

# 인터넷 멀티미디어 시나리오를 모델하기 위한 페트리 넷 소개

임재걸  
동국대학교 컴퓨터학과

## A Petri Net for Modeling Internet Multimedia Scenario

Jaegool Yim  
Dept. of Computer, Dongguk Univ.  
yim@wonhyo.dongguk.ac.kr

### 요 약

멀티미디어 동기화를 명시하기 위한 수단으로 개발된 페트리 넷의 예로 [1]에 제안된 Object Composition Petri Net (OCPN), [2]에 소개된 Time Stream Petri Net (TSPN), [3]에 소개된 변형된 Petri Net, 등을 들 수 있다. 이들은 모두 멀티미디어 시스템 동기화를 모델링하기 위하여 제안된 Petri net들이다. 기존의 Petri net을 바탕으로 본 논문은 WWW에서 상영될 멀티미디어 시나리오를 표현하기 위하여 사용될 수 있는 Petri net을 제안한다.

### 1. 서론

멀티미디어 저작 도구를 사용하여 시나리오를 작성하면 손쉽게 다양한 멀티미디어 타이틀을 제작할 수 있다. 이와 비슷한 방법으로 멀티미디어 시나리오를 WWW 상에서 작성하고, 이를 웹브라우저를 통하여 보여 줄 수 있다면 인터넷 사용자에게 더욱 생동감 있게 정보를 전달할 수 있다. WWW 상에서 멀티미디어 시나리오를 작성하여 보여 주는 기능을 제공하려는 시도가 이미 이루어지고 있지만 지금까지의 시도는 타임라인에 출력될 멀티미디어 데이터를 순서대로 나열하는 수준이다.

이와 같은 방법에서는 인터넷 통신량의 변화에 따른 전송 시간의 변화나, 사용자의 위치에 따른 데이터 전송에 필요한 시간의 차이를 반영시킬 수가 없다. 본 논문은 우선 기존의 멀티미디어 관련 Petri net을 소개하고, 멀티미디어 시나리오를 표현할 수 있으며, 출력에 필요한 시간을 동적으로 변화시킬 수 있는 Petri net을 제안한다.

### 2. 기존의 연구

멀티미디어 프레젠테이션은 여러 미디어가 동시에 병렬 출력하는 것, 각 미디어가 일련의 출력물을 연속 출력하는 것, 그리고 미디어간의 연속 출력 등을 포함한다. 이러한 현상을 잘 나타낼 수 있는 도구로 페트리 넷을 들 수 있으며, 페트리 넷은 이분-유향 그래프로 다음과 같이 정의된다.

페트리 넷은 이분-유향 그래프로 다음과 같이 정의된다.

$N = (T, P, A)$ , where  $T = \{t_1, t_2, \dots, t_n\}$ ,  
 $P = \{p_1, p_2, \dots, p_m\}$ , and  $A: \{T \times P\} \cup \{P \times T\} \rightarrow I$ ,  
 $I = \{1, 2, \dots\}$ .

$T$ 는 트랜지션(막대)의 집합,  $P$ 는 places(원)의 집합,  $A$ 는 유향간선의 집합이다. Marked Petri net,  $N_m$ 은 다음과 같이 정의된다.  $N_m = \{T, P, A, M\}$ , where  $(T, P, A)$ 는 페트리넷이고  $M$ 는 token (점)을 place에 해당하는 다음과 같은 함수이다.

$M: P \rightarrow I, I = \{0, 1, 2, \dots\}$

어떤 transition의 모든 입력 place에 token이 놓이면 그 transition은 fire할 수 있으며, Fire에 필요한 경과 시간은 0이다. 이러한 페트리 넷은 시간을 표현하는 기능을 갖고 있지 않다.

출력에 사용되는 미디어 유형과 시간 관계를 나타내기 위하여 페트리 넷의 Place에 시간과 미디어 자원을 부과한 변형된 페트리 넷이 Object Composition Petri Net(OCPN) 이며, 다음과 같이 정의된다.

$C_{OCPN} = \{T, P, A, D, R, M\}$  where,  $D: P \rightarrow R$  and  
 $R: P \rightarrow \{r_1, r_2, \dots, r_k\}$ .

$D$ 는 place를 Real numbers(경과 시간)에 매핑하는 함수이고,  $R$ 는 place를 자원(resources)으로 매핑하는

함수이다.

OCPN의 격발 (fire) 규칙은 다음과 같다.

- 1) Transition  $t_i$ 는 모든 입력 place들이 unlocked token을 갖고 있으면 즉각 fire한다.
- 2) 격발의 결과  $t_i$ 의 입력 place로부터 토큰 한 개를 제거하고, 출력 place에 토큰 한 개를 더한다.
- 3) 토큰이 place  $p_i$ 에 놓이면  $p_i$ 는  $\tau_i$  시간 동안 활동 상태로 있게 된다. 이 기간 동안 그 토큰은 locked된다.  $\tau_i$  시간이 경과하여, 그 place가 비활동 상태가 되면 토큰은 unlocked 된다.

분산 비동기 시스템에서는 통신, 처리, 출력에 따른 시간적 비결정성이 필연적이다. 그럼에도 불구하고 OCPN은 nominal 경과 시간만을 표기함으로 허용 가능한 시간적 jitter의 표현이 불가능하다. 시간을 transition에 사상하여 놓는 페트리넷을 Time Petri net이라고 하며, 이때 nominal 시간 대신 시간 interval을 사용하면 jitter를 표현하는 것이 가능하다. 그런데 Time Petri net은 fire가 늦게 도착하는 토큰에 의하여 주도됨으로 멀티미디어 시나리오를 모델링할 때, Time Petri net으로는 표현이 불가능한 경우가 존재한다

### 3. Internet Multimedia Petri Net (IMPN)

본 논문이 제안하는 Internet Multimedia Petri net (IMPN)의 목적은 인터넷에서 상영될 수 있는 멀티미디어 시나리오를 표현하는 것이다. 따라서, 기존의 Petri net에 상영될 멀티미디어 데이터를 표현하는 기능을 추가시켜야 하고, 시간의 경과에 따른 시나리오 진행을 표현할 수 있어야 한다. TSPN의 경우와 마찬가지로 출력시간은 간선에 사상하고, 동기화 정책을 transition에 표현한다. 단 TSPN은 jitter를 고려한 동기화 정책을 나타내는 것이 목적인데 반하여 본 논문의 목적은 시나리오를 표현하는 것이 목적이므로 동기화 정책을 달리해야 한다. 또한, 출력될 멀티미디어 데이터를 IMPN의 place에 사상하여 나타내고, 통신 속도를 고려하여 간선에 사상된 활성화 기간을 동적으로 변화시킨다. 이러한 기능이 가미된 IMPN의 정의는 다음과 같다.

IMPN = (P, T, B, F, M, SYN, SYN<sub>p</sub>, MT, IM, S\_POS, W\_POS, RECT, URL, F\_MT, F\_S\_POS, F\_W\_POS, F\_RECT, F\_URL)

P = {p<sub>1</sub>, p<sub>2</sub>, ..., p<sub>n</sub>} place의 집합. 그래프에서 원으로 표현됨.

T = {t<sub>1</sub>, t<sub>2</sub>, ..., t<sub>n</sub>} transition의 집합. 그래프에서 사각형으로 표현됨.

B ⊆ {P × T} transition에 대한 입력간선의 집합.

F ⊆ {T × P} transition에 대한 출력간선의 집합.

SYN = T → {and, or, weak-and, strong-or, master, or-master, and-master, strong-master, weak-master},

SYN(t<sub>j</sub>)는 t<sub>j</sub>의 진행 정책(디폴트는 AND).

SYN<sub>p</sub>: T → POW(P), POW(P)는 P의 멱집합, SYN<sub>p</sub>

는 주어진 transition의 진행 정책에서 참고하는 place들의 집합.

MT = {MT\_start, MT\_end, MT\_au, MT\_txt, MT\_mpg, MT\_jpg}. 멀티미디어 데이터 유형의 집합.

IM: A → Q' × Q' × (Q' ∪ ∞), a<sub>i</sub> → (α, n, β), α = 최소, n = nominal β = 최대 출력시간이다.

S\_POS = {(X\_pos, Y\_pos)} X\_pos와 Y\_pos는 음이 아닌 정수로 시나리오 작성시 스크린 상의 위치.

W\_POS = {(X\_pos, Y\_pos)}. 윈도우의 위치.

RECT = {(Width, Height)} Width와 Height는 음이 아닌 정수로 윈도우의 크기.

URL 인터넷 주소와 파일 이름.

F\_MT, F\_S\_POS, F\_W\_POS, F\_RECT, F\_URL은 각각 place에 연관된 MT, S\_POS, W\_POS, RECT, URL를 찾아 주는 함수임.

SYN은 트랜지션을 진행정책과 사상하는 함수이며, 각 진행정책의 의미는 다음과 같다:

- 1) AND : 입력 간선에 사상된 모든 IM이 중복되는 시간에 격발 가능하다.
- 2) OR : IM의 어느 시간에도 격발가능하다.
- 3) weak-and: IM의 α 중 max와 β 중 max 사이에 격발 가능하다.
- 4) strong-or: IM의 α 중 min과 β 중 min 사이에 격발 가능하다.
- 5) master: 지정된 간선의 α와 β 사이에 격발 가능하다.
- 6) or-master: IM의 α 중 min과 지정된 간선의 β 사이에 격발 가능하다.
- 7) and-master: IM의 α 중 max와 지정된 간선의 β 사이에 격발 가능하다.
- 8) strong-master: 지정된 간선의 β와 IM의 β 중 min 사이에 격발 가능하다.
- 9) weak-master: 지정된 간선의 α와 IM의 β 중 max 사이에 격발 가능.

단, SYN<sub>p</sub>에 나열된 place는 고려 대상에서 제외한다.

IMPN의 상태는 마킹 M과 간선들의 IM값에 의하여 결정된다. 시나리오의 진행은 IMPN의 transition 격발로 시뮬레이션된다. IMPN의 격발은 마킹뿐만 아니라 간선의 IM 값도 동적으로 변화시킨다. IMPN의 격발 규칙은 다음과 같다.

M은 마킹, I는 enable된 간선의 활성화 시간의 list로 정의된다. 이 list의 entry 수는 마킹 M에서 enable된 간선의 수와 같다. 현재 상태를 S라 하고, transition t가 현재 상태에서 θ 시간이 흐른 다음 격발 가능하다고 가정하자. 그러면 θ에서 t를 격발하여 얻는 다음 상태 S'=(M', I')은 다음과 같이 계산하여 얻는다.

1. IMPN은 초기에 F\_MT(p<sub>i</sub>)가 MT\_start인 place p<sub>i</sub>에만 한 개의 토큰을 갖는다.
2. F\_MT(p<sub>i</sub>)가 MT\_end인 place p<sub>i</sub>에 토큰이 놓

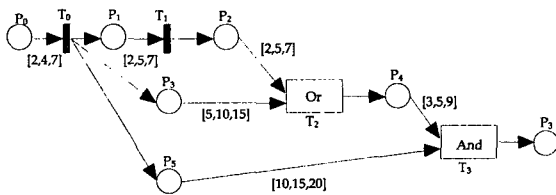
이런 처리 과정을 종료한다.

3. place  $p_i$ 에 토큰이 놓이면  $F\_W\_POS(p_i)$ ,  $F\_RECT(p_i)$  크기의 윈도우를 만들고, 여기에  $F\_URL(p_i)$ 에 지정된 데이터를 출력한다. 동시에  $p_i$ 의 out going 간선 ( $p_i, t_k$ )를 enable시킨다.
4. 모든 입력 간선이 enable된 변천  $t_k$ 는 입력 간선들의 활성화 시간과 transition의 진행정책을 이용하여 격발 가능 시간을 구한다.
5.  $t_k$ 의 모든 입력 장소에 사상된 데이터의 출력이  $t_k$ 가 enable된지  $\theta$ 시간만에 모두 종료하고,  $\theta$ 가  $SYN(t_k)$ 의 중간 값보다 작으면  $t_k$ 는 다음과 같이 격발한다. 단  $SYN(t_k)$ 의 중간 값을  $Mid$ 라 하고,  $R = \theta / Mid$ 라 한다.
- 5.1 현재 마킹  $M$ 으로부터 새로운 마킹  $M'$ 을 일반적인 Petri net의 경우처럼 다음과 같이 얻는다.  

$$M'(p) = M(p) - B(t,p) + F(t,p).$$
- 5.2  $t_k$ 의 격발에 의하여 disable되는 모든 간선  $a_i$ 에 대하여  $IM(a_i)$ 를 활성화 시간 리스트에서 삭제한다.
- 5.3 모든 활성화 시간 ( $\alpha, n, \beta$ )에 대하여, 새로운 ( $\alpha', n', \beta'$ )을 다음과 같이 구한다:  

$$n' = n * R, \alpha' = n' * \alpha / n, \beta' = n' * \beta / n.$$
- 5.4 enable 상태를 유지하는 모든 간선  $a_k$ 의 IM 값을  $\theta$ 로 다음과 같이 갱신한다.  $IM(a_k) = (x_k, n_k, y_k)$  라면 이것을  $(\max(0, x_k - \theta), n_k - \theta, \max(0, y_k - \theta))$ 로 대체한다.
- 5.5 격발 결과 enabled 된 간선의 IM 값으로 활성화 시간 리스트를 만든다.
- 5.6 변천  $t_k$ 의 격발을 종료하고 단계2로 간다.
6. 데이터 출력이 종료하지 않아도 단계 3에서 구한 격발 가능 시간이 넘으면 다음과 같이 격발한다.
- 6.1  $t_k$ 의 모든 입력 place  $p_i$ 에 대한 미디어 전송률의 평균  $R$ 을 구한다.
- 6.2 모든 활성화 시간 ( $\alpha, n, \beta$ )에 대하여, 새로운 ( $\alpha', n', \beta'$ )을 다음과 같이 구한다:  

$$n' = n * 1/R, \alpha' = n' * \alpha / n, \beta' = n' * \beta / n.$$
- 6.3 5.1부터 5.5를 수행한다.



(그림 2) IMPN 예

(그림 2)에 보이는 IMPN에는 멀티미디어 데이터에 대한 정보가 생략되었다. IMPN의 초기 상태  $S_0$ 는 다음과 같다.  $S_0=(M_0, I_0)$  where  $M_0=(p_0(1))$ ,  $I_0=((2,4,7))$ . 상태  $S_0$ ,  $\theta=4$ 에서  $t_0$ 가 격발하였다고 하

자. 그러면 다음과 같은 새로운 상태  $S_1$ 을 얻는다.  $S_1=(M_1, I_1)$  where  $M_1=(p_1(1), p_3(1), p_5(1))$ ,  $I_1=((2,5,7), (5,10,15), (10,15,20))$ . 이 상태에서는  $t_1$ 이 2와 7 단위 시간 사이에서 격발 가능하다.  $\theta=6$ 에서 격발하면 상태는 다음과 같은  $S_2$ 로 변환한다.

$S_2=(M_2, I_2)$  where  $M_2=(p_2(1), p_3(1), p_5(1))$ ,  $I_2=((2,5,7), (\max(0, 5-6), 10-6, \max(0, 15-6)) = (0, 4, 9), (\max(0, 10-6), 15-6, \max(0, 20-6)) = (4, 9, 14))$ .  $t_2$ 는 or 임으로  $p_2$ 나  $p_3$  중 먼저 출력이 종료하는 데이터에 의하여 격발된다.  $\theta=1$ 에서  $t_2$ 가 격발하였다면 다음과 같은  $S_3$ 를 얻는다.  $S_3=(M_3, I_3)$ , where  $M_3=(p_4(1), p_5(1))$ ,  $I_3=((3,5,9), (3,8,13)) = (4-1, 9-1, 14-1)$ .  $t_3$ 는 "and" 임으로 모든 출력이 끝나야 격발된다.

이상은 통신이 원활한 경우의 예다. 만일  $t_1$ 이 7 단위 시간에 격발할 때까지  $p_1$ 의 데이터가 반 밖에 출력되지 못하였다면, 이를 반영하여 모든 활성화 시간이 변화한다. 예를 들어 [5, 10, 15]는 [10, 20, 30]으로 바뀐다.

#### 4. 결론

멀티미디어 동기화 현상을 모델하기 위하여 많은 Petri net이 개발되었다. 그러나, 기존의 Petri net은 인터넷 상에서 상영될 멀티미디어 시나리오를 표현하기에는 부적절하다. 왜냐하면 활성화 시간의 MAX 시간 이전에 전송된 데이터가 전혀 없을 만큼 전송 속도가 느릴 경우에는 아무것도 보여 주지 못하는 경우가 발생할 수 있기 때문이다. 활성화 시간의 MAX를 무한대로 놓으면 이러한 현상을 방지할 수는 있다. 그러나 이렇게 하면 시나리오 진행이 가장 늦게 도착하는 데이터에 의하여 결정됨으로 사용자에게 지루함을 주게된다.

인터넷은 속도가 불규칙하고 사용자가 전 세계에 퍼져 있기 때문에 사용자와 서버간의 거리에 따른 전송 속도의 편차가 크다. 따라서, 속도의 변화를 동적으로 반영하는 Petri net이 필요하다. 본 논문은 인터넷에서 상영될 수 있는 멀티미디어 시나리오를 표현하기 위하여 사용될 수 있는 Petri net을 제안하였다. 제안된 Petri net은 WWW 상에서 멀티미디어 시나리오를 제작하고, 이것을 해석하여 브라우저에 출력하는 소프트웨어 개발에 사용될 수 있다.

#### 5. 참고문헌

- [1] T. Little and A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects," IEEE Journal on Selected Areas in Communications, vo 1. 8, No. 3, April 1990, pp. 413-427.
- [2] M. Diaz and P. Senac, "Time Stream Petri Nets A Model for Timed Multimedia Information," A Application and Theory of Petri Nets 1994, Springer-Verlag, Berlin, pp. 219-238.
- [3] Y. Zhou and T. Murata, "Fuzzy-Timing Petri Net Model for Distributed Multimedia Synchronization,"