

멀티미디어 응용의 명세, 분석 및 재생제어를 위한 베타분포형 시간 패트리넷 모형

이진석[†] · 이강수^{**}

요 약

본 연구에는 멀티미디어 타이틀의 저작과 동기화 문제를 효과적으로 명세하고 분석할 수 있으며, 타이틀의 재생을 직접 제어할 수 있는 통합모형인 BTPN모형과 패러다임들을 제시한다. 타이틀내의 객체들간의 모든 관계들을 BTPN구조와 BTPN언어로 명세하는 방법과 잔여시간형 도달성그래프, 타이밍다이아그램 등을 이용하여 BTPN을 분석하는 방법을 제시하였다. 특히, PERT/CPM방법으로부터 도입한 임계객체경로나 최장경로 개념은 타이틀 저작 및 재생시의 불확실성을 모형화하며 재생시간의 조정도 가능하게 하고 있다.

A Beta-distributed Timed Petri Net Model for Specification, Analysis and Playout Control of Multimedia Titles

Lee, Jin Seok and Lee, Gang Soo Lee

ABSTRACT

In this paper, we propose a BTPN (Beta-distributed Timed Petri Net) model which is not only an effective multimedia synchronization and authoring specification model, but also a direct control model for playout of a title. Methods of specification of relationships among all media objects in a title by using the BTPN structure and language, as well as methods of analysis of the BTPN by means of a Remaining Timed Reachability Graph and Timing diagram, are proposed. A concept of critical object path, coming from PERT/CPM, is useful for modeling the uncertainty of playout of a multimedia title and editing of title

1. 서 론

멀티미디어 타이틀의 저작(authoring) 패러다임은 다음과 같이 전통적인 프로그래밍 패러다임과 대응시킬 수 있다: 시나리오 또는 스토리보드 명세 \approx 알고리즘 명세; 미디어객체(=자료파일+제어기) \approx 객체(=자료구조+메소드); 저작 또는 명세언어 \approx 프로그래밍 언어; 저작도구 \approx 개발도구.

그러나, 타이틀내의 미디어객체(media object)들간의 시간 및 공간 동기화(temporal and spatial synchronization)를 위한 명세(specification) 및 제어

문제는 전통적인 프로그래밍 패러다임에는 대응하지 않으며 실시간/병행(realtime/concurrent) 프로그래밍 패러다임에 부분적으로 대응시킬 수 있다. 따라서, 시간 동기화 명세[1,2]를 위해 CSP, 패트리넷 및 Statechart[22]등과 같은 기존의 실시간/병행 명세모형들을 이용하여 왔으며, 특히 패트리넷[3]은 하이퍼텍스트 저작 및 브라우징[10,11]과 멀티미디어 동기화에 주로 사용되고 있다.

기존의 패트리넷기반 동기화 모형, OCPN [4], TSPN [5], HTSPN [6], MHPN [7], XOCPN [8], AOCPPN [9], Telrellis [10], MORENA [11], IPN [12], PG [13], RTSM [14], DSPN [15], SSTS[19], MRL [20] 및 PNDT[21]들은 주로 시간적 동기화 명세를 위해 제시되었으며, 사용자와의 대화(즉, 선택), 미디

[†] 한남대학교 대학원 컴퓨터공학과 박사과정 수료

^{**} 한남대학교 컴퓨터공학과 교수

어객체간의 모든 부분관계의 명세, 공간동기화 및 타이틀 재생시(playout)의 불확실성과 같은 동적인 특성을 명세하기에는 불충분하다.

또한, 병행시스템의 명세 및 분석도구인 패트리넷은 주로 타이틀의 명세만을 위한 목적으로 사용되고 있으며, 기존의 명세결과로부터 타이틀의 재생을 위한 제어에는 활용하지 못하고 있다. 이는 기존의 모형에서는 미디어객체의 재생시간을 플레이스나 아크에 부여했기 때문으로 판단된다.

본 논문에서는 기존의 연구들에서는 부분적으로만 다루고있는 패트리넷기반 멀티미디어 동기화 문제에 관련된 10가지의 문제를 해결하고자 한다.

- Q1. 미디어객체간의 가능한 모든 부분관계들의 정형명세
- Q2. 시간과 공간 동기화를 동시에 명세할 수 있는 모형의 개발
- Q3. 미디어객체들간(또는, 내부의)의 계층적 동기화 방법 개발
- Q4. 도형적, 구문적 및 시간축 방식의 명세를 동시에 할 수 있는 통합적 명세모형의 개발
- Q5. 타이틀의 명세뿐 아니라 수행(또는, 분석, 리허설, 에디팅)과 재생제어를 위한 통합적 모형의 개발
- Q6. 타이틀의 시나리오의 명세와 미디어객체명세의 분리 방법
- Q7. 타이틀 재생시의 지연시간의 불확실성을 모형화하는 방법
- Q8. 패트리넷을 타이틀의 명세뿐 아니라 명세의 분석 및 재생제어에 활용하는 방법
- Q9. 타이틀 재생중 사용자의 개입을 처리하는 방법
- Q10. 타이틀의 재생시간을 줄이거나 늘이기 위한 타이틀의 편집방법

본 논문에서는 이들 문제를 BTPN(Beta-distributed Timed Petri Net)기반 저작 및 명세 패러다임을 통해 해결하며, 기존의 OCPN에 기반을 둔 모형들의 단점들을 해결한다. 본 논문의 2장에서는 멀티미디어객체관계를 정형화하며, 3장에서는 BTPN 모형과 BTPN기반 저작 패러다임을 제시한다. 4장에서는 타이틀의 BTPN명세방법을 제시하며 5장에서는 분석, 재생 및 편집방법을 제시한다. 6장에서는 BTPN

의 사례를 보이며 7장에서는 기존의 패트리넷기반 모형들과의 비교 평가 결과를 보이며 결론을 맺는다.

2. 멀티미디어 객체 관계

2.1 타이틀과 객체

‘타이틀’은 미디어객체와 그들간의 관계(또는, 시나리오)로 구성되며, ‘미디어객체’(이하 ‘객체’로 약칭함)는 자료파일과 재생을 위한 하드웨어 또는 소프트웨어 연산으로 구성된다. 객체는 이름, 상태, 시간 및 위치속성을 가지며 한 개 이상의 원자적 객체인 ‘미디어스트림’으로 구성된다.

객체의 ‘타입’(또는, 자료 파일)의 예로서 비디오(*.avi, *.mpg), 이미지(*.tiff, *.pcx, *.jpg, *.bmp), 그래픽(*.gif, *.tga, *.eps, *.bmp, *.pcx, *.tiff), 오디오(*.mid, *.wav, *.rol, *.sog), 텍스트(*.hwp, *.text, *.html) 및 애니메이션객체 등이 있다.

객체의 ‘상태’는 타이틀의 재생동안 펜딩, 런, 정지, 리뷰, 프리뷰 중의 하나이며 사용자의 개입 또는 객체제어기로부터의 사건에 의해 상태가 전이된다. 각 객체에는 재생 기간(duration)이 부여되며, 비디오 등과 같은 가시적 객체는 디스플레이 상의 재생위치가 부여된다. 타이틀의 저작자는 리허설 또는 분석을 통해 객체상의 자료파일을 편집함으로써, 객체의 재생시간을 변화시킬 수 있으며 객체의 크기나 위치를 변화시킬 수 있다.

예컨대, [sample, video, run, C:\app\sample.mpg, HW-mpeg, 10 min, (20,0)-(600,700)]은 sample이라는 객체는 비디오 타입이고 런 상태이며 디렉토리 “C:\app\sample.mpg”에 저장되어있고 하드웨어 MPG에 의해 10분간 작동되며 화면상의 하단 좌표 (20,0)부터 상단좌표 (600,700)사이에서 디스플레이 됨을 의미한다.

2.2 객체 관계

객체관계(즉, 스토리보드 또는 시나리오)는 타이틀상의 객체들간의 논리적, 시간적 또는 공간적 관계로서 다음과 같이 정의한다.

[정의 1] 객체간 관계(Inter-object relations, IOR) : 두 객체간의 시간 독립적인 부분관계이다. O_i, O_j, O_k 및 O_l 를 객체라 하고 @, @_i 및 @_j들을 동기화

를 위한 가상객체라 할때,

- *sequence* ($O_i \rightarrow O_j$) : O_i 는 O_j 에 논리적으로 선행한다.
- *merge* ($O_i \bowtie O_j$) : ($O_i \rightarrow @$) \wedge ($O_j \rightarrow @$)
- *choice*
 - *free-choice* ($O_i \bowtie O_j$) : [($@ \rightarrow O_i$) \otimes ($@ \rightarrow O_j$)] = [\neg ($@ \rightarrow O_i$) \wedge $@ \rightarrow O_j$] \vee [($@ \rightarrow O_i$) \wedge \neg ($@ \rightarrow O_j$)] (\otimes 는 exclusive-or 연산자)
 - *restricted-choice* ($O_i \bowtie O_j$) : [($O_i \bowtie O_j$) \wedge ($@_i \rightarrow O_i$)] \vee [($O_i \bowtie O_j$) \wedge ($@_j \rightarrow O_j$)]
- *parallel*
 - *pure-parallel* ($O_i \parallel O_j$) : \neg ($O_i \rightarrow O_j$) \wedge \neg ($O_j \rightarrow O_i$)
 - *semi-parallel* ($O_i \parallel O_j$) : [($O_i \parallel O_j$) \wedge ($O_i \bowtie @_i$)] \vee [($O_i \parallel O_j$) \wedge ($O_j \bowtie @_j$)]
 - *joined-parallel* ($O_i \dashv O_j$) : ($O_i \rightarrow @$) \wedge ($O_j \rightarrow @$) \wedge ($O_i \parallel O_j$)
 - *forked-parallel* ($O_i \vdash O_j$) : ($@ \rightarrow O_i$) \wedge ($@ \rightarrow O_j$) \wedge ($O_i \parallel O_j$)

객체는 계층적으로 정의되므로, 원자객체 또는 미디어스트림들 간에도 IOR을 갖는다. 이 경우, IOR은 객체내(intra-object) 관계라 할 수 있다.

2.3 객체 시간적 관계

객체상의 사건 θ 는 그 개시점 θ_i^α 과 종료시점 θ_i^β 으로 나타낸다. 사건의 발생시점인 타임스탬프 Ψ 는 실수이며, 다음과 같은 시간함수(즉, 시계) Ω 에 의해 측정된다.

$$\Psi_i^\alpha = \Omega(\theta_i^\alpha), \quad \Psi_i^\beta = \Omega(\theta_i^\beta)$$

구간(interval) Δ 는 두 타임스탬프상의 범위를 나타내며 다음과 같은 연속적 기간(period)으로 정의한다.

$$\Delta_i = [\Psi_i^\alpha, \Psi_i^\beta] \quad (\text{즉, } \Delta_i \in \{ \Psi \mid \Psi_i^\alpha \leq \Psi \leq \Psi_i^\beta \})$$

객체 O_i 상의 구간 Δ_i 의 지속시간(duration) τ_i 는 $\Psi_i^\beta - \Psi_i^\alpha$ 로 계산되며, O_i 의 재생상태는 시간축상에서 $\langle \Psi_i^\alpha, \tau_i \rangle$ (즉, 개시점과 지속)을 직선으로 나타낼 수 있다.

[정의 2] 객체구간 관계(Object interval relation, OIR) : 두 객체 O_i 와 O_j 의 재생구간들 간의 시간 종속적(즉, temporal)이진 부분관계이다.

- Δ_i *equal* Δ_j : ($\Psi_i^\alpha = \Psi_j^\alpha$) \wedge ($\Psi_i^\beta = \Psi_j^\beta$)
- Δ_i *precede* Δ_j : $\Psi_i^\beta < \Psi_j^\alpha$
- Δ_i *meet* Δ_j : $\Psi_i^\beta = \Psi_j^\alpha$
- Δ_i *overlap* Δ_j : $\Psi_i^\alpha < \Psi_j^\alpha < \Psi_i^\beta < \Psi_j^\beta$
- Δ_i *start* Δ_j : $\Psi_j^\alpha = \Psi_i^\alpha < \Psi_i^\beta < \Psi_j^\beta$
- Δ_i *during* Δ_j : $\Psi_j^\alpha < \Psi_i^\alpha < \Psi_i^\beta < \Psi_j^\beta$
- Δ_i *end* Δ_j : $\Psi_j^\alpha < \Psi_i^\alpha < \Psi_i^\beta = \Psi_j^\beta$

객체는 계층적으로 정의되므로, 원자객체 또는 미디어스트림들 간에도 OIR을 갖는다. 이 경우, OIR은 객체내(intra-object) 구간관계라 할 수 있다. 구간관계는 OCPN[4]과 그 확장 모형들[5~13]에서 사용되고 있다.

2.4 객체 공간관계

오디오 객체를 제외한 동시에 재생되는 객체(즉, *equal*, *overlap*, *start*, *during* 및 *end* 관계)간의 공간 관계는 다음과 같이 정의된다.

[정의 3] 객체 공간 관계(Object spatial relation, OSR) : $\#O_i$ 는 객체 O_i 의 위치이며 아래-왼쪽 좌표 (LLx_i, LLy_i)와 위-오른쪽 좌표 (URx_i, URy_i)로 구성될 때, O_i 의 디스플레이 범위는 ($URx_i - LLx_i$) \times ($URy_i - LLy_i$)가 된다. O_i 와 O_j 의 *spatial overlap* 관계의 필요·충분조건은

$$[(LLx_i < URx_j \leq URx_i) \vee (LLx_i \leq LLx_j < URx_i)] \wedge [(LLy_i < URy_j \leq URy_i) \vee (LLy_i \leq LLy_j < URy_i)] \text{이다.}$$

3. BTPN모형과 타이틀 프레임워크

3.1 BTPN 모형의 정의

BTPN모형은 타이틀의 명세와 분석을 위한 수행 가능성(*executable*) 모형이며, 도형적명세(즉, BTPN 구조에 의한 명세) 및 구문적명세(즉, BTPN언어에 의한 명세)를 동시에 할 수 있다.

[정의 4] BTPN구조 = ($P, T, I, SI, O, SO, Mo, OBj$)

- $P = (NP, CP, SP)$: 플레이스 클래스
 - NP : 정규(normal)플레이스 집합

- CP : 조건(condition)플레이스 집합(타이틀의 재생시 사용자 개입의 명세용)
- SP : 공간(spatial)플레이스 집합(점으로된 원으로 표시하며 공간관계의 명세용)
- $T = (ST, OT)$: 트랜지션 클래스
 - ST : 동기(synchronization)트랜지션(얇은 막대로 표시하며 객체간의 동기점을 명세함. 즉시(immediate) 트랜지션의 일종임(즉, 지연시간이 0임))
 - OT : 객체트랜지션 집합(진한 막대로 표시하며 객체의 재생을 명세함)이며 OBJ함수에 의해 객체와 연관 됨.
- $I \subset P \times T$: 입력함수 집합(트랜지션으로의 입력아크)
- $SI \subset SP \times T$: 입력함수 집합(입력 공간아크이며 점선으로 표시)
- $O \subset T \times P$: 출력함수 집합(트랜지션의 출력아크)
- $SO \subset T \times SP$: 출력함수 집합(출력 공간아크이며 점선으로 표시)
- K : 토큰이며 타이틀의 제어흐름으로 해석
- Mo : 초기마킹임
- OBJ : $PT \rightarrow \Sigma^* \times \Sigma^* \times \Sigma^* \times \Sigma^* \times \Sigma^* \times ((Q^+ \cup \infty) \times (Q^+ \cup \infty) \times (Q^+ \cup \infty)) \times ((Q^+ \cup \infty) \times (Q^+ \cup \infty) \times (Q^+ \cup \infty) \times (Q^+ \cup \infty))$ (Σ^* 는 임의 길이의 심볼스트링, Q^+ 은 양의 유리수임)

• $OBJ(pt_i) = [name_i, type_i, state_i, data-file_i, operation, (a_i, m_i, b_i), (LLx_i, LLy_i) - (URx_i, URy_i)]$,

여기서, $name_i$ 는 ot_i 에 연관된 객체 O_i 의 이름; $type_i$ 는 O_i 의 타입(예; 비디오, 이미지); $state_i$ 는 O_i 의 상태(예; 런, 리뷰); $data-file_i$ 는 객체의 파일명; $operation_i$ 는 미디어 드라이버의 명칭(예; MPEG, JPEG); (a_i, m_i, b_i) 는 O_i 의 재생 지속시간(duration)이며 a_i, m_i 및 b_i 는 각각 최악, 최적 및 최악의 지속시간이며 베타분포의 모수들; $(LLx_i, LLy_i) - (URx_i, URy_i)$ 는 디스플레이상의 O_i 의 위치와 크기이다.

[정의 5] BTPN언어 : 부록 A에서 기술하였다.

BTPN언어는 BTPN구조와 논리적으로 동일하며 BTPN모형은 다음과 같은 특성을 갖는다. ① *Timed transition Petri net*이다. 즉, 트랜지션의 점화지연 시간은 베타분포를 가지며 객체의 재생시의 불확실성과 객체의 편집을 모형화할때 유용하다. ② *Place/transition net*이다. 즉, OCPN모형과 그 확장 모형들과는 달리 구조적 제한이 없다. ③ *Object linked transition net*이다. 즉, OCPN모형과는 달리, 객체가 객체트랜지션에 연관된다. 이를 통해, 미디어 관계(즉, 시나리오)와 미디어 객체를 분리하여 명세할 수 있다.

3.2 BTPN 기반 저작 패러다임

BTPN기반 저작 패러다임은 그림 1에서 보이며 그림 2는 저작 및 재생시의 상태전이를 보인다. BTPN모형은 다음과 같이 모든 종류의 명세가 가능하다. ① BTPN구조(즉, 도형적 명세) 또는 BTPN언어(즉, 스크립트형 명세)에 의한 정적명세. ② BTPN의 점화로부터 얻은 RTRG(잔여시간형 도달성그래프이며 상태 또는 사건기반 명세)와 타이밍다이어그램(시간축형 명세)에 의한 동적명세.

또한, 저작자는 타이틀을 리허설해 볼 수도 있으며 시나리오를 변경할 수도 있음을 의미한다. 이를 통해, 저작자는 타이틀을 단계적 정제화 과정을 거쳐 개발할 수 있다. 표 1은 BTPN모형과 저작시의 개념간의 대응성을 보인다.

4. 명세 방법

4.1 객체 관계의 명세

모든 IOR들에 대한 BTPN 명세결과들은 그림 3에서 보인다. *sequential, free-choice* 및 *merge* 관계만을 가지는 패트리넷을 Free-choice net이라 하며[3], *parallel* 관계를 가지는 멀티미디어 동기화의 명세에는 부적합하다. 또한, *sequential, joined-parallel* 및 *forked-parallel* 관계만을 가지는 것을 Marked graph라 하며[3] OCPN이나 시간축 명세모형들은 동기화 명세에는 적합하지만 사용자 개입(즉, choice)을 명세하기 어렵다. 그러나, BTPN은 구조적 제약이 없으므로 이들의 문제점을 해결하고 있다.

free-choice 관계는 다른 조건을 고려하지 않고 객체

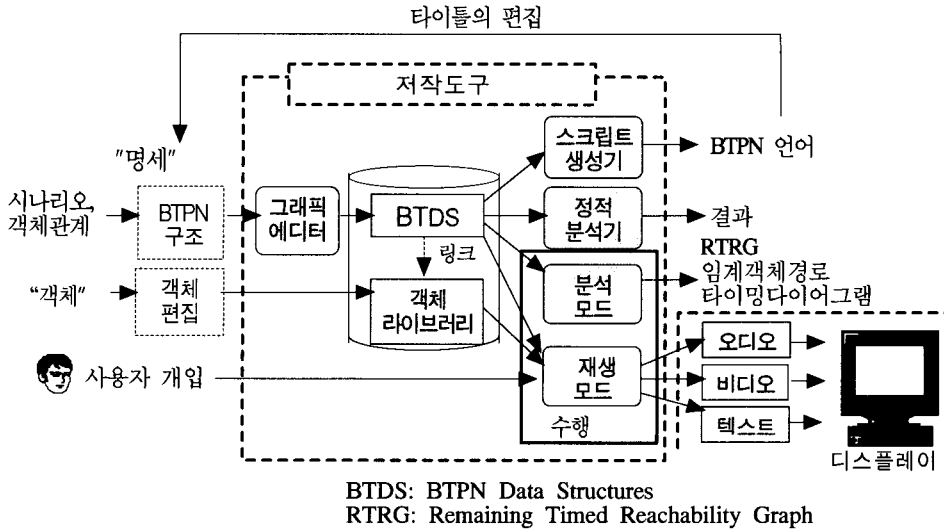


그림 1. BTPN기반 저작 패러다임의 골격

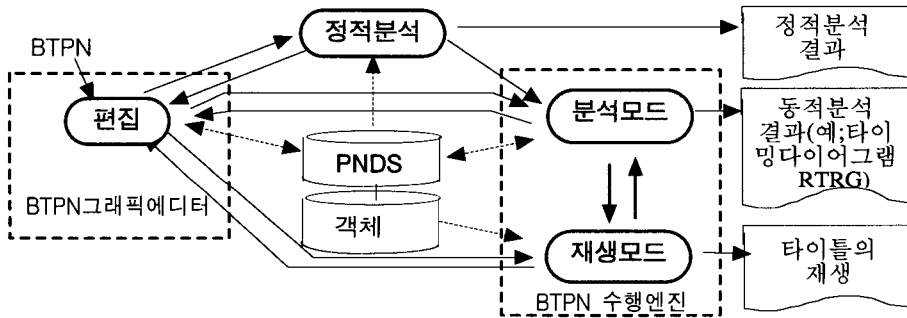


그림 2. BTPN기반 저작시의 상태전이

를 선택하는 상황을 명세하며, *restricted-choice* 관계는 다른 조건을 고려하여 선택해야 하는 상황을 명세한다. 또한, *pure-parallel* 관계는 객체들의 동시 재생을 의미하며, *semi-parallel* 관계는 객체가 동시에 재생되거나 순서적으로 재생될 수도 있는 상황을 명세한다.

4.2 객체시간 관계의 명세

객체트랜지션 O_i 에 점화 지속시간 τ_i 를 레이블링 함으로써, 모든 OIR들을 BTPN으로 명세할 수 있다. 그림 3의 (k)와 (l)은 *sequence* 및 *forked-parallel* 관계 세그먼트를 통해 모든 OIR을 명세한 결과를 보인다.

4.3 객체공간 관계의 명세

공간동기화는 그림 4-(a) 및 (b)처럼 디스플레이 상의 동시객체들의 재생 중복성 여부를 명세하는 것이다. 동시에 재생되는 객체들간에 *spatial overlap* 관계를 가질 때, 그림 4-(c)처럼 BTPN으로 이 관계를 명세하지는 않는다(예; 동영상 객체와 문자 객체를 동시에 재생). 그러나, *spatial non-overlap* 관계를 가질 때, 저작자는 그림 5-(d)와 같이 BTPN에 공간플레이스와 공간아크를 통해 이 사실을 명세해야 한다. 이 경우, 공간플레이스내의 토큰은 디스플레이의 사용권을 나타낸다. 여기서, 공간플레이스, 공간아크 및 토큰은 *spatial non-overlap* 관계의 명세용으로만 사용하며 BTPN의 수행시에는 고려하지 않는다.

표 1. BTPN과 저작시의 개념들간의 대응성

명 세	BTPN모형	저작
정적특성	BTPN 구조 BTPN 언어 NP(정규플레이스) CP(조건플레이스) SP(공간플레이스) ST(동기트랜지션) OT(객체트랜지션) OBJ(연관함수) I, O, SI 및 SO (아크) 토큰 M ₀ (초기마킹)	도형적 명세 스크립트형 명세 사전 및 사후조건 사용자 개입 공간 조건 객체의 동기화 멀티미디어 객체 객체링크 객체간의 부분관계 제어 또는 자료 초기상태
동적 특성	트랜지션의 점화가능(enabling) 트랜지션의 점화(firing) BTPN의 수행 - 분석모드 - 재생모드 잔여시간형 도달성그래프 타이밍 다이어그램 임계 객체경로 및 스케줄링	객체의 재생준비 완료 객체의 재생(run, pause, forward, rewind) 타이틀의 수행 - 분석, 리허설, 객체의 편집 - 재생제어 상태 및 사건기반명세 시간축기반 명세 임계객체경로 및 편집

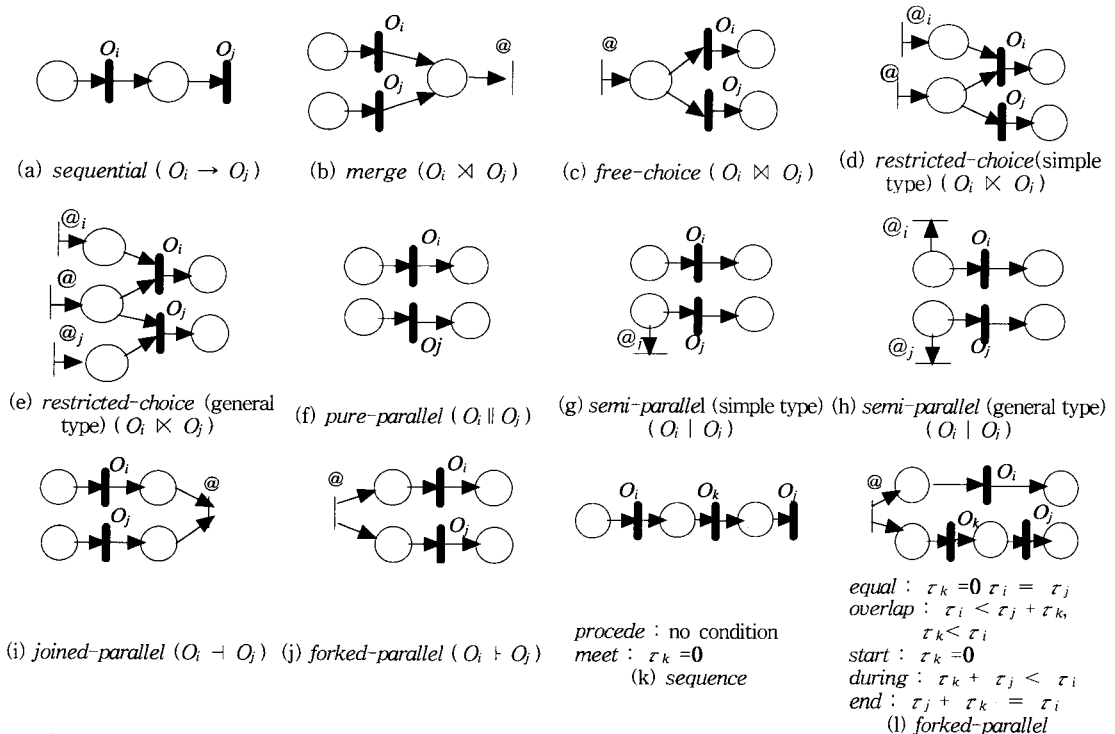


그림 3. 객체관계와 객체시간 관계들의 BTPN 명세방법

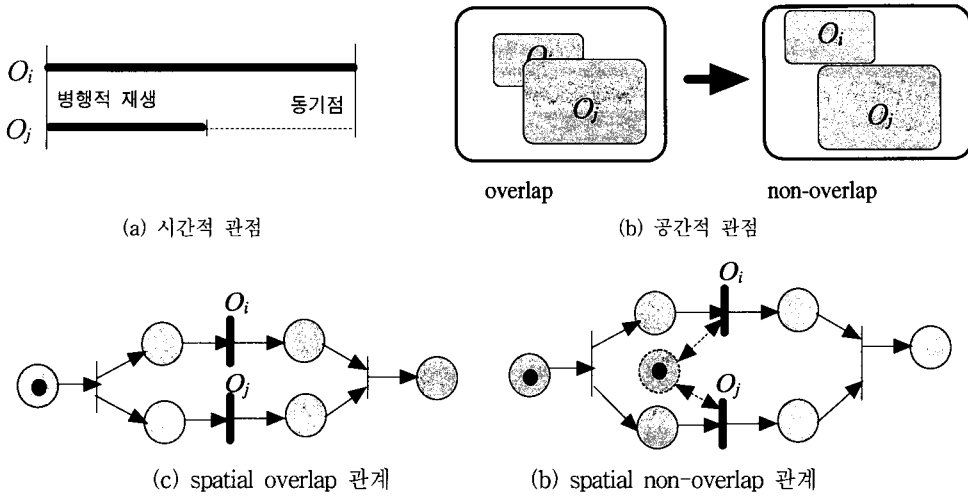


그림 4. 공간관계의 BTPN 명세방법

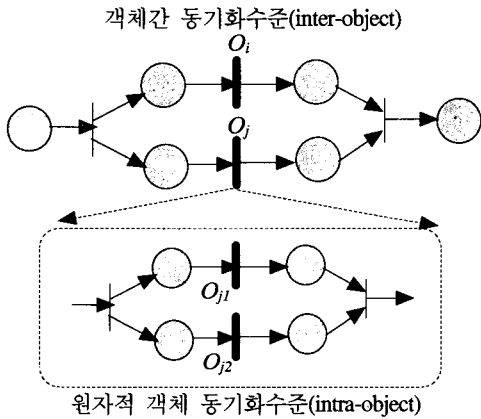


그림 5. BTPN의 계층적 명세방법($O_i = O_{i1} \cup O_{j2}$)

4.4 계층적 명세방법

패트리넷, Statechart 및 Dataflow 다이어그램들은 합성 컴포넌트나 매크로 트랜지션을 통해 모형의 추상화 수준을 선택할 수 있다. BTPN에서도 그림 5처럼 단계적 정제방법을 통해 명세할 수 있다. 여기서, 최하위 수준의 명세를 원자적 객체동기화 수준이라 하며 객체내부(intra-object)의 동기화 명세가 되며, 상위 수준의 명세들은 객체간(inter-object) 동기화 명세에 해당한다.

5. 분석, 편집 및 재생제어 방법

BTPN은 타이틀의 명세모형일 뿐 아니라, 분석 및

재생제어 모형이기도하다. 특히, 분석을 위해 BTPN을 수행할 때를 '분석모드'라 하며, 객체의 재생을 제어하기 위해 수행할 때를 '재생모드'라 한다.

5.1 분석 모드 및 편집

분석모드에서는 BTPN을 수행하므로써, BTPN으로 명세화된 타이틀의 생존성(liveness), 도달성(reachability), 보존성(conservability)에 대한 정보가 포함된 잔여시간형 도달성그래프(RTRG)를 생성한다. 또한, RTRG를 통해 타이밍다이아그램(TD)과 임계객체경로(COP)를 생성할 수 있다. RTRG와 TD는 타이틀의 동적인 특성 정보를 포함하고 있으며, 타이틀의 저작자(또는, 분석자)는 이들 정보를 통해 BTPN의 초안을 개선할 수 있다. 그림 6은 분석모드의 개념과 상태의 전이를 보인다.

(1) 분석모드에서의 BTPN 수행 알고리즘

분석모드에서는 BTPN을 수행할 때 "선선택형 병행점화규칙"(pre-choiced concurrent firing rule)을 사용한다. 이 규칙은 병행점화규칙(concurrent firing rule, CFR)과 선선택규칙(pre-choice rule, PCR)을 결합한 것이다. CFR은 타이틀내의 객체들의 재생의 동시성을 나타내며 BTPN분석시의 상태폭발 문제를 줄이기 위한 것이며 PCR은 CFR의 문제점을 해결하기 위한 규칙이다.

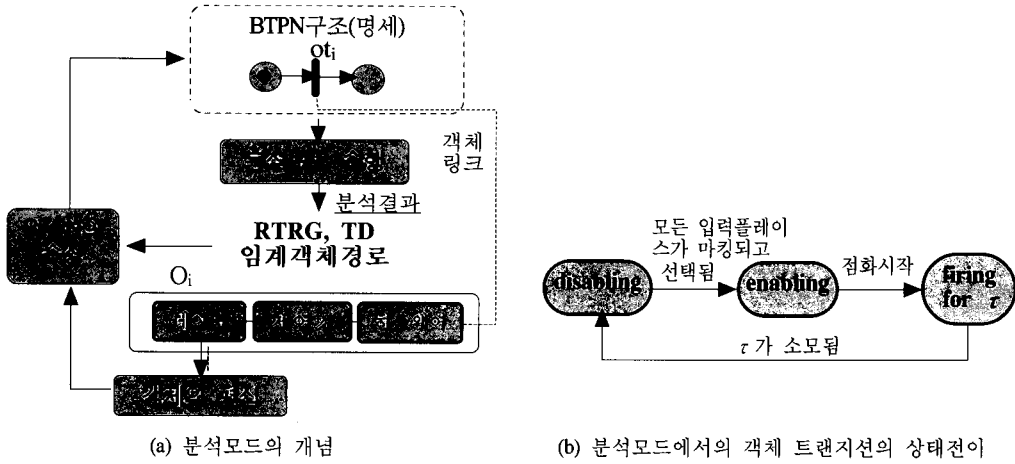


그림 6. BTPN의 분석모드의 개념과 상태전이

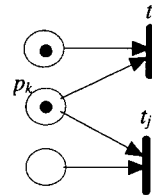
[정의 6] 선선택규칙(PCR) : BTPN내에 *restricted-choice* 관계를 가진 모듈 내에 1개의 트랜지션만 점화가능(enabling)된다면, 모듈을 그림 7처럼 선선택형으로 변환한다.

표준적인 점화규칙[3]에 의하면 그림 7-(a)에서 t_i 만 점화가능하다. 그러나, 선선택규칙을 적용하면 그림 7-(b)와 (c)처럼 t_i 는 점화되거나 안될수도 있다. 예컨대, 그림 8로부터 다음과 같은 사실을 관찰할 수 있다. ① 만일 t_2 와 t_3 가 병행적으로 점화되면, 데드락을 발견할 수 없게된다. ② 만일 P_2 가 t_2 를 선선택한다면, t_1 과 t_2 는 병행적으로 점화된다. 만일 p_2 가 t_4 를 선선택한다면, t_1 은 우선 점화되며 t_4 가 다음으로 점화된다. 따라서, 데드락을 발견 할 수 있게되며 CFR의 문제점을 해결 할 수 있다.

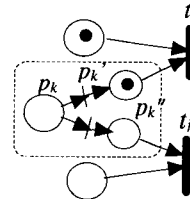
[정의 7] 배타적 점화가능 트랜지션(Exclusively firable transition, EFT) : $\bullet t_i$ 를 t_i 의 입력 플레이스 집합이라 하고, $\#p_k$ 를 p_k 내의 토큰수라할때,

- $\forall p_k \in \bullet t_i$ 에 대해 $[\#p_k \geq 1]$ 이면, t_i 는 점화가능(enabled) 트랜지션이다.
- $\forall t_i, t_j \in CFR$ 에 대해, t_i 와 t_j 가 점화가능(enabled)일때, $\bullet t_i \cap \bullet t_j = \emptyset$ and $t_i \parallel t_j$ (즉, *parallel* 관계)이면, t_i 와 t_j 는 병행 점화가능 트랜지션(*Concurrently firable transition, CFT*) 집합의 원소이다.
- $\forall CFT_i, CFT_j \in EFT, \forall t_i \in CFT_i, \forall t_j \in CFT_j$ 에 대해, $\bullet t_i \cap \bullet t_j \neq \emptyset, t_i \leftrightarrow t_j$ (즉, *choice*

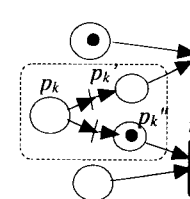
관계)이며, t_i 와 t_j 는 배타적 점화가능 트랜지션(*Exclusively firable transition, EFT*) 집합의 원소이다.



(a) *restricted-choice* 관계를 가진 모듈



(b) p_k 는 t_i 를 선선택함



(c) p_k 는 t_j 를 선선택함

그림 7. 선선택 규칙

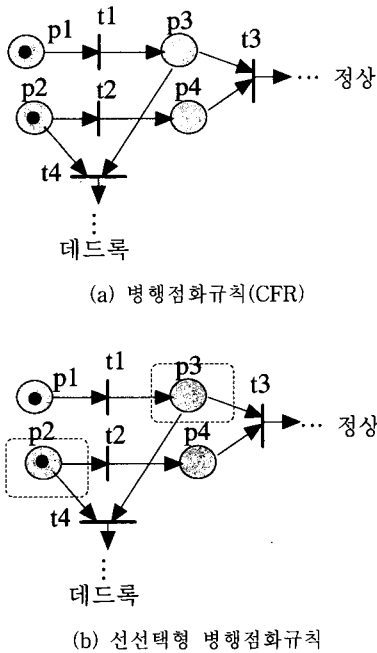


그림 8. CFR의 문제점과 해결방법

EFT내의 CFT들은 상호배타적으로 점화되며 CFT내의 트랜지션들은 병행적으로 점화할 수 있다. 예를 들어, t_1, t_2, t_3 및 t_4 가 $(t_1 \leftrightarrow t_2) \parallel (t_3 \leftrightarrow t_4)$ 관계를 가지면, $EFT = \{CFT1, CFT2, CFT3, CFT4\}$, $CFT1 = \{t_1, t_3\}$, $CFT2 = \{t_1, t_4\}$, $CFT3 = \{t_2, t_3\}$, $CFT4 = \{t_2, t_4\}$ 가 된다.

[정의 8] 잔여시간형 도달성그래프(Remaining timed reachability graph, RTRG) : RTRG는 노드집합 N (등근 박스로 표시)과 레이블된 아크 LA 로 구성된다. 각 노드 $NODE^i$ 는 마킹셋 MS^i 와 EFT^i 를 나타낸다. CFT^i 내의 각 t_j 는 " $t_j : \delta_j$ "로 표시한다 (여기서, t_j 는 점화트랜지션이며 δ_j 는 점화종료까지의 잔여기간이다). 점화 종료 트랜지션집합 ft_j 는 각 LA 에 레이블링된다. ft_j 는 " $t_i \parallel \dots \parallel t_n$ "로 나타낸다 (여기서, $n \geq 1$, t_i, \dots, t_n 들은 ft_j 의 원소이다).

그림 9는 RTRG의 노드의 타입을 보이며 부록 B에는 RTRG의 생성 알고리즘(즉, BTPN의 수행 알고리즘)을 보인다. BTPN내의 각 객체트랜지션 ot_i 에는 점화지속시간으로서 베타분포의 3개의 모수(즉, a_i, m_i, b_i)가 부여되지만, 상수형 점화지속시간 τ_i 로 점화한다. 여기서, $\tau_i = \mu_i = (a_i + 4m_i + b_i) / 6$ (즉,

베타분포에서의 평균치). 이 방법을 통해 타이틀 재생시의 지연의 불확실성을 명세할 수 있으며 RTRG를 통해 타이틀을 쉽게 분석할 수 있다.

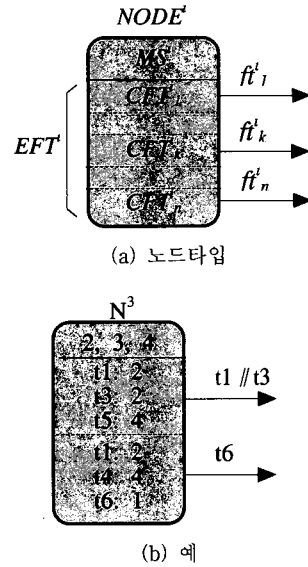


그림 9. RTRG의 노트 타입

(2) 타이밍 다이어그램과 임계객체 경로

타이틀상에서의 재생경로는 소프트웨어의 런(또는, 쓰래드)개념과 유사하며 재생중 사용자의 개입에 의해 경로가 선택된다. 경로는 타이밍 다이어그램으로도 나타낼 수 있으므로, RTRG로부터 여러개의 재생경로(또는 타이밍 다이어그램)을 구할 수 있다. 부록 C에는 RTRG내의 최장경로를 구하는 알고리즘을 보인다.

[정의 9] 타이밍 다이어그램(Timing diagram, TD)은 수직축(객체 축)과 수평축(시간축)으로 형성된 평면상의 수직 및 수평선의 집합이다. 시간구간 Δ_k 는 (ψ_k^α, k') 부터 (ψ_k^β, k') 까지의 직선으로 나타낸다. 여기서, k' 는 객체 O_k 의 좌표이다.

TD는 간트차트와 유사하며 모든 객체구간 관계를 표현할 수 있으며 RTRG상의 한 재생경로로부터 쉽게 구할 수 있다.

(3) 분석 결과

RTRG와 TD로부터 다음과 같은 타이틀의 특성을 구할 수 있다. ① 정량적 특성 : 특정한 객체는 재생이

가능한가(즉, 도달성문제)? 타이틀의 재생시 데드록은 발생할 수 있는가? ② 정성적 특성 : 타이틀 재생시 객체의 최대 병행성은(시스템내의 해당 객체의 제어기 숫자보다는 작아야함)? 타이틀의 재생 종료시간은?

최장경로와 임계객체경로는 타이틀의 재생시간을 늘이거나 단축할 때 사용된다. 예컨대, 30분으로 편집된 타이틀을 20분으로 단축하여 재편집할 경우 타이틀내의 임계객체경로상의 객체들을 편집함으로써 재생시간을 줄이면 된다. 이 개념은 공정관리 모형인 PERT/CPM[18]을 활용한 것이다.

5.2 재생모드

BTPN은 타이틀의 재생을 직접 제어하기 위해 다시 한번 수행된다. 이를 재생모드라하며 PLC(Programmable Logic Control)과 패트리넷 기반 제어기의 명세 표준인 Grafset[16]의 개념을 응용한 것이다. Grafset는 Sequential Foundation Chart(SFC)의 기본을 이루며 1987년에 국제표준으로 사용되고 있다.

그림 10은 재생모드의 개념과 객체트랜지션의 상태전이를 보이며 부록 D에는 재생모드에서의 수행 알고리즘을 보인다. 수행알고리즘에는 *spatial non-overlap* 관계를 해결하기 위한 공간동기화의 제어절차를 포함하고있다. 따라서, BTPN은 타이틀의 동기화 명세뿐 아니라 직접제어에도 활용될 수 있다.

6. BTPN기반 명세와 분석의 예

저작하려는 대화형 의류 광고물의 시나리오는 다음과 같다. 광고의 헤드라인 이미지가 초기스크린에 재생된 후(t1), 회사로고와 회사의 소개가 문자형태로 스크린의 하단에 표시된다(t4). 배경음악을 스피커로 재생한다(t2). 시청자는 패션쇼비디오(t6) 또는 제품비디오(t7)를 선택할 수 있으며 각각에 대해 동시에 소개말(t5)이 재생된다. 다음으로, 이미지(t9)가 팝업되며 시청자는 옵션버튼을 통해 제품에 대한 세부사항을 선택 할 수 도 있다. 스킵버튼을 누르면, 제품에 대한 소개 비디오가 재생된후 다른 제품에

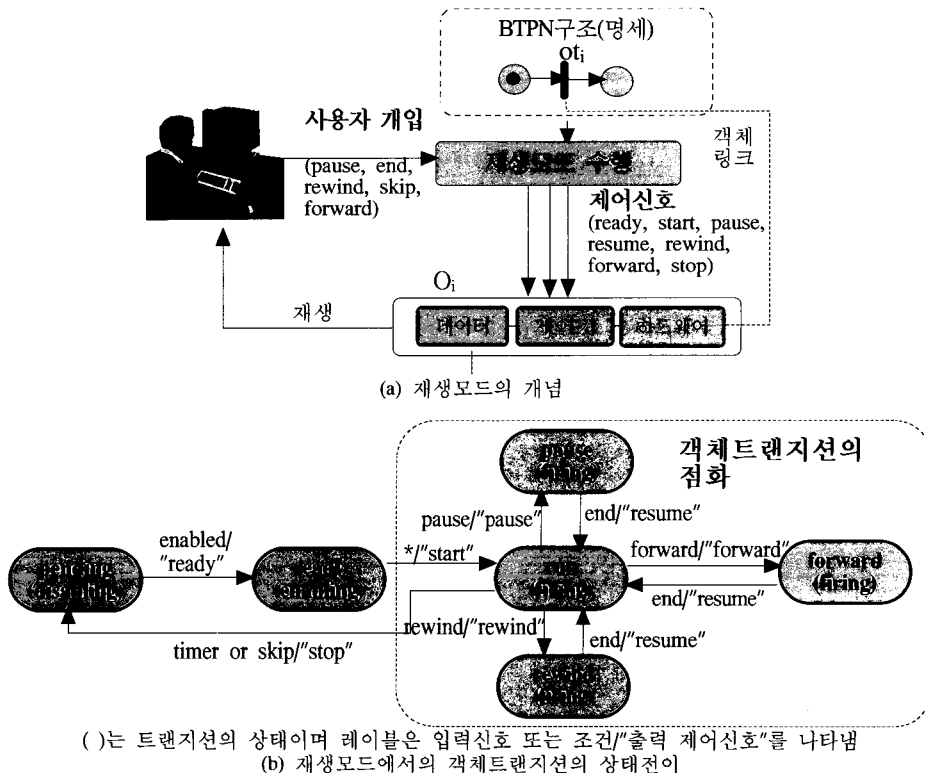


그림 10. BTPN의 재생모드의 개념과 상태전이

대한 이미지가(t13) 계속 재생된다. 계속버튼을 누르면, 제품의 세부설명 비디오(t11)가 스크린 중앙에 회사의 텍스트(t4)와 동시에 재생된후 t11과 t4의 재생이 완료되면 감사의 내용을 담은 새로운 비디오(t12)가 재생된후 다른 제품에 대한 이미지(t13)가 계속 재생된다.

저작자는 11개의 단위 객체들을 준비 및 편집하여 재생시간 등의 모수들과 함께 객체라이브러리에 등록하고 객체별로 해당 재생기(MPG, JPG등)들과 링크시킨다. 객체간의 관계와 재생시간을 고려하여 그림 11과같이 BTPN을 작성한다.

BTPN으로부터 타이틀의 cyclomatic 복잡도는 아크수 - (플레이스수 + 트랜지션수) + 2 = 4임을 알 수 있다. 분석모드에서 BTPN을 수행하므로써 RTRG를 구한다(그림 12).

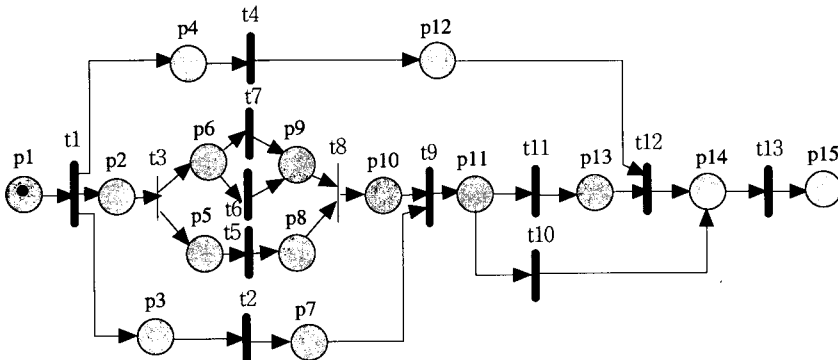
RTRG로부터 병행 재생되는 객체의 최대수는 4임을 알 수 있다.(즉, t2 || t4 || t5 || t6 또는 t2 || t4 || t5 || t7) 또한, RTRG로부터 임계객체경로(즉, t1 → t12 → t13)를 구하고 이에 대한 타이밍 다이어그램도 구할 수 있다(그림 13). 임계객체경로로부터 타이틀의 총 재생시간은 26.8분이며, 이를 20분으로 줄이기 위해서는 임계객체경로상의 t1, t4, t12 및 t13에 부여된 객체(O1, O4, O12, O13)들을 재편집하면 된다.

7. 평가 및 결론

패트리넷은 하이퍼텍스트의 저작 및 브라우징을 위한 모형[10, 11])들에서 뿐만아니라, 멀티미디어 동기화를 위한 모형[5~9, 12~15, 21]로서 활용되고있으며 OCPN[4]을 모체로 하고 있다.

(1) 모형화 능력 : 모형의 구조적 제한은 모형화 능력에 영향을 미친다[3]. OCPN[4], TSPN[5], MHPN[7], XOCPN [8], PG[13], RTSM[14] 및 TFPN[15]과 같은 OCPN기반 모형들은 Marked graph이므로, 객체간의 병행성을 명세 할 수 있지만 choice와 merge관계를 명세하기 어렵다. AOCPN[9]에서도 객체의 skip, freeze, reverse 및 scaling을 위한 제한된 수준의 choice와 merge관계를 명세 할 수 있지만, 이는 명세적 편의를 위해 도입된 것이다. BTPN을 포함한 IPN[12]과 HTSPN[6]은 구조적 제한이 없는 Place/transition net이므로 모든 종류의 객체관계들을 명세할 수 있다.

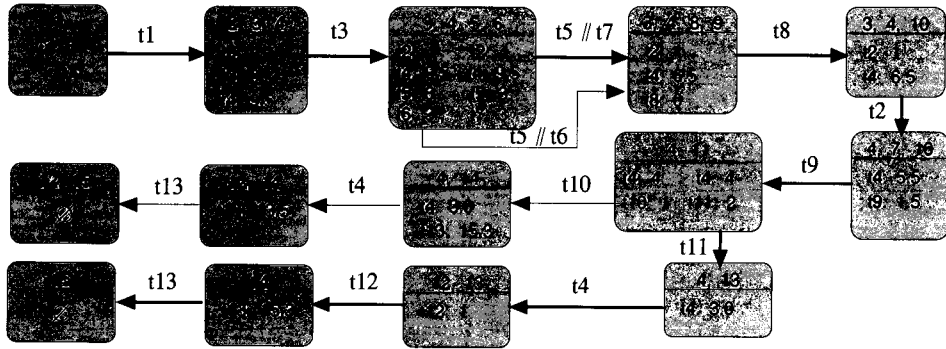
(2) 객체의 할당 : OCPN기반모형[4~15]들에서는 플레이스로 객체를 명세하고 있다. 참고문헌 [5]의 4장에서는 트랜지션에 시간을 부여한 패트리넷(예; Merlin의 Time Petri net)은 동기화 명세를 할수 없



트랜지션	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13
속성													
name	O1	O2	@	O3	O4	O5	O6	@	O7	O8	O9	O10	O11
type	image	audio		text	audio	video	video		image	audio	video	video	image
a	1	3	0	7	2	2	2	0	1	1	1	1	12
m	1	4	0	9	3	3	3	0	1.5	1	2	1	12
b	1	5	0	14	4	4	4	0	2	1	3	1	20
μ	1	4	0	9.5	3	3	3	0	1.5	1	2	1	15.3

(주: 시간단위는 분이며, state, data-file, operation, LLx, LLy, URx, URy 들은 생략함)

그림 11. 사례 타이틀의 BTPN명세



(주: 진한 경로는 임계객체경로임)

그림 12. 사례 타이틀의 RTRG

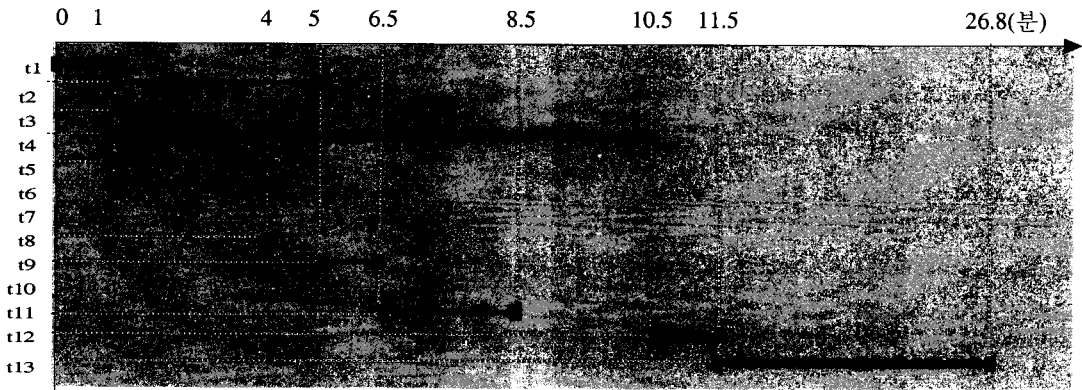
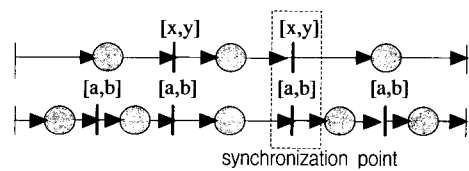


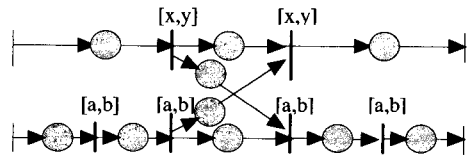
그림 13. RTRG상의 최장 경로에대한 타이밍다이아그램

다고 주장하였으나 이는 잘못된 것임을 그림 14에서 보인다. 따라서, BTPN처럼 트랜지션으로 객체를 명세하는 것은 문제가 없으며, 도달성그래프를 쉽게 구할 수 있으며 좀더 자연스럽다는 장점이 있다. 대부분의 OCPN모형들은 도달성그래프를 생성하지 않으며 타이틀의 분석보다는 단순히 명세만을 위한 것이다.

(3) 객체의 지속시간 : 기존의 패트리넷 기반 동기화 모형들은 다음과 같이 분류할 수 있다. ① 지속시간의 형태별 : 상수형(OCPN, PG, MHPN, XOCPN, AOCPN, RTSN, IPN), 구간형(HTSPN). before와 after지속시간형(XOCPN and PG), 확률분포형(BTPN) ② 객체를 명세한 요소별: 트랜지션(IPN, BTPN), 플레이스(OCPN, XOCPN, AOCPN, TFPN, RTSN, PG), 아크(TSPN, HTSPN), 아크와 플레이스(MHPN)



(a) Time Petri net에의한 동기화의 예



(b) 모형화 결과

그림 14. 참고문헌 [5]에대한 반례

기존의 모형들은 원격 객체의 통신이나 재생시 발생할 수 있는 지터나 통신지연으로 인한 객체의 재생

지속시간의 불확실성을 모형화하지 못하고 있지만 BTPN에서는 베타분포형 지속시간 개념을 통해 이를 해결하고 있다. 베타분포는 불확실성이 있는 PERT/CPM 등과같은 프로젝트 관리 모형에서 이용되고있다.

(4) 분석과 직접제어 : 기존의 OCPN기반 모형들에서는 패트리넷의 모형화 능력만을 활용할 뿐 분석능력(예; 도달성 분석 등)을 활용하지 못하고 있다. 또한, OCPN기반 모형들은 시간형패트리넷이므로 도달성그래프를 구하기가 어려우며 각 논문에서도 이에 대한 해결 방법을 제시하지 않고 있다.

그러나, BTPN모형은 패트리넷의 모든 기능을 최대한으로 활용하고 있으며 명세와 분석뿐 아니라 타이틀의 직접제어에도 활용하고 있다. 특히, BTPN의 분석모드의 수행결과인 RTRG와 TD는 타이틀의 동적인 명세모형이라 할 수 있다. RTRG상의 임계객체경로는 타이틀의 재편집시 활용할 수 있다.

(5) 점화규칙과 상태전이 : OCPN기반모형들에서의 각 플레이스는 그림 15처럼 "3 상태" 마킹규칙을 적용하기 때문에 도달성그래프를 생성하기가 어렵지만, BTPN에서는 "2 상태" 마킹규칙을 적용하고

있으므로 도달성그래프를 생성하기가 쉽다.

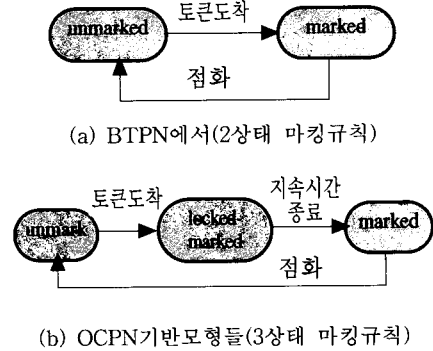


그림 15. 플레이스의 상태전이

결론적으로, 표 2와 같이 BTPN모형은 멀티미디어 타이틀의 저작과 동기화문제를 효과적으로 명세하고 분석할 수 있는 명세모형일 뿐 아니라, 타이틀의 재생을 위한 제어기의 역할도 함을 알 수 있다. 특히, 타이틀을 구성하는 객체들간의 모든 관계를 명세할 수 있으며 기존의 패트리넷기반 동기화 모형들의 단점들을 부분적으로 해결하고 있다. PERT/CPM방법으로부터 도입한 임계객체경로나 최장경

표 2. 문제들에 대한 본 연구의 해결방법과 기존의 해결방법의 비교

문제	본 연구의 방법		기존의 해결방법
	BTPN	BTPN에서 응용한 모형	
Q1	<ul style="list-style-type: none"> 구조적 제한을 두지 않음 IOR, OIR 및 OSR BTPN명세방법 	-	Marked graph수준(선택관계 명세 없음) OIR만 정의됨
Q2	공간플레이스와 공간아크 이용	-	없음
Q3	원자적동기화 수준 및 객체동기화 수준	-	TELLES, MORENA, TSTN, HTSPN, X OCPN, PG와 동일
Q4	BTPN구조, BTPN언어 및 TD	-	구조만 존재함
Q5	분석모드와 재생모드	Grafcet (패트리넷기반 제어언어)	<ul style="list-style-type: none"> 명세만 함 Trellis에서도 도달성그래프를 제시했지만 동기화를 위한 것은 아님
Q6	트랜지션에 객체를 링크	-	<ul style="list-style-type: none"> 플레이스에 객체를 링크 IPN는 BTPN과 유사
Q7	베타분포형 지속시간	PERT/CPM에서의 가정	TSPN과 HTSPN에서는 3개의 모수 이용 (베타분포 아님)
Q8	BTPN의 분석모드수행, RTRG 및 TD생성	-	타이틀의 명세만 실시
Q9	재생모드 수행	Grafcet (패트리넷기반 제어언어)	없음
Q10	임계객체경로상의 객체의 편집	프로젝트관리와 PERT/CPM	[18]에서는 시간적 그래프를 사용

로 개념은 타이틀 저작 및 재생시의 불확실성을 모형화하며 재생시간의 조정도 가능하게 하고있다. 본 논문에서는 간략하게 다루었던 공간동기화 문제를 향후 연구과제로 남기고 있다.

참 고 문 헌

[1] R. Steinmetz, "Synchronization properties in multimedia systems", *IEEE Comm.*, Vol. 8, No. 3, April, pp. 401-412, 1990.

[2] G. Blakowski, R. Stinmetz, "A multimedia synchronization survey : reference model, specification, and case Studies", *IEEE selected area in Communications*, Vol. 14-1, pp. 5~35, Jan. 1996.

[3] J. Peterson, *Petri nets theory and the modeling of system*, Prentice-Hall, 1981.

[4] T. Little, A. Ghafoor, "Synchronization and storage models for multimedia objects", *IEEE Selected Area on Comm.*, Vol. 8, No. 3, pp. 414~427, 1990.

[5] M. Diaz, et al., "Time stream Petri nets: a model for timed multimedia information", *LNCS 815, APN'95*, pp.219~238, 1995.

[6] P. Senac, et al., "Hierarchical timed stream Petri net: a model for hypermedia systems", *LNCS 815, APN'95*, pp. 451~470, 1995.

[7] H. Wang, et al., "Interactive hypermedia titles: a model and its implementation", *Software Practice and Experience*, Vol.25(9), pp. 1045~1063, Sep. 1995.

[8] N. Qazi, M. Woo, A. Ghafoor, "A synchronization and communication model for distribution multimedia objects", *Proc. ACM Multimedia '93*, pp. 147~155, 1993.

[9] B. Prabhakaran, S. Raghavan, "Synchronization models for multimedia presentation with user participation", *Proc. ACM Multimedia '93*, pp. 157~166, 1993.

[10] P. Stott and R. Furata, "Petri net based hypertext: document structure with browsing semantics", *ACM Tran. on Information System*, Vol.7-1, pp. 3~29, Jan. 1989.

[11] R. Botafogo, et al., "The MORENA model for hypermedia authoring and browsing", *Proc. Int. Con. on Hypermedia Computing and System*, pp. 42~49, May 1995.

[12] Al-Salqan, C. Chang, "Temporal relations and synchronization agent", *IEEE Multimedia*, pp. 30~39, Summer 1996.

[13] D. Adjeroh, M. Lee, "Synchronization mechanisms for distributed multimedia presentation systems", *Int. Workshop on Multimedia DBMS*, pp. 30~37, Aug. 1995.

[14] C. Yang, J. Huang, "A multimedia synchronization model and its implementation in transport protocols", *IEEE selected area in Communications*, 14-1, pp. 212~225, Jan. 1996.

[15] Sang-shin Yoo, et al., "Transition function Petri net model for multimedia synchronization", *Journal of Electrical Engineering and Information Science*, Vol.1, No.4, pp. 1~7, 1996.

[16] R. David, "Grafcet: a powerful tool for specification of logic controllers", *IEEE tran. on Control Systems Technology*, Vol. 3-3, pp. 253~268, Sep. 1995.

[17] J. Allen, "Maintaining knowledge about temporal intervals", *Comm. ACM*. Vol. 26, No. 11, pp. 832~843, 1983

[18] M. Y. Kim, J. Song, "Multimedia documents with elastic time", *Proc. ACM Multimedia'95*, pp. 143~154, 1995.

[19] S. Kang and J. Nang, "A Multimedia Synchronization Specification Method for Temporal and Spatial Events", *Journal of KISS(A)*, Vol. 24, No.1, pp.1~12, 1997.1.

[20] I.K. Yeu and D.H. Hwang, "Design and Implementation of Multimedia Authoring System using Temporal/Spatial Synchronization Manager", *The Transactions of KIPS*, Vol.4, No.4, pp. 2679~2689, 1997.11.

[21] J.G. Yi, and G. Lee, Estimation of the Time to Ececute A Hypermedia Scenario Using Petri Net, Vol.5, No.5, pp. 1119~1129, 1998.5.

[22] F. Paulo, et al., "Hypercharts: extended statechart to support hypermedia specification", IEEE

tran. S.E., Vol.25, No.1, pp. 33~49, Jan./Feb. 1999.

부록 A. BTPN언어의 정의

```

BTPN ::= <Transition_definition> <Place_definition>
<Arc_definition>
<Transition_definition> ::= begin_TRANSITION
    <transition_list>
end_TRANSITION
<transition_list> ::= <s_transition_list> | <o_transition_list>
<s_transition_list> ::= <transition_number> : (<input_place_list>
=> <output_place_list>)
<o_transition_list> ::= <transition_number> <object_link> :
    (<input_place_list> => <output_place_list>)
<object_link> ::= <name>, <type>, <state>, <data_file>,
    <operation>, <b_duration>, <s_space>
<name> ::=  $\Sigma^+$  (i.e., positive closure of string)
<type> ::= <image_list> | <motion_pic_list> | <sound_list>
<image_list> ::= jpeg | bmp | tiff | gif
<motion_pic_list> ::= mpeg | avi | mov
<sound_list> ::= wav | midi
<state> ::= ready | run | pause | review | preview | block
<data_file> ::=  $\Sigma^+$ 
<operation> ::= <image_drive_list> |
    <motion_pic_drive_list> | <sound_drive_list>
<image_drive_list> ::= jpeg | bmp | tiff | gif
<motion_pic_drive_list> ::= mpeg | avi | mov
<sound_drive_list> ::= wav | midi
<b_duration> ::= beta(<R+, R+, R+>)
<s_space> ::= <upper_x_axis>, <upper_y_axis>,
    <lower_x_axis>, <lower_y_axis>
<upper_x_axis> ::= N, <upper_y_axis> ::= N
<lower_x_axis> ::= N, <lower_y_axis> ::= N
<input_place_list> ::= <input_place>, <input_place>
<input_place> ::= <place_number>
<output_place_list> ::= <output_place>, <output_place>
<output_place> ::= <place_number>
<place_number> ::= N, <transition_number> ::= N
<Place_definition> ::= begin_PLACE
    <place_list>
end_PLACE
<place_list> ::= <n_place_list> | <c_place_list> | <s_place_list>
<n_place_list> ::= <place_number> : <capacity>
<c_place_list> ::= <place_number> : <capacity>
<s_place_list> ::= <place_number>, <capacity> ::= N
<Arc_definition> ::= begin_ARC
    <arc_list> | <spatial_arc_list>
end_ARC
<arc_list> ::= <arc_number> : <coordination_list>
<coordination_list> ::= <coordinate_element>
<spatial_arc_list> ::= <arc_number> : <coordination_list>
<coordination_list> ::= <coordinate_element>
<coordinate_element> ::= <transition_list>, <place_list> |
    <place_list>, <transition_list>
    
```

부록 B. 분석모드 수행 알고리즘

Procedure Analysis_mode-firing(BTPN)

begin

```

L1: Get_node(BTPN, NODEα) (i.e., MSα and EFTα) from initial marking.
Set remaining time to all ti in EFTα (i.e., for each ti ∈ EFTα do δi = τi = μi = (ai + 4mi + bi)/6 enddo).
Select a CFTkα among EFTα.
If EFTα has unselected CFTα's, then push NODEα.
Get ftα which have minimum δ from CFTkα.
Fire ftα (i.e., For ∀ pk ∈ ftα • and ∀ pj ∈ • ftα, do #pk = #pk + 1 and #pj = #pj - 1 enddo).
Get_node(BTPN, NODEβ) (i.e., MSβ and EFTβ) from current marking.
If NODEβ is duplicated in RTRG or CETβ is empty, then
    Draw a labeled arc with ftα from NODEα to NODEβ
    Pop a NODEα (If stack is empty then stop.) and goto L1 endif
Set remaining time to all ti in EFTβ (i.e., for each ti ∈ EFTβ, if ti is a new firing transition
    then δi = τi = μi = (ai + 4mi + bi)/6, else δi = δi - δ of ftα endif )
Draw a labeled arc with ftα from NODEα to NODEβ.
α = β.
goto L1
end
    
```

Procedure Get_node(BTPN, NODE)

begin

```

Get marking MS.
Apply pre-choice rule, if necessary.
Get EFT by using the Distribution Raw of choice and parallel relations.
end
    
```

부록 C. 타이밍다이아그램, 임계객체경로 및 최장경로 생성 알고리즘

```

Procedure Generation_of_a_Timing_diagram(RTRG)
begin
  For all paths in RTRG do
    base = 0, i = 0
    repeat
      for new transition  $t_j \in CFT^i$  of  $NODE^i$  in a path do
        draw  $t_j$ 's line from base to  $base + \delta_j$  enddo
      base =  $base + \text{Min}\{\delta_j \text{ of } t_j \in CFT^i\}$ 
      i = i + 1
    until no more nodes in a path
  enddo
end

```

```

Procedure Finding_the_Critical_Object_Path(TD)
begin
  i = 1
  Get latest firing transition  $lt_i$  from a TD
  repeat
    Get latest transition  $lt_{i+1}$  which meets the transition  $lt_i$ 
    i = i + 1
  until  $lt_i$  is initial transition
   $\{t_{i-1}, t_{i-2}, \dots, t_2, t_1\}$  is the Critical object path in the TD.
end

```

```

Procedure Finding_the_longest_path(RTRG)
begin
  For all pathi in RTRG do
    tot = 0
    For each  $ft_j$  in pathi do tot =  $tot + \delta_j$  of  $ft_j$  enddo
    pathi = pathi  $\cup$  tot
  enddo
  Find pathk whose tot in path is maximum.
  pathk is the longest path.
end

```

부록 D. 재생 모드 수행 알고리즘

```

begin
  Repeat
    Get current concurrently firable transitions(CFT) from BTPN.
    If there is spatial non-overlap relation among transitions in CFT, then select one transition among
      the non-overlap transitions, and delete unselected transition(s) from CFT.
    For all  $t_i$  in CFT, do //firing//
      send "start" signals to  $O_i$ ,  $t_i.state = run$ ,  $t_i.timer = \tau_i = \mu_i = (a_i + 4m_i + b_i) / 6$ .
    enddo
    When  $t_i.state = run$  :  $t_i.timer$  is decreased
      If pause signal is received from user, then send "pause" signal to  $O_i$  and  $t_i.state = pause$ .
      If forward signal is received from user, then send "forward" signal to  $O_i$  and  $t_i.state = forward$ .
      If rewind signal is received from user, then send "rewind" signal to  $O_i$  and  $t_i.state = rewind$ .
      If skip signal is received from user or  $t_i.timer=0$ , then send "stop" signal to  $O_i$  and
        For  $\forall op_k \in t_i \bullet$  and  $\forall ip_j \in \bullet t_i$  do  $\#op_k = \#op_k + 1$  and  $\#ip_j = \#ip_j - 1$  enddo
    When  $t_i.state = pause$  :  $t_i.timer$  is blocking
      If end signal is received from user, then send "resume" signal to  $O_i$  and  $t_i.state = run$ .
    When  $t_i.state = forward$  :  $t_i.timer = t_i.timer - \text{time length of forward}$ 
      If end signal is received from user, then send "resume" signal to  $O_i$  and  $t_i.state = run$ .
    When  $t_i.state = rewind$  :  $t_i.timer = t_i.timer + \text{time length of rewind}$ 
      If end signal is received from user, then send "resume" signal to  $O_i$  and  $t_i.state = run$ .
  Until CFT =  $\emptyset$ 
end

```




이진석

1986년 국립대전산업대학 전자계산학 졸업
1990년 한남대학교 대학원 컴퓨터공학과 석사과정 졸업
1997년 한남대학교 대학원 컴퓨터공학과 박사과정 수료
관심분야 : 소프트웨어공학(시험), 패트리넷 응용, 정보보호시스템



이강수

1981년 홍익대학교 컴퓨터공학과 졸업
1983년 서울대학교 대학원 전산학과 석사과정 졸업
1989년 서울대학교 대학원 전산학과 박사과정 졸업
1985~1987년 국립대전산업대학교 전자계산학과 전임강사
1992~1993년 미국일리노이대학교 객원교수
1995년 한국전자통신연구원 초빙연구원
1987년~현재 한남대학교 컴퓨터공학과 전임강사~교수
1998~현재 한남대학교 멀티미디어학부장
관심분야 : 소프트웨어공학, 병행시스템 모델링 및 분석, 정보보호시스템 평가, 멀티미디어교육 커리큘럼