

시공간 동기화 관리기를 이용한 멀티미디어 저작 시스템의 설계 및 구현

여 인 국[†] · 황 대 훈^{††}

요 약

본 논문에서는 멀티미디어 타이틀 제작을 쉽고 효율적으로 할 수 있도록, 시공간 동기화 관리기를 이용한 멀티미디어 저작 시스템을 설계하고 구현하였다. 이를 위하여 타이틀 저작자의 설계 의도를 별도의 변화 과정을 거치지 않고 실제 타이틀 구성 로직으로 표현하는 흐름도 방식의 로직 생성기와 생성된 로직을 번역하여 실행하는 로직 인터프리터를 설계하였다. 아울러 멀티미디어 프리젠테이션에 필요한 미디어 데이터 간의 시-공간 동기화 정보를 관리하기 위한 동기화 관리기를 설계하였다.

특히 미디어 객체들 간의 시간적 관계를 동기화하기 위하여 시간 명세 모델과 그 모델의 형식 언어인 MRL을 설계하였다. MRL은 복잡한 시간 관계를 간결한 형태로 표현할 수 있는 좋은 표현력을 가지고 있으며 사용자의 의도에 따라 효과적으로 멀티미디어 프리젠테이션을 동기화시키는 것이 가능하다. 공간적 동기화를 위하여 표시 매체의 공간 상의 가시적인 크기와 접속점을 일치시키는 표시 화면 편집기를 구현하였다.

Design and Implementation of Multimedia Authoring System using Temporal/Spatial Synchronization Manager

In-Kook Yeu[†] · Dae-Hoon Hwang^{††}

ABSTRACT

In this paper, a multimedia authoring system using temporal/spatial synchronization manager is designed and implemented to support easy and efficient generation of multimedia title. For this goal, a flowchart-oriented logic generator which represents a title author's design intent into a practical title composition logic without extra translation process, and a logic interpreter which translate and implement the generated title logic, are designed. Furthermore, a temporal/spatial synchronization manager which manages temporal/spatial synchronization information between media data for multimedia representation, is designed.

Especially, a temporal specification model and MRL, a formal language for the model, are designed to synchronize the temporal relation between media objects. The MRL represents a complex temporal relation by simple and clear form, and synchronizes efficiently multimedia representation according to the author's intent. A presentation frame editor which makes coincidence between visible size of representation media and attachment point, is implemented for spatial synchronization.

† 준 회 원: 생산기술연구원 선임연구원

†† 정 회 원: 경원대학교 전자계산소장

논문접수: 1997년 5월 19일, 심사완료: 1997년 10월 14일

1. 서 론

하드웨어 및 정보 처리기술의 발달로 컴퓨터에서 멀티미디어 정보를 표현할 수 있게 됨에 따라 멀티미디어 타이틀의 필요성이 대두되고 있다. 멀티미디어 타이틀이란 멀티미디어 시스템에서 동작하는 응용 소프트웨어로서, 멀티미디어 데이터를 사용하여 정보에 대한 인지도를 높이며 상호 대화적(interactive)으로 실행되므로 적시에 필요로 하는 정보를 얻을 수 있도록 한다.

이상의 멀티미디어 타이틀을 간편하고 효율적으로 제작할 수 있도록 지원하는 소프트웨어를 저작도구(authoring tool)라 하며, 오늘날 흐름도(flowchart) 방식, 시간선(timeline) 방식, 책(book) 방식의 세가지 방식이 주종을 이루고 있다.

본 논문에서는 기존의 흐름도 방식의 저작도구에 기초하여, 타이틀 제작자의 설계 의도를 별도의 변환 과정 없이 효율적으로 표현할 수 있는 흐름도 방식의 로직 생성기와 멀티미디어 프리젠테이션에 필요한 미디어 데이터 간의 시·공간 동기화를 관리하기 위한 시·공간 동기화 관리기 및 생성된 타이틀 로직을 번역하여 실행하는 로직 인터프리터를 설계하고 구현하였다.

본 논문에서 설계한 로직 생성기는 책 방식과 시간선 방식에서 스크립트 언어로써 제어의 흐름을 표현하는 것과는 달리 시각 프로그래밍(icon-oriented visual programming) 기법을 적용하여 제어의 흐름과 상호작용을 정의하였다. 아울러 기존의 시·공간 동기화에 대한 연구 결과를 분석하고 이를 토대로 멀티미디어 데이터의 효율적인 시·공간 동기화를 위한 모델을 제안하였다.

본 논문의 구성은 다음과 같다. 2장에서는 프로그래밍 모델에 따른 멀티미디어 저작 도구의 특징과 시·공간 동기화 모델에 대하여 고찰하고, 3장에서는 로직 생성기와 표시 화면 편집기 및 시간 동기화를 위한 시간 동기화 모델 및 멀티미디어 화일 그리고 로직 인터프리터를 설계하였다. 4장에서는 로직 생성기와 시·공간 동기화 관리기 그리고 로직 인터프리터의 구현을 위한 알고리즘을 설명하며, 5장에서 결론을 맺는다.

2. 관련 연구

2.1 프로그래밍 모델에 따른 저작도구의 분류

멀티미디어 저작도구에 이용되는 프로그래밍 모델에는 흐름도 방식, 책 방식 그리고 시간선 방식이 있다^{15, 17).}

흐름도 방식은 타이틀의 제어의 흐름을 중시하여 타이틀이 흐름도로 구성되며, 타이틀 설계자는 시나리오에 따라 흐름도를 구성하는 아이콘들을 흐름도에 삽입하여 흐름도를 구성한다. 이 방법은 백과사전 등을 구성하는데 적합한 하이퍼미디어 형태의 정적인 타이틀보다는 설계자의 설계 목적에 따른 시나리오에 따라 동적으로 구성되므로 프리젠테이션이나 교육용 타이틀 제작에 적합하다. 이러한 저작도구에는 Icon Author, Artware, Authorware Professional 등이 있다^{16, 17).}

책 방식은 타이틀을 여러 페이지로 구성된 책이라 가정하고 각 페이지가 포함하는 매체들을 기술한다. Multimedia ToolBook이 여기에 해당되며 제어의 흐름은 스크립트 언어로서 표시한다^{15).}

시간선 방식은 가로축으로 여러 채널을 시작점을 맞추어 나열한 다음 타이틀이 왼쪽부터 오른쪽으로 실행된다고 보고 각 시점에 실현할 미디어들을 각 채널의 해당 시점에 위치시키므로써 저작한다. Director가 이 방식의 저작도구이며, 입력에 따른 다양한 분기가 많지 않은 프리젠테이션이나 애니메이션 위주의 타이틀을 저작하는데 유용하다^{15).}

2.2 시·공간 동기화 모델

시간 동기화 모델들은 시간 간격을 가지는 두가지 이상의 행위(action)에 대하여 before, equal, meet, overlap, during, start, finish와 equal을 제외한 나머지의 역관계로 정의하는 Allen²⁾의 시간 관계 모델에 기반한다. Allen의 시간 논리는 두 시간 간격에 대한 가능한 모든 상대적인 시간 관계성을 표현할 수 있다는 장점은 있으나, 두 가지 행위의 관계성에 내재된 시점들을 정확하게 표현하기 어렵다는 단점을 지니고 있다^{1), 6, 7).}

Allen의 13가지 시간 관계성은 시스템에서 지연 객체를 지원하고 멀티미디어 객체는 지연 객체와 결합

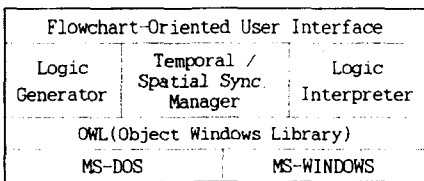
할 수 있다는 조건이 만족된다면 순차 관계성과 병행 관계성만으로도 보다 간결한 형태로 정보를 표현할 수 있다^{4, 7}. 그러나 이들이 제안한 방법은 모든 시간 관계적 요소들이 객체를 표시할 때 발생하는 시간 지연 등에 대해서는 완전한 보장이 어렵다^{5, 6}.

공간 동기화(spatial synchronization)에서는 멀티미디어 프리젠테이션 편집시 사용자에게 의해 결정되는 표시 매체의 가시적 크기(visible size)와 접속점(attachment point)을 일치시키기 위해 표시 매체의 크기를 조절할 수 있는 스케일링 기능을 정의하고 있다⁸. 따라서 MHEG에 정의된 스케일링 기능을 이용하여 표시 매체의 가시적 크기와 접속점을 일치시키기 위해서는 비트맵으로 저장되어 있는 이미지와 비디오의 표시 매체 데이터의 크기를 가시적 크기로 조절할 수 있는 스케일링 기능을 시스템에서 제공해야만 한다¹³.

3. 시스템의 설계

3.1 시스템 구성

(그림 1)은 본 연구의 시스템 구성도이다. 본 시스템은 메모리 관리 및 사용자 인터페이스를 효율적으로 지원하기 위해 MS-WINDOWS에서 구현하였으며, GUI 환경 구축을 위하여 Borland사의 OWL을 사용하였다.

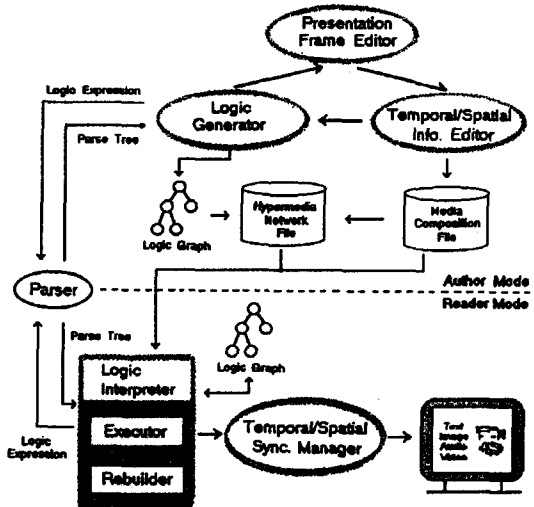


(그림 1) 시스템 구성도
(Fig. 1) System architecture

계층 3은 흐름도 방식의 로직 생성기와 시·공간 동기화 관리기 그리고 로직 인터프리터로 구성된 본 시스템의 엔진 부분이다. 계층 4는 본 시스템의 사용자 인터페이스 계층으로서 사용하기에 편리하도록 흐름도 방식의 인터페이스를 제공한다. (그림 2)는 (그림 1)의 계층 3에 대한 상세도이다.

3.2 Icon 중심의 로직 생성기

로직 생성기에서는 타이틀의 흐름을 중시하여 설계자가 타이틀을 구성하고자 하는 의도대로 아이콘을 흐름도 위에 동적으로 배열하여 로직을 생성한다. 이를 위하여 인간 사고의 연상 작용을 모방하여 무작위로 노드를 연결하는 하이퍼미디어 구조와는 달리, 설계하는 타이틀의 흐름을 구조적 프로그래밍 기법의 순차, 분기, 반복의 세가지 제어 구조를 이용하여 표현한다. 이를 위하여 아이콘 집합은 기존의 프로그래밍 언어에서 나타날 수 있는 행동양식을 흐름도에서 표현할 수 있도록 설계하였다.



(그림 2) 시스템 계층 3의 상세도
(Fig. 2) Detailed architecture of the system layer 3

본 시스템에서는 제어 구조를 표현하기 위하여 조건(분기) 아이콘과 반복 아이콘만을 설계하였다. 순차제어 구조의 경우에는 흐름도 상에서 시작점부터 순차적으로 나열된 아이콘들을 실행하는 방법을 통하여 표현할 수 있으므로, 특별히 아이콘을 설계하지 않았다. 분기 제어구조와 반복제어 구조의 경우에는 두 제어 구조가 모두 조건문을 가진다는 공통점이 있으나 흐름의 차이점을 고려하여 조건 아이콘과 반복 아이콘을 두었다.

생성된 로직은 이진 트리로 구성되며 트리의 각 노드는 자신의 타입과 위치 좌표, 해당 아이콘의 속성 값 등을 가진다.

3.3 시·공간 동기화 모델의 설계

멀티미디어 데이터에 함축된 의미를 사람에게 효과적으로 전달하기 위해서는 미디어 데이터 간의 시·공간 정보를 효율으로 처리할 수 있는 기법이 필요하다. Allen이 제안한 시간 관계 모델들은 시간 동기화(temporal synchronization)를 위하여 두 미디어 데이터 간의 시간 관계를 단편적으로 나타낼 수는 있으나, 여러 미디어 데이터 간의 시간 관계를 복합적이고 유기적으로 나타낼 수 없다^{3, 6}. 또한 공간 동기화(spatial synchronization)를 위해서는 표시 매체의 가시적인 크기와 접속점을 일치 시키기 위해 표시 매체의 데이터의 크기를 가시적인 크기로 조절 할 수 있는 스케일링 기능이 시스템에서 제공되어야 한다^{9, 13}.

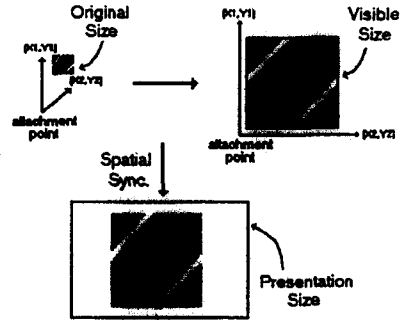
본 논문에서는 멀티미디어 데이터를 표시하는데 필요한 미디어 데이터들을 결합하여 이들 간의 복합적인 시간 관계 구조를 효율적으로 표현할 수 있는 멀티미디어 표현 언어인 MRL(Multimedia Representation Language)을 설계하였고, 멀티미디어 프리젠테이션에서 편집한 표시 매체의 크기를 가시적인 크기로 스케일링할 수 있는 표시 화면 편집기를 개발하여 공간 동기화를 수행하도록 설계하였다.

(1) 공간 객체 합성 모델

공간 동기화란 programming by demonstration이라 불리는 가시화 기술⁸에 근거하여 사용자에 의해 인식될 표시 매체들의 공간 상에서의 가시적 크기와 접속점을 일치시키는 것이다.

공간 상의 가시적인 크기는 사용자에 의해 설정된 공간으로 표시 매체가 출력 장치를 통해 표시될 때 사용자에 의해 인식될 수 있는 표시 매체의 표시 공간을 의미하며, 접속점은 출력 장치의 표시 공간을 나타내는 표시 크기(presentation size)안에서 가시적 크기를 결정하는 좌표이다. 이와 같은 표시 매체의 가시적 크기와 접속점의 일치를 공간 동기화라 한다^{9, 11}.

그래픽 프리미티브의 경우 접속 시작점과 접속 끝점의 좌표가 바로 가시적 크기에 해당되고 이들 접속점에 따른 가시적 크기에 대한 스케일링이 필요하지 않은 반면 이미지, 비디오와 같은 매체는 표시될 영역의 접속 시작점과 접속 끝점에 따라 상대적으로 표시될 매체의 데이터들을 가시적인 크기로 스케일링해 주어야 한다. (그림 3)은 공간 동기화의 처리 과정



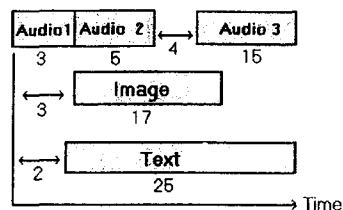
(그림 3) 공간 동기화 처리 모델
(Fig. 3) Spatial synchronization process model

을 모형화한 것이다.

(2) 시간 동기화 모델

MRL에서는 Allen이 제안한 13가지의 시간 관계 명세 중에서 before, meets, overlap, finish, during의 시간 관계는 overlap으로 표현하고, equal, starts의 시간 관계는 equal로 표현하여 기존의 Allen이 제안한 시간 관계를 두 가지 형태로 구분하여 다양한 시간 관계 명세를 단순화 시켰다. (그림 4)는 멀티미디어 시나리오의 예를 나타내고 있다.

(그림 4)에서 Audio1의 데이터가 3초 간 출력되는 동안 시점으로 부터 2초가 되는 순간에 Text 데이터가 25초 동안 출력된다. 그리고 Audio1의 데이터가 완전히 출력됨과 동시에 Audio2의 데이터가 5초 동안 출력되며, 같은 시간에 화면의 좌측 상단에 Image가 17초 동안 출력된다. 또한 Audio2의 데이터가 완전히 출력된 4초 후에 Audio3와 Video 데이터가 15초 동안 출력되어 프리젠테이션의 시점으로 부터 27초가 되는 시간에 Text 데이터와 함께 종료된다.



(그림 4) 멀티미디어 시나리오의 예
(Fig. 4) An example of multimedia scenario

위의 시나리오를 Allen이 제안한 시간 관계성으로 표현하면 다음과 같다.

(Audio1 overlap (Text overlap ((Audio2 equal Image) overlap (Audio3 equal Video))))

본 논문에서 제안한 멀티미디어 표현 언어인 MRL은 사용자의 표현 시간 명세를 단순한 리스트 형태로 나타냄으로써 매체 상호간의 복잡한 시간 관계에 대해서 보다 효율적인 표현력을 제공한다. (그림 5)는 MRL의 정규형을 나타낸 것이다.

```

stmt -> (OP)
OP -> OPcode lterm term' | ε
OPcode -> O | E
lterm -> [
rterm -> ]
term -> oper' rterm | oper' rterm OP
oper -> ( PT, ET, ED, DP )

PT : 객체 표시 준비시작 시간
ET : 객체 표시 시작시간
ED : 객체의 표시기간
DP : 객체의 데이터 포인터
    
```

(그림 5) MRL의 정규형
(Fig. 5) Regular expression of the MRL

(3) 매체 합성 화일의 설계

본 연구에서는 멀티미디어 데이터를 효율적으로 저장하기 위해 별도의 매체 합성 파일(media composition file)을 설계하였다.

매체 합성 파일의 각 레코드는 멀티미디어 프리젠테이션시 하나의 프레임에 해당되는 데이터들을 저장하기 위한 데이터 블록을 나타내는 것으로 사운드를 제외한 다른 매체들은 공간 상에 표시될 가시적 크기와 접속점들을 가지고 있다. 이 가시적 크기와 접속점을 이용하여 공간 동기화를 수행한다. 사운드, 이미지, 비디오 등은 독립적으로 존재하는 화일의 저장된 위치를 나타내는 화일 경로를 포함하는 구조로 설계하였다⁷⁾.

텍스트의 경우 내용의 크기를 1024 바이트로 고정하였고 나머지 매체들은 가변적인 크기를 갖도록 설계

하였다. 해당 프레임에서 미디어들 간의 시간 관계를 나타내는 동기화 정보는 블록의 마지막에 저장된다.

3.4 로직 인터프리터의 설계

로직 인터프리터(logic interpreter)는 로직 생성기에서 생성된 로직 그래프를 분석하여 제어의 흐름에 따라 타이틀을 실행하고 표시 매체(presentation medium)인 화면에 표시하는 기능을 가지며, 구문 분석기, 로직 실행기 및 로직 재구성기로 구성된다.

구문 분석기에서는 로직 그래프의 제어문에 대한 구문 트리를 생성하여 로직 실행기로 전달하고, 로직 실행기에서는 로직 그래프를 실행할 때 구문 트리를 이용하여 제어문을 평가하여 수행한다. 로직 재구성기는 화일에 저장되어 있는 로직 그래프 정보를 읽어 들여 로직 그래프를 재구성하여 로직 실행기로 전달한다.

(1) 구문 분석기

구문 분석기(parser)는 타이틀 제어의 흐름을 표현하는 조건·반복문이나 변수에 값을 할당하는 배경문을 입력받아 토큰을 분리하여 구문 트리를 구성한다. 구성된 구문 트리(parse tree)에 의거하여 문장의 문법을 검사한다.

이를 위하여 올바른 구문을 갖는 문장에 대해서는 생성된 구문 트리에 대하여 자료형 검사를 수행한다. 자료형 검사는 구성된 구문 트리에서 연산자 노드의 양쪽 노드의 자료형의 일치 여부와 연산자 노드의 양쪽 노드가 연산자로 연산할 수 있는 자료형인가 하는 것이다. 오류가 없으면 생성된 구문 트리를 로직 실행기로 전달하여 타이틀 실행 정보로 활용한다.

(2) 로직 실행기

로직 실행기(logic executor)는 로직 생성기에서 생성된 로직 그래프나 로직 재구성기에서 재구성된 로직 그래프를 분석하여 로직 그래프의 실행 순서에 따라 노드의 타입에 해당하는 아이콘 객체에게 메시지를 전송하여 타이틀을 실행한다.

로직 그래프는 root→right→child의 순서로 실행하며, 실행할 노드가 매체 합성 노드이면 페이지 식별자를 동기화 관리자(synchronization manager)로 전달하여 매체 합성 화일에 저장된 페이지 정보를 분석

한 후 표시 배치 상에 페이지를 구성하는 매체들을 표시한다.

실행할 노드에 포함된 객체의 타입이 조건·반복·배정 아이콘일 경우에는 노드에 포함된 조건·반복문이나 배정문을 구문 분석기로 전달하여 오류를 검사하고, 생성된 구문 트리를 바탕으로 의미를 분석하여 수행한다.

(3) 로직 재구성기

로직 재구성기(logic rebuilder)는 하이퍼미디어 네트워크 화일(hypermedia network file)에 저장되어 있는 타이틀 정보를 읽어들이 실행할 경우에 화일을 분석하여 로직 그래프를 재구성하여 로직 실행기로 전달한다.

로직 실행기는 로직 생성기에서 생성된 로직 그래프를 입력으로 하여 로직의 실행 순서에 따라 노드 타입에 해당되는 아이콘에게 메시지를 전송하여 타이틀을 실행한다.

로직 재구성기는 로직 생성기에서 생성된 로직 그래프를 실행할 경우에는 필요치 않으나 화일로 저장되어 있는 타이틀을 읽어들이 실행할 경우에 저장되어 있는 로직을 분석하여 로직 트리를 재구성하고 로직 실행기에서 타이틀을 실행할 수 있도록 한다.

4. 시스템의 구현

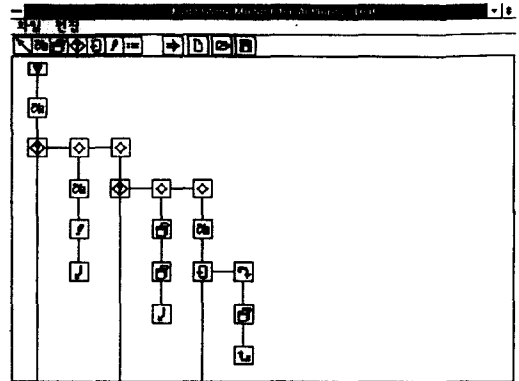
4.1 로직 생성기

로직 생성기에서는 설계자의 의도에 의해 제작되어 있는 흐름도를 바탕으로 로직을 생성할 수 있다는 것과 마우스 조작에 익숙한 사용자들을 고려하여 drag and drop 방식 대신에 한번 선택한 아이콘을 흐름도에 여러번 삽입할 수 있는 click and drop 방식을 택하였다.

로직 생성기는 도구 윈도우(tool window)와 표시 윈도우(display window)로 구성되는데, (그림 6)은 로직 생성기의 실행 화면을 나타낸 것이다.

(1) 도구 윈도우

도구 윈도우는 아이콘 객체들로 구성된다. 아이콘 객체들은 도구 클래스(tool class)를 아이콘 클래스의 일반화 클래스로 둬으로써 객체지향 기법으로 설계



(그림 6) 로직 생성기의 사용자 인터페이스
(Fig. 6) User interface of the logic generator

하여 전체적인 복잡도를 줄일 수 있었다. 또한 정의된 여러 객체 간의 상호 관계를 통한 객체 속성의 상속(inheritance)이 가능하고 각 객체의 정보를 하나로 묶어 캡슐화(encapsulation)함으로써 시스템 조작 및 운용이 용이하며 객체 단위의 재사용성(reusability)을 증진시켰다.

(2) 표시 윈도우

표시 윈도우는 도구 윈도우에서 선택된 아이콘 객체의 아이콘 객체에게 메시지를 보내 로직 그래프를 갱신(update)하게 하고 갱신된 그래프에 대응하는 흐름도를 화면 상에 도시한다. 도구 윈도우에서 선택된 아이콘이 객체 선택 아이콘일 경우에는 선택된 아이콘 객체로 메시지를 전송한다.

(3) 로직 그래프의 생성

로직 그래프는 각 아이콘 객체들의 속성값을 할당 한 노드들을 이진 트리로 구성한 것이다. 각 노드는 자신의 오른쪽에 위치한 노드의 주소와 자신의 아래쪽에 위치한 노드의 주소를 포함하고 해당 위치에서의 아이콘의 속성값과 자신이 표시된 위치의 좌표값을 갖는다. (그림 7)은 로직 그래프를 구성하는 알고리즘이다.

로직 그래프의 생성 과정은 흐름도의 확장 과정이다. 따라서 흐름도가 화면 상에서 수직 확장/수평 확장되는 것과 대응되도록 로직 그래프 내 노드들의 X 좌표와 Y 좌표를 재배치하는 것이 고려되어야 한다.

```

Procedure GenerateGraph
begin
  if (Header==NULL) then Create Header
    // Generate header
  else if (CurrentPosition<HeadPosition)
    // If the position of mouse is smaller than
    // the position of header
    {
      Create NewNode
      NewNodeChild=HeaderChild
      Header=NewNode
    }
    // Set new node to a header of logic graph
  else if (IsPrevNodeExist)
    {
      Temp=PrevNode
      // Determine existing node
      // which connect new node
      Create NewNode
      NewNodeChild=TempChild
      TempChild=NewNode
    }
end
    
```

(그림 7) 로직 그래프 생성 알고리즘
(Fig. 7) An algorithm for generation of a logic graph

로직 그래프가 이진 트리로 구성되어 있으므로 X 좌표는 로직 그래프를 전위 탐색(preoder search)법을 이용하여 재배치하고, Y 좌표는 후위 탐색(postoder search)법을 이용하여 재배치한다.

(그림 8)은 노드의 X 좌표를 재배치하는 알고리즘을 pseudo 코드로 나타낸 것이다. Y 좌표를 재배치하는 알고리즘은 (그림 8)의 알고리즘과 거의 유사한 알고리즘으로 구현할 수 있다.

생성된 로직 그래프의 화일 형식에서 Node는 로직 그래프의 노드 객체를 의미하며 Node 레코드의 Attr (객체 속성) 필드는 각 아이콘마다 다른 형식을 가진다. 예를 들어 매체합성 객체 노드는 Attr 필드가 표시 화면 편집기(presentation frame editor)에서 생성한 페이지 식별자(page identifier)를 저장하고, 조건 객체 노드는 로직 구성시 설정한 조건문(condition statement) 이 문자열(string) 형태로 저장되며, 반복 객체 노드의 경우에는 반복 조건문이나 반복 회수를 저장한다.

4.2 시·공간 동기화 관리기

멀티미디어 프리젠테이션시 최적의 시·공간 동기화 조건을 만족하기 위해서는 멀티미디어 데이터 간

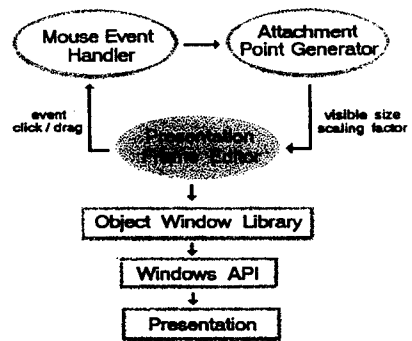
```

Procedure AdjLocationX(Temp,LocationX,max)
begin
  Temp->LocationX=LocationX;
  if (Temp->Child) // If child node exist
    AdjLocationX(Temp->Child,Temp->LX,max)
  if (Temp->Right) // If right node exist
    if (ChildNode exist)
      // If current node has child node
      AdjLocationX(Temp->Right,LocalMaxX(Temp,
        Temp,LX),max)
      // Compute max. X coordinate of child node
    else
      AdjLocationX(Temp->Right,NextLX,max)
    else
      AdjLocationX(Temp->Right,NextLX,max)
end
    
```

(그림 8) 노드의 X 좌표를 재배치하는 알고리즘
(Fig. 8) An algorithm for replacement of each node's X-coordinate

의 시·공간적 오차가 발생해서는 안된다. 즉 저장된 멀티미디어 데이터들을 컴퓨터 내부에서 표현 가능한 형태로 부호화하며 부호화된 멀티미디어 데이터들을 프리젠테이션의 시나리오에 맞추어 표시의 작업을 수행시 발생하는 시간 지연 및 가시적 크기와 접속점의 일치를 위해 표시 매체의 크기를 스케일링 할 수 있어야 한다^{6,13).}

이를 위하여 본 논문에서는 표시 매체의 준비 시간과 실제 이들 매체가 표시될 때 발생하는 동적인 지연시간을 보장하기 위한 알고리즘을 개발하고 표시 매체의 크기를 스케일링 할 수 있는 표시 화면 편집기를 구현하였다.



(그림 9) 표시화면 편집기의 실행 메카니즘
(Fig. 9) Execution mechanism of the presentation-frame editor

(1) 표시 화면 편집기의 구현

이벤트 중심(event oriented)의 윈도우 상에서 가시적 크기와 접속점의 추출은 마우스 이벤트 핸들러(mouse event handler)를 통한 이벤트의 핸들링으로 가능하다. (그림 9)는 표시 화면 편집기의 실행 메카니즘을 도형화한 것이다.

여기에서 마우스 이벤트 핸들러에 의해 인식된 이벤트는 접속점 생성기(attachment point generator)에 의해 가시적 크기와 접속점이 산출된다. 산출된 가시적 크기와 접속점은 표시 화면 편집기에서 공간 동기화에 필요한 스케일링 인자(scaling factor)로 사용하여 공간 동기화를 수행한다.

(2) 시간 동기화의 구현

주변 장치에 저장된 멀티미디어 데이터들을 컴퓨터 내부에서 표현 가능한 형태로 부호화하며 부호화된 미디어 데이터의 표시 작업 수행시 발생하는 시간 지연 등에 대한 대책이 마련되어야 한다. 이를 위하여 본 논문에서는 표시 매체의 준비 시간과 실제 이들 매체가 표시될 때 발생하는 동적인 지연시간을 보정하기 위한 알고리즘을 제안한다.

① 준비시간 수집 알고리즘

매체가 표시에 이르기 까지에는 많은 준비시간이 요구되며 각 매체의 속성에 따라 그 시간은 정확한 예측이 불가능하기 때문에⁴⁾, 본 연구에서는 다양한 특성의 매체를 결합하기 위해 다음과 같은 준비시간 수집 알고리즘을 제시하였다.

첫째, 준비시간 수집 알고리즘에서는 시간적으로 부담이 가장 큰 매체 데이터의 전송 시간에 우선하여 스케줄링한다.

둘째, 각 멀티미디어 매체의 전송 시간은 표시될 매체의 데이터 크기에 비례하며, 저장 장치의 seek time, access time 및 데이터 전송률은 모든 매체에 대하여 동일하다.

(그림 10)은 준비시간 수집 알고리즘을 기술한 것이고, (그림 11)은 표시 시작시간 보정 알고리즘을 기술한 것이다.

② 표시 시작시간 보정 알고리즘

데이터의 압축률 및 분해능에 따라 디코딩 소요 시

```

get So
Sn=sum(So)/lastET
// determine preparation duration
for(O1,O2,...,On) {
    switch (node type) {
        case 0 : PD=So/Sn
                PT=ET-PD
        case E : PD(LC)=So(LC)/Sn
                // for left child
                PT=PT(LC)=ET(LC)-PD(LC)
                PD(RC)=So(RC)/Sn
                // for right child
                PT=PT(RC)=ET(RC)-(PD(RC)+PD(LC))
    }
}
// determine preparation starting time
for(O1,O2,...,On) {
    if (PT>beforeET) break
    // beforeET = ETr-1, ETr-2, ..., ET1
    else PT=PT-sum(beforePD)
    // beforePD = PDr-1, PDr-2, ..., PD1
}
// adjusting PT for all object
PT=PT+min(abs(PT))
    
```

(그림 10) 준비시간 수집 알고리즘
(Fig. 10) An algorithm for collection of make-ready time

```

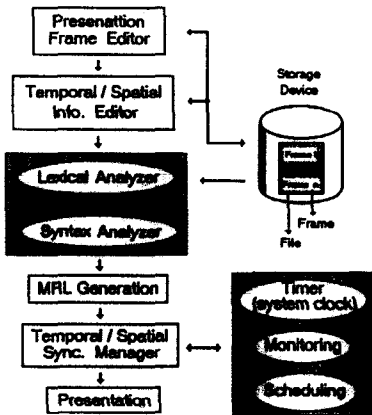
while {
    switch (node type) {
        case 0 : dET=ET'-ET
                // dET is presentation time difference
                ET=ET+dET
                rET=dET/ET-PT
                // rET is difference ratio
        case E : dET(LC)=ET'(LC)-ET(LC)
                dET(RC)=ET'(RC)-ET(RC)
                dET=max(dET(LC),dET(RC))
                ET(LC)=ET(LC)+dET
                ET(RC)=ET(RC)+dET
                rET=max(dET(LC)/ET(LC)-PT(LC),
                        dET(RC)/ET(RC)-PT(RC))
    }
    afterET=afterET*(1+rET)
    // afterET = ETr+1, ..., ETr-1, ET1, ET2
}
    
```

(그림 11) 표시 시작시간 보정 알고리즘
(Fig. 11) An algorithm for compensation of presentation-
beginning time

간이 가변적일 뿐 아니라 오디오 또는 비디오 디바이스 드라이버의 하드웨어적인 동작 특성은 사용자가 요구하는 미디어를 정확한 시간에 표시하는데 큰 변수로 작용한다.

표시 시작시간 보정 알고리즘은 멀티미디어의 데이터 표시시에 나타나는 지연 시간을 해결하기 위해 미디어 매체를 실행할 때마다 매체의 준비시간 수집 알고리즘에서 계산된 값과 실제로 측정된 준비 소요 시간의 차를 구하여 그 차이에 비례하는 시간만큼 시작 시간을 계속해서 보정해주도록 하는 구조이다⁶⁾.

(3) 시·공간 동기화 관리기의 실행



(그림 12) 시공간 동기화 정보의 처리 과정

(Fig. 12) Procedure of the temporal/spatial synchronization information

(그림 12)는 시·공간 동기화 관리기의 전반적인 수행 과정을 나타낸 것으로서, 다음과 같은 과정을 수행한다. 표시 화면 편집기를 이용하여 하나의 프레임을 작성한 후 시간 편집기에서 각 미디어들의 시간 관계 명세를 기술한다.

기술된 시간 관계 명세는 어휘 분석기를 통하여 올바르게 기술되었는지 분석하고 의미 해석기를 거쳐 준비시간 수집 알고리즘 수행을 위한 세멘틱(semantics)을 분석하여 MRL을 생성한다. 그 후 생성된 MRL을 이용하여 시공간 실행기에서는 생성된 MRL의 시작 시간을 체크하기 위해 시스템 클럭을 이용한 모니터링 기법을 이용하여 계산된 동적인 실행 시간에 맞추어 표시를 수행한다.

4.3 로직 인터프리터의 구현

(1) 구문 분석기

현재 실행할 노드가 포함하는 객체 타입이 조건·반복·배정 아이콘 객체일때 구문 분석기는 분기·반복 조건문과 배정문에 대하여 구문 분석을 수행한다. 구문 분석기의 작업 과정은 다음과 같다.

Step 1: 표현식에서 토큰을 분리한다. 본 시스템에서는 변수명, 숫자나 문자, 문자열 상수, 연산자를 토큰으로 정의하였다. 분리된 토큰은 연산자 우선순위에 따라 후위 표기법(postfix notation)으로 변환한다. 이때 후위 표기법은 토큰의 리스트로 구성된다.

Step 2: 후위 표기법으로 변환된 수식으로 구문 트리를 구성한다. 구문 트리의 중간 노드에는 연산자가 위치하고 있 노드에는 변수나 상수가 위치하게 된다.

Step 3: 구성된 구문 트리에 의거하여 표현식의 문법 오류를 검사한다. 문법의 오류를 검사할 때의 점검사항은 다음과 같다. 다음 점검 사항 중 해당 항목이 있으면 표현식의 문법에 오류가 있다는 것을 의미한다.

- ① 구문 트리가 완전히 구성된 후에도 후위 표기 리스트에 토큰 노드가 남아있는 경우
- ② 구성된 구문 트리에서 연산자인 중간 노드의 양쪽 노드의 자료형이 다른 경우
- ③ 구성된 구문 트리에서 중간 노드인 연산자의 양쪽 노드가 연산자로 연산할 수 없는 자료형인 경우
- ④ 앞 노드의 토큰이 문자일 때, 토큰이 문자 상수나 문자열 상수가 아니고, 변수가 아닌 경우

(2) 로직 재구성기

로직 그래프를 구성할 때의 순서는 로직 그래프를 저장할 때와 같다. 로직 그래프의 저장시에 저장된 "CR" 정보를 분석하여 "CR" 리스트를 구성한 후 로직 그래프의 재구성시에 노드가 적합한 위치에 연결되도록 한다.

화일에서 노드 정보를 읽어들이며 하나의 노드를 생성하고 로직 그래프에 연결시킬 때는 이전까지의 "CR" 리스트를 검색하여 "CR" 리스트의 노드 중에

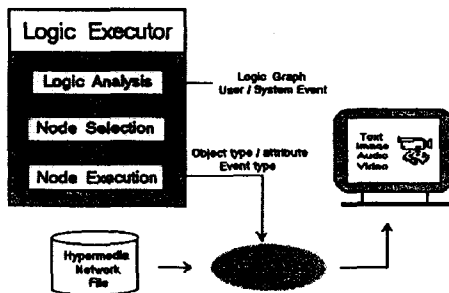
서 가장 뒤에 C나 R이 존재하는 노드까지의 수를 기억한다. 그리고 로직 그래프를 탐색하면서 하나의 노드를 만날 때마다 기억한 수를 감소시킨다. 기억한 수가 0이 되는 노드에서 해당 노드의 C가 세트되어 있으면 노드의 자식 노드로 연결하고 R이 세트되어 있으면 오른쪽 노드로 연결한다. 연결한 후에는 연결한 노드에 해당되는 "CR" 리스트의 노드의 C나 R을 삭제한다.

C와 R이 동시에 세트되어 있는 경우는 로직 그래프의 저장 순서가 root→child→right이므로 C가 R에 우선한다. 새로 화일에서 읽어 생성하여 로직 그래프에 추가된 노드의 "CR" 정보는 "CR" 리스트에 추가시킨다.

(3) 로직 실행기

로직 그래프를 구성하는 노드들은 아이콘에 전송할 속성값을 포함하므로 로직 그래프에서 한 노드를 실행할 때는 노드에 포함된 아이콘의 속성값을 해당 아이콘 객체에 전송하고 발생한 이벤트에 따라 아이콘 객체를 호출한다.

아이콘 객체는 전송받은 값을 자신의 속성으로 설정한 후에 발생한 이벤트를 처리하는 멤버 함수를 실행한다. 로직 그래프는 화면상의 흐름도의 시작점에서부터 끝점까지 수직방향으로 실행되며, 실행 중 오른쪽 노드가 존재하는 노드를 만나면 오른쪽 노드를 아래쪽 노드보다 먼저 실행한다.



(그림 13) 로직 실행기의 실행 과정
(Fig. 13) Execution procedure of the logic executor

실행할 노드가 포함된 아이콘 객체의 종류에 따라 다른 동작을 수행한다. 매체 합성 객체는 노드 식별자

에 따라 다매체 합성 화일을 읽어 실행 윈도우에 출력하며 이벤트 객체는 실행 윈도우상에서 마우스 이벤트를 받아들여 설정된 영역과 일치하는지 검사한다.

조건 객체, 반복 객체, 배정 객체의 경우는 구문 분석기를 먼저 수행하여 구문 트리를 구성하고, 구성된 구문 트리를 바탕으로 표현식에 따른 작업을 수행한다. 특히 반복 객체는 타이틀 실행시에 한번 이상 표현식이 평가되므로 반복 아이콘을 실행할 때마다 구문 트리를 구성하고 표현식 수행을 반복 실행한다. (그림 13)은 로직 실행기의 작업 과정을 나타낸 것이다.

5. 결 론

본 논문에서는 흐름도 방식으로 멀티미디어 타이틀을 저작할 수 있는 흐름도 방식의 저작도구인 KMAT 시스템을 설계하고 구현하였다. 이를 위하여 타이틀 로직을 흐름도로 표현할 수 있도록 아이콘 위주의 사용자 인터페이스를 지원하는 로직 생성기와 미디어 간의 시·공간 동기화를 효율적으로 관리하는 시·공간 동기화 관리자 그리고 로직 생성기에서 생성된 로직을 번역 및 실행하는 로직 인터프리터를 설계/구현하였다.

본 시스템은 타이틀 저작자가 설계한 타이틀 로직을 쉽게 흐름도 방식으로 표현할 수 있도록 하였다. 이를 위하여 아이콘 위주의 시각 프로그래밍 기법을 이용하여 제어의 흐름과 상호작용을 정의하였다. 또한 객체지향 기법으로 설계하여 사용자 인터페이스를 설계하는 것과 동시에 프로그램의 구현이 가능하였으며, 흐름도 방식의 저작도구의 사용에 익숙치 않은 사용자라도 사용하기 쉽도록 단순하고 기본적인 기능만을 구현하였다. 기능이 추가로 필요할 경우에는 해당 기능을 클래스로 정의하여 시스템에 추가하여 기능을 확장할 수 있으므로 시스템의 확장이 용이하다.

또한 본 논문에서는 멀티미디어의 시간적 동기화 관계를 표현할 수 있는 효율적인 동기화 표현 언어인 MRL을 이용하여 다양한 시간적 관계성 명세를 단순한 형태로 표현할 수 있도록 하였으며, 표시 매체의 가시적 크기와 접속점을 일치시키는 공간 동기화의 기능을 수행 하는 표시 화면 편집기를 개발하였다. 아울러 매체 표시할 때의 준비 시간과 시작 시간의

동적 생성을 위한 알고리즘을 제안하여, 사용자에게 기술된 매체의 불완전한 시작 시간을 표시 시작시간 보정 알고리즘에 의해서 완전하게 보정하여 표시할 수 있도록 하였다.

참 고 문 헌

[1] Ralf Steinmetz, "Synchronization Properties in Multimedia Systems", IEEE Journal on Selected Areas in Communication., Vol. 8, No. 3, April 1990, pp. 401-412.

[2] J.F. Allen, "Maintaining Knowledge about Temporal Intervals", Comm. of the ACM, November 1983, pp. 832-843.

[3] T.D.C. Little, Arif Ghafoor, "Multimedia Object Models for Synchronization and Database", Intl Conf. on Data Engineering, 1990, pp. 20~27.

[4] B. Prabhakaran and S.V. Raghavan, "Synchronization Models for Multimedia with User Participation", ACM Multimedia, 1993, pp. 157~166.

[5] T.D.C. Little, Arif Ghafoor, "Synchronization and Storage Models for Multimedia Objects", IEEE Journal on Selected Areas in Communication., Vol. 3, No. 3, April 1990, pp. 413~427.

[6] 여인국, 황대훈, "지연시간의 변화를 고려한 연속 미디어 표현 메카니즘의 설계 및 분석", 한국정보처리학회 논문지, Vol. 2, No. 5, pp. 787~796, Sep. 1995.

[7] 이동재, 여인국, 황대훈, "멀티미디어의 동기화를 위한 시간 관계 명세 모델 설계", '94 춘계 학술 발표 논문집, 제17권 1호, 대한전자공학회, 1994, pp. 650~653.

[8] Shi-Kuo Chang, "Visual Languages: A Tutorial and Survey", IEEE Software, pp. 29~39, Jan. 1987.

[9] Seong Bae Eun, Eun Suk No, "Specification of Multimedia Composition and A Visual Programming Environment", ACM Multimedia 93, pp. 167~173.

[10] Pankaj K. Garg, "Abstraction Mechanisms In Hypertext", CACM, Vol. 31, No. 7, 862~870, Jul. 1988.

[11] Rei Hamakawa, Jun Rekimoto, "Object Composition and Playback Models for Handling Multimedia Data", ACM Multimedia 93, pp. 273~281, Aug. 1993.

[12] Roger Price, "MHEG: An Introduction to the Future International Standard for Hypermedia Object Interchange", ACM Multimedia 93, pp. 121~126, Aug. 1993.

[13] Information Technology-Coded Representation of Multimedia and hypermedia Information Objects (MHEG).

[14] P.K. Andleigh, K. Thakrar, "Multimedia Systems Design", Prentice-Hall, 1996.

[15] 이만재, "멀티미디어 저작시스템", 정보과학회지, pp. 52~59, Oct. 1992.

[16] Lynda Hardman, Guido vanRossum, Dick C.A. Bulterman, "Structured Multimedia Authoring", ACM Multimedia 93, pp. 283~289, Aug. 1993.

[17] John A. Waterworth, "Multimedia technology and applications", Ellis Horwood, 1991.



여 인 국

1984년 명지대학교 전자공학과 졸업(공학사)
 1995년 경원대학교 대학원 전자계산학과(공학석사)
 1984년~1991년 국방과학연구소 연구원
 1991년~현재 한국생산기술연구원 선임연구원

관심분야: 멀티미디어 시스템, 영상처리, 컴퓨터조직 응용



황 대 훈

1977년 동국대학교 수학과 졸업(학사)
 1983년 중앙대학교 대학원 전자계산학과(공학석사)
 1991년 중앙대학교 대학원 전자계산학과(공학박사)
 1983년~1985년 한국산업경제기술연구원(KIET) 연구원

1987년~현재 경원대학교 전자계산학과 부교수
 1995년~현재 경원대학교 전자계산소장
 관심분야: 멀티미디어 시스템, FA, 가상현실 등