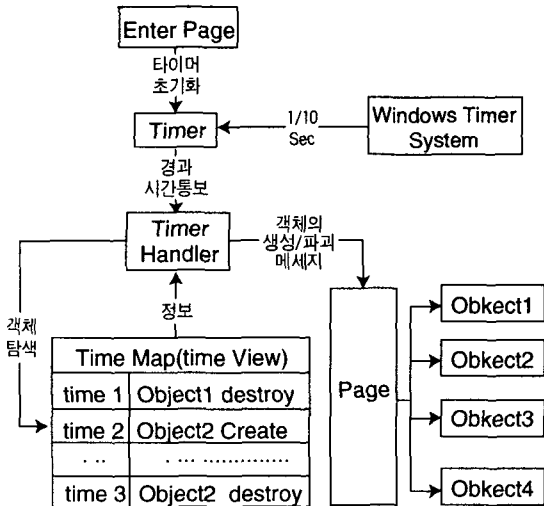


그림 4. 객체 존재 시간 처리
Fig. 4. The process of object life time

2) 페이지

페이지 객체는 실제로 저작이 이루어지는 작업 영역으로 설계한다. 그 내부는 그룹 객체나 일반적인 객체 및 하위 페이지를 소유하고 복과 객체를 연결해주는 중간자 역할을 하게 된다. 즉, 복이 객체에 메시지를 전달하려고 할 때 복 자체가 직접 객체에 전달하지 않고 페이지를 통하여 객체에 전달하게 된다. 또한 페이지는 페이지와 동일한 속성 및 기능을 수행하는 하위 페이지를 소유할 수 있다.

3) 객체

본 논문에서 말하는 객체는 일반적으로 내부에 다른 객체를 포함하지 않는 객체를 의미한다. 각 객체의 종류는 다음과 같다.

1. 벡터 그래픽 객체(vector graphics objects) : 다각형 (polygon), 선(line), 베지어 곡선(bezier curve), 타원 (ellipse), 다각선(polyline), 아크(arc)로 구성된 객체.
2. 윈도우즈 컨트롤 객체(windows control objects) : 버튼(button), 콤보박스(combobox), 텍스트(text), 리스트박스(listbox), 체크 박스(checkbox), 라디오버튼(radiobutton)으로 구성된 객체.
3. 멀티미디어 객체(multimedia objects):웨이브(wave) 미디(MIDI), CD 오디오, 디지털 비디오(digital video), 비트맵 그래픽(bitmap graphic), 애니메이션(animation)으로 구성된 객체.

3. 메시지 처리

본 저작도구는 이벤트 구동 방식의 프로그래밍을 제공한다. 따라서 본 시스템에서는 별도의 이벤트 핸들러를 구현하여 사용자 또는 시스템, 그리고 객체로부터 발생하는 메시지들을 관리할 수 있도록 하였다. 메시지를 분류하면 다음과 같다.¹⁶⁾

- ① 사용자에게 의한 키보드 및 마우스 입력을 감시하는 메시지
- ② 타이머를 위한 타이머 메시지
- ③ 시스템을 관리하기 위한 메시지
- ④ 각 객체들이 자신의 정보를 나타내기 위해 발생시키는 메시지
- ⑤ 사용자가 객체에 특정한 기능을 부여하기 위한 사용자 정의 메시지

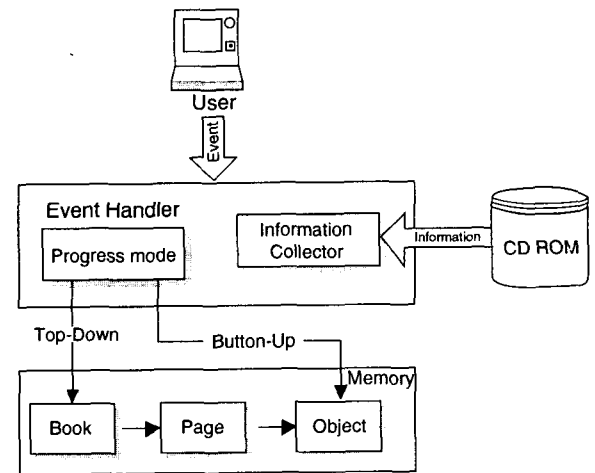


그림 5. 메시지 전달 과정
Fig. 5. The process of message

이벤트 핸들러에서는 입력된 이벤트를 처리하기 위해서 실행기가 페이지를 읽어 올 때 저작 객체들에 대한 속성을 검색하여 정보를 얻어 온 후 실행시 발생하는 메시지에 따라 반응하게 된다.

(그림 5)는 메시지의 발생 및 처리과정을 나타낸다. 메시지는 다음과 같은 두 가지 방법으로 전달된다. 첫 번째 경우 『페이지 준비(prepare page)와 같은 메시지의 경우 메시지는 복을 거쳐 페이지로, 페이지에서 다시 객체로 전해지는 반면, 『버튼 클릭(button click)』과 같은 메시지는 그 반대인 객체에서 페이지를 통해 복으로 전해지게 된다.

만일 발생된 객체에서 그 메시지에 대한 처리 속성이 존재할 경우 그것을 처리하고 그렇지 않으면 그 객체의 상위 객체로 발생된 메시지를 전달하여 준다.

IV. 스크립트 해석기 구현

1 스크립트 해석기의 설계

스크립트 해석기(script interpreter)는 사용자에게 의하여 입력된 스크립트 소스를 분석하여 각 토큰을 분리하는 어휘 분석기(lexical analyzer), 생성된 토큰을 입력으로 문법

을 검색하고 실행시 사용될 테이블을 생성하는 구문 분석기(syntactic analyzer), 실행 테이블을 입력으로 객체에 해당되는 행동을 실행하는 실행기, 스크립트의 분석 과정 및 실행 과정에서 발생하는 오류에 대한 처리를 담당하는 오류 처리기로 구성된다.

(그림 6)은 사용자와 객체 표현 모듈 그리고 스크립트 인터프리터간의 동작 관계를 나타낸다. 모든 객체에는 스

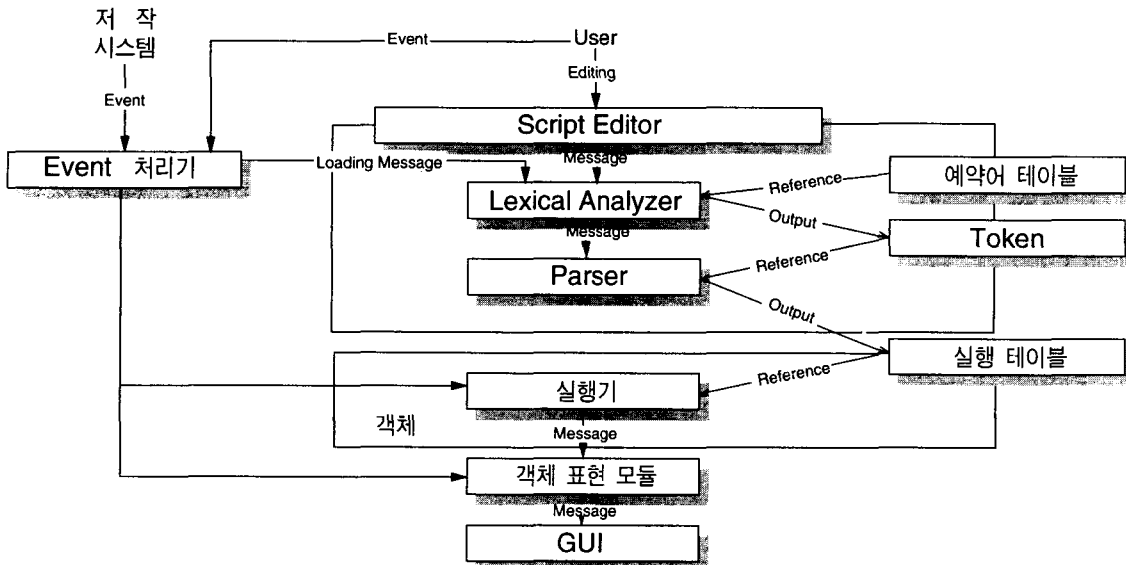


그림 6. 스크립트 인터프리터의 동작도
Fig. 6. The operation diagram of script interpreter

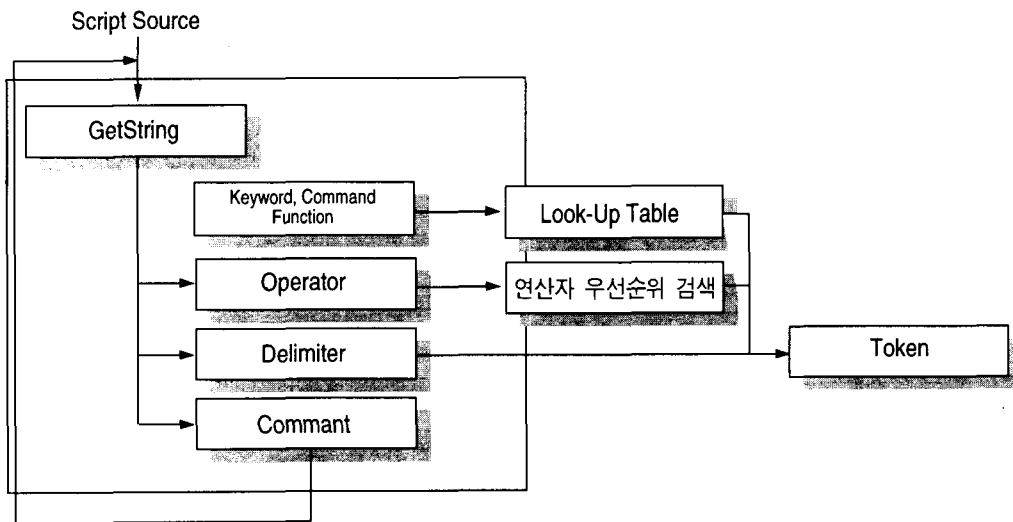


그림 7. 어휘 분석 과정
Fig. 7. The process of Lexical Analysis

스크립트 소스와 실행 테이블을 저장하는 장소를 가지고 있고 실행테이블을 자원으로 객체를 실행 할 수 있는 실행기를 내장하고 있다.

객체가 소유하고 있는 스크립트 소스가 구문 분석 및 어휘 분석되는 경우는 두 가지로 구분 될 수 있다. 첫 번째는 사용자가 특정 객체를 선택하여 스크립트 편집기(script editor)를 호출 한 후 스크립트를 부여하여 스크립트 에디터에 있는 스크립트 분석에 대한 메뉴를 선택하는 경우이다. 이 경우 스크립트 에디터는 인터프리터의 구문 분석기에 분석 메시지를 보내게 되고 최종적으로 객체의 행동을 위한 실행 테이블을 생성하고 저장하게 된다.

두 번째는 기존에 생성되어진 페이지가 적재될 때 이벤트 처리기는 페이지 내의 모든 객체를 대상으로 적재된 메시지를 보내게 된다. 메시지를 받은 모든 객체는 자신이 스크립트 소스를 가지고 있는 지의 여부를 검색하고 스크립트 소스가 있을 경우 분석 과정을 수행하여 앞의 경우와 마찬가지로 객체에 실행 테이블을 생성하고 저장한다.

2. 어휘 분석기의 설계 및 구현

어휘 분석기는 스크립트 에디터를 통하여 입력된 스크립트 소스를 예약어 테이블과 비교함으로써 각 속성에 해당하는 토큰을 분리하여 테이블 형태로 저장한다.^{[4][5][8][13]} (그림 7)은 어휘 분석과정을 도식화한 것이다. 어휘 분석 과정에서 보다 편리한 변수 관리를 위하여 변수명에 대한 한글 처리를 첨가하였다. 토큰의 속성은 다음과 같다.

- ① 기본 속성
 - 토큰 입력 순서, 토큰 스트링 길이, 토큰 스트링 선언된 토큰 타입 정보(연산자, 키워드 함수 등)
 - 오류 발생시 스크립트 에디터 상에서의 토큰의 위치에 대한 정보.
- ② 연산자 속성
 - 연산자 우선 순위
- ③ 객체 속성
 - 객체 타입 정보

이와 같은 토큰에 대한 클래스를 다음과 같이 정의한다

```

// Token Table
class token : public CObject
{
public :
  int token_index; // token 입력 순서...
  int strln; // token string 길이
  CString ch_string; // token string.
  CString INFO_token; // 선언된 token 타입.
  CString INFO_DECL; // 토큰 정의 타입 정보.
  int INFO_OBJ; // Operator priority, Object.
  int INFO_Object; // Object 타입에 관한 정보.
  // error 발생시 highlight를 위한 속성 값
  int line_index; // y value
  int index; // x value
};

```

3. 구문 분석기의 설계 및 구현

본 논문에서 구현한 구문 분석기는 프리딤티브 파서(predictive parser) 알고리즘으로 구현한다.^{[5][8][13]}

본 저작도구는 각 이벤트가 수행하는 기본 행위에 대한 스크립트와 전역적으로 사용되는 상수형 선언 자료 등을 기본적으로 가지고 있다. 이러한 기본적인 스크립트는 저작 시스템이 처음 실행될 때 분석과정을 거쳐 실행 테이블을 생성한다. 또한 사용자가 직접 구현 할 수 있는 함수(user define function)의 재사용을 위하여 스크립트를 별도의 파일에 저장할 수 있게 하였다.

사용자 정의 함수로 구성된 파일은 『import』라는 키워드를 사용하여 사용자 함수를 호출하는 스크립트의 처음에 포함된다.

4. 실행 테이블 관리

본 시스템의 모든 객체들은 실행 테이블을 가지고 있다. 이러한 실행테이블은 사용자가 특정 객체에 스크립트를 부여하여 어휘 및 구문 분석을 수행한 결과로 생성되거나 페이지가 처음 적재될 때 페이지 내의 모든 객체들의 스크립트를 분석함으로써 생성된다. 이렇게 생성된 테이블은 실행시 실행기에 의해서 참조되어 이벤트에 해당하는 스크립트를 실행한다. 구문 분석과정에서 생성한 테이블들은 각각 저장되는 장소에 따라서 전역 테이블과 지역 테이블로 구성시킨다.

1) 전역 테이블

전역 테이블은 복에 저장되고 스크립트가 사용자 정의 함수이거나 전역 변수 및 상수에 해당되는 경우이며 그 종류는 다음과 같다.

- ① 사용자 함수 테이블 : 예약된 함수를 제외한 사용자가 임의로 정의한 모든 함수를 저장한다. 변수, 연산자, 상수토큰에 대한 배열 및 사용자 함수 이름과 상위 객체의 이름으로 구성되어 있으며 『Procedure』 키워드로 함수를 선언한다.
- ② 사용자 함수 정보 테이블 : 선언된 사용자 함수에 대한 여러 가지 정보를 저장한다. 함수에 대한 인자, 인자 개수, 반환값 저장 장소, 반환형으로 구성한다.
- ③ 전역 변수, 전역 상수 테이블 : 복의 모든 영역에서 참조될 수 있는 형태의 변수로 전역상수의 경우 정의된 이후 새로운 값을 재정의 하지 못한다. 변수에 대한 값과 타입 등으로 구성되며, 『Procedure』 또는 『Event』 키워드 앞부분에 선언한다. 전역으로 정의되는 테이블들은 직접 실행되지 못한다. 단지 참조 가능한 라이브러리 파일 형태로 지역 테이블에서 호출되어야만 실행이 가능하다. 다음은 각 전역 테이블에 대한 클래스이다.

```

. 사용자 함수 테이블

class CGlobalTable : public CObject
{
public :

    CObArray ident_table// Identifier table
    CObArray op_table; // Operator table
    CObArray token_table;
    CObArray const_table;

    CString TBL_name;
    CString BookPageName;
    CString Msg_name;
}

. 상수 테이블
// 상수 테이블에 대한 정의...
class CConstTable : public CObject
{
public :
    double mConstValue;
    CString mConstName;
    int mType;
}

```

```

. 사용자 함수 정보 테이블

class UserFuncINFO : public CObject
// 사용자 함수에 대한 정보
{
public :
    int ParaCount// parameter Count...
    CString FuncName// 함수 이름...
    int RtnType// 반환값 타입...
    void operator = (UserFuncINFO *);
    DataStorage mRtnValue;
}

```

2) 지역 테이블

모든 객체들은 이벤트에 따른 독립적인 지역 테이블을 가지고 있으며 사용자가 『Event』 키워드로 선언함으로써 정의된다.

지역 테이블은 선언된 이벤트가 발생할 경우 직접 실행될 수 있으며 전역 테이블에 저장된 사용자 함수를 호출하여 실행할 수 있다.

```

. 사용자 함수 테이블

class CGlobalTable : public CObject
{
public :

    CObArray ident_table // 변수 테이블
    CObArray op_table; // 연산자 테이블
    CObArray token_table;
    CObArray const_table;

    CString TBL_name;
    CString BookPageName;
    CString Msg_name;
}

. 변수 테이블

class CId_table : public CObject
{
public:
    token* Id_name; // Identifier 이름...
    token* Id_definition; // Identifier 선언...
    int Id_value; // Identifier table
    int INFO_type; // Data Type flag
    DataStorage mDataSt;// 데이터 저장..
    BOOL mColorFlag;// Line.Brush Color
    int mTextFlag// Text 종류
    BOOL mParaFlag// Parameter flag
    BOOL mArray;// 배열
    int ArrayFlag// 배열의 typeflag.
    int ArrayMax;// 배열의 크기...
    int ArrayPos;// 배열의 포지션...
}

```

```

. 연산자 테이블

class COp_table : public CObject
{
public:
    token* OP_left// 왼쪽 토큰의 인덱스...
    token* OP_string;
    token* OP_right// 오른쪽 토큰 인덱스....
    int OP_flag;
}

```

5. 실행기의 구현

본 논문의 저작도구는 저작모드와 실행모드의 두 가지 모드로 구성한다.⁶⁾ 저작모드는 사용자가 객체들을 임의로 생성 또는 이동 할 수 있고 객체에 스크립트를 할당 할 수 있다.

실행모드는 객체에 대한 수정 및 스크립트 할당이 금지되며 이벤트의 발생과 이에 해당하는 스크립트의 실행으로 객체에 변화를 줄 수 있다. 각 스크립트가 실행되는 시점은 실행 모드에서 객체가 스크립트에 해당하는 메시지를 받을 경우이다. 해당 메시지는 사용자 및 시스템에 의하여 발생한 이벤트를 이벤트 처리기에 의하여 해당 메시지로 맵핑되며 이벤트 처리기는 실행기와 객체 표현 모듈로 메시지를 보낸다. 또한 페이지가 처음 적재될 경우에는 객체의 실행을 위한 테이블 생성을 위하여 어휘 분석에 메시지를 보낸다. 메시지를 받은 객체는 자신이 소유하고 있는 스크립트를 검색하여 해당 스크립트인 경우 실행기를 호출하고 해당되지 않는 경우는 자신의 상위 객체로 이벤트를 옮겨 준다. 스크립트의 실행은 이벤트가 발생된 객체의 지역 테이블을 검색하여 연산자 테이블을 순차적으로 처리함으로써 실행하게 되며 연산자의 종류에 따라 객체연산, 조건 연산, 사칙 연산, 함수 연산의 네 가지 부분으로 실행한다. (그림 8)은 이벤트 처리기로부터 메시지를 받아서 스크립트를 실행하는 과정을 나타낸다.

6. 실행

본 절에서는 간단한 스크립트 소스를 분석하여 실행하는 과정에 대하여 설명한다. (그림 9)는 마우스 왼쪽버튼 클릭 메시지가 발생했을 경우 발생한 버튼에 'btn1'이라는 이름을 주고 이미 사용자에게 의해서 작성되어 있는 'CreateButton.ssf'라는 파일을 임포트해서 실행 즉, 버튼(Button) 객체를 그려주는 행위를 하는 스크립트 소스를 나타낸다. 마우스를 드래그 하여 (그림 10)에서와 같은 버튼 객체와 비트맵 객체를 생성하고 생성된 버튼 객체에 (그림 9)의 스크립트를 부여한 후, 스크립트 에디터의 분석 메뉴를 선택함으로써 스크립트 소스를 분석한다. 상기 과정을 거친후 툴바의 실행 버튼을 눌러서 실행 모드로 전환 한다. 실행 모드에서 스크립트가 부여된 버튼생성 버튼을 누르면 새로운 버튼을 생성한다. 또한 새로 생성된 버튼을 누르면 사전에 생성되어 있는 페이지로 네비게이션 된다. (그림 10)은 위와 같은 과정을 나타낸 것이다.

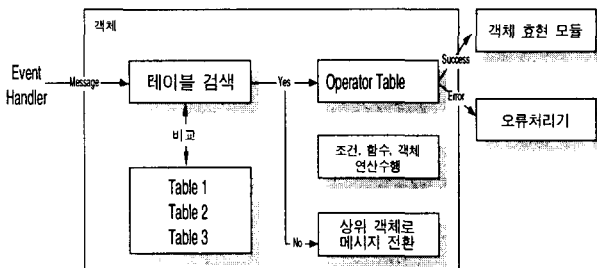


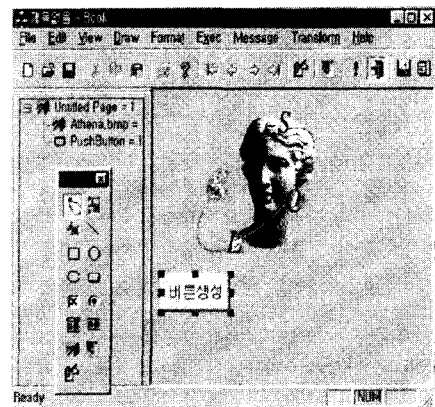
그림 8. 스크립트 실행 과정
Fig. 8. The process of script execution

```

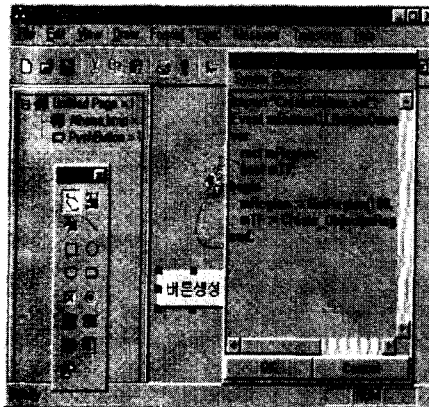
. File name : CreateButton.ssf
Procedure Create_Object(rect mRegion) : bool;
var
    PushButton btn1;
begin
    btn1.Position := mRegion;
    btn1.Color := GetRGB(255, 0, 255); /* 기본 함수*/
    if (btn1.Show = TRUE) then
        btn1.Show := FALSE;
    else
        btn1.Show := TRUE;
    Create_Object := TRUE;
    btn1.Script := /* 새로운 스크립트 할당 */
        "Event btn1.LButtonDown();
        begin
            goto(2);
        end.";
    btn1.Caption := "다음 페이지로";
end.

. File name : CallButton.ssf
import "CreateButton.ssf";
Event mButtonC.LButtonDown(int nFlag, point pt);
var
    rect mRegion;
    bool mTF;
begin
    mRegion := GetRegion(100, 100, 300, 300);
    mTF := Create_Object(mRegion);
end.
    
```

그림 9. 스크립트 소스
Fig. 9. Script source



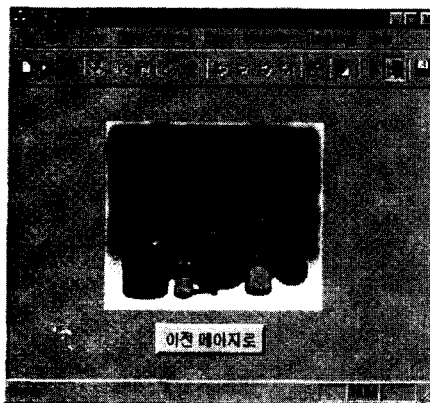
객체 생성



스크립트 할당



스크립트 실행



생성된 버튼을 클릭하여 다음 페이지로 이동

그림 10. 저작 예
Fig. 10. Example of authoring

V. 결 론

본 논문에서는 저작도구를 위한 다양한 객체를 설계 및 구현하고 스크립트를 정의 및 그 실행을 위한 실행기를 설계 및 구현하였다. 모든 객체의 구현이 객체 지향적 언어인 C++로 설계되었기 때문에 확장성이 가능하다. 모든 객체는 자신만의 단일한 속성을 소유하고 있다. 그러므로 사용자는 자신이 필요로 하는 객체를 생성하고 그 속성을 조작하거나 스크립트를 작성함으로써 객체에 동적인 변화를 줄 수 있다. 또한 객체의 자동 생성 및 제거를 지원하기 위하여 자체 타이머를 구현하였고 각 객체에 자신의 존재 시간에 관한 속성을 부여하였다. 또한 본 저작도구는 전문가적인 사용자와 초보적인 사용자층을 고려하여 프로그래밍 언어 형태의 스크립트와 객체 기본 행동에 대하여 정의하였다. 스크립트는 표준 파스칼 형태를 갖추고 있어서 프로그래머에게 익숙한 환경을 제공한다. 또한 이벤트 발생에 따른 함수만을 작성해 줌으로서 각 객체에 대하여 독자적인 실행을 가능하게 하였고 텍스트 형태의 파일을 임포트 하여 스크립트에 대한 별도의 라이브러리 기능을 제공함으로써 소스를 재사용 가능하게 하였다. 초보자를 위하여 각 객체에 대한 기본 행동을 라디오 버튼(radio button)과 체크 박스(check box) 형태로 제공함으로써 초보자가 보다 쉽게 저작 할 수 있는 환경을 제공하였다.

본 저작도구는 객체 및 스크립트 속성을 중심으로 구현하였기 때문에 사용자가 객체의 상태를 모니터링(monitoring) 할 수 있는 인터페이스를 구현하지 못하였다.

향후 연구 과제로는 사용자 인터페이스를 보강하고 초보적인 사용자라도 쉽게 사용할 수 있는 라인 단위의 실행기 설계 및 구현과 네트워크 환경에서 공동으로 저작할 수 있는 저작 환경을 설계 및 구현할 예정이다.

참 고 문 헌

- [1] Asymetrix Multimedia ToolBook 3.0, Asymetrix Corporation, 1993.
- [2] Judith Jeffcoate, *Multimedia in practice-technology and application*, Prentice-Hall, 1995.
- [3] Songbae Eun, Eun Suk No, Hyung Chul Kim, Hyunsoo Yoon, and seung Ryoul Maeng, "Eventor :an authoring system for interactive multimedia applications," *Multimedia systems*, vol. 2, no. 1-6, ACM press, pp. 129-140, 1994.
- [4] A. V. Aho, R. Sethi, and J. D. Ullman, *Compilers principles, techniques and tools*, Addison-Wesley, pp. 181-195, 1988.
- [5] T. W. Parson, *Compiler construction programming*, AP Professional, 1994.
- [6] R. N. Taylor, N. Medvidovic, K. M. Anderson, and

E. J. Whitehead Jr, "A component and message_based architectural style for GUI software," *IEEE Transactions on Software Engineering*, vol. 22, no. 6, 1996.

[7] Koji Takeda, Mitsuyuki Inaba, and Kazuo Sugihara, "User interface and agent prototyping for flexible workinhttp://www.nb.net/~lrc/page6.htmlg," *IEEE Multimedia*, pp. 40-50, 1996.

[8] S. J. Gibbs and D. C. Tsichritzis, *Multimedia programming*, Addison-Wesley, ACM Press, 1994.

[9] 양옥렬, 정영식, 이용주, "멀티미디어를 기반으로 하는 저작도구 틀북에서 객체 자동 변환을 이용한 자동 프리젠테이션 시스템 개발", *한국정보처리학회 논문지*, 제4권, 제5호, pp. 1182-1195, 1997.

[10] 이종인, 차현성, 한광록외, "스크립트 인터프리터 개발을 위한 이벤트 처리와 객체관리에 관한 연구", *한국정보처리학회 춘계 학술발표 논문집*, 제4권 1호, pp. 325-329, 1997.

[11] 한혜현, 박상욱, 김정학, 최제욱, "Multimedia Tool-Book 3.0", 남양문화, 1994.

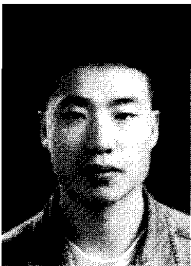
[12] 성미영, "분산 멀티미디어 환경에서의 공동 저작", *한국정보처리학회 멀티미디어 시스템 연구회 제 4회 멀티미디어 산업기술 학술대회 논문집*, pp. 117-126, 1995.

[13] C. W. Fraser and D. R. Hanson, "A retargetable C compiler: Design and implementation," The Benjamin/Cummings, 1995.

[14] F. C. Heeman, I. Herman, and Graham Reynolds, "Interaction objects the MADE multimedia environment," *Multimedia/Hypermedia in Open Distributed Environments, Proceeding of the Eurographics Symposium, Springer-Verlag Wein New York*, pp. 264-277, 1994.

[15] M. E. Hodges and R. M. Sasnett, *Multimedia computing, case studies from MIT project at the na*," Addison-Wesley, 1993.

저 자 소 개



차 현 성

1996년 2월 호서대학교 컴퓨터공학과 졸업(공학사)
 1998년 2월 호서대학교 대학원 컴퓨터공학과 졸업(공학석사)
 주관심분야: 멀티미디어 저작시스템, 데이터 베이스, 정보검색, 자연언어처리 등



한 광 록

1984년 2월 인하대학교 전자공학과 졸업
 1986년 2월 인하대학교 대학원 전자공학과(공학석사)
 1989년 8월 인하대학교 대학원 전자공학과(공학박사)
 1991년 3월 ~ 현재 호서대학교 컴퓨터공학과 부교수
 주관심분야: 자연언어처리, 정보검색, 멀티미어 저작 시스템 등