멀티미디어 저작도구의 번역기 설계 및 구현

이계영 임재걸 김유리° 감창수
동국대학교 전자계산학과

A Design and Implementation of a Translater for a Multimedia Authoring Tool

Lee, Gyeyoung Yim, Jaegeol Kim. Yuree° Kam, Changsu
Div. of Computer and Information Science, Dongguk University
leegy@wonhyo.dongguk.ac.kr vim@wonhyo.dongguk.ac.kr kyr@www.dongguk.ac.kr

요 약

본 논문에서는 스크립트로 표현된 멀티미디어 시나리오를 윈도우95 API를 이용하는 C++ 프로그램으로 번역하여 주는 스크립트 번역기를 설계하고 구현한다. 스크립트 에디터로 직접 작성하거나, 혹은 비주얼 에디터에서 작성된 시나리오를 자동 변환하여 작성할 수 있다. 본 연구에서는 시나리오가 화면 단위로 구성된다고 가정하여, 스크립트 파일 역시 화면 Pi 단위로 정의되어 있으며, 한 개의 화면은 다수의 오브젝트로 구성된다. 각 객체를 표현하는 데 한 개의 윈도우가 필요하므로, 번역기는 각 Pi마다 하나의 Main Window와 여러 개의 Child Window를 생성하여 멀티미디어 오브젝트를 출력한다. 따라서, 본 논문은 스크립트 파일의 구성 내용을 정의하고, 저작도구의 스크립트 번역기를 설계하고 구현하는 방법에 대하여 기술한다.

1. 서론

멀티미디어 저작도구란 멀티미디어 타이틀을 쉽게 제작할 수 있도록 지원하는 소프트웨어 개발 도구이다. 이러한 저작도구에 사용되는 프로그래밍 모델에는 흐름도 방식, 책 방식 및 시간선 방식 등이 있다. 흐름도 방식에서는 타이틀의 기본 구조가 흐름도로 표현되며, 저작자는 흐름도를 구성하는 도구 아이콘들을 저작할 시나리오에 맞추어 흐름선에 삽입함으로써 흐름도를 구성한다. 책 방식은 타이틀을 여러 페이지로 구성된 책이라 보고 각 페이지에 나타날 미디어들을 기술하며, 타이틀의 책 흐름이나 사용자의 입력에 대한 동작 등은 스크립트 언어로 표현한다. 시간선 방식은 음악 책에서 악보를 기술하는 방식과 비슷한 방식으로 가로축으로 여러 채널을 시작점을 맞추어 나열하고 프리젠테이션이 왼쪽 끝점에서부터 오른쪽으로 실행된다고 보고 각 시점에 실연할 미디어들을 각 채널의 해당 시점에 삽입하여 넣는다. 흐름도 방식의 저작도구는 아트웨어, Authorware Professional Icon Author등이 있고, 책 방식에는 ToolBook, 시간선 방식에는 MacroMedia사의 Director가 있다[1].

본 논문은 멀티미디어 저작도구의 스크립트 번역기를 설계 및 구현한다. 스크립트 에디터에서 사용자가 스크립트로 직접 시나리오를 구성하면, 본 번역기는 그것을 멀티미디어 프로그램을 할 수 있는 실행 파일로 변환시킨다. 본 번역기의 프로그래밍 모델은 책 방식을 따른다. 그래서, 저작물에 대해서 화면 단위로 스크립트 파일을 작성한다. 2절에서 멀티미디어 저작도구의 일반적인 구조를 보이고, 3절에서는 시나리오에 대한 스크립트 파일의 구성 요소를 정의하고, 4절에서 멀티미디어 저작물의 기본구성과 번역기에 대하여 기술한다.

2. 멀티미디어 저작도구 구조

멀티미디어 저작도구는 일반적으로 에디터(Visual Editor와 Script Editor)와 번역기(Translator), 실행기(Player)의 세 부분으로 구성된다(그림 1)[5]. 에디터는 멀티미디어 저작물에 대한 시나리오를 작성하는 부분으로, 시나리오를 그림으로 작성하는 Visual Editor와 스크립트 파일 형식으로 작성하는 Script Editor가 있다. 에디터에서 작성한 시나리오는 다음 절의 표 1과 같은 형식의 스크립트 파일로 저장된다. 그러면, 번역기는 그 정보를 읽어들여 멀티미디어 프로그램에 대한 실행 화일을 생성한다. 본 논문은 스크립트 파일의 형식을 정의하고, 정의된 방법으로 표현된 시나리오를 번역하여 실행 화일을 생성하는 번역기의 구현에 대하여 중점적으로 기술한다.

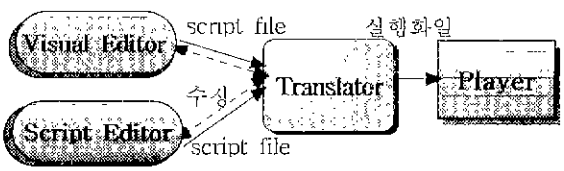

그림 1. 멀티미디어 저작도구의 구조

실행기는 번역기에서 생성된 저작물의 실행 화일을 실행시켜 사용자가 보여주고, 대화하는 역할을 하는 모듈로 Prev와 Next의 Button Control로 저작물을 제어할 수 있다. 다음 절에서 시나리오에 대한 스크립트 파일에 대하여 자세히 기술한다.

3. 스크립트 화일

본 논문에서 사용하는 스크립트 화일은 저작물을 화면 단위로 정의하며, 각 화면은 멀티미디어 객체로 구성된다. 본 스크립트 화일에 사용되는 객체는 다음과 같이 분류된다.

① PICTURE
그림 화일에 관한 정보로서, 각 화일은 다양한 색상(16, 256, 16 bit 등)을 가질 수 있다. 본 번역기는 그림 화일의 헤더를 분석하여 알 맞는 색상과 해상도로 DISPLAY 한다

② TEXT FILE
화면에 표시되는 그림 또는 동영상, SOUND에 대한 설명을 나타내는 TEXT FILE을 화면에 나타낼 수 있다. TEXT FILE은 기본적으로 상하, 좌우 스크롤이 가능한 윈도우에 출력된다.

③ 동영상
본 번역기는 다양한 TYPE의 동영상 FILE을 지원한다

④ SOUND
본 번역기는 지정된 SOUND FILE을 연주한다.

⑤ BUTTON
다음화면, 이전화면 또는 EXIT 등의 화면 이동과 종료에 이용될 수 있는 BUTTON 사용이 가능하다.

⑥ STRING
해당 화면의 제목이나 주제에 대한 글을 화면에 나타낼 수 있다. 문자열에 대한 폰트와 글자의 크기, 글씨체 등을 지정할 수 있다.

에디터에서 작성된 시나리오의 스크립트 파일을 본 논문에서는 INI 파일이라 칭한다. INI 파일의 구성은 표 1과 같다. 표 1에서, <INI_FILE>은 <LOGO_SECTION>과 <INIT_SECTION>, <BUTTON>, <SECTION>으로 구성되어 있다 <LOGO_SECTION>과 <INIT_SECTION>은 멀티미디어 저작물의 초기화면과 제목, 메인 윈도우를 설정하는 것으로. <MAXSCREEN_L>은 저작물을 구성하는 전체 화면의 수를 나타낸다 <BUTTON>은 시나리오에서 화면 이동을 하는 방법으로, 'Prev' 또는 'Next' 버튼을 지정한다 <SECTION>은 저작물에서 한 화면에 해당하고, <SECTION_ID>와 <ELEMENT_L>, <TIMECONTROL_L>, <OBJECT_SECTION>으로 구성된다. 여기에서, <SECTION_ID>는 저작물의 화면 번호이며, [P1]~[P*n*]의 차례로 각각 첫 번째 화면부터 *n* 번째 화면의 내용을 정의한다. <ELEMENT_L>는 화면을 구성하는 오브젝트의 리스트인데, 각 <OBJECT_ID>는 오브젝트의 종류에 따라서 표 2의 <OBJECT_TYPE>을 포함한다 COMMA(',')에 의해 나누어진 문자열은 각각 다른 <OBJECT_SECTION>으로 관리된다 그리고, <TIMECONTROL_L>은 시나리오에서 화면 이동을 하는 방법 중 하나로서, 화면 P_i는 TIMECONTROL에서 지정한 시간 동안 DISPLAY된 후, 다음 화면으로 전환된다.

<OBJECT_SECTION>은 해당 화면에 출력될 모든 멀티미디어 오브젝트 각각에 대하여, 필요한 윈도우를 비롯한 관련된 모든 정보를 포함한 것으로. <OBJECT_SECTION_ID>와 <MEMBER_LIST>로 구성된다 특히, <MEMBER_LIST>는 오브젝트의 종류에 따라서 각각 다른 윈도우 설정 정보로 구성된다. <PICFILE_L>, <TEXTFILE_L>, <SOUNDFILE_L> 및 <MOVEFILE_L>은 해당 오브젝트의 화일 이름에 관한 정보이고, <CAPTION_L>은 문자열이다 <X_MAP>과 <Y_MAP>. <WIDTH_L>, <HEIGHT_L>은 각각 생성될 윈도우의 X, Y좌표 및 폭과 높이에 관한 정보이다. <ITALIC_L>과 <UNDEERLINE_L>. <SIZE_L>, <THIC KNESS_L>, <FONT_L>은 윈도우에 출력될 글자의 폰트에 관한 정보를 지정한다

```
<INI_FILE> ::= <LOGO_SECTION> <INIT_SECTION>
              <BUTTON> <SECTION> {<SECTION>}*
<SECTION> = <SECTION_ID> <ELEMENT_L> <TIMECONTROL_L>
<OBJECT_SECTION> {<OBJECT_SECTION>}
<SECTION_ID>  = '['P'<DIGIT>']'
<ELEMENT_L>::='ELEMENT' '=' '"'<OBJECT_ID>
                {,<OBJECT_ID>} '"'
<TIMECONTROL_L> ::= 'TIMECONTROL' '=' <INT>
<LOGO_SECTION> ::= <LOGO_SECTION_ID> <L_MERBER_LIST>
<LOGO_SECTION_ID> ::= '[' 'LOGO' ']'
<INIT_SECTION> ::= <INIT_SECTION_ID> <L_MERBER_LIST>
<INIT_SECTION_ID>::= '[' 'PO' ']'
<OBJECT_SECTION> ::= <<OBJECT_SECTION_ID><MEMBER_LIST>
<OBJECT_SECTION_ID> ::= '['<SECTION_ID> '_' <OBJECT_ID> ']'
<OBJECT_ID> ::= <OBJECT_TYPE> <DIGIT>
<OBJECT_TYPE>  ::='A'|'a'|'T'|'t'|'P'|'p'|'C'|'c'|'S'|'s'
<MEMBER_LIST> ::= <P_MEMBER_LIST> | <T_MEMBER_LIST> |
          <S_MEMBER_LIST> | <A_MEMBER_LIST> |
                    <C_MEMBER_LIST> | <W_MEMBER_LIST>
<P_MEMBER_LIST>  = <PICFILE_L> <X_MAP> <Y_MAP>
                    <WIDTH_L> <HEIGHT_L>
<T_MEMBER_LIST> ::= <TEXTFILE_L> <X_MAP> <Y_MAP>
      <WIDTH_L> <HEIGHT_L> <ITALIC_L>  <UNDEERLINE_L>
      <SIZE_L> <THICKNESS_L><FONT_L>
<S_MEMBER_LIST>  =<SOUNDFILE_L> <LOOP_L>
<A_MEMBER_LIST> ::=<MOVEFILE_L> <X_MAP> <Y_MAP>
                    <WIDTH_L><HEIGHT_L>
<C_MEMBER_LIST> ::= <CAPTION_L> <X_MAP> <Y_MAP>
      <WIDTH_L> <HEIGHT_L> <ITALIC_L> <UNDEERLINE_L>
      <SIZE_L> <THICKNESS_L><FONT_L>
<W_MEMBER_LIST>  =<CAPTION_L> <X_MAP> <Y_MAP>
                    <WIDTH_L> <HEIGHT_L>
<L_MERBER_LIST>  = <LOGOFILE_L> <WIDTH_L>
                    <HEIGHT_L> <TIMECONTROL_L>
<L_MERBER_LIST> ::= <CAPTION_L> <X_MAP> <Y_MAP>
                  <WIDTH_L> <HEIGHT_L><MAXSCREEN_L>
<BUTTON>  = <PREV_BTN_DCL> <NEXT_BTN_DCL>
<PREV_BTN_DCL>  = <PREV_BTN_ID> <PREV_BTN_MEMBER>
<PREV_BTN_ID>  = '[' 'PREV_BTN' ']'
<PREV_BTN_MEMBER>  = <P_CAPTION> <X_MAP> <Y_MAP>
                    <WIDTH_L> <HEIGHT_L>
```

표 1 멀티미디어 스크립트 파일 형식

OBJECT	TYPE
동영상	'A' or 'a'
TEXT FILE	'T' or 't'
그림(PICTURE)	'P' or 'p'
문자열(STRING)	'C' or 'c'
SOUND	'S' or 's'

표 2. 멀티미디어 오브젝트와 TYPE

4. 번역기

4.1 멀티미디어 저작의 기본 구성

저작도구의 에디터는 스크립트 형태로 멀티미디어 시나리오에 관한 정보를 저장한다. 이 정보를 번역기에서 멀티미디어 프로그램으로 번역하여 실행 화일을 생성한다. 기본적으로, 멀티미디어 저작은 화면 단위로 이루어짐으로, 스크립트 파일 역시 화면 P_i 단위로 정의되어 있으며, 한 개의 화면은 다수의 오브젝트로 구성된다 각 객체를 표현하는 데 한 개의 윈도우가 필요하므로, 번역기는 그림 2와 같이 각 P_i마다 하나의 Main Window와 여러 개의 Child Window를 생성하여 멀티미디어 오브젝트를 출력한다.

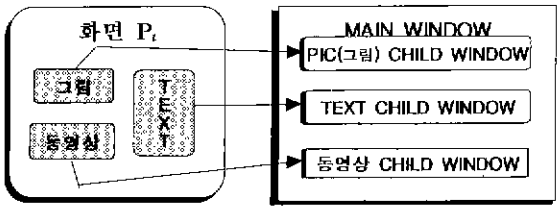

그림 2. 멀티미디어 저작의 단위(화면)

1) Main Window

Main Window는 상하 스크롤 바를 갖는 WINDOW TYPE으로 생성된다. 그 외에도 마우스 포인트와 커서의 모양 및 윈도우의 배경색을 지정할 수 있다. Main Window에서 발생하는 각종 이벤트를 처리하기 위한 메시지 처리 루틴을 'WndProc'으로 정의한다.

2) 오브젝트별 Child Window

본 번역기에서 오브젝트를 관리하는 방식은 하나의 Main Window 외에 나타나는 모든 오브젝트를 Main Window에 대한 Child Window로 관리하는 방식을 택했다. 이와 같이함으로써, 각 오브젝트는 해당 화면이 DISPLAY되는 시점에서 생성되어 다른 화면으로 전환되는 시점에 삭제된다.

3) 멀티미디어 객체의 자료형

각각의 멀티미디어 객체에 대한 INI 화일의 정보는 번역기에 의해 다음의 구조체로 읽힌다. SECTIONINI는 저작될 저작물의 화면 정보에 관한 구조체로서, 구성원은 각각 한 화면 내에서 표현되는 멀티미디어 오브젝트의 수를 나타내고, TimeControl은 한 화면이 실행되는 시간에 관한 정보를 포함한다. 예를들어 동영상에 관한 자료형인 MOVEINI를 살펴보면 저작물이 실행될 때, 그 실행 화면에서의 동영상에 관한 윈도우의 X, Y좌표와 폭과 넓이를 지정해 줌으로써, szFile에 지정된 동영상이 정해진 공간상에 정확히 출력되도록 한다. 그래서, 여러 멀티미디어 오브젝트 사이의 공간 동기화를 간단히 할 수 있다.

```
typedef struct tagSECTIONINF
{
    char    szElement[256],
      // 화면의 OBJECT LIST
    int     PicCount,
    int     TextCount,
    int     MoveCount,
    int     SoundCount,
    int     StringCount,
    int     TimeControl,
} SECTIONINI,
```

```
typedef struct tagMOVEINF
{
  char    szFile[256]
  // 동영상 화일명
  int     x,
  int     y,
  int     Width,
  int     Height,
} MOVEINI,
```

표 3. 멀티미디어 저작물의 화면을 정의하는 자료형(좌)과 동영상에 대한 윈도우를 정의하는 자료형(우)

4.2 번역기

본 번역기는 INI 화일의 내용을 읽어서, Main Window를 생성하는 프로그램으로 번역하고, SECTIONINI의 정보에 따라서 해당 오브젝트의 Child Window를 생성하고, 멀티미디어 오브젝트를 출력하는 C++ 프로그램으로 번역한다. 예를 들어, INI 화일로부터 동영상 오브젝트에 관한 정보를 읽어서 Child Window를 생성하려고 한다고 하자. 동영상 오브젝트의 윈도우에 대해서 발생하는 각각의 이벤트를 처리하는 루틴을 AviChildWndProc()으로 정의하고, Get_Move_Information()을 사용하여 INI 화일에서 해당하는 <OBJECT_SECTION_ID>의 정보를 얻는다. 여기에서, GetPrivateProfileString()은 원하는 섹션에서 원하는 키의 값을 가져오는 함수이고, 키의 값이 숫자일 때는 GetPrivateProfileInt()는 원하는 섹션의 키 값이 숫자일 때 키의 값을 가져오는 함수이다. 다음은 Get_Move_Information()과 AviChildWndProc() 루틴에서 윈도우가 생성될 때, 동영상 플레이어 윈도우를 생성하는 구문의 일부분이다.

```
case WM_CREATE
  sprintf (Section, "P%d_A%02d", iScreenNo+1, 1),
  MI = Get_Move_Information (szIniFile, Section),
       // 윈도우가 생성될 때 avi 플레이어 윈도우를 생성
  hAviWnd = MCIWndCreate (
     hWnd,     // 현재 윈도우의 핸들을 지정
     hInst,    // 인스턴스 핸들
     MCIWNDF_NOMENU, //윈도우의 속성을 지정
     MI.szFile),
  return 0;
```

표 5. 동영상 Child Window를 생성하는 루틴

```
MOVEINI Get_Move_Information (char *szIniFile, char *Section)
{
 MOVEINI MI;
 GetPrivateProfileString (Section, "MOVEFILE", "",
                          MI.szFile, sizeof (MI.szFile), szIniFile),
 MI.x = GetPrivateProfileInt (Section, "X", MI.x, szIniFile);
 MI.y = GetPrivateProfileInt (Section, "Y", MI.y, szIniFile),
 MI.Width = GetPrivateProfileInt(Section, "WIDTH", MI.Width, szIniFile);
 MI.Height=GetPrivateProfileInt(Section, "HEIGHT", MI.Height, szIniFile),
 return (MI), }
```

표 6. INI 화일로부터 동영상 오브젝트에 관한 정보를 읽는 루틴

5. 결론 및 향후 연구과제

본 논문은 저작도구로 작성된 시나리오에 대한 스크립트의 번역기를 구현하였다. 그림 3은 3절에서 기술한 스크립트 화일의 형식으로 작성된 '영화 . 타이타닉'을 소개하는 시나리오를 번역하여 실행한 예를 보인 것이다.

그림 3 시나리오의 실행 예

본 번역기는 객체 지향적 프로그래밍 언어인 Visual C++ 5.0을 사용하여 구현하였고, 저작 환경을 위한 운영체제는 Windows 95를 이용하였다[2,3,4]. 본 논문에서 구현한 스크립트 번역기의 장점은 스크립트의 작성이 용이하고, 간단하게 멀티미디어 저작을 할 수 있다는 것이다. 또, 공간 동기화에 대한 표현이 비교적 쉽다는 장점이 있다. 본 논문에서는 화면 전환에 관한 컨트롤을 TIMECONTROL과 BUTTON을 이용하여 구현하였는 데, 앞으로 시간 동기화에 대한 표현을 좀더 세분화하여 스크립트 번역기를 구현하려고 한다.

참고 문헌

[1] 황태동, 멀티미디어 시스템, 정익사 1997

[2] MacDonald h Jackson, J Eric Baldschwieler, and Lawrence A Rowe, "Berkeley Continuous Media Toolkit API," http:// www.cs.berkeley.edu, 1996.3, pp.1-13

[3] Fadaday, Pete, Sutcliffe, Alistair, "Multimedia Design for the 'Moment',", The ACM international Multimedia Conference, 1997.11, pp.183-192

[4] Eleftheriadis, Alexandros, "Flavor A Language for media Representation," The ACM international Multimedia Conference, 1997.11, pp.1-9

[5] 차현성, 한광록, "멀티미디어 저작을 위한 스크립트 인터프리터의 설계 및 구현," 한국정보처리학회 논문지, 제 5권 제 5호, 1998.4, pp. 1098-1108.