韓國 컴퓨터情報學會 論文誌
第12卷 第3號, 2007. 7.

2007-12-3-1-2

페트리 넷 기반 시나리오 분석 및 제어를 통한 재생기 구현

임 재 걸*, 이 강 재**

Implementation of a Player via Petri Net-Based Scenario Analysis and Control

Jaegeol Yim*, Kangjai Lee**

요 약

본 논문은 페트리 넷 기반의 멀티미디어 프로그래밍 방법을 제안한다. 이를 위하여 멀티미디어 시나리오를 표현할 MPN(Multimedia Petri Net)을 정의하고, 또한 MPN을 분석하여 시나리오의 결함을 찾아낼 수 있는 방법을 소개한다. 멀티미디어 프로그램은 MPN을 해석하여 멀티미디어 시나리오를 재생한다. 이렇게 작성한 우리의 멀티미디어 프로그램은 정지, 빠른 재생, 되감기, 빠른 되감기 등의 조작 기능을 제공한다. 멀티미디어 시나리오 모델링을 위해 페트리 넷을 이용한 사례는 존재하지만, 이들은 모두 시간적 동기화 분석을 위한 것으로, 멀티미디어 시나리오의 재생을 목적으로 사용된 사례는 아직 없다. 이에 반하여, MPN은 시나리오의 흐름은 물론 시나리오를 구성하는 데이터들의 정보까지도 모두 표현함으로써 재생에 필요한 모든 정보를 나타낼 수 있다.

Abstract

This paper introduces a Petri Net-based multimedia programming method. For this purpose, we are proposing MPN(Multimedia Petri Net) which can be used for representing a multimedia scenario. We are also introducing methods to analyze a MPN with which we can detect some kinds of design faults in the scenario. A multimedia program replays the scenario by interpreting the MPN. A method to implement such a multimedia program is also discussed. Our multimedia program provides the manipulation functions of stop, play, fast forward, rewind, and fast rewind. There are many varieties of Petri Net. Several of them are for modeling multimedia scenarios. They all have been used for synchronization analysis. But none of them were used for replaying multimedia scenario. We have extended these nets to MPN. A MPN model contains not only the flow of a scenario but also all the information associated with the data units. Therefore, our player can play the multimedia scenario by interpreting the MPN.

▸ Keyword : Petri Net, Multimedia Petri Net, multimedia programming, multimedia scenario

• 제1저자 : 임재걸 • 교신저자 : 이강재
• 접수일 : 2007.5.18, 심사일 : 2007.5.19, 심사완료일 : 2007.6.25.
* 동국대학교(경주) 컴퓨터멀티미디어학과 교수, ** 수원과학대학 컴퓨터정보과 교수

Ⅰ. 서 론

소프트웨어 개발에서 요구 사항 분석의 오차는 전체 개발 단계별로 막대한 오류 수정의 비용을 초래한다[1]. 그래서 요구 사항의 무결성을 검증하는 연구가 무수히 수행되어 왔다. 예를 들어, Use Case 모델링 과정에서 생성되는 시나리오를 이용하여, UML 객체지향 분석모델의 일관성과 완전성을 검증하는 방법이 [1, 2]에 소개되었으며, 요구 사항 분석을 위한 데이터 획득과 시나리오 기반의 분석 방법과 목표기반의 분석 방법을 통합한 요구 사항 분석 방법이 [3, 4]에 제안되었고, ATM을 이용한 실시간 시스템의 명세, 분석 및 검증 방법이 [5]에 소개되었다. 소프트웨어 시스템의 검증 방법으로 페트리 넷(Petri Net)이 사용된 예도 많이 찾아 볼 수 있다. Use Case로 묘사된 시스템 요구 사항을 페트리 넷으로 나타내고, 시스템의 완전성과 무결성을 분석하는 방법이 [6]에 소개되었으며, Workflow 시스템 구현에 앞서 권한 위임의 안전성을 검증하기 위한 페트리 넷 방법이 [7]에 소개되었다.

본 논문은 멀티미디어 시나리오를 페트리 넷으로 표현하고[8], 이를 분석하여 시나리오의 무결성을 검증하며, 나아가서 이 페트리 넷을 해석함으로써 시나리오를 상영하는 멀티미디어 프로그램을 작성하는 방법을 소개한다. 멀티미디어 시스템의 페트리 넷 모형은 [9]에 최초로 소개되었는데, 여기에서는 멀티미디어 출력 데이터에 대한 동기화 전략이 집중적으로 다루어져 있다. 출력에 소요되는 시간은 환경에 따라 차이가 발생하게 되는데, 이러한 오차를 가미하여 모델링의 유연성을 향상시키는 연구가 [10, 11]에 소개되었으며, 모델링 대상을 멀티미디어에서 하이퍼미디어로 확장시킨 것이 [12]에 소개되었다. 이처럼 기존의 연구들은 모두 동기화 전략에 대한 모델링에 목적을 두고 있는데 반하여, 본 논문은 번역기에 의하여 해석되고 실제 재생될 수 있는 모델을 위한 페트리 넷을 제안한다.

제안된 페트리 넷을 멀티미디어 페트리 넷(Multimedia Petri Net, MPN)이라고 하며, MPN을 분석하여 시나리오에 나타날 수 있는 결함을 제거하는 방법을 제시한다. 또한, MPN을 해석하고 상영하는 시나리오 재생기의 구현을 소개한다. 소개한 재생기는 플레이 도중에 '정지', '빨리 가기', '되감기', '빨리 되감기' 등의 기능을 갖는다.

Ⅱ. 멀티미디어 페트리 넷

기존의 멀티미디어 시스템의 페트리 넷 모형은 동기화를 집중적으로 다룬 반면에[9-12], 본 논문은 멀티미디어 시나리오를 표현할 수 있는 페트리 넷을 제안하고, 제안된 페트리 넷으로 표현된 시나리오를 해석하고 상영할 수 있는 재생기를 구현한다. 따라서, 제안된 페트리 넷은 시나리오의 모든 구성 요소들을 포함하여야 한다. 즉, 출력되는 데이터 자체에 대한 모든 정보, 출력에 관련된 정보, 데이터간의 동기화 현상, 더 나아가서 비동기적인 흐름까지도 표현 가능해야 한다.

제안한 페트리 넷을 MPN(Multimedia Petri Net)이라고 하며, <표 1>은 MPN을 정의한 것이다.

표 1. MPN의 정의
Table 1. Definition of MPN

[정의] MPN=(P, MT, FMT, POS_P, POS_W, SIZE_W, PATH_P, T, POS_T, F_{syn}, E)
P={p_1, p_2, ..., p_m} //place의 집합으로 원으로 표현
MT={Start, End, Bitmap, Avi, Wave, Midi, Mpeg, τ}
//데이터 유형의 집합. τ는 지연 시간
FMT : P → MT
//place를 데이터 유형으로 매핑하는 함수
POS_P : P → X_Y, where X_Y={(X_POS, Y_POS) | X_POS∈integer, Y_POS∈integer}
//시나리오 편집기에서 place의 스크린 위치 지정 함수
POS_W : P → X_Y
//데이터 출력 창의 스크린 위치를 지정 함수
SIZE_W : P → X_Y
//데이터 출력 창의 가로와 세로 지정 함수
PATH_P: P → PATH
//데이터의 경로 및 이름 지정 함수
T={t_1, t_2, ..., t_n}
//transition의 집합으로 그래프에서는 막대로 표현
POS_T : T → X_Y
//시나리오 편집기에서 transition의 스크린 위치
//지정 함수.
Fsyn : T → P*
//t_i는 Fsyn(t_i)에 나열된 place들을 격발 조건에서
//무시, 이들을 Exclude 리스트라고 부름
E ⊆ {T × P} ∪ {P × T}
//간선의 집합으로, 토큰의 흐름을 나타냄

시나리오를 페트리 넷으로 표현할 때 데이터는 플레이스(place)에, 시나리오의 흐름은 트랜지션(transition)으로 표현하는 것이 자연스럽다. 플레이스의 집합을 P라고 하고 트랜지션의 집합을 T라고 한다. 따라서, 제안된 페트리 넷의 플레이스는 플레이스 식별자(Place_id), 플레이스가 나타내는 멀티미디어의 유형(MT: Start, End,

Bitmap, Avi, Wave, MiDi, MPeg, Time 등의 유형이 있음), 데이터가 출력될 윈도우의 위치(W_POS), 윈도우의 크기(RECT), 데이터를 찾기 위한 경로(PATH) 등의 정보를 갖고 있다. Multimedia type에서 Start, End, Time은 각각 시나리오의 시작, 종료, 휴지기간을 나타낸다. 한편, 트랜지션은 식별자(Transition_id), 동기화 전략(Synchronization strategy)에 대한 정보를 갖고 있다. E는 간선의 집합이다.

플레이스와 트랜지션의 합집합인 정점 V에서, (v_i, v_j)가 간선일 때 v_i를 v_j의 입력 정점이라 하고, v_j를 v_i의 출력 정점이라 하면, 어떤 트랜지션 t_i는 입력 정점에 연합된 데이터의 출력이 모두 끝날 때까지 출력 정점에 연합된 데이터의 출력을 지연시키고 있다가, 입력 정점의 데이터 출력이 끝났을 때 비로소 출력 정점에 연합된 데이터의 출력을 동시에 시작한다. 이 때, $F_{syn}(t_i)$의 원소인 입력 플레이스는 무시한다.

이와 같은 데이터 출력의 천이는 토큰의 움직임으로 나타내게 되는데, 어떤 플레이스에 연합된 데이터가 출력 중이면 그 플레이스에 토큰을 놓아 출력 중임을 나타낸다. 그러므로 시나리오의 흐름은 $FMT(p_i)$가 Start인 플레이스 p_i에 한 개의 토큰을 놓음으로써 시작하게 된다. 페트리 넷 분야에서는 트랜지션의 입력 플레이스에 있는 토큰들이 출력 장소로 옮겨가는 것을 격발(firing)이라고 한다. MPN에서는 플레이스에 토큰이 놓이면 해당 데이터의 출력을 시작하고, 트랜지션의 모든 입력 장소에 연합된 데이터의 출력이 종료하게 되면 트랜지션을 격발함으로써 입력 장소의 토큰을 제거하여 출력 장소에 옮겨 놓은 후, 새로운 데이터의 출력을 시작시킨다. 〈표 2〉는 출력의 천이를 제어하는 MPN의 트랜지션 격발 규칙을 보인 것이다.

표 2. MPN의 격발 규칙
Table 2. The Firing Rule of MPN

1) MPN은 초기에 $FMT(p_i)$가 Start인 place p_i에 한 개의 토큰을 갖는다.
2) place p_i에 토큰 tok_k가 놓이면 $POS_W(p_i)$에 $SIZE\_W(p_i)$ 크기의 윈도우를 띄우고, 여기에 $PATH_P(p_i)$에 지정된 데이터를 출력시킨다. 출력이 완료되면 p_i의 out going 간선 (p_i, t_j)를 enable시킨다. 이때 p_i가 $F_{syn}(t_j)$의 원소이면 p_i의 데이터를 출력시킴과 동시에 (p_i, t_j)를 즉시 enable시키고, tok_k에 (p_i, t_j)라는 레이블을 붙인다.
3) 모든 입력 간선이 enabled된 트랜지션 t_j는 즉시 격발한다. 격발은 t_j의 입력 장소 p_i의 토큰과 (p_i, t_j)라는 레이블을 갖는 모든 토큰을 제거하고, 출력 장소에 한 개씩의 토큰을 놓는다.
4) $FMT(p_i)$가 End인 place p_i에 토큰이 놓일 때 격발을 종료한다.

간단한 멀티미디어 시나리오로 화면 1에서는 밭갈이 도구 중의 쟁기를, 화면 2에서는 써레를 설명하고, 쟁기와 써레에 대한 설명과는 독립적으로 화면 1부터 2에 걸쳐 소를 몰고 밭갈이하는 동영상을 화면의 상반부에 계속 출력함으로써 밭갈이하는 모습을 전체적으로 보여주는 시나리오를 MPN으로 나타내면 〔그림 1〕과 같다.

〔그림 1〕의 원들에 대해서 좌에서 우로, 위에서 아래로 p_1, p_2, …, p_7이라 하고, 같은 방법으로 트랜지션들은 왼쪽으로부터 t_1, t_2, t_3라고 임의로 지정하자. 플레이스 p_1은 시나리오의 시작($FMT(p_1)$ = Start)을 나타내고 p_2는 비디오를, p_3와 p_4는 각각 쟁기 bmp와 wav를, p_5, p_6는 각각 써레 bmp와 wav를, 마지막으로 p_7은 시나리오의 종료($FMT(p_7)$ = End)를 나타낸다. 〔그림 1〕에는 p_2에만 데이터 유형 avi, 출력창의 위치 (100, 0), 출력창의 크기 (100, 100), 데이터 파일 이름 '밭갈이.avi'라고 표시되어 있지만, 실제 MPN에는 모든 플레이스에 이와 같은 정보가 표시되어 있어야 한다. 플레이스 p_2의 비디오는 p_3, …, p_6의 데이터와는 비동기적으로 출력되고 시나리오의 진행에 아무런 영향을 주지 않으므로 $F_{syn}(t_3) = p_2$이다. 〔그림 1〕에는 t_3 옆에 p_2가 있어 이를 나타낸다.

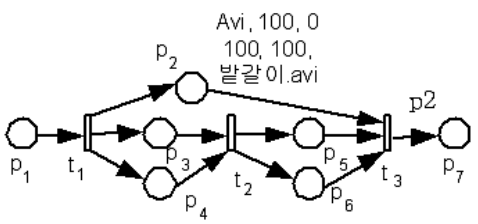

그림 1. 간단한 MPN 모형
Figure 1. A simple example MPN

여기서 트랜지션 t_1이 격발하면 p_2, p_3, p_4에 토큰이 놓여 비디오를 비롯한 쟁기 bmp와 wav 데이터가 동시에 출력하게 되고, t_3의 동기화 전략에 p_2가 있기 때문에 출력과 함께 간선 (p_2, t_3)가 enabled된다. 그리고, 쟁기 bmp와 wav에 대한 출력이 모두 종료하면 t_2가 격발하여 p_5와 p_6에 토큰이 놓이게 되고, 이에 따라 써레 bmp와 wav 데이터가 출력을 시작하게 되며, 이들의 출력이 종료하면 t_3가 격발하여 End인 p_7에 토큰이 놓여 종료하게 된다.

페트리 넷의 간선 정보는 Backward incidence matrix B와 Forward incidence matrix F에 저장하

며, B와 F는 n×m (n는 트랜지션의 수, m은 플레이스의 수) 행렬이다. 예를 들면, 〔그림 1〕에 대한 MPN의 incidence matrix는 〔표 3〕과 같다.

표 3. 〔그림 1〕의 B matrix와 F matrix
Table 3. B matrix and F matrix of Figure 1

t	p_1	p_2	p_3	p_4	p_5	p_6	p_7
t_1	1						
t_2			1	1			
t_3		1			1	1	

(a) B Matrix

t	p_1	p_2	p_3	p_4	p_5	p_6	p_7
t_1		1	1	1			
t_2					1	1	
t_3							1

(b) F Matrix

〔표 3〕에서 행렬 B의 일반항 i-th 행, j-th 열의 엔트리 B〔i〕〔j〕는 p_j에서 t_i로 나가는 간선의 수이고, 행렬 F의 일반항 F〔i〕〔j〕는 t_i에서 p_j로 들어오는 간선의 수이다.

III. MPN 분석 및 재생기 구현

1. MPN의 성질

시나리오의 시작 상태와 종료 상태는 각각 유일하므로, MPN은 다음과 같은 성질을 갖는다.

〈성질 1〉 어떤 MPN에는 FMT(p_i) = Start인 플레이스(Start place)와 FMT(p_j) = End(End place)인 플레이스가 꼭 하나씩 존재한다.

모든 MPN은 반드시 〈성질 1〉을 만족해야 한다. 〈성질 1〉을 만족하는지 검증하려면 모든 플레이스들의 FMT 값을 살펴보아야 한다. FMT 값을 'Start'와 비교하고 또 'End'와 비교하는데 걸리는 시간은 상수 시간이므로 〈성질 1〉을 체크하는데 걸리는 시간은 플레이스의 수 m에 비례한다.

MPN에서는 비동기적인 진행을 표시하기 위하여 트랜지션을 인수로 하는 F_{syn}이라는 함수를 사용한다. 이러한 플레이스들을 모두 제거한 MPN의 격발 습성과 원래 MPN의 격발 습성은 동일하다.

〈성질 2〉 주어진 MPN과 그 MPN의 '동기MPN'은 격발 습성이 동일하다.

멀티미디어 시나리오의 진행은 결정적이다. 따라서, 다음의 〈성질 3〉과 〈성질 4〉가 성립한다.

〈성질 3〉 End 플레이스를 제외한 '동기 MPN'의 모든 플레이스 p_i는 유일한 출력 트랜지션을 갖는다.

〈성질 3〉을 검증하는 방법 : B matrix의 정의에 의하여, B〔i〕〔j〕는 p_j에서 t_i로 나가는 간선의 수이다. 따라서, End 플레이스에 해당하는 열을 제외한, B Matrix의 모든 열에 1이 꼭 한 개씩 있어야 한다. 이 검증은 B matrix의 모든 항을 조사하여 1인지 비교해야 하므로 n×m 시간이 걸린다.

〈성질 4〉 Start 플레이스를 제외한 '동기 MPN'의 모든 플레이스 p_i는 유일한 입력 트랜지션을 가진다.

〈성질 4〉를 검증하는 방법 : F matrix의 정의에 의하여, F〔i〕〔j〕는 t_i에서 p_j로 들어오는 간선의 수이다. 따라서, Start 플레이스에 해당하는 열을 제외한, F Matrix의 모든 열에 1이 꼭 한 개씩 있어야 한다. 이 검증은 F matrix의 모든 항을 조사하여 1인지 비교해야 하므로 n×m 시간이 걸린다.

〈성질 3〉과 〈성질 4〉에 의하면 Start 플레이스와 End 플레이스를 제외한 모든 플레이스는 입력간선과 출력간선이 꼭 한 개씩인 것을 알 수 있다. '동기 MPN'에 여벌의 트랜지션 t_a를 가미하여 End 플레이스와 Start 플레이스를 〔그림 2〕와 같이 연결한 그래프를 '가미 동기 MPN'이라고 한다.

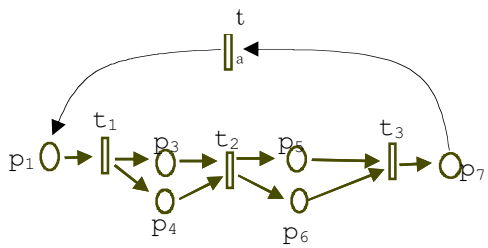

그림 2. t_a가 가미된 '가미 동기 MPN'
Figure 2. An 'augmented synchronous MPN'

〈성질 5〉 Start 플레이스에만 하나의 토큰이 놓인 상태에서 격발을 진행하면 모든 트랜지션을 꼭 한번 씩 격발한 다음 End 플레이스에만 꼭 하나의 토큰이 놓인 상태에 도달하게 된다.

〈성질 5〉를 검증하는 방법 : 각 트랜지션들의 격발 횟수를 나열한 벡터를 '격발 횟수 벡터'라 한다. 트랜지션 t_i가 격발하면 t_i의 입력 플레이스에서 토큰을 제거하고 출력 플레이스에 토큰을 더 놓음으로 마킹이 변화한다.

이때 제거하는 토큰의 수는 B matrix에, 더 놓는 토큰의 수는 F matrix에 수록되어 있음을 알 수 있다. 이들 두 matrix를 하나로 모아 놓은 것을 incidence matrix A라고 하며 A = F matrix - B matrix로 정의한다. 예를 들면, 〔그림 2〕에 대한 MPN의 A matrix는 〈표 4〉와 같다.

표 4. 〔그림 2〕에 대한 MPN의 A matrix
Table 4. The matrix A of (Figure 2)

t_i / p_i	p_1	p_3	p_4	p_5	p_6	p_7
t_1	-1	1	1			
t_2		-1	-1	1	1	
t_3				-1	-1	1
t_a	1					-1

여기서, A matrix의 각 행은 해당 트랜지션의 격발 결과 토큰이 어떻게 움직이지를 잘 나타낸다. 예를 들어, t_1행은 t_1의 격발 결과 p_1에서 토큰이 하나 없어지고, p_3와 p_4에 각각 하나씩의 토큰이 더하여지며, 나머지 플레이스에는 변화가 없음을 나타낸다. 따라서, 현재 마킹 M_i에서, '격발 횟수 벡터' F로 나타낼 수 있는 격발 과정으로 도착하는 마킹 M_j를 A matrix를 이용하여 다음과 같이 (식 1)로 표현할 수 있다.

$$Mj = Mi + AT \times F \quad \cdots\cdots (1)$$

〈성질 5〉에서 모든 트랜지션을 꼭 한 번씩 격발한다는 말에 트랜지션 t_a는 물론 포함되지 않는다. 따라서, '가미 동기 MPN'에서 t_a의 순서를 제일 마지막으로 하면 모든 트랜지션을 꼭 한번 격발하는 '격발 횟수 벡터'는 (1, …, 1, 0)이 된다. 또한 Start 플레이스에만 토큰이 있는 마킹은 (1, 0, 0, …, 0)이고, End 플레이스에만 토큰이 있는 마킹은 (0, 0, …, 0, 1)이다. 따라서, (식 1)에 의하여 MPN은 다음의 (식 2)를 반드시 만족한다. 즉, 주어진 MPN의 '가미 동기 MPN'의 A matrix가 (식 2)를 만족하지 않으면 MPN이 표현하는 시나리오에 오류가 있다고 결론을 내릴 수 있다.

$$\begin{pmatrix}0\\ \cdot\\ \cdot\\ \cdot\\ 0\\ 1\end{pmatrix} = \begin{pmatrix}1\\ 0\\ \cdot\\ \cdot\\ \cdot\\ 0\end{pmatrix} + A \begin{pmatrix}1\\ \cdot\\ \cdot\\ \cdot\\ 1\\ 0\end{pmatrix} \quad \cdots\cdots (2)$$

예를 들어, 다음에 보는 바와 같이 〔그림 2〕는 (식 2)를 만족한다. (식 2)를 만족하는 것은 〈성질 5〉의 필요조건이다.

$$\begin{pmatrix}0\\0\\0\\0\\0\\1\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\\0\\0\end{pmatrix} + \begin{bmatrix}-1 & 0 & 0 & 1\\ 1 & -1 & 0 & 0\\ 1 & -1 & 0 & 0\\ 0 & 1 & -1 & 0\\ 0 & 1 & -1 & 0\\ 0 & 0 & 1 & -1\end{bmatrix} \begin{pmatrix}1\\1\\1\\0\end{pmatrix}$$

〔정리1〕 (식 2)는 〈성질 5〉의 필요조건이다.

MPN은 격발 횟수 벡터 (1, …, 1, 0)가 나타내는 각 트랜지션의 격발 횟수만큼 각 트랜지션을 실제로 격발하는 것을 보장해야 한다. 현재 마킹에서 도달 가능한 어떤 마킹에서라도 트랜지션 t_i를 격발할 수 있는 격발과정이 존재하면 트랜지션 t_i는 '살아 있다'라고 한다. MG(Marked Graph)는 각 플레이스의 입력 간선과 출력 간선이 모두 유일하므로 어떤 순회(circuit)에 놓인 토큰의 수는 불변이다. 그래서, 어떤 순회에 초기 마킹에 의하여 놓인 토큰의 수가 0이면 그 순회상의 트랜지션은 결코 격발할 수가 없게 된다. 역으로, 어떤 트랜지션이 격발 불가능하다면 토큰이 없는 플레이스들을 역추적하여 토큰이 없는 순회를 만들 수 있다. 따라서, MG의 모든 트랜지션이 '살아있을' 필요충분 조건은 초기 마킹이 모든 순회에 하나 이상의 토큰을 놓는 것이다.

〔정리 2〕 〈성질 5〉를 만족할 필요충분 조건은 '가미 동기 MPN'의 모든 순회가 Start 플레이스와 End 플레이스를 포함하고 모든 트랜지션이 어떤 순회에 반드시 포함되는 것이다.

〈증명〉 (→) '〈성질 5〉를 만족하면 '가미 동기 MPN'의 모든 순회가 Start 플레이스와 End 플레이스를 포함한다'를 모순법으로 증명하자. 즉, "〈성질 5〉를 만족하고 Start 플레이스나 End 플레이스를 포함하지 않는 순회가 있다고 가정하자." 이 순회를 't_1, p_1, t_2, p_2, …, '라 하자. 그러면, 이 순회에는 t_a가 포함되지 않으며, '가미 동기 MPN'은 MG이기 때문에 이 순회를 구성하는 트랜지션 중에 출력 간선의 수가 2 이상인 t_i와 입력 간선의 수가 2 이상인 트랜지션 t_j가 존재하고, 초기마킹에서 이 순회에 놓인 토큰의 수는 0 개다. 즉, t_j는 격발이 불가능한 트랜지션이다. 그런데, 〈성질 5〉는 모든 트랜지션을 한 번씩 격발한다고 하였으므로 모순이다.

(←) '가미 동기 MPN'의 모든 순회가 Start 플레이스와 End 플레이스를 포함하면, 〈성질 5〉 격발 회수 벡터

(1, 1, …, 1, 0)이 존재함을 증명하자.' 모든 순회가 Start 플레이스와 End 플레이스를 포함하면, Start 플레이스에서 출발하여 End 플레이스에 도착하는 경로 상에 사이클이 존재하지 않는다. 그러면, 모든 트랜지션, t_i에 대하여 Start 플레이스로부터의 유일한 거리 d_i를 구할 수 있다. 이때, 거리가 같은 트랜지션들은 다수 존재할 수 있다. 거리가 1인 트랜지션들의 집합을 L_1, 거리가 2인 트랜지션들의 집합을 L_2, …, 라고 하자. L_i에 속한 트랜지션들의 입력 플레이스들은 j 〈 i인 어떤 j에 대한 L_j에 속한 어떤 트랜지션의 출력 플레이스이다. 따라서, 'L_i의 트랜지션은 L_j의 트랜지션이 격발되기 전에 격발할 수 없다.' --- 〔보조정리 1〕

MG의 '살아있을' 필요충분 조건에 의하여 모든 트랜지션이 어떤 순회에 속하고 모든 순회가 Start 플레이스와 End 플레이스를 포함하는 '가미동기 MPN'은 초기마킹 (1, 0, …, 0)에서 살아있다. --- 〔보조정리 2〕

보조정리 1과 2에 의하여 L_i의 트랜지션이 격발되기 전에 L_j의 트랜지션들이 모두 몇 번인가는 격발한다는 것과 그 회수는 반드시 한 번이라는 것을 알 수 있다.

2. 시나리오 재생 알고리즘

제안한 멀티미디어 재생 프로그램은 MPN을 해석하여, MPN의 격발 규칙에 따른 시나리오 재생 알고리즘 〈표 5〉에 따라 시나리오를 재생한다.

표 5. 멀티미디어 시나리오 흐름 제어 알고리즘
Table 5. Algorithm to control the flow of a Multimedia Scenario

```
// B는 B Matrix, F는 F Matrix를 의미한다.
Control_flow(i)
begin
  B[j][i] = 1인 j에 대하여 // j는 유일함
  B[j][i] := 0  //pi의 출력이 종료되었음
                if B의 j_th row에 1 entry가 없으면
                                  //격발 j
                                  then
    unlock B;
    For every pk such that F[j][k] = 1
      //모든 출력 장소에 대하여
      if FMT(pk) = END
        //End place라면 종료
      then exit
      else
        BeginThread
        //출력장소 하나에 하나씩 Exclude list에
        //속한 place는 동기화 조건에서 제외함
        pk ∈ Fsync(th)에 대하여 B[h][k] := 0;
        Display data associated with pk
          //출력이 완료된 후 ...
          if B is unlocked
          then lock(B) and Control_flow(k);
 else  //1 entry, 즉 출력이 종료되지 않은 place ...
    unlock B;
    return;  //thread 종료
end.
```

알고리즘은 FMT(p_i)가 Start인 p_i를 찾아 Control_flow(i)로 호출한다. 이러한 호출은 p_i의 데이터를 출력하게 되고, 출력이 완료되면 B[j][i]를 0으로 바꾸어 출력 완료를 표시한다. 단, 최초에는 Start 플레이스를 인수로 하기 때문에 아무런 출력이 없이 그대로 B[j][i]를 0으로 변환한다.

알고리즘에서 5행은 트랜지션 t_j가 enable되었는지 검사하는 명령이다. Enable되었으면 F 행렬의 j행에서 항목이 1에 해당하는 플레이스들의 데이터들을 동시에 출력시킨다. 7행은 B를 다른 프로세스가 사용할 수 있도록 허용하는 명령인데, 이 명령이 필요한 이유는 17행에서 쓰레드마다 B 행렬을 lock하기 때문이다. 이는 4행과 5행을 배타적으로 수행하기 위한 장치다. 8행부터는 F[j][k] = 1인 모든 p_k에 대하여 멀티쓰레드를 생성하고 각 쓰레드에서 p_k가 나타내는 데이터를 출력하는 작업을 수행한다. 그리고, p_k의 유형이 'end'이면 10행에서 재생을 종료한다. 13행에서는 p_k가 비동기적인 플레이스면 B[h][k]를 0으로 바꾸어 줌으로써 t_h의 격발 조건에서 p_k를 제외시킬 수 있도록 하여 준다. p_k의 해당 데이터의 출력이 종료하면 17행에서 Control_flow를 재귀적으로 호출한다.

Ⅳ. 멀티미디어 재생기의 구현

멀티미디어 시나리오 흐름 제어 알고리즘을 Pentium V, Windows XP 시스템에서 Microsoft Visual C++, MFC 개발 환경에서 구현하고, 멀티미디어 시나리오 재생 프로그램을 작성하여 보았다. 작성한 멀티미디어 프로그램은 CDMAEngine, Run(), CTPManager, CIMatrix, CWord2DArray, CDIB, CMpeg, CAVI, CWave 등으로 구성된다.

다음은 〔그림 3〕을 실험 데이터로 하여 멀티미디어 재생 프로그램을 수행시켜 보았다. 멀티미디어 시나리오의 예로서 화면에 '중력'이라는 제목이 그려진 bmp 파일과 스카이다이버 그림인 bmp 파일을 〔그림 4〕처럼 출력하고 있는 동안, 중력을 설명하는 wav 파일을 출력하고 〔그림 5〕처럼 무중력 동영상을 출력하는 시나리오를 MPN으로 나타내면 〔그림 3〕과 같다. 〔그림 3〕의 원들에 대해서 좌에서 우로, 위에서 아래로 p_0, p_2, …, p_{11}이라 하고, 같은 방법으로 트랜지션들은 왼쪽으로부터

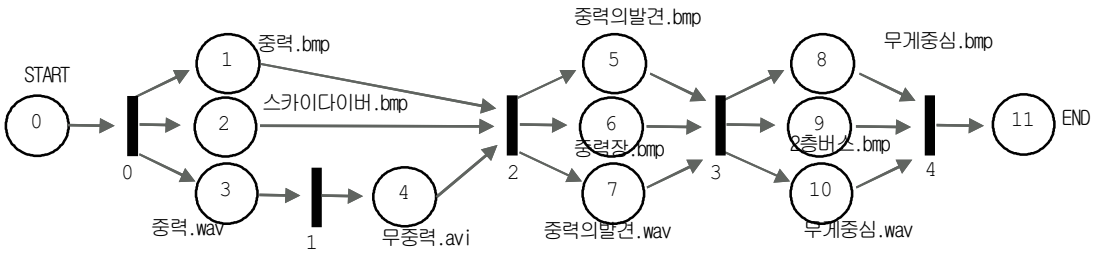

그림 3. '중력'을 설명하는 시나리오의 MPN 모델
Figure 3. MPN model for a multimedia scenario Explaining 'Gravity'

t_0, ..., t_4라고 하자. 그림에는 생략되어 있지만 각 플레이스에는 출력창의 위치와 크기 등도 포함되어 있다.

시나리오가 처음 시작되면 〔그림 4〕처럼 0번의 시작 플레이스가 실행이 되고, 그 다음 1번, 2번, 3번의 순서로 쓰레드가 만들어져 각각이 병렬로 수행된다. 그리고, 3번의 wave 파일이 끝나면 〔그림 5〕처럼 4번 avi파일이 수행되게 된다.

또한, 구현한 멀티미디어 시나리오 재생기의 편리성을 위해 일반 CD 플레이어에서 제공하는 재생, 빠른 재생, 정지, 되감기, 빠른 되감기 등 다섯 가지 재생 방법을 부가하는 방안을 정의하고 구현하여 보았다. 기본 흐름인 '재생(Play)'을 기준으로 '역방향 재생(Backward Play)'은 '재생'과 흐름의 방향만 반대이고 나머지는 같은 것으로 정의하였다.

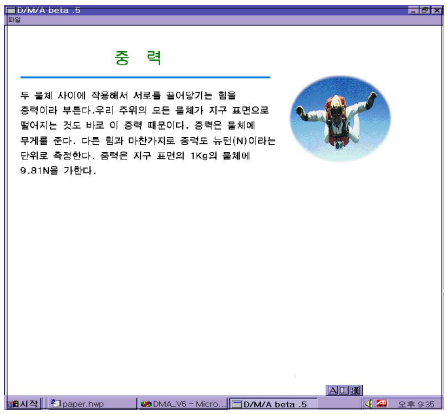

그림 4. 시작 후 1, 2, 3번 플레이스가 실행되고 있는 화면
Figure 4. Place 1, 2, 3 are running

그림 5. 플레이스(avi) 4번 실행 화면
Figure 5. Place 4(avi) is running

'정지(Stop)'는 트랜지션 상에서만 이루어지는 것으로 정의하였다. 즉, 수행 중인 플레이스를 강제로 종료시키고 정지할 수는 없다는 말이다. 이것은 이미지 플레이스의 경우 시간성을 가지고 있지 않고, 소리나 동영상의 경우는 시간성을 가지므로 수행 완료를 플레이스(미디어) 단위로 통일하여 흐름을 제어하기 위해서이다. '빠른 재생(Fast Play)'이나 '빠른 역방향 재생(Fast Backward Play)'은 시간성을 가지는 플레이스들을 실행하지 않고 건너뛰어 흐름을 빠르게 하는 것으로 정의하였다.

이렇게 정의된 5가지의 재생 방법을 나타내기 위해, 재생 방향을 나타내는 플래그인 bForward와 빠른 수행 여부를 나타내는 플래그인 bFast, 그리고 정지 상태를 나타내는 플래그인 bStop 등 3개의 플래그를 둔다. 이들 플래그로 5가지 재생 모드를 나타내면 〈표 6〉과 같다.

표 6. 재생 모드 플래그
Table 6. Flags for Playing Modes

플래그 \ 재생모드	◀◀	◀	■	▶	▶▶
bForward	F	F	F or T	T	T
bFast	T	F	F	F	T
bStop	F	F	T	F	F

이러한 제어 방법을 실현하기 위하여 현재 진행 중인 시나리오의 위치를 격발 행렬에 기록할 필요가 있다. 이를 위하여 entry 3을 추가한다. 예를 들면, 어떤 플레이스 p_i에 해당하는 데이터 출력이 종료하고 그 플레이스의 출력 트랜지션이 t_j일 때, 격발행렬 IMatrix[j, i]에 3을 넣는다.

쓰레드 함수 Run()은 트랜지션이 활성화되고 해당 플레이스를 활성화시키는 과정에서 플래그들을 체크하도록 변형한다. 예를 들면, 플래그 bStop이 true로 세트되어 있으면, 현재 활성화된 트랜지션 ID를 스택에 저장하고, 쓰레드를 종료한다. 플레이스 실행 시 미디어 별로 해당 함수를 플레이스 ID로 호출해 주는 함수도 bFast 플래그에 따라 시간성 플레이스는 건너뛰도록 수정된다. 예를 들면 bFast플래그가 true로 세팅되어 있는 경우는 wave나 avi, mpeg 플레이스는 실행하지 않는다.

아래 [그림 6]과 같은 재생기 흐름 제어 툴바에서 사용자가 원하는 재생 모드를 선택하면, <표 6>에서 명시한 바와 같이 플래그들이 세팅된다. 세팅된 플래그에 따라 격발 행렬을 처리하는 함수는 각기 다른 동작을 만들어 낸다.

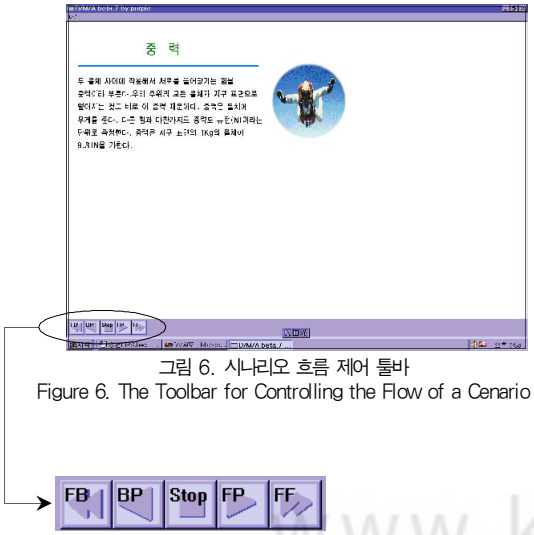

그림 6. 시나리오 흐름 제어 툴바
Figure 6. The Toolbar for Controlling the Flow of a Cenario

V. 결 론

본 논문은 페트리 넷을 이용한 시나리오 재생기 구현 방법을 제안하였으며, 이를 위하여 시나리오를 모델링할 수 있는 MPN을 제시하였다. 페트리 넷을 이용하는 멀티미디어 시나리오 모델링에 관한 기존의 연구는 동기화 현상을 분석하는 것을 목적으로 하여, 동기화와 관련된 정보만을 다루었는데 반하여, MPN의 특징은 재생기에 의하여 해석됨에 따라 멀티미디어 시나리오가 상영될 수 있도록, 시나리오의 흐름뿐만 아니라 시나리오를 구성하는 멀티미디어 데이터에 관한 모든 정보를 포함하는 것이다. 나아가서 데이터의 동기적인 진행뿐만 아니라 비동기적 진행까지도 나타낼 수 있다는 점이 기존의 페트리 넷과 다른 점이다.

MPN을 분석하여 시나리오의 무결성을 검증하는 방법도 역시 제시하였다. 소프트웨어 시스템 개발에서 개발 초기에 결함을 제거하지 못하면, 이를 수정하기 위하여 나중에 엄청난 비용을 치르게 된다는 것은 잘 알려진 사실이다. 멀티미디어 프로그래밍에서도 역시 시나리오 자체에 결함이 있는 시스템을 개발하면 나중에 이를 수정하기 위하여 엄청난 비용을 치르게 된다. 제시한 무결성 검증 방법은 멀티미디어 프로그래밍을 시작하기 전에 시나리오 자체를 분석한다는 점에서 그 가치가 크다고 하겠다.

제안한 MPN의 장점을 기존의 그래픽 멀티미디어 시나리오 모델링 도구와 비교한다면, 기존의 재생기는 여러 화면에 걸쳐 출력이 계속되는 데이터를 표현할 수 없는데 반하여 MPN은 이를 자연스럽게 표현할 수 있다는 점이다. 예를 들어, 어떤 비디오가 화면 1에서 출력을 시작하여 화면 1을 구성하는 다른 데이터의 진행과는 관계없이 화면 2에서도 재생을 계속한다면, 기존의 화면 단위 시나리오 작성기로는 이를 표현할 수가 없었다. 왜냐하면 화면2에서는 모든 구성 데이터의 출력을 처음부터 다시 시작하기 때문에 비디오 데이터까지도 처음부터 다시 시작하기 때문이었다. 이에 반하여, 본 MPN에서는 시나리오 구성 요소가 데이터 단위이기 때문에 이러한 현상을 자연스럽게 표현할 수 있었다.

이처럼 본 논문에서는 MPN 모델을 채택한 멀티미디어 시나리오 재생 알고리즘을 새롭게 제시하였으며, 구현한 멀티미디어 재생기를 통해 가상의 시나리오를 재생하는 예를 보였다. 또한, 정상적인 재생 외에도 일반 CD

플레이어에서 볼 수 있는 정지, 빠른 재생, 되감기, 빠른 되감기 등의 기능을 부가한 결과도 소개하였다.

끝으로, 구현한 멀티미디어 시나리오 재생기의 성능 분석과 보다 향상된 고성능 MPN 모델에 대한 연구가 더 필요하다고 본다.

참고문헌

〔1〕 조진형, 배두환, "UML 객체지향 분석 모델의 완전성 및 일관성 진단을 위한 시나리오기반 검증기법", 정보과학회논문지: 소프트웨어 및 응용, 제 28권 제 3호, 2001년 3월, pp. 211-223.

〔2〕 이성대, 박휴찬, "요구분석을 위한 UML 다이어그램 저장관리 시스템", 정보과학회논문지: 컴퓨팅의 실제 제8권 제6호, 2002년 12월, pp. 657-668.

〔3〕 김진태, 김동선, 박수용, "목표와 시나리오 기반의 통합적 요구 사항 분석 방안", 정보과학회논문지: 소프트웨어 및 응용 제31권 제5호, 2004년 5월 pp. 543-554

〔4〕 최순황, 김진태, 박수용, 한지영, "목표 및 시나리오 기반 요구사항을 이용한 기능점수 분석", 정보과학회논문지: 소프트웨어 및 응용 제33권 제8호, 2006년 8월 pp. 655-667

〔5〕 박지연, 이문근, "추상 시간 기계를 이용한 실시간 시스템의 도달성에 대한 검증 방법", 정보과학회논문지: 소프트웨어 및 응용, 제 28권 제 3호, 2001년 3월, pp. 224-238.

〔6〕 Woo Jin Lee, Sung Deok Cha, and Yong Rae Kwon, "Integration and Analysis of Use Cases Using Modular Petri Nets in Requirements Engineering", IEEE Transactions on Software Engineering, Vol. 24, No. 12, Dec. 1998 ,pp. 1115-1130.

〔7〕 V. Atluri and W. Huang, "A Petri net based safety analysis of work flow authorization models", Journal of Computer Security 8 (2000) pp. 209-240.

〔8〕 James L. Peterson, Petri Net Theory and the Modeling of Systems, Prentice-Hall, 2000.

〔9〕 T. Little, A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects", IEEE Journal on Selected Areas in Communications, 8, (3), April, 1990.

〔10〕 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.

〔11〕 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.

〔12〕 P. Senac et. al, "Hierarchical Time Stream Petri Net: a Model for Hypermedia Systems", Application and Theory of Petri Nets 1995, Springer-Verlag, 1995, pp. 451-470.

저 자 소 개

임재걸

1981년 동국대학교 전자계산학과 학사

1987년 University of Illinois 석사

1990년 University of Illinois 박사

1992년 - 현재 동국대학교(경주) 컴퓨터멀티미디어학과 교수

〈관심분야〉 멀티미디어시스템, 컴퓨터시스템 분석, 위치기반 서비스

이강재

1981년 동국대학교 전자계산학과 학사

1983년 동국대학교 대학원 전자계산학과 석사

1997년 동국대학교 대학원 컴퓨터공학과 박사

1986년 - 현재 수원과학대학 컴퓨터정보과 교수

〈관심분야〉 데이터베이스 응용, 데이터마이닝, 멀티미디어시스템