

인터넷상에서 페트리 넷을 이용한 하이퍼-프리젠테이션의 명시 표현 및 불일치성을 검증하기 위한 방법 연구

우 시 연[†] · 조 종 근^{††} · 임 영 환^{†††} · 임 기 옥^{††††}

요 약

하이퍼-프리젠테이션 개념은 하이퍼-미디어 개념을 확장한 것으로, 동적으로 연출되고 있는 상황에서 링크개념을 설정하여 특정시점이나 지점에서 원하는 다음 정보를 보기 위해서 비순차적으로 연출할 수 있도록 하는 연출이다. 이러한 비순차적인 연출방법인 하이퍼-프리젠테이션 개념을 도입함으로써, 자유로운 멀티미디어 연출이 가능하게 되었지만, 이러한 동기화 방법에 의해서 혼합된 여러 프리젠테이션들간에는 일관성 문제가 발생할 수 있다. 본 논문에서는 하이퍼-프리젠테이션 명시 프로그램의 일관성 검증을 위해서 Petri Net을 이용한 하이퍼-프리젠테이션 명시 방법을 제시하고, 그것을 이용한 명시 프로그램의 일관성을 검증하는 방법을 제안한다.

A Method for Describing a Hyper-Presentation Specification and Checking It's Inconsistency using Petri-Net in a Internet

Si-Yeon Woo[†] · Jong-Keun Cho^{††} · Younghwan Lim^{†††} · Kee-Wook Rim^{††††}

ABSTRACT

The concept of a hyperpresentation, as an extension of a hypermedia, is the presentation in which time-varying multimedia presentations are dynamically linked together and a hyperlink's context can be changed over time at any time during a continuous presentation. While this concept provides users with a flexible and powerful tool for their descriptions of the multimedia presentation, inconsistency problems in the synchronized presentations connected by hyper links can be introduced. This paper introduces a method of describing the hyperpresentation using Petri Net and proposes a methods of detecting inconsistent hyper link loop in the hyperpresentation description.

키워드 : 페트리 넷(Petri Net), 하이퍼 프리젠테이션(Hyper-Presentaion), 멀티미디어(Multimedia)

1. 서 론

기존의 멀티미디어 시스템은 시간적 동기화에 관심을 기울여 왔다.

멀티미디어 연출을 위한 시간적 동기화 명시 방법은 광역 시간(Global-Time)대에 미디어의 연출 시간을 명시하는 시간축 기반 명시방법(Time Axes-based Specification)과, 연출할 미디어간의 관계(relation)를 기술함으로써 시간적 우선순위를 명시하는 간격기반 명시방법(Interval-based Specification) 등이 있다. 이러한 기존의 동기화 명시 방법과 더불어 하이퍼-프리젠테이션을 위한 새로운 동기화 명시방법을 제안한다.

지금까지의 멀티미디어 시스템은 미디어 스트림의 순차적인 연출(presentation)을 위주로 하는 동기화 방법에 초점

을 맞추어 왔다. 이러한 순차적인 연출 시스템에 하이퍼-프리젠테이션의 개념을 도입함으로써 비순차적인 연출은 물론 사용자의 요구에 따른 다양한 표현을 제공할 수 있게 되었다. 하이퍼-프리젠테이션의 개념은 기존의 멀티미디어 연출에 하이퍼-미디어의 개념을 확장한 것이라고 보면 된다. 즉, 멀티미디어의 연출을 순차적으로만 하는 것이 아니라 하이퍼 링크를 설정하여 특정 지점에서 원하는 다음 정보를 볼 수 있도록 비순차적인 연출을 가능하게 하는 것이다.

하이퍼-프리젠테이션에서의 동기화 방법은 자유로운 연출을 가능하게 하지만, 이러한 동기화 방법에 의해 혼합된 여러 프리젠테이션들간에 일관성(consistency) 문제가 발생할 수 있으며, 그들간의 동기화 명시 방법을 Petri Net으로 모델링하여 하이퍼-프리젠테이션 명시 프로그램의 일관성을 검증할 수 있다.

명시 프로그램의 일관성 오류로 인하여, 하이퍼-프리젠테이션의 연출이 불가능한 경우나 무한반복 상태가 발생할 수 있다. 따라서, 명시 프로그램 상의 일관성을 연출 전에 검증하여 주어진 하이퍼-프리젠테이션의 연출가능성을 미

† 정 회 원 : (주)쌍용정보통신 교육센터 강사
†† 준 회 원 : 숭실대학교 대학원 컴퓨터학과
††† 종 신 회 원 : 숭실대학교 미디어학부 교수
†††† 종 신 회 원 : 선문대학교 공과대학 지식정보 산업공학전공 교수,
ETRI 컴퓨터 소프트웨어 연구소장
논문접수 : 2002년 7월 24일, 심사완료 : 2002년 9월 27일

리 판단할 수 있는 방법이 필요하다.

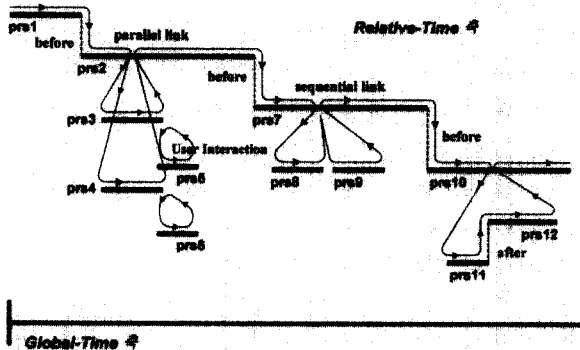
2. 하이퍼-프리젠테이션 모델

2.1 하이퍼-프리젠테이션의 예 : 뉴스 진행 시나리오

하이퍼-프리젠테이션의 간단한 예로 (그림 2)와 같은 뉴스 진행 예제를 들어 보겠다. 뉴스를 진행하는 앵커의 화면과 음성, 배경 음악으로 구성되는 prs 1, prs 2, prs 7, prs 10이 주 흐름이 되며, 진행 도중 사건에 대한 보도나 일기예보를 담은 프리젠테이션들이 순서에 따라 연결된다.

- prs 1 : 뉴스 앵커의 시작 멘트
- prs 2 : 의원선거 연설현장에 대한 설명과 prs 3, 4, 5, 6이 끝난 후 정리 멘트
- prs 3 : 후보 A의 연설 장면
- prs 4 : 후보 A에 대한 자세한 설명
- prs 5 : 후보 B의 연설 장면
- prs 6 : 후보 B에 대한 자세한 설명
- prs 7 : 다음 사건에 대한 보도와 prs 8, 9가 끝난 후 정리 멘트
- prs 8 : 리포터의 사건 설명
- prs 9 : 사건 정황 화면
- prs 10 : 내일의 날씨와 prs 11, 12가 끝난 후 뉴스의 마감 멘트
- prs 11 : 지역별 날씨
- prs 12 : 해상의 날씨와 한주간의 날씨

다음의 (그림 1)는 비디오 스트림과 오디오 스트림들로 구성되어 있는 하이퍼-프리젠테이션 연출이 가능하도록 작성한 명시 프로그램을 도식적으로 표현한 그림이다.



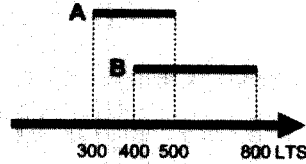
(그림 1) 뉴스 진행 예제

2.2 하이퍼-프리젠테이션을 위한 동기화 명시 방법

2.2.1 Global LTS(Logical Time System)

논리적 시간 체계(LTS)는 스트림들의 동기화(synchronization)를 위한 타이밍(timing) 방법으로 내부의 tick 간격(interval)을 조정하여 단위 시간을 결정할 수 있다.

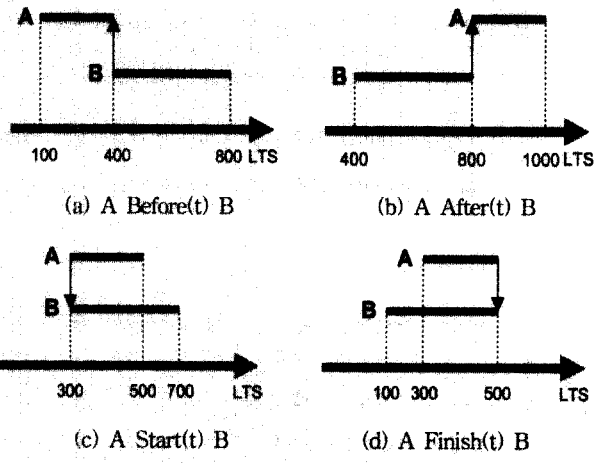
(그림 2)는 LTS 시간축에서 시작시간과 종료시간이 설정된 하이퍼-프리젠테이션이다. 프리젠테이션 A는 LTS 300에서 시작되며 500에서 종료된다. 또한 프리젠테이션 B는 LTS 400에서 시작되어 800에서 종료된다. LTS 400에서 500 사이의 시점에서 보면, 두 프리젠테이션 A와 B가 동시에 수행되나, 서로 독립적인 관계를 갖는다.



(그림 2) 시간축 기반 명시방법

2.2.2 큐(Cue)

기존의 간격기반 명시방법[7, 13, 15, 17]은 미디어들간의 시간적인 관계(temporal relationship)를 Before, Starts, Finishes, Overlaps, During, Meets, Equals, Before⁻¹, Starts⁻¹, Finishes⁻¹, Overlaps⁻¹, During⁻¹, Meets⁻¹ 등으로 정의하여 동기화를 맞추었으며, CUE 방법이 LST와 다른 점은 아래의 (그림 3)와 같이 두 프리젠테이션 A와 B의 시간적 관계를 명시하여 큐를 설정해 놓는다.



(그림 3) 큐를 이용한 동기화 명시 방법(단, t = 0)

2.2.3 하이퍼 링크(Hyper Link)

기존의 순차적인 멀티미디어 연출에 하이퍼-미디어의 개념을 확장해서 적용하면, 현재 수행되고 있는 프리젠테이션을 일시 중지하고 다른 프리젠테이션을 수행한 이후에 다시 원래의 프리젠테이션 연출을 계속하게 된다. 하이퍼 링크는 다음의 세 가지 구성 요소로 정의된다.

(1) 위치(Location)

임의의 프리젠테이션 위에 하이퍼 링크의 위치를 설정하기 위한 방법으로서, LTS 값으로 시간적 위치 "At"과 "During"이 제공된다.

(2) 조건(Condition)

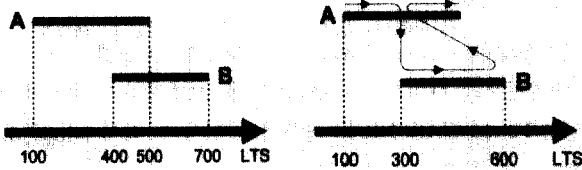
하이퍼 링크의 위치와 액션에 부여되는 조건으로서, "Active"와 "Inactive", "User_Selection"이 제공된다.

(3) 액션(Action)

하이퍼 링크가 수행하게 될 액션을 말하며, 프리젠테이션의 "Play" 또는 외부 프로그램의 "Execute" 등이 있다.

(그림 4)는 프리젠테이션 A의 수행 도중 프리젠테이션 B를 수행하도록 프리젠테이션 A에 하이퍼 링크를 설정한 그림이며, 다음과 같이 계산할 수 있다.

- $EndTime(A) = StartTime(A) + Duration(A) + Duration(B) = 800$
- $EndTime(B) = StartTime(B) + Duration(B) = 600$



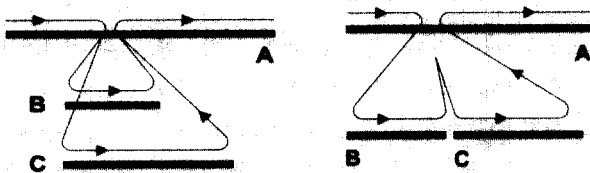
(a) 하이퍼 링크 설정 전 (b) 하이퍼 링크 설정 후
(그림 4) 하이퍼 링크를 이용한 동기화 명시 방법

2.3 중첩된 하이퍼 링크

프리젠테이션의 동일한 위치에 두 개 이상의 하이퍼 링크가 중첩된 경우, 두 프리젠테이션 B와 C 경우로 나누어 정의할 수 있다.

(그림 5(a))는 두 개의 프리젠테이션 B와 C를 하이퍼 링크의 시점에서 동시에 수행하는 경우이다. 이 때, B와 C의 수행을 마친 후 프리젠테이션 A를 계속 수행하기 위한 조건은 두 가지가 가능하다. 첫째는, 수행시간이 더 긴 프리젠테이션 C가 종료된 이후에 A를 계속해서 수행하는 방법이고, 둘째는 수행시간이 짧은 프리젠테이션 B가 종료되면 곧바로 A의 수행으로 넘어가는 방법이다. 이 두 가정 모두 가능하다고 본다.

(그림 5(b))는 프리젠테이션 B와 C를 차례로 수행하는 경우이다. 따라서 B와 C의 수행 순서가 중요한 조건이 된다.

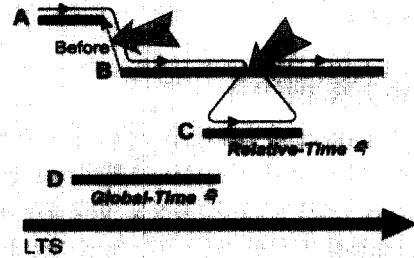


(a) 병렬적으로 처리 (b) 순차적으로 처리
(그림 5) 중첩된 하이퍼 링크

2.4 프리젠테이션들간의 동기화

하이퍼-프리젠테이션은 LTS와 큐, 그리고 하이퍼 링크

로 프리젠테이션들간의 동기화를 명시하여 표현할 수 있으며, 상대적인 시간(Relative-Time)축을 형성한다.

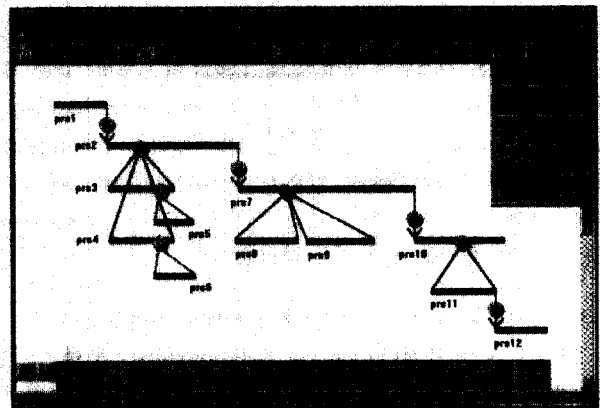


(그림 6) 하이퍼-프리젠테이션의 동기화 Time 축

(그림 6)을 보면, 프리젠테이션 A는 프리젠테이션 B에 "Before" 큐로 설정되어 있으므로 관계성에 묶이게 된다. 프리젠테이션 C 또한 프리젠테이션 B에 하이퍼 링크로 연결되어 있다. 그러므로 광역시간축을 기반으로 수행되는 프리젠테이션은 B와 D 뿐이다.

2.5 문제점 제시

(그림 7)은 VIP(Visual Interface Player)를 사용하여 뉴스 진행 시나리오를 작성한 하이퍼-프리젠테이션이다.



(그림 7) VIP로 작성한 뉴스 진행 시나리오

시나리오의 설정에 따라 프리젠테이션들이 동기화되어 연출되도록 작성하였으나, 경우에 따라서는 사용자가 의도했던 대로 연출되지 않을 수도 있다. 하이퍼-프리젠테이션 명시 프로그램을 작성하다 보면 실제로 연출이 불가능한 경우가 발생하게 된다. 이러한 문제는 명시 프로그램의 일관성 오류에서 비롯되는데, 즉 사용자가 프리젠테이션들의 동기화를 명시하였지만, 연출 의도에 없었던 사이클이 발생하여 프리젠테이션이 무한반복 되거나 아예 수행조차 되지 않는 경우가 발생할 수 있다. 그러므로, 연출 전에 사이클을 찾아내어 이를 해결함으로써 하이퍼-프리젠테이션의 일관성을 유지시켜야 한다.

본 논문에서는 이에 대한 해결 방안으로, 하이퍼-프리젠테이션 명시 프로그램을 Petri Net으로 모델링한 후 일관성 검증

을 위한 알고리즘을 제안하고, 하이퍼-프리젠테이션의 QoS에 영향을 미치는 Critical Path를 판단하는 알고리즘을 제안한다.

2.6. 관련 연구

2.6.1 Petri Net 모델

2.6.1.1 Petri Net의 정의

1962년 독일 Bonn 대학의 C. A. Petri의 박사 학위 논문에서 시작된 Petri Net은 시스템의 병행적(concurrent) 또는 병렬적(parallel) 프로세스들의 특성을 연구하기 위해 고안된 도식적, 수학적 모델링 도구이다. Petri Net 모델은 (P, T, F, B, μ_0)와 같은 튜플(tuple)로 정의되며, 각 속성(attribute)은 다음과 같다[3, 4].

- P는 place들의 유한집합이다.
 $P = \{ p_1, p_2, p_3, \dots, p_m \}, P \neq \emptyset$
- T는 transition들의 유한집합이다.
 $T = \{ t_1, t_2, t_3, \dots, t_n \}, T \neq \emptyset, P \cap T = \emptyset$
- F는 정방향영향함수(forward incidence function)이다.
- B는 역방향영향함수(backward incidence function)이다.
- μ_0 는 초기 마킹(initial marking)으로서 초기 상태를 나타낸다.

2.6.1.2 Petri Net 그래프

Petri Net은 시스템의 병행적 또는 병렬적 프로세스들을 표현하는 모델링 도구로 개발되었다. 프로세스는 조건(condition), 사건(events) 및 그들간의 관계를 서술하는 규칙(rule)으로 구성된다고 가정할 수 있으며, Petri Net 그래프에서는 이들 조건과 사건을 각각 place(원으로 표시)와 transition(선분 또는 박스로 표시)으로 나타낸다.

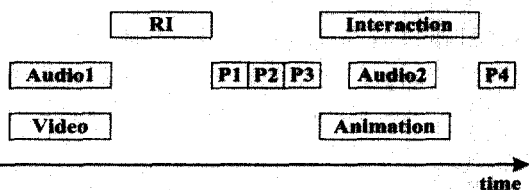
2.6.2 Petri Net을 이용한 동기화 명시 기법

2.6.2.1 시간 관계(Temporal Relationships) 모델

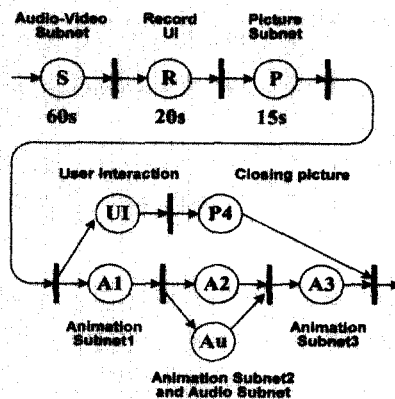
Allen[7]의 연구에서는 두 객체간의 시간 간격을 표현하는 7가지 관계로 분류하여 정의하였는데, 다음은 이러한 간격기반 명시방법에서의 시간적 관계를 Petri Net으로 표현한 그림이다[5, 6, 9, 13].

2.6.2.2 Petri Net을 이용한 UI 표현

아래의 (그림 8)는 시간축을 기반으로 미디어 데이터와 User Interaction을 표현한 동기화 예제로서, 이를 Petri Net으로 모델링한 것은 (그림 9)이다[2, 8, 13].



(그림 8) 동기화 예제



(그림 9) Petri Net을 이용한 동기화 명시 방법

3. Petri Net을 이용한 하이퍼-프리젠테이션 명시 프로그램의 모델링

3.1 Multidimensional Timed Petri Net(MTPN) 모델 정의
 다차원적인 하이퍼-프리젠테이션을 모델링하기 위하여 기존의 Timed Petri Net을 확장하여 다음과 같이 Multi-dimensional TPN을 정의한다.

[정의 1] Multi-dimensional Timed Petri Net

$$MTPN C = (P, T, F, B, \mu_0, \tau, \varepsilon)$$

- $P = \{ p_1, p_2, p_3, \dots, p_m \}, P \neq \emptyset$
- $T = \{ t_1, t_2, t_3, \dots, t_n \}, T \neq \emptyset, P \cap T = \emptyset$
- $F : P \times T \rightarrow N$ (단, N은 음수를 제외한 정수 집합)
- $B : P \times T \rightarrow N$
- $\mu_0 : P \rightarrow N$
- $\tau : T \rightarrow N$
- $\varepsilon : P \rightarrow N$

Petri Net 모델은 place의 유한집합 P, transition의 유한집합 T, 역방향 영향 함수 B, 정방향 영향 함수 F, 초기 마킹 μ_0 , 시간 함수 τ , 상태 함수 ε 과 같은 튜플로 구성된다.

transition $t_j \in T$ 가 주어진 마킹 μ 에서 모든 $p_i \in P$ 에 대해 $\mu(p_i) \geq F(p_i, t_j)$ 인 경우, t_j 를 마킹 μ 에 대해 점화가능한 transition이라 한다. 또한, 임의의 마킹 μ 에 대해 점화가능한 모든 transition들의 집합 $E(\mu)$ 을 마킹 μ 에 대한 점화가능 집합이라 부른다. 이 때, $t_j \in E(\mu) \Leftrightarrow \forall p_i \in P, \mu(p_i) \geq F(p_i, t_j)$ 가 성립한다. 현재 마킹이 μ 이고 transition $t_j \in T$ 가 점화가능한 transition일 때, 다음 마킹(next marking) μ' 는 다음과 같이 정의되며, 이 때, 마킹 μ' 는 마킹 μ 로부터 도달가능하다고 부른다.

$$(\forall p_i \in P) \mu'(p_i) = \mu(p_i) - B(p_i, t_j) + F(p_i, t_j)$$

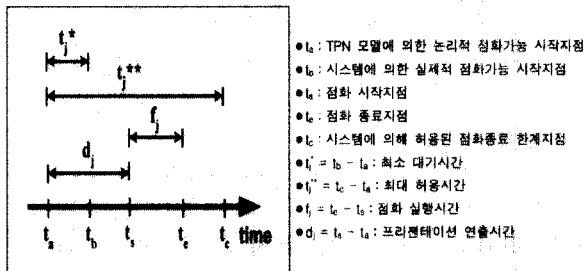
3.1.1 MTPN 모델의 시간 함수

시간 함수 τ 는 각 transition에 부여된 시간 조건을 표현하는 함수이다. 다음은 기존의 Timed Petri Net에서의 시간 조건에 대한 정의이다.

Carrier[14]은 Timed Petri Net의 시간 조건으로 점화실행시간(firing execution time)을 제안하였는데, 이는 점화시작시점(starting firing time)에서 점화종료시점(ending firing time)까지의 시간을 나타낸다. t_i 의 점화로 인한 input place로부터의 토큰 제거는 점화시작시점(t_s)에서 발생하고, output place로의 토큰 첨가는 점화종료시점($t_s + f_i$)에서 발생한다[10-12, 14].

본 논문에서는 transition의 실제 대기 시간을 MTPN 모델의 시간 조건으로 정의하고 (그림 10)은 프리젠테이션의 연출시간(duration)을 이에 적용한다.

$$\forall (p_i, t_j) \in (P \times T), \tau(t_j) = \text{Duration}(p_i) = d_j$$



(그림 10) Timed Petri Net 모델에서의 시간 조건

3.1.2 MTPN 모델의 상태 함수

Event는 프리젠테이션 연출과는 다른 시간영역(time domain)을 갖으므로 event의 발생 예측이 불가능하다. 그러므로 다차원적인 하이퍼-프리젠테이션을 모델링하기 위하여 다음과 같이 상태 함수 ε 를 정의한다. 이 때, R_0 는 0과 ∞ 를 포함하는 양의 실수 집합이다.

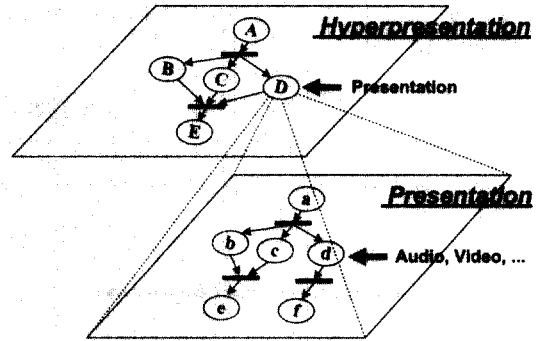
$$\varepsilon(p_i) = e \quad (p_i \in P, e \in R_0, 0 \leq e \leq \tau(t_j))$$

여기서 e는 event가 발생한 시간으로서, 프리젠테이션 연출의 전 구간에서 발생 가능하다. 따라서 상태 함수 ε 는 event에 의한 수행을 마친 후에 원래의 프리젠테이션을 계속 연출(resume)하기 위한 시점을 나타낸다.

3.2 MTPN을 이용한 동기화 명시 프로그램의 모델링

프리젠테이션 자체를 하나의 객체로 보고 그들간의 동기화를 명시함으로써, 프리젠테이션을 구성하는 미디어 스트림이나 그들간의 동기화에 영향을 받지 않고서도 하이퍼-프리젠테이션을 표현할 수 있다. 프리젠테이션을 place로 표현하면, 토큰을 포함하고 있는 place는 현재 수행중인 프리젠테이션으로 간주할 수 있다. 그러므로, 입력 place에서 출

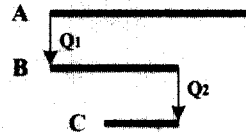
력 place로의 점화는 현재 수행중인 프리젠테이션의 종료와 더불어 다음 프리젠테이션의 수행을 의미하므로 (그림 11)처럼 하이퍼-프리젠테이션의 표현이 가능하다.



(그림 11) MTPN을 이용한 하이퍼-프리젠테이션의 모델링

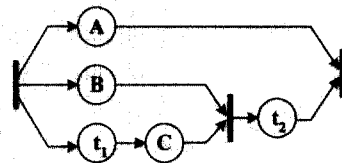
3.2.1 큐의 모델링

큐는 두 프리젠테이션간의 시간 관계를 명시함으로써 동기를 표현한다. 다음의 (그림 12)는 큐로 연결된 프리젠테이션들의 시간적 우선순위를 명시하여 하이퍼-프리젠테이션을 구성한 그림이다.



(그림 12) 큐로 연결한 프리젠테이션들간의 동기화 표현

다음은 (그림 12)와 같은 동기화 명시 방법을 MTPN으로 모델링한 (그림 13)이며, <표 1>은 큐의 관계에 의해 동기화되는 종속 프리젠테이션의 시작시간을 계산하기 위한 공식이다.



(그림 13) MTPN을 이용한 큐의 시간 관계 모델

<표 1> 큐의 관계에 따른 종속 프리젠테이션의 시작시간 계산공식

큐의 속성	종속 프리젠테이션의 시작시간
A Start(t) B	StartTime(B) = StartTime(A) + t
A Finish(t) B	StartTime(B) = EndTime(A) - Duration(B) - t
A Before(t) B	StartTime(A) = StartTime(B) - Duration(A) - t
A After(t) B	StartTime(A) = EndTime(B) + t

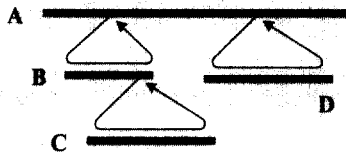
3.2.2 하이퍼 링크의 모델링

하이퍼-프리젠테이션은 다차원적인 멀티미디어 표현방식

을 제공한다. 프리젠테이션 위에 event point를 설정하기 위한 방법으로는

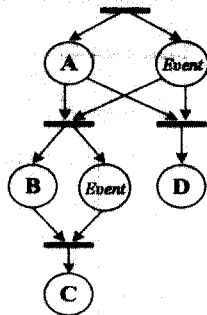
- LTS에 의한 시간값을 event point로 설정하는 경우
- 연속 미디어 상의 핫스팟(hot-spot)을 event point로 설정하는 경우

본 논문에서는 이러한 하이퍼 링크의 개념을 MTPN을 이용하여 모델링하는 방법을 제안하고자 한다. (그림 14)는 하이퍼 링크로 연결한 프리젠테이션들간의 동기화를 표현한 그림이다.



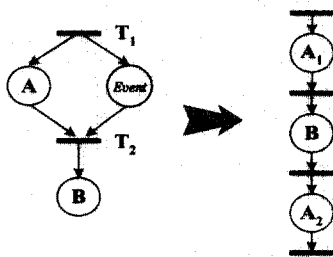
(그림 14) 하이퍼링크로 연결한 프리젠테이션들간의 동기화 표현

다음은 (그림 14)의 하이퍼-프리젠테이션을 MTPN으로 모델링한 (그림 15)이다. 하이퍼 링크의 표현을 위해 "Event"라는 place를 추가하여 모델링하였다.



(그림 15) MTPN을 이용한 하이퍼 링크의 모델링

Event point는 LTS에 의해 정적(static)으로 설정할 수 있으며, 또한 핫스팟에 의해 동적(dynamic)으로도 설정할 수도 있다. 물론 사용자에 의한 event(UI) 역시 연출 도중 동적으로 설정된 경우이다. 이러한 event들은 프리젠테이션 연출과는 다른 시간영역을 갖고 발생하기 때문에 시간 예측이 어렵다. 또한 event가 발생하면 그 시점에서 프리젠테이

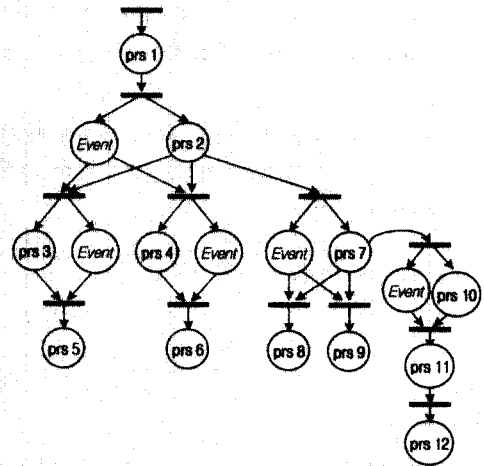


(그림 16) 동적 실행 모델

션 A를 일시 중지하고 프리젠테이션 B를 수행한 이후에 A의 연출을 계속해야 하므로 run-time에서 실행되는 동작 모델이 필요하다. (그림 16)은 이를 표현한 동작 모델로서, "Static Description Model"에서 "Dynamic Execution Model"로의 동적 실행 모델이다.

3.2.3 뉴스 진행 시나리오의 모델링

다음은 (그림 7)의 뉴스 진행 예제를 모델링한 (그림 17)이다.



(그림 17) 뉴스 진행 시나리오의 모델링

4. 하이퍼-프리젠테이션 명시 프로그램의 일관성 검증

4.1 동기화 명시 프로그램의 일관성 문제

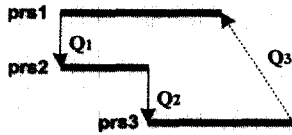
하이퍼 프리젠테이션들의 동기화를 명시하였지만, 연출 의도에 없었던 사이클이 발생하여 프리젠테이션이 무한 반복 되거나 아예 수행조차 되지 않는 경우가 발생할 수 있다. 그러므로, 연출 전에 사이클을 찾아내어 이를 해결함으로써 하이퍼-프리젠테이션의 일관성을 검증해야 한다. 본 논문에서는 이에 대한 해결 방안으로, 하이퍼-프리젠테이션 명시 프로그램을 MTPN으로 모델링하여 주어진 하이퍼-프리젠테이션의 연출가능성을 미리 판단하고, 또한, 멀티미디어 프리젠테이션의 특성 및 네트워크 문제로 인해 하이퍼-프리젠테이션의 동기화가 제대로 이루어지지 않거나 멀티미디어 연출의 QoS가 저하되는 경우도 발생하므로, QoS 요구사항이 명시된 하이퍼-프리젠테이션에서 Critical Path를 찾는 알고리즘을 제안하고자 한다.

4.2 동기화 명시 프로그램의 일관성 검증

4.2.1 MTPN을 이용한 동기화 명시 프로그램의 모델링

큐나 하이퍼 링크로 프리젠테이션간의 동기화를 명시할 경우 사이클이 발생하는 일관성 문제는, 이를 MTPN으로 표현한 모델에서도 마찬가지로 사이클이 발생하므로 MTPN 모델에서 사이클을 검사함으로써 명시 프로그램의 일관성을

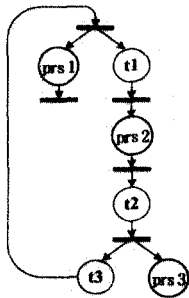
검증할 수 있다. 다음의 (그림 18)은 큐의 연결에서 사이클이 발생한 경우이다.



(그림 18) 큐 연결에서의 사이클

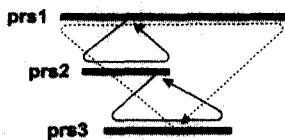
그림에서 보면, prs1과 prs3을 큐(Q3)로 연결할 경우 사이클이 발생한다. 큐의 관계(Finish)에 따라 prs1의 시작시간과 종료시간이 변경되는데, 다시 Q1의 연결에서 보면, 기존 프리젠테이션인 prs1의 시작시간이 변화하였으므로 prs2의 시작시간과 종료시간이 변경된다. 마찬가지로 Q2의 연결에서 보면, 기존 프리젠테이션인 prs2의 시작시간이 변화하였으므로 역시 prs3의 시작시간과 종료시간이 변경된다. 그러므로, Q3에 의해 prs1의 시간값이 또다시 변경되는 결과가 계속 반복된다.

다음은 (그림 18)의 하이퍼-프리젠테이션을 MTPN으로 모델링한 (그림 19)이며, 큐의 연결로 인해 프리젠테이션간에 사이클이 발생함을 쉽게 알 수 있다.



(그림 19) MTPN을 이용한 큐의 모델링

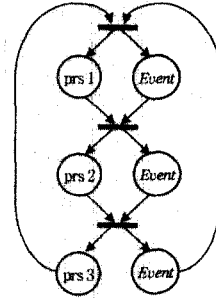
다음은 프리젠테이션에 설정한 하이퍼 링크로 인해 프리젠테이션들간에 사이클이 발생한 경우이다.



(그림 20) 하이퍼 링크 연결에서의 사이클

(그림 20)을 보면, prs3에 하이퍼 링크를 설정하여 액션으로 prs1을 수행하도록 할 경우 사이클이 발생한다.

MTPN으로 모델링한 (그림 19)에서 프리젠테이션간의 사이클이 발생하는 것을 알 수 있으며, (그림 21) 또한 MTPN으로 표현했을 때 사이클이 발생하는 것을 알 수 있다. 그러므로, 사이클이 발생하는 부분을 미리 찾아서 시스템에 심각한 문제를 초래하는 것을 해결할 수 있을 것이다.



(그림 21) MTPN을 이용한 하이퍼 링크의 모델링

4.2.2 MTPN 모델 상에서 사이클을 검출하기 위한 알고리즘

MTPN $C = (P, T, B, F, \mu_0, \tau, \epsilon)$ 의 정의에서 place의 집합 P는 다음과 같이 표현된다.

$$P = \{p_1, p_2, p_3, \dots, p_m\}$$

또한, place와 transition을 각각 노드 s로 표현하면, 후속 노드(successor)의 집합을 다음과 같이 정의할 수 있다.

$$s^o = \{v \in PUT, (s, v) \in (P \times T) \cup (T \times P)\}$$

s가 place인 경우 s^o 는 s의 output transition들의 집합이고, 반대로 s가 transition인 경우에는 s^o 가 output place들의 집합이다. 따라서, s^o 는 다음과 같이 표현된다.

$$s^o = \{\Gamma(s, 1), \Gamma(s, 2), \dots, \Gamma(s, |s^o|)\}$$

$$P_{i-1} = \{p_1, p_2, \dots, p_{i-1}\}$$

여기서, $\Gamma(s, i)$ 는 s의 i번째 후속 노드이다. 이와 같은 표현을 근거로 MTPN 모델 상에서 사이클을 검출하기 위한 알고리즘을 제안한다.

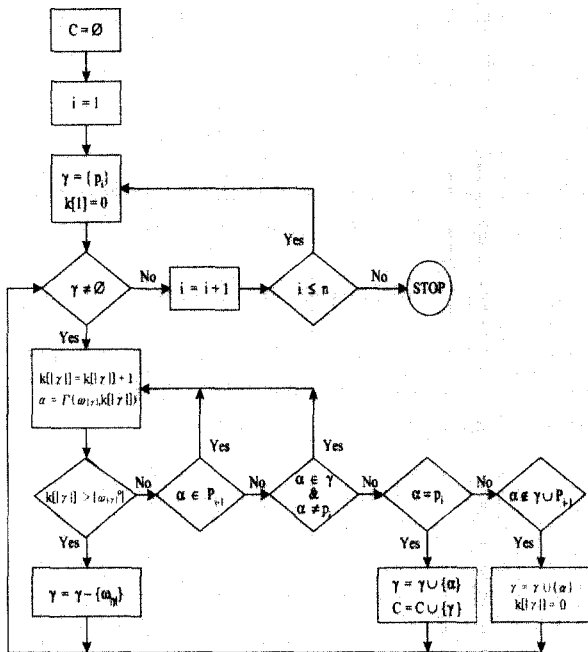
1. [Initialize the set of circuits] $C = \emptyset$
2. [For $i = 1$ to n , compute the set of circuits that contain place p_i but do not contain any place belonging to $P_{i-1} = \{p_1, p_2, \dots, p_{i-1}\}$ For this purpose, consider the set of elementary paths starting from p_i . Let ω_j be the j -th node of v , i.e. $v = \{\omega_1, \omega_2, \dots, \omega_{|v|}\}$ and let ω_{j+1} be the $k[j]$ -th successor of node ω_j , i.e. $\omega_{j+1} = \Gamma(\omega_j, k[j])$.
 - 2.1 Initialize: $v = \{p_i\}$, $k[1] = 1$
 - 2.2 while $v \neq \emptyset$, do
 - 2.2.1 $k[|v|] = k[|v|] + 1$
 - 2.2.2 [Consider the $k[|v|]$ -th successor α of node $\omega_{|v|}$, i.e. $\alpha = \Gamma(\omega_{|v|}, k[|v|])$. We distinguish the five following cases:]
 - Case 1: $k[|v|] > |\omega_{|v|}|$
[all successors of $\omega_{|v|}$ have been examined.
Then remove $\omega_{|v|}$.]
 $v = v - \{\omega_{|v|}\}$
 - Case 2: $\alpha \in P_{i-1}$
[α is one of places p_1, p_2, \dots, p_{i-1} . Then proceed to the next successor of $\omega_{|v|}$.]

Case 3 : $a \in \gamma$ and $a \neq p_i$
 [a is already exists in path γ and it is not the origin of γ . Then proceed to the next successor of ω_{i+1} .]

Case 4 : $a = p_i$
 [γ is a new circuit, add $\forall U(a)$ to C , and proceed to the next successor of ω_{i+1} .]
 $C = C \cup \{\gamma\}$

Case 5 : $a \notin \gamma \cup P_{i-1}$
 [a is a new node.]
 $\gamma = \gamma \cup \{a\}$, $k[i+1] =$

(그림 22)은 위의 사이클 검출 알고리즘을 순서도로 표현한 그림이다.



(그림 22) 순서도

MTPN의 모델상에서 사이클을 검출하기 위해서는 처음으로 실행되는 place의 집합 P를 프리젠테이션간의 시간함수 τ 로 할당하고, 이 값을 시간함수 τ 가 하나씩 이동할 때마다 시간함수 τ 의 집합원소인 ω_j 즉, ω_j 와 $k[j]$ 를 비교한 결과값을 a 에 넣는다. 그리고, 위에서 제한한 사이클 검출 알고리즘은 $\omega_{j+1} = \Gamma(\omega_j, k[j])$ 의 수식에서 보는것처럼 Γ (델타) 즉, 5가지의 경우로 적용함으로써 사이클을 검출할 수 있다. 또한, 사이클을 검출한 후에는 다시 시간함수를 하나 증가시킴으로써 다른 사이클을 검출한다.

그리고, 사이클을 검출하기 위한 조건은 $C = \emptyset$ 이 아닌 경우이다.

4.3 하이퍼-프리젠테이션에서의 Critical Path

대부분 주어진 네트워크 전송에 대한 지연시간 상한을 알 수 없기 때문에 통계적인 방법을 이용하여 가장 현실적인

시간을 지연시간에 대한 대표값으로 설정하고, 이를 근거로 모든 패킷들에 대한 전송 스케줄을 작성하는 방법을 사용하고 있다. 그러므로, Critical Path를 판단하여 멀티미디어 연출의 품질을 보장할 수 있어야 한다.

4.3.1 Critical Path 판단 알고리즘

Petri Net을 이용한 하이퍼-프리젠테이션 명시 프로그램의 모델에서 각 place, 즉 프리젠테이션에 대한 사용자의 QoS 요구수준을 정형화된 값으로 받아 이를 근거로 하이퍼-프리젠테이션에서의 Critical Path를 판단하고자 한다. 다음은 Critical Path 판단 알고리즘과 이에 사용된 용어에 대한 정의이다.

- S : start place
- E : end place
- QoS(p) : place p의 QoS
- PP(p) : p의 previous place 집합
- N(p) : PP(p)의 원소 개수
- A(p) : $\max\{A(PARENT(p)) + QoS(p)\}$

1. [This step initializes a variable N(p) to id p for every place $p \neq S$. The variable N(p) will be used to count the number of places adjacent to p that have not yet been labeled.]
 For every place $p \neq S$, let $N(p) =$ the number of PP(p).
2. [A queue Q is initialized]
 $Q = \emptyset$.
3. [This step labels S and updates N(p) for all places p adjacent from S. Further, those places p for which $N(p) = 0$ are added to Q, which consists of places that are ready to be labeled.]
 3.1 $A(S) = 0$.
 3.2 For every place p such that $S \in PP(p)$,
 let $N(p) = N(p) - 1$, PARENT(p) = S and add p to Q.
4. [This step deletes a place q from Q, labels q, determines PARENT(q), updates N(p) for all places p adjacent from q, and adds to Q those places p for which $N(p) = 0$.]
 let $N(p) = N(p) - 1$.
 If $N(p) = 0$, then add p to Q.
5. [This step determines whether T has been labeled.]
 If T has been labeled, then continue ; Otherwise, return to Step 4.
6. [This step finds a critical path in PN.]
 6.1 CP : T = p_0 .
 6.2 $k = 0$.
 6.3 [Suppose that
 CP : $p_k, p_{k-1}, \dots, p_1, p_0$ has been determined, where
 $p_i = PARENT(p_{i-1})$ for $1 \leq i \leq k$.
 Then this step determines whether CP is already a critical path. If we have not yet found a critical path, then this step extends P.]
 If $p_k = S$, then output CP and A(T), and stop.
 Otherwise, let $p_{k+1} = PARENT(p_k)$
 and CP : $p_{k+1}, p_k, \dots, p_1, p_0$.
 6.4 $k = k + 1$ and return to Step 6.3.

Critical Path란 자원의 소비가 가장 심하여 병목(bottlen-

ect)이 유발될 가능성이 높은 경로를 말한다. 일반적으로 설정된 지연 시간값보다 더 큰 지연시간으로 전송된 패킷들은 요구된 시점에 프리젠테이션 될 수 없기 때문에 손실된 패킷으로 처리되고, 손실된 패킷이 많을수록 프리젠테이션 품질이 저하된다. 따라서, 손실된 패킷을 가능한 한 최소화 할 수 있는 전송 스케줄을 작성하는 것이 매우 중요하다.

위에 보이는 Critical Path 알고리즘은 먼저 place의 집합 p중에서 previous place의 p들에 먼저 id를 부여하고, 실행하는 place(S)에 label을 주고 이 근처에 있는 Previous place의 p값들을 다시 변경한다. 변경한 Previous place의 p 값이 S의 원소이면 하나씩 감소해 나가면서, 이렇게 해서 큐(Queue)에 들어 있는 값들을 모두 처리하면서 Path경로 들을 결정하고, 이 Path들중에 Critical Path를 찾는다.

6. 결 론

본 논문에서는 하이퍼-프리젠테이션 명시 프로그램을 Petri Net으로 모델링하여 주어진 하이퍼-프리젠테이션의 연출 가능성을 미리 판단하고자 하였으며, 하이퍼-프리젠테이션의 연출가능성을 판단하기 위하여 MTPN을 정의하였고, 이 모델을 이용하여 동기화 명시 프로그램을 모델링하였다. 또한 하이퍼-프리젠테이션의 MTPN 모델에서 사이클을 검출하는 알고리즘을 제안하여 동기화 명시 프로그램의 일관성을 검증할 수 있도록 하였다. 또한, 사용자의 QoS 요구와 실제 resource의 능력을 고려한 하이퍼-프리젠테이션의 연출가능성을 판단하기 위하여 MTPN 모델에서 Critical Path를 판단하는 알고리즘을 제안하였다.

참 고 문 헌

[1] 임영환, "ComBiStation : 분산 멀티미디어 컴퓨팅 환경을 위한 컴퓨팅 플랫폼", 정보과학회논문지, 제2권 제1호, pp.6-8.
 [2] Patrick Senac, Michel Diaz, Alain Leger and Pierre de Saque-Sannes, "Modeling Logical and Temporal Synchronization in Hypermedia Systems," IEEE Journal on Selected Areas in Communications, Vol.14, No.1, pp.94-103, January, 1996.
 [3] M. Ajmone Marsan, G. Balbo and G. Conte, "Modeling with Generalized Stochastic Petri Nets," WILLY, 1993.
 [4] T. Murata, "Petri Nets : Properties, Analysis and Applications," Proceedings of the IEEE, Vol.77, No.4, pp.541-576, April, 1989.
 [5] Thomas D. C. Little and Arif Ghafoor, "Interval-Based Conceptual Models for Time-Dependent Multimedia Data," IEEE Transactions on Knowledge and Data Engineering, Vol.5, No.4, pp.551-563, August, 1993.
 [6] Ralf Steinmetz, "Synchronization Properties in Multimedia

Systems," IEEE Journal on Selected Areas in Communications, Vol.8, No.3, pp.401-412, April, 1990.

[7] J. F. Allen, "Maintaining Knowledge About Temporal Intervals," CACM, Vol.26, No.11, pp.832-843, Nov., 1983.
 [8] S. V. Raghavan and B. Prabhakaran, "Synchronization Models for Multimedia Presentation with User Participation," ACM/Springer-Verlag Multimedia Systems, Vol.2, No.2, pp.53-62, August, 1994.
 [9] M. Hack, "Decidability Questions for Petri Nets," Ph.D. dissertation, MIT, Massachusetts, Dec., 1975.
 [10] 손진곤, 황종선, 백두권, "동적 표현능력을 갖는 Timed Petri Net 형식론", 정보과학회논문지, 제19권 제1호, pp.34-47, January, 1992.
 [11] Stephen S. Yau and Mehmet U. Caglayan, "Distributed Software System Design Representation Using Modified Petri Nets," IEEE Transactions on Software Engineering, Vol. SE-9, No.6, pp.733-745, November, 1983.
 [12] Bernard Berthomieu and Michil Diaz, "Modeling and Verification of Time Dependent Systems Using Time Petri Nets," IEEE Transactions on Software Engineering, Vol.17, No.3, pp.259-273, March, 1991.
 [13] G. Blakowski and R. Steinmetz, "A Media Synchronization Survey : Model, Specification, and Case Studies," IEEE Journal on Selected Areas in Communications, Vol.14, No.1, pp.5-35, January, 1996.
 [14] J. Carlier, P. Chretienne and C. Girault, "Modelling Scheduling Problems with Timed Petri Nets," in Advanced in Petri Nets 1984, Lecture Notes in Computer Science, Springer-Verlag Pub. Co., pp.62-82, 1985.



우 시 연

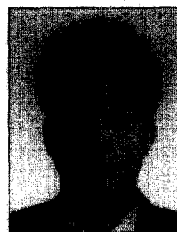
e-mail : silky@mildsky.com

1995년 숭실대학교 전자계산학과 졸업 (학사)

1999년 숭실대학교 대학원 전자계산학과 졸업(석사)

1999년~현재 (주)쌍용정보통신 교육센터 강사

관심분야 : 멀티미디어 스트리밍, 사용자 인터페이스



조 종 근

e-mail : jkdang@empal.com

1998년 성결대학교 컴퓨터공학과 졸업 (학사)

2001년 숭실대학교 대학원 컴퓨터학과 졸업(석사)

2001년~현재 숭실대학교 대학원 컴퓨터학과 박사 재학중

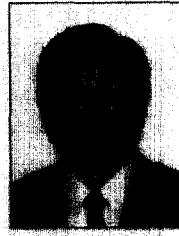
관심분야 : 멀티미디어 분산 스트리밍, 컴퓨터 그래픽스



임영환

e-mail : yhlhm@computing.soongsil.ac.kr
1977년 경북대학교 수학과 졸업(학사)
1979년 한국과학원 전산학과 졸업(석사)
1985년 Northwestern University 전산학과
(박사)
1979년~1996년 한국전자통신연구소 책임
연구원

1996년~현재 숭실대학교 미디어학부 부교수
관심분야 : 멀티미디어



임기욱

e-mail : nim@etri.re.kr
1969년~1977년 인하대학교 공과대학 전자
공학과 졸업
1984년~1987년 한양대학교 전자계산학
석사
1990년~1994년 인하대학교 전자계산학
박사

1977년~1983년 한국전자기술연구소 선임연구원
1983년~1988년 한국전자통신연구소 시스템소프트웨어 연구실장
1988년~1989년 미 캘리포니아 주립대학(Irvine) 방문연구원
1989년~1997년 한국전자통신연구원 시스템연구부장
주전산기(타이컴) III, IV 개발 사업책임자
1997년~1999년 정보통신부 정보통신 연구진흥원 정보기술전문
위원
2000년~현재 선문대학교 공과대학 지식정보산업공학전공 교수
2001년~현재 ETRI 컴퓨터 소프트웨어 연구소장
관심분야 : 실시간데이터베이스 시스템, 시스템 아키텍처, 운영
체제