

# 페트리 넷으로 표현된 멀티미디어 시나리오의 재생기 구현

한 승협\*, 임 재걸, 이 계영

(동국대학교 전자계산학과)

## Implementation of the Player for Petri-Net-based Multimedia Scenario

Seunghyup Han, Jaegool Yim, and Gyeoung Lee

Dept. of Computer Science Dongguk University at Kyung-Ju

### 요 약

동기화 전략을 명시하는 방법으로 시간 구간 명시, 시간축 명시, 레퍼런스 포인트를 두는 방법, 페트리넷을 이용하는 방법 등 매우 다양한 연구 결과가 소개되었다. 본 논문은 기존의 멀티미디어 시나리오의 동기화 명시를 위한 페트리넷 방법[1-3]을 확장하여, 페트리넷 동기화 명시를 실현한 멀티미디어 시나리오를 재생하여 주는 재생 시스템을 구현하고, 자료 구조, 멀티프로세싱, 동기화 기법 등을 중심으로 본 재생 시스템을 소개한다. 본 시스템의 특징은 미디어 단위의 시나리오 진행이 가능한 것이다. 멀티미디어 프로그램이 학습에 많이 이용되므로, 물리의 '중력'을 간단하게 설명하는 예제와 더불어 어떻게 실행되는가를 설명한다.

### 1. 서 론

멀티미디어 프로그램 분야에서 가장 뜨거운 연구 분야 중의 하나가 동기화이다[4, 5]. 멀티미디어 동기화 모델링 방법으로 페트리넷에 기반을 둔 방법이 [1-3]에 소개되었다. 한편, 멀티미디어 프로그램을 손쉽게 작성하는 제작 툴 중에는 MINOS, Multos, Toolbook, 세빛[6] 등이 있다. 세빛은 흐름도 방식에 입각한 저작 도구 중 하나이다. 세빛은 먼저 화면 단위의 흐름도를 작성한 다음 각 화면의 구성 요소를 배치한다. 이러한 화면 단위의 작성 방법은 여러 화면에 걸쳐 독립적으로 출력되는 데이터를 표현하는 데 어려움이 있다. 이런 어려움을 해결하기 위하여 변형된 페트리넷을 이용한 시나리오 모델링 방법이 [7]에 소개된다. 본 연구에서는 [7]에 소개된 방법으로 표현된 멀티미디어 시나리오를 해석하고 재생하는 재생기 구성 방법을 제시하며 구현 결과를 보인다.

### 2. 페트리넷으로 표현된 멀티미디어 시나리오(MPN)

변형된 페트리넷(MPN)을 이용한 멀티미디어 시나리오 모델링 방법은 [7]에 소개되었으므로 MPN에 대

한 정형적인 소개는 생략하고, 그림1을 중심으로 예를 들어 보인다. 그림1은 '천재들의 과학백과'라는 학습용 CD-ROM 타이틀에서 중력을 설명하는 부분인데 첫 화면에는 두 개의 이미지 파일(중력.bmp, 스카이다이버.bmp)과 한 개의 오디오 파일(중력.wav)이 동시에 출력되고, 중력.wav의 출력이 종료되면 비디오 파일(무중력.avi)이 출력된다. 이미지 파일들은 비디오 파일이 출력되는 동안에도 계속 화면에 남아 있는다. 이러한 출력 데이터들은 그림1의 원들의 레이블로 표기되어 있음을 알 수 있다.

두 번째 화면과 세 번째 화면은 각각 두 개의 이미

t \ p	0	1	2	3	4	5	6	7	8	9	10	11
0	1	2	2	2								
1				1	2							
2			1	1		1	2	2	2			
3						1	1	1	2	2	2	
4										1	1	1

[표.1] 예제의 격발 행렬

[Table.1] Incidence Matrix of Example

지 파일과 한 개의 오디오 파일로 구성된다. 그림1에는 나타나 있지 않지만 각 원에는 레이블로 지정된 데이터가 출력될 화면상의 위치, 크기 등의 데이터도

연합되어 있다. 이 시나리오 예제의 격발행렬은 [표.1]과 같다. 여기서 플레이스(원)에서 트랜지션(막대)으로의 유향간선은 1로, 그 반대는 2로 표현된다.

### 3. 구현

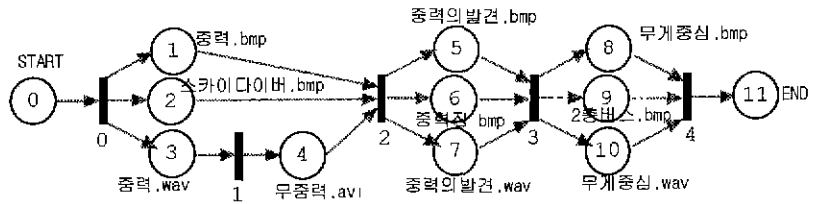
본 논문에서 구현한 시나리오 재생기는 Pentium II - 200MHz, Windows95 시스템에서 Microsoft Visual C++ 5.0, MFC 4.0의 개발 환경에서 구현되었다. 재생기를 이루는 클래스는 크게 데이터와 제어에 관계된 '데이터/제어 클래스'와 멀티미디어의 재생에 관련된 '미디어 재생 클래스'로 나눌 수 있다.

데이터/제어 그룹에 속하는 클래스로는 CDMAEngine, CTPManager, CIMatrix, CWord2DArray 등이 있고, 미디어 재생 클래스 그룹은 각 매체당 하나의 클래스를 가지며 CDIB, CMpeg, CAVI, CWave가 있다. 이들 클래스 이외에 가장 중요한 쓰레드 함수로 Run()이 있다. 이것은 2차원 배열로 구현된 페트리 넷의 격발 행렬을 처리하여 멀티프로세싱시키고 동기화 시키는 함수이다.

#### 3.1 데이터/제어 클래스 그룹

이 그룹의 클래스에는 CDMAEngine, CTPManager, CIMatrix, CWord2DArray 등이 있다. [리스트.1]에 보이는 바와 같이, 클래스 CDMAEngine은 클래스 CIMatrix와 클래스 CTPManager를 public 멤버 변수로 가지고 있다. CIMatrix는 격발 행렬을 표현하는 클래스로 페트리 넷의 흐름에 대한 정보를 갖고 있다.

```
class CDMAEngine
{
private:
    CDC *m_pDC;
    HWND m_hMainWindow;
    // 격발 행렬(Incidence Matrix)에 대한 접근
    // 제어를 위한 동기화 클래스 변수.
    CCriticalSection m_csRead;
    CCriticalSection m_csWrite;
    int m_nReadCounter;
public:
    // 격발 행렬 클래스 변수
    CIMatrix m_cimx;
    // 플레이스/트랜지션 관리자 클래스 변수
    CTPManager m_ctpm;
    CDMAEngine();
    bool SetDC(CDC *);
    bool SetWindow(HWND);
    // 쓰레드가 격발 행렬을 읽기 위해 임계영역을
    // 확인후 대기 또는 획득을 한다.
    void ReadLock(void);
    // 격발행렬을 다 읽은 쓰레드는 판독자 수를 하나
    // 줄인다
    void ReadUnlock(void);
    // 미디어별로 해당 루틴을 호출한다.
    void Distribute(CPlace *);
    void MpegPlayer(CPlace *);
};
```



[그림.1] '중력'을 설명하는 간단한 페트리 넷 시나리오 예제  
[Fig.1] Simple Petri Net-based Multimedia Scenario Example that Explains 'Gravity'

```
void AviPlayer(CPlace *);
void BmpPlayer(CPlace *);
void WavePlayer(CPlace *);
```

1;  
[리스트.1] 클래스 CDMAEngine의 프로토타입  
[List.1] Prototype of Class CDMAEngine

격발 행렬을 다루는 클래스인 CIMatrix, 시나리오를 이루는 미디어 요소(플레이스)들에 관한 정보와 이 플레이스들의 구성 순서와 동기화 그룹, 나아가 동기화 전략(본 논문에는 포함되지 않았다) 등의 정보를 담은 트랜지션들을 저장하고 참조하는 관리자 클래스인 CTPManager, 등 다른 클래스에 대한 설명은 지면이 부족하여 생략한다.

#### 3.2 미디어 재생 클래스 그룹

멀티미디어 저작물에서 프리젠테이션되는 미디어는 각각 하나의 클래스로 구현되어 다뤄진다. 설정과 재생의 단순한 인터페이스를 가지며, AVI, MPEG, Wavc 미디어는 Windows API에서 제공하는 고수준의 멀티미디어 제어 인터페이스(MCI; Multimedia Control Interface)를 이용하였다. 예를들어 MPEG의 경우, Open()이라는 멤버함수에서 디스플레이할 윈도우 핸들, 좌표 등을 파라미터로 설정해주고, Play()라는 멤버함수로 프리젠테이션 한다. 그림의 경우, 소리나 동영상과는 달리 시간성이 없다 따라서 소멸의 처리에 별도의 추가적인 작업이 필요하다.

#### 3.3 쓰레드 함수 Run()

격발 행렬을 읽어 페트리 넷의 흐름을 수행하고 복수 개의 쓰레드를 생성하며 플레이스 재생 함수를 호출하는 역할을 수행하는 함수이다.

```
Run(플레이스 ID);
begin
    플레이스실행(플레이스 ID);
    토른을발사할트랜지션을찾아라();
    if (트랜지션이 없다)
        리턴;
    if (도착하지않은토른이었다)
        리턴;
    병행처리
    begin
```

```

Run(트랜지션이 활성화할 플레이스들);
end

리턴;
end

[리스트 2] 쓰레드 함수 Run()의 알고리즘
[List.2] Algorithm of Thread Function Run()
    
```

이 함수는 플레이스 ID 하나를 받고, 정상적인 수행 시 1을 비정상적인 수행시 0을 반환한다. 먼저 파라미

랜지션을 찾는다. 해당 트랜지션을 찾아 모든 플레이스로부터 토큰이 도착했는지를 검사한다. 만약 하나라도 아직 도착하지 않은 토큰이 있다면, 지금의 쓰레드는 그냥 종료함으로써 트랜지션 활성화(플레이스 쓰레드를 생성하여 처리하는 작업)를 마지막으로 도착하는 쓰레드에게 남긴다.

마지막으로 도착한 쓰레드는 더 이상 기다릴 토큰이 없음을 인지하고 플레이스 수 만큼의 쓰레드를 생성한다. 이때 쓰레드 함수의 전달인자로 해당 플레이스 ID를 넘겨 줌으로써 다시 Run()이 호출 된다

#### 4. 예제실험

[그림.1]을 실험 데이터로 하여 본 재생 시스템을 수행시켜 보았다. [그림.2]에 보이듯 처음 시작되면 0번의 시작 플레이스가 실행(시작 플레이스는 아무일도 하지 않는다.)이 되고, 그 다음 1번, 2번, 3번의 순서로 쓰레드가 만들어져 각각이 병행적으로 수행된다. 3번의 wave 파일이 끝나면 4번 avi파일이 수행된다. [그림.3]

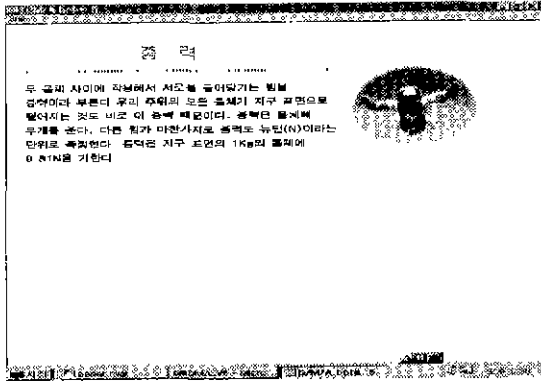
#### 5. 결론

페트리넷을 이용한 멀티미디어 동기화 명시에 대한 연구가 많이 이루어졌다. 본 연구에서는 이러한 기존의 연구를 확장하여 동기화 정보 뿐만 아니라, 시나리오 재생에 필요한 모든 정보를 명시하는 페트리넷 표현 방법을 소개하였으며, 이렇게 표현된 시나리오를 재생하는 재생 시스템 구현 및 실험을 소개하였다.

향후 연구 계획은 재생 시스템을 확장하여 정지, 되감기, 역방향 재생, 빠르게 가기, 등의 기능을 첨가하는 것이다.

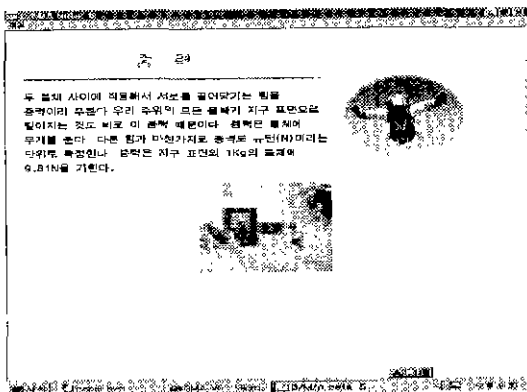
#### 6. 참고문헌

- [1] T Little, A Ghafoor, "Synchronization and Storage Models for Multimedia Objects," IEEE Journal on Selected Areas in Communications, 8, (3), April, 1990.
- [2] M. Diaz, P. Senac, "Time Stream Petri Nets a Model for Timed Multimedia Information," *Application and Theory of Petri Nets 1994*, Springer-Verlag, 1994, pp. 219-238.
- [3] S. Yoo, W. Lee, D. Kim, "Transition Function Petri Net Model for Multimedia Synchronization Specification," Journal of Electrical Engineering and Information Science Vol 1, No 4, Dec. 1996. pp. 1-8.
- [4] R. Steinmetz and K. Nahrstedt, *Multimedia Computing, Communications and Applications*, Reading, Prentice Hall, 1995.
- [5] S.J. Gibbs and D.C. Tschritzis, *Multimedia Programming*, Reading, Addison-Wesley, 1995.
- [6] 정성부 외, *멀티미디어 저작 도구 새빛을 배우자*, 교학사, 1996.
- [7] 임재실, 이계영, "페트리넷을 이용한 멀티미디어 시스템 시나리오 작성기 설계," 동국논집, 제 17집 1호, 1998, pp. 61-74.



[그림.2] 시작후, 1,2,3번 플레이스가 실행되고 있는 화면  
[Fig.2] After Started, Place 1,2 and 3 are being run

터로 받은 플레이스 ID를 트랜지션/플레이스 관리자(TPM; Transition/Place Manager)에게 전달하여 플레이스 클래스에 대한 포인터를 받아온다. 이 포인터를 가지고 다시 CDMAEngine의 Distribute 멤버함수를 호출한다. 이 함수는 플레이스의 종류에 따라 적절한 플레이어 함수를 호출하여 수행하도록 한다. 플레이스 수행이 끝나면, 이 플레이스가 토큰을 전달할 트



[그림 3] 3번 플레이스(wave)가 종료되고 4번 플레이스(avi)가 실행되고 있는 화면  
[Fig.3] After Place 3(wave) finished, Place 4(avi) is being run.