시간 페트리네트를 DEVS 형식론으로 변환하는 알고리즘*

김영찬**, 김탁곤***

Algorithm for Transformation of Timed Petri Nets to DEVS Formalism

Young Chan KIM, Tag Gon KIM

Abstract

Petri nets is a widely used formalism for specification and analysis of concurrent systems which is a subclass of discrete event systems. The DEVS (Discrete Event System Specification) formalism provides a general framework for specification of discrete event systems in a hierarchical, modular form. Often, modeling a discrete event system may employ both Petri Nets and DEVS formalism. In such a case low-level operational logics are modeled by Petri Nets and high-level managements by the DEVS formalism. Analysis of the system requires simulation of the overall system. This paper presents an algorithm for transformation of Petri Nets to DEVS formalism. The transformation enables modelers to simulate an overall system, which consists of DEVS models and Petri Nets models, in a unified DEVS simulation environment such as DEVSim++. An example for such transformation will be given.

Key Words: Timed Petri Nets, DEVS, Formalism, Formalism Transformation

* 이 논문은 1998년도 대전산업대학교 교내학술연구비 지원을 일부 받았음.

** 한밭대학교 정보통신-컴퓨터공학부

*** 한국과학기술원 전자전산학과

1. 서론

시간 페트리네트(Timed Petri Nets) [1-5]는 동시 발생적인 (concurrent) 시스템을 명세하고 성능을 평가하기 위해 사용되는 중요한 형식론중의 하나인 반면에, DEVS 형식론은 이산 사건으로 표현할 수 있는 모델을 명세할 수 있는 수학적 토대를 제공한다.

DEVS 형식론[6,7]은 이산사건 모델을 명세하기 위해 Zeigler에 의해 소개된 형식론이다. 이 형식론은 모델과 시뮬레이션 프로그램 개발, 고급 시뮬레이션 언어 설계, 시뮬레이션 프로그램 검증 등에 대한 토대로도 활용된다. 집합 이론적 형식론인 DEVS는 계층적이고 모듈적인 형태로 모델을 명세할 수 있다. 한 개의 이산사건 시스템을 모델링 할 경우, 시간 페트리네트와 DEVS 형식론을 모두 사용할 경우가 종종 생길 수 있다. 이 경우, 시간 페트리네트는 시스템의 하위레벨의 동작논리를 기술하기에 적합하고, 반면에 DEVS 형식론은 상위레벨의 관리 규칙을 명세하기에 적합하다. 이렇게 두개의 모델링 방법으로 한 개의 시스템을 모델링 했을 경우, 전체의 시스템을 시뮬레이션하는 것이 어렵게 된다. 한 가지 해결책으로 한 개의 형식론을 다른 형식론으로 변환하는 것이 제시될 수 있다.

이 논문에서는 시간 페트리네트를 DEVS 모델로 변환하는 알고리즘을 제시한다. 이는 학술적인 의미 외에도 DEVSim++[8]와 같은 DEVS 형식론이 지원하는 다양한 환경 및 도구를 이용할 수 있는 장점이 있다.

이 논문의 구조는 다음과 같다. 2절에서는 이 논문을 이해하기 위해 필요한 기본적인 정의와 결과들을 설명한다. 3절에서는 시간 페트리네트를 DEVS 모델로 변환하는 알고리즘을 제시한다. 4절에서는 예제를 통해 알고리즘의 동작을 보여준다. 5절에서 이 논문의 결론과 연구방향을 제시한다.

2. 관련 연구

본 논문에서 N은 음수가 아닌 정수를, R은 음수가 아닌 실수를, n는 페트리네트의 플레이스의 개수를($n=|P|$), m는 트랜지션의 개수를($m=|T|$), $e[j]$를 j번째만 1이고 나머지는 0인 m-벡터를 나타내기 위해서 사용된다. 또한 원소 a가 2개 원소 b가 1인 가방(bag)을 $\{a:2, b:1\}$로 표현한다.

2.1 시간 페트리네트 (Timed Petri Nets)

페트리네트 이론은 동시 발생적인 시스템을 명세하기 위해 사용되는 잘 알려진 형식론중의 하나이다. 이 절에서 페트리네트 이론의 기본적인 개념을 간단히 살펴본다. 자세한 내용은 [1-4]을 참조하라.

페트리네트

이 논문에서는 [1]에서 정의된 페트리네트 형식론에 약간 수정한 것을 채용한다.

페트리네트는 다음의 구조를 갖는다.

$$PN=\langle P, T, I, O\rangle$$

P 플레이스(place)의 집합
T 트랜지션(transition)의 집합
$I: T\rightarrow P^{\infty}$ 입력함수
$O: T\rightarrow P^{\infty}$ 출력함수

I와 O는 트랜지션 T로부터 장소의 가방(bag 혹은 multiset) P^{∞}로의 매핑(mapping)이다. 입력 I와 출력함수 O는 필요에 따라서 두 개의 동등한 표현인 행렬 D^{-}와 D^{+}로 각각 사용될 수도 있다. 입출력 행렬과 입출력 함수는 다음과 같이 연관되어 진다.

$$D^{-}[j,i] = \#(p_i, I(t_j))$$
$$D^{+}[j,i] = \#(p_i, O(t_j))$$

여기서 $\#(p_i, I(t_j))$는 p_i로부터 t_j로의 입력 모서리(arc) 개수이고, $\#(p_i, O(t_j))$는 t_j로부터

p_i로의 출력 모서리 개수이다. 페트리네트의 행렬형태는 다음의 구조를 갖는다:

$$PN=\langle P, T, D^-, D^+\rangle$$

마킹(marking) $\mu: P\rightarrow N$은 플레이스로부터 자연수로의 매핑이다. 마킹은 플레이스에 토큰(token)의 할당을 의미한다. 일반적으로 μ는 n-벡터로 표현된다.

$$\mu=(\mu_1, \mu_2, \cdots, \mu_n)$$

여기서, $\mu_i=\mu(p_i)$는 플레이스 p_i에 있는 토큰의 개수를 나타낸다. 마크된 페트리네트(marked Petri net)는 페트리네트 PN과 마킹 μ의 튜플 즉 (PN, μ)로 정의된다.

페트리네트의 실행은 그 페트리네트가 가지는 토큰의 수와 분포에 의해 제어된다. 페트리네트는 트랜지션을 fire하여 실행된다. 트랜지션의 fire는 그 트랜지션의 입력 플레이스에서 토큰을 제거하고 출력 플레이스에 토큰을 추가하는 과정을 거친다. 임의의 트랜지션 t_j가 fire할 수 있으려면 t_j는 반드시 enable되어 있어야 한다. t_j는 다음의 조건을 만족할 때 enable되었다고 정의된다:

$$\mu(p_i) \geq \#(p_i, I(t_j)) \quad \text{for all } p_i \in P$$

행렬식으로 표현한 동등한 조건은 다음과 같이 표현된다:

$$\mu \geq e[j] \cdot D^-$$

페트리네트의 상태공간

마크된 페트리네트 (PN, μ)이 주어졌을 때, 그 네트의 상태는 마킹함수 μ에 의해 정의된다. n-플레이스 페트리네트의 상태공간은 모든 가능한 마킹의 집합이다. 상태 변화는 트랜지션의 firing의 의해 기술된다. 상태전이함수 δ는 다음과 같이 정의된다: 만약 $\delta(\mu, t_j)=\mu'$이라면,

$$\mu'(p_i)=\mu(p_i)-\#(p_i, I(t_j))+\#(p_i, O(t_j)) \quad \text{for each } p_i \in P$$

행렬식으로 표현한 상태전이함수는 다음과 같이 표현된다:

$$TPN_1=(P, T, I, O, \mu, \theta)$$

$P = \{p_1, p_2, p_3, p_4, p_5, p_6\}$
$T = \{t_1, t_2, t_3, t_4, t_5\}$
$\mu = (2,1,2,0,1,0)$
$\theta = (1,3,2,0,1)$

$I(t_1) = \{p_2:1, p_3:1, p_6:1\}$	$O(t_1) = \{p_1:1\}$
$I(t_2) = \{p_6:1\}$	$O(t_2) = \{p_5:1\}$
$I(t_3) = \{p_2:1, p_3:1, p_5:1\}$	$O(t_3) = \{p_4:2\}$
$I(t_4) = \{p_4:1\}$	$O(t_4) = \{p_3:1\}$
$I(t_5) = \{p_1:2\}$	$O(t_5) = \{p_6:1\}$

<그림 1> 시간 페트리네트: TPN_1

$$\mu' = \mu - e[j] \cdot D^- + e[j] \cdot D^+$$

초기 마킹 μ로부터 하나 혹은 여러 개의 트랜지션을 fire하여 도달할 수 있는(reachable) 모든 마킹 집합을 $R(\mu)$로 나타낸다.

시간 페트리네트 (Timed Petri Nets)

시간 페트리네트는 다음과 같이 정의된다.

$$TPN=\langle PN, \mu, \theta\rangle$$

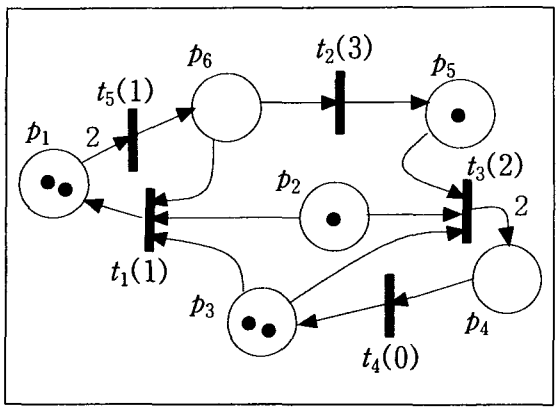
여기서 $\langle PN, \mu\rangle$는 마크된 페트리네트이고 θ는 시간구조(clock structure)이다. 시간구조 θ는 트랜지션의 집합과 연관된다. 다른 말로 표현하면, θ는 트랜지션의 집합으로부터 음수가 아닌 실수로의 매핑이다 ($\theta: T\rightarrow R$).

$$\theta(t_j)=\theta_j$$

여기서 θ_j는 트랜지션 t_j가 가지는 지연시간이다. 시간 페트리네트에서 θ_j는 다음의 의미를 가진다.

트랜지션 t_j가 enable될 때, 그것은 즉시 fire하지 않고 θ_j에 의해 주어진 지연시간만큼 지연된다. 이 지연동안 토큰은 t_j의 입력 플레이스에 남아 있는다.

일반적으로 θ는 m-벡터로 표현된다.



<그림 2> TPN_1의 그림형태

$$\theta=(\theta_1, \theta_2, \cdots, \theta_n)$$

<그림 1>는 시간 페트리네트 TPN_1의 명세 예를 보여주고 있다.

<그림 2>는 <그림 1>과 동등한 명세방법인 그림 형태로 표현된 TPN_1을 보여주고 있다. 이 시간 페트리네트는 6개의 플레이스와 4개의 트랜지션을 가진다. 트랜지션 위의 레이블 $t_j(d)$는 트랜지션 t_j가 $\theta_j=d$을 가지는 것을 나타낸다. 예를 들면, $t_5(1)$는 트랜지션 t_5는 지연시간 θ_5으로 1을 가진다는 것을 나타낸다. 모서리 (u, v) 위의 정수 k는 그 모서리의 무게(weight) 즉 u에서 v로 k개의 병렬 모서리가 존재하는 것을 나타낸다. 예를 들면, (p_1, t_5)는 무게로 2를 가지는 것은 p_1에서 t_5로 두 개의 모서리가 있는 것을 나타낸다.

시간 페트리네트의 상태공간

시간 페트리네트의 역학(dynamics)은 비 시간적 페트리네트의 역학보다 좀 더 복잡하다. 시간 페트리네트의 상태는 마킹함수 μ와 시간 구조 θ로 구성된다. $n=|P|$이고 $m=|T|$이라고 할 때, 상태공간은 $N^n\times R^m$이 된다. 상태전이함수 $\delta: N^n\times R^m\times T\rightarrow N^n\times R^m$는 다음과 같이 정의된다: $\delta((\mu, \theta), t_j)=(\mu', \theta')$이라고 가정하면,

$$\mu'=\mu-e[j]\cdot D^-+e[j]\cdot D^+$$
$$\theta'(t_k)=\theta(t_k)-\theta(t_j) \quad \text{for each } t_k\in T$$

2.2 DEVS 형식론

DEVS 형식론은 이산 사건 시스템을 계층적이고 모듈화된 형태로 표현하는 수학적인 도구이다. DEVS 형식론으로 시스템을 표현하기 위해서는 시스템을 atomic DEVS 모델과 coupled DEVS 모델로 구성하여야 한다. Atomic DEVS 모델은 다음과 같이 표현된다.

$$M=\langle X, S, Y, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, ta\rangle$$

여기서,

X	입력사건들의 집합
S	일련의 상태들의 집합
Y	출력 사건들의 집합
$\delta_{\text{int}}: S\rightarrow S$	내부 (상태)전이 함수
$\delta_{ext}: Q\times X\rightarrow S$	외부 (상태)전이 함수
$\lambda: S\rightarrow Y$	출력함수
$ta: S\rightarrow R$	시간진행함수

여기서, R는 음수가 아닌 실수이고, Q는 M의 전체 상태를 나타내는 것으로서 다음과 같이 주어진다.

$$Q=\{(s, e)\,|\,s\in S,\ 0\le e\le ta(s)\}$$

다음의 형태는 coupled DEVS 모델로 구성요소가 되는 몇 개의 모델들이 결합되어 새로운 모델을 만드는 것을 표현한다. 그런데 이 coupled DEVS 모델은 그 자체가 다시 보다 큰 모델의 구성요소가 될 수 있다. 이것을 이용하여 복잡한 모델을 계층적으로 구성할 수 있게 된다. coupled DEVS 모델은 다음과 같이 정의된다.

$$DN=\langle D, \{M_i\}, \{I_i\}, \{Z_{i,j}\}, \text{select}\rangle$$

여기서,

D 구성요소들의 이름의 집합

D의 각 원소 i에 대하여,

M_i D의 i번째 구성요소의 DEVS

I_i i번째 모델의 influencee DEVS들의 집합

I_i의 각 원소 j에 대하여,

$Z_{i,j}$: $Y_i \rightarrow X_j$ D의 i번째 DEVS의 출력을 j번째 DEVS의 입력으로 연결하는 함수

select: $2^D \rightarrow D$ 여러 구성요소 DEVS들이 같은 시간에 스케줄을 원할 때 그것들을 순서화시키는 함수

Atomic DEVS 모델과 coupled DEVS 모델에 대한 보다 상세한 것은 [5]을 참조하여라.

DEVS의 상태공간

DEVS의 상태공간은 위에서 정의된 Q이다. DEVS의 상태전이함수 $\delta : Q \times (X \bigcup \{\varepsilon\}) \rightarrow Q$는 다음과 같이 정의된다:

$$\delta((s,e),x) = \begin{cases} (\delta_{\text{int}}(s), 0) & \text{if } x = \varepsilon \\ (\delta_{\text{ext}}(s,e),x) & \text{otherwise} \end{cases}$$

2.3 시간페트리네트와 DEVS 형식론의 비교

초기의 페트리네트는 시간이나 성능(performance)에 대한 고려없이 시스템의 논리적 행위(behavior)만을 표현하기 위해서 사용되었다. 이 분야에서 상당한 연구성과를 거두운 결과로 도달성(reachability), 교착상태(deadlock)등과 같은 행위 속성 (behavioral property)을 분석하기 위한 다양한 기법들이 존재한다 [2]. 하지만, 초기의 페트리네트는 시스템의 동적인 속성(dynamic property)을 분석하지 못하는 단점이 있었고, 페트리네트 형식론에 시간 개념을 추가할려는 시도가 있어 왔다. 최초의 시간 페트리네트 (Timed Petri Nets)는 Ramchandani [3]에 의해 소개되었다. Ramchandani는 일반적인 페트리네트의 트랜지션에 지연시간을 추가하는 방식을 사용하였다. 이 후로 많은 연구자들이 시간 페트리네트에 관한 연구를 발표하였다.

하지만 이들의 시간 페트리네트는 형식론 내에서 시스템의 역학(dynamics)을 명시적으로 표현할 수 없다. 즉, 현재 상태에 머문 시간이 어느 정도인지, 앞으로 얼마나 시간이 경과해야 상태전이를 할 것인가와 같은 시스템의 동적 속성을 명시할 수 없는 단점이 있다.

또한 페트리네트는 계층적으로 모델링할 수 없기 때문에 대규모 시스템을 다루는 면에서 매우 불리하다.

한편 DEVS형식론은 시뮬레이션을 위해 개발된 형식론답게 계층적 모델링 및 시스템의 동적 역학을 잘 분석할 수 있는 기능을 제공한다. 하지만, DEVS형식론으로 모델링된 시스템의 행위속성을 분석하는 연구가 시간 페트리네트에 비하여 매우 부족한 것이 현실이다.

따라서 표현력은 DEVS형식론이 우세하지만 분석론은 시간 페트리네트가 유리하기 때문에 시스템 속성 분석을 위해서는 시간 페트리네트를, 동적 속성을 분석하기 위해서는 DEVS형식론을 사용하는 것이 편리하다.

3. 변환 알고리즘

3.1 기본 아이디어

변환 알고리즘을 설명하기 위해 필요로 하는 정의들은 다음과 같다.

정의 1: (모듈러 페트리네트)

모듈러 페트리네트는 다음의 조건을 만족하는 페트리네트이다.

$$|O(p_i)| = 1 \quad \text{for all } p_i \in P$$

즉 모듈러 페트리네트가 되기 위해서는 각 플레이스는 정확하게 한 개의 출력 트랜지션을 가져야 한다.

모듈러 페트리네트는 보통의 페트리네트의 하위분류(subclass)에 속한다. 저자들은 연구를 통해 DEVS 모델로 변환하려는 페트리네트가 모듈러 페트리네트이면 DEVS 모델로 변화하기가 쉽다는 것을 알게 되었다. 하지만 모듈러 페트리네트는 일반 페트리네트의 하위분류이므로, 보통의 페트리네트를 모듈러 페트리네트로 변환하는 방

법은 존재하지 않을 것이다.

대신 이 논문에서는 의사 모듈러 페트리네트를 제시한다. 의사 모듈러 페트리네트는 보통의 페트리네트 정의를 일시적으로 확장하지만 (즉 플레이스에서 트랜지션으로의 모서리중 일부가 무게로 음수를 가질 수 있게 정의를 확장함), DEVS 모델로 변환하는 과정 내에서만 사용이 한정되므로 사용자에게는 보이지 않고 변환 알고리즘을 매우 간편하게 하는 장점이 있다.

정의 2: (의사 모듈러 페트리네트)

의사 모듈러 페트리네트의 다음의 정의와

$$PN=\langle P, T, I, O\rangle$$

P 플레이스(place)의 집합

T 트랜지션(transition)의 집합

$I: T\rightarrow P^{\infty}$ 입력함수

$O: T\rightarrow P^{\infty}_{-\infty}$ 출력함수

다음의 제한조건을 갖는다:

$$|O(p_i)|=1$$

여기서, p^{∞}는 일반적인 가방(bag)이고, $p^{\infty}_{-\infty}$는 원소(element)의 개수가 음수일 수 있도록 본 논문에서 확장한 가방(bag)을 나타낸다.

3.2 변환 알고리즘

변환 알고리즘의 간략한 스케치는 다음과 같다:

1. 주어진 시간 페트리네트가 모듈러 페트리네트가 아니면 의사 모듈러 페트리네트로 변환한다.
2. (의사) 모듈러 페트리네트에서 각각의 트랜지션과 그 트랜지션의 입력 플레이스들과, 그 플레이스들에게 출력모서리로 연결되는 트랜지션들로 구성되는 부 페트리네트를 부-모듈로 정의한다. 정의된 각각의 부-모듈을 atomic DEVS 모델로 변환한다.
3. 각 부-모듈에 존재하는 하나의 출력 트랜지션에 대해, (1) 그 부-모듈의 출력단자로 그 트랜지션의 이름을 사용하고, (2) 그 트랜지션의 출력 플레이스를 가지는 다른 부-모듈에 대해서는 입력단자로 그 트랜지션의 이름을 사용한다. (3) 위의 (1)과 (2)의 단계에서 생성된 정보는 바로 부-모듈들을 연결하는 정보이므로, 이것을 이용하여 coupled DEVS 모델로 변환한다.

페트리네트를 DEVS 모델로 변환하는 정확한 알고리즘은 다음과 같다.

알고리즘: TPN2DEVS

1. 만약 $TPN=\langle P, T, I, O, \mu, \theta\rangle$이 모듈러 페트리네트가 아니면, (의사) 모듈러 페트리네트로 변환하여라.

   상세한 과정은 다음과 같다. $|O(p_i)|>1$인 각 p_i에 대해서 다음의 연산을 수행한다:

   (a) p_i의 입력 트랜지션이 다음과 같이

   x개이고

   $$I(p_i)=\{t_{i_1}, t_{i_2}, \cdots, t_{i_x}\}$$

   p_i의 출력 트랜지션이 다음과 같이 y개가

   $$O(p_i)=\{t_{o_1}, t_{o_2}, \cdots, t_{o_y}\}$$

   있다고 가정한다.

   (b) p_i을 제거한 후, y개의 새로운 플레이스인 $\{p_{i_1}, p_{i_2}, \cdots, p_{i_y}\}$을 추가한다. 이 때 새로 추가된 각 플레이스는 p_i가 가지고 있는 토큰과 같은 개수의 토큰을 가진다.

   (b) p_i에 연결되어 있던 모서리들을 제거한 후, 새로운 모서리를 다음과 같이 추가한다.

   $1\le v\le x, 1\le w\le y$인 각 v, w에 대해서, 웨이트가 $\#(p_i, O(t_{i_v}))$인 t_{i_v}에서 p_{i_w}로 연결되는 모서리를 추가한다.

   (d) $1\le w\le y$인 각 w에 대해서, 웨이트가 $\#(p_i, I(t_{o_w}))$인 p_{i_w}에서 t_{o_w}로 연결되는 모서리를 추가한다.

   (e) $1\le w\le y$인 각 w에 대해서,

$-\#(p_i, I(t_{o_w}))$의 웨이트를 가지고 각 $w'(1 \le w' \le y \wedge w' \neq w)$에 대해서 t_{o_w}에서 $p_{i_{w'}}$로 연결되는 모서리를 추가한다.

2. 위의 과정에서 얻어진 (의사) 모듈러 페트리네트에서 부 페트리네트들을 $\{TPN_{t_j}\}$들은 다음과 같이 구성된다.

   각 트랜지션 $t_j \in T$에 대해 다음과 같이 TPN_{t_j}를 구성한다:

   (a) 트랜지션 t_j과

   (b) t_j의 입력 플레이스들 $I(t_j)$과

   (C) t_j와 $I(t_j)$사이의 입력 모서리들과

   (c) $I(t_j)$에 출력 모서리로 연결되는 각 t_k들과

   (d) 각 t_k에서 플레이스 $I(t_j)$에 연결되는 출력 모서리들

3. 각 부 페트리네트 TPN_{t_j}을 atomic DEVS 모델 M_{t_j}로 다음과 같이 변환한다.

$$TPN_{t_j} = \langle P_{t_j}, T_{t_j}, I_{t_j}, O_{t_j}, \mu_{t_j}, \theta_{t_j} \rangle$$
$$M_{t_j} = \langle X_{t_j}, Y_{t_j}, S_{t_j}, \delta_{\text{int}.t_j}, \delta_{\text{ext}.t_j}, ta_{t_j}, \lambda_{t_j} \rangle$$
$$X_{t_j} = I(P_{t_j})$$
$$Y_{t_j} = T_{t_j} = \{t_j\}$$
$$S_{t_j} = \{\mu_{t_j}' \mid \mu_{t_j}' \in R(\mu_{t_j})\}$$

$$\delta_{\text{ext}.t_j}(\mu_{t_j}, t_k) = \begin{cases} \mu_{t_j} + e_{t_j}[k] \cdot D_{t_j}^{+} & \text{if } t_k \in X_{t_j} \\ \text{undefined} & \text{otherwse} \end{cases}$$

$$\delta_{\text{int}.t_j}(\mu_{t_j}) = \mu_{t_j} - e_{t_j}[j] \cdot D_{t_j}^{-}$$

$$ta_{t_j}(\mu_{t_j}) = \begin{cases} \theta_{t_j}(t_j) = \theta_j & \text{if } \mu_{t_j} \ge e_{t_j}[j] \cdot D_{t_j}^{-} \\ \infty & \text{otherwise} \end{cases}$$

$$\lambda_{t_j}(\mu_{t_j}) = t_j$$

4. 각 atomic DEVS 모델의 입출력 단자(port)를 연결하는 coupled DEVS 모델을 다음과 같이 구한다.

$$CM = \langle D, \{M_i\}, \{I_i\}\{Z_{ij}\}, \text{select} \rangle$$

$$D = \{1, 2, \cdots, m \mid m = |T|\}$$
$$M_i = DEVS_{t_i} \quad \text{for each } i \in D$$
$$I_i = \{j \mid t_i \in X_{t_j}\} \quad \text{for each } i, j \in D \wedge i \neq j$$
$$\{Z_{i,j}\} = \{(M_i.t_i, M_j.t_i) \mid M_i.t_i \in Y_{t_i} \wedge M_j.t_i \in X_{t_j}\} \quad \text{for each } i \in D, j \in I_i$$

여기서 $M_i.t_i$는 M_i컴포넌트의 (출력)단자 t_i을 나타내고 $M_j.t_i$는 M_j컴포넌트의 (입력)단자 t_i을 나타낸다.

4. 예제

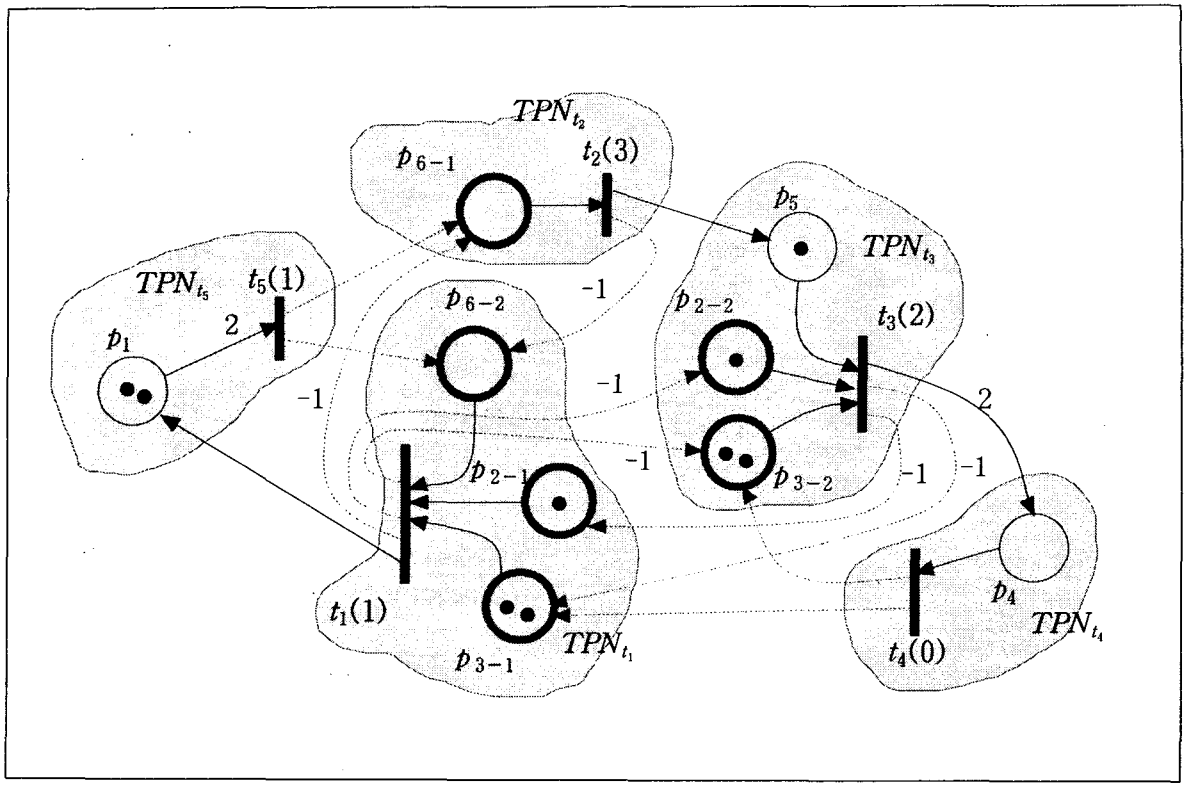
<그림 1>의 TPN_1을 사용하여 DEVS 모델로 변환하는 과정을 보여주고자 한다. TPN_1의 수학적 명세를 편리를 위해 여기에 다시 기술하였다.

$$TPN_1 = \langle P, T, I, O, \mu, \theta \rangle$$
$$P = \{p_1, p_2, p_3, p_4, p_5, p_6\}$$
$$T = \{t_1, t_2, t_3, t_4, t_5\}$$
$$\mu = (2, 1, 2, 0, 1, 0)$$
$$\theta = (1, 3, 2, 0, 1)$$

$I(t_1) = \{p_2{:}1, p_3{:}1, p_6{:}1\}$ $O(t_1) = \{p_1{:}1\}$
$I(t_2) = \{p_6{:}1\}$ $O(t_2) = \{p_5{:}1\}$
$I(t_3) = \{p_2{:}1, p_3{:}1, p_5{:}1\}$ $O(t_3) = \{p_4{:}2\}$
$I(t_4) = \{p_4{:}1\}$ $O(t_4) = \{p_3{:}1\}$
$I(t_5) = \{p_1{:}2\}$ $O(t_5) = \{p_6{:}1\}$

TPN_1이 모듈러 페트리네트가 아니므로 의사 모듈러 페트리네트로 변환하여야 한다. <그림 3>은 <그림 2>의 페트리네트를 의사 모듈러 페트리네트로 변환한 결과를 보여주고 있다. 변환 과정을 자세히 살펴보면 다음과 같다.

1. p_6는 출력 트랜지션이 두 개 (t_2, t_1)이므로 p_6대신 p_{6-1}와 p_{6-2}로 대체한다.
2. p_6의 입력 모서리를 p_{6-1}와 p_{6-2}의 입력 모서리로 대체한다.
3. p_6의 출력 모서리를 p_{6-1}와 p_{6-2}의 출력 모서리로 대체한다.
4. 원래의 페트리네트에서는 출력 트랜지션들



<그림 3> 페트리네트를 의사 모듈러 페트리네트로 변환하기

중에서 한 순간에는 한 트랜지션만 fire하므로, 의사 페트리네트에서도 똑 같은 행동을 보장하기 위해서는 p_{6-1}나 p_{6-2}의 출력 트랜지션이 fire할 때 복사된 다른 플레이스에서는 토큰을 제거해야 한다. 즉 t_2가 fire한다면 p_{6-2}의 토큰을 하나 제거해야 하기 때문에 (t_2, p_{6-2}) 모서리를 추가하였다. (t_1, p_{6-1}) 모서리는 같은 이유로 추가되었다.

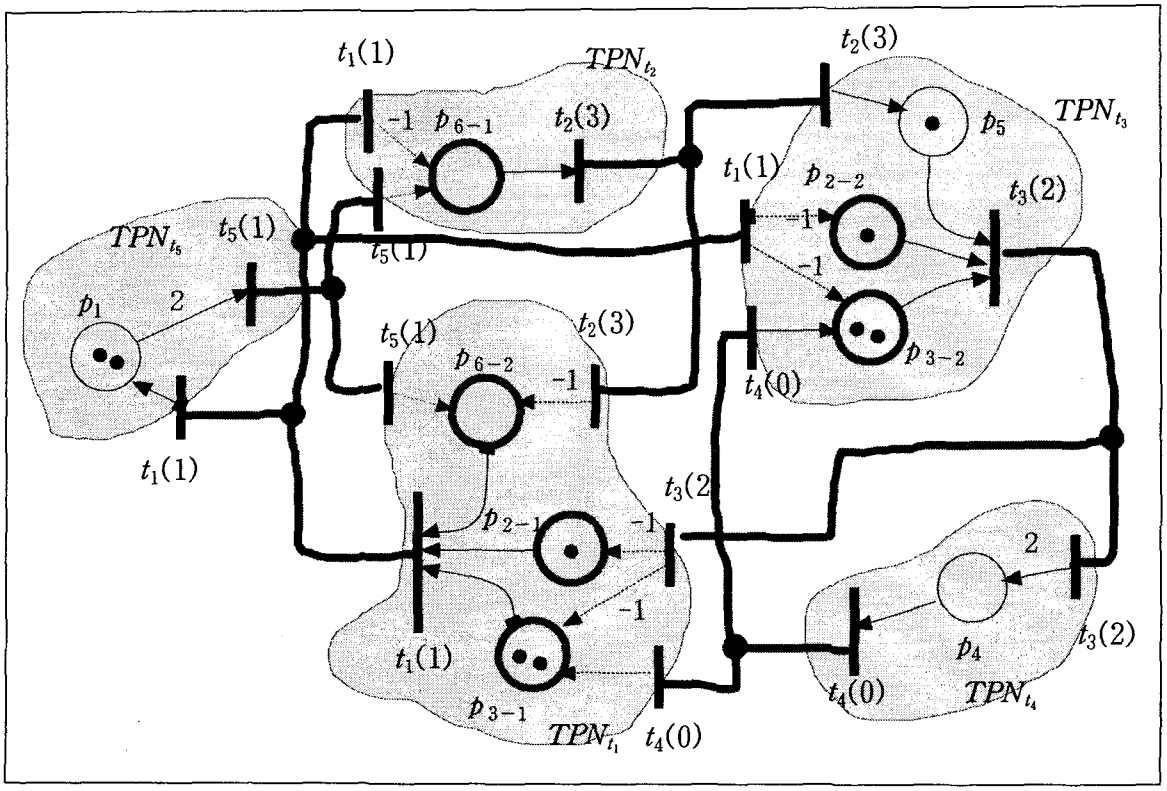
5. p_2나 p_3도 위와 같은 과정으로 p_{2-1}, p_{2-2}와 p_{2-1}, p_{2-2}로 각각 대체되었다.

변환된 의사 페트리네트의 수학적 모델은 다음과 같다.

$$TPN_2 = (P, T, I, O, \mu, \theta)$$

$$\begin{aligned}
P &= \{p_1, p_{2-1}, p_{2-2}, p_{3-1}, p_{3-2}, p_4, p_5, p_{6-1}, p_{6-2}\} \\
T &= \{t_1, t_2, t_3, t_4, t_5\} \\
\mu &= (2,1,1,2,2,0,1,0,0) \\
\theta &= (1,3,2,0,1) \\
I(t_1) &= \{p_{2-1}:1, p_{3-1}:1, p_{6-2}:1\} \\
O(t_1) &= \{p_1, p_{2-2}:-1, p_{3-2}:-1, p_{6-1}:-1\} \\
I(t_2) &= \{p_{6-1}:1\} \\
O(t_2) &= \{p_5:1, p_{6-2}:-1\} \\
I(t_3) &= \{p_{2-2}:1, p_{3-2}:1, p_5:1\} \\
O(t_3) &= \{p_4:2, p_{2-1}:-1, p_{3-1}:-1\} \\
I(t_4) &= \{p_4:1\} \\
O(t_4) &= \{p_{3-1}:1, p_{3-2}:1\} \\
I(t_5) &= \{p_1:2\} \\
O(t_5) &= \{p_{6-1}:1, p_{6-2}:1\}
\end{aligned}$$

<그림 4>는 다섯 개의 부 페트리네트들 (TPN_{t_1}, TPN_{t_2}, TPN_{t_3}, TPN_{t_4}, TPN_{t_5})을 보여



<그림 4> 모듈러네트를 DEVS로 변환하기

주고 있다. TPN_{t_1}과 TPN_{t_5}의 수학적 명세는 다음과 같다:

$$TPN_{t_1}=(P_{t_1}, T_{t_1}, I_{t_1}, O_{t_1}, \mu_{t_1}, \theta_{t_1})$$

$P_{t_1} = \{p_{2-1}, p_{3-1}, p_{6-2}\}$
$T_{t_1} = \{t_1, t_2, t_3, t_4, t_5\}$
$\mu_{t_1} = (\mu_{2-1}, \mu_{3-1}, \mu_{6-2}) = (1,2,0)$
$\theta_{t_1} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = (1,3,2,0,1)$

$I_{t_1}(t_1) = \{p_{2-1}:1, p_{3-1}:1, p_{6-2}:1\}$
$O_{t_1}(t_1) = \{\}$
$I_{t_1}(t_3) = \{\}$
$O_{t_1}(t_3) = \{p_{2-1}:-1, p_{3-1}:-1\}$

$I_{t_1}(t_2) = \{\}$ $O_{t_1}(t_2) = \{p_{6-2}:-1\}$
$I_{t_1}(t_4) = \{\}$ $O_{t_1}(t_4) = \{p_{3-1}:-1\}$
$I_{t_1}(t_5) = \{\}$ $O_{t_1}(t_5) = \{p_{6-2}:-1\}$

$$TPN_{t_5}=(P_{t_5}, T_{t_5}, I_{t_5}, O_{t_5}, \mu_{t_5}, \theta_{t_5})$$

$P_{t_5} = \{p_1\}$
$T_{t_5} = \{t_1, t_5\}$
$\mu_{t_5} = (\mu_1) = (2)$
$\theta_{t_5} = (\theta_1, \theta_5) = (1,1)$

$I_{t_5}(t_1) = \{\}$ $O_{t_5}(t_1) = \{p_1:1\}$
$I_{t_5}(t_5) = \{p_1:2\}$ $O_{t_5}(t_5) = \{\}$

다른 부 페트리네트들도 비슷하게 명세된다. 부 페트리네트 TPN_{t_1}과 TPN_{t_5}에 각각 대응하는 atomic DEVS 모델 M_{t_1}과 M_{t_5}는 다음과 같이 명세된다:

$M_{t_1} = \langle X_{t_1}, Y_{t_1}, S_{t_1}, \delta_{\text{int.}t_1}, \delta_{\text{ext.}t_1}, ta_{t_1}, \lambda_{t_1} \rangle$

$X_{t_1} = I(P_{t_1}) = \{t_2, t_3, t_4, t_5\}$
$Y_{t_1} = \{t_1\}$
$S_{t_1} = \{\mu_{t_1}' \mid \mu_{t_1}' \in R(\mu_{t_1})\}$

$$\delta_{ext.t_1}(\mu_{t_1}, t_k) = \begin{cases} \mu_{t_1} + e_{t_1}[k] \cdot D_{t_1}^{+} & \text{if } t_k \in X_{t_1} \\ \text{undefined} & \text{otherwse} \end{cases}$$

$$\delta_{int.t_1}(\mu_{t_1}) = \mu_{t_1} - e_{t_1}[1] \cdot D_{t_1}^{-}$$

$$ta_{t_1}(\mu_{t_1}) = \begin{cases} \theta_1 = 1 & \text{if } \mu_{t_1} \geq e_{t_1}[1] \cdot D_{t_1}^{-} \\ \infty & \text{otherwise} \end{cases}$$

$$\lambda_{t_1}(\mu_{t_1}) = t_1$$

$$M_{t_5} = \langle X_{t_5}, Y_{t_5}, S_{t_5}, \delta_{int.t_5}, \delta_{ext.t_5}, ta_{t_5}, \lambda_{t_5} \rangle$$

$$X_{t_5} = I(P_{t_5}) = \{t_1\}$$

$$Y_{t_5} = \{t_5\}$$

$$S_{t_5} = \{\mu_{t_5}' \mid \mu_{t_5}' \in R(\mu_{t_5})\}$$

$$\delta_{ext.t_5}(\mu_{t_5}, t_k) = \begin{cases} \mu_{t_5} + e_{t_5}[k] \cdot D_{t_5}^{+} & \text{if } t_k \in X_{t_5} \\ \text{undefined} & \text{otherwse} \end{cases}$$

$$\delta_{int.t_5}(\mu_{t_5}) = \mu_{t_5} - e_{t_5}[5] \cdot D_{t_5}^{-}$$

$$ta_{t_5}(\mu_{t_5}) = \begin{cases} \theta_5 = 1 & \text{if } \mu_{t_5} \geq e_{t_5}[5] \cdot D_{t_5}^{-} \\ \infty & \text{otherwise} \end{cases}$$

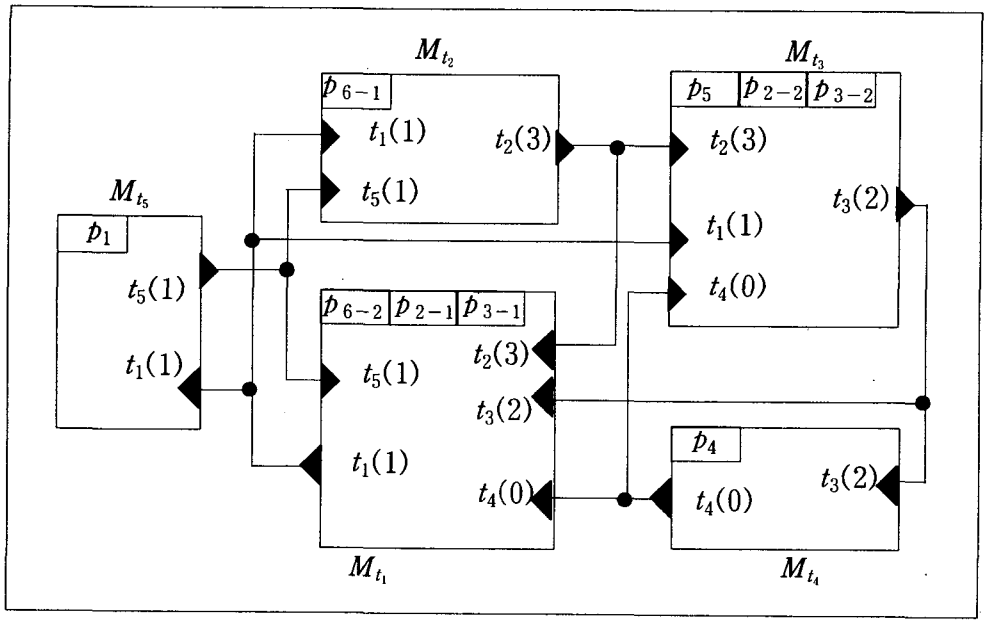
$$\lambda_{t_5}(\mu_{t_5}) = t_5$$

<그림 4>는 또한 다섯 개의 부 페트리네트가 의 입출력 단자와 부 페트리네트 사이의 연결 관계를 보여주고 있다.

기본적인 아이디어는 다음과 같다. 각 부 페트리네트의 트랜지션은 출력 단자 역할을 하고, 플레이스들은 상태변수 역할을 하게 하는 것이다. 따라서 각 부 페트리네트의 트랜지션이 fire하면 토큰이 다른 부 페트리네트의 플레이스로 전달되는 것을 DEVS 모델에서는 그 트랜지션의 이름과 같은 출력단자에서 이벤트가 발생하고 그 트랜지션의 출력 플레이스에 해당되는 다른 DEVS 모델의 입력단자로 이벤트가 전달되는 것으로 해석한다.

다른 관점으로 설명을 하면, 페트리네트의 각 트랜지션을 각 입출력 모서리 개수만큼 분할하여 해당하는 DEVS 모델의 입출력 단자로 활용하고 이 분할된 트랜지션을 연결하면 <그림 5>와 같은 결과가 나온다. coupled DEVS 모델의 수학적 정의는 다음과 같다.

$$DN = \langle D, \{M_i\}, \{I_i\}\{Z_{ij}\}, \text{select} \rangle$$



<그림 5> DEVS 모델로 완전히 변화된 모습

$$D = \{1,2,3,4,5\}$$
$$M_i = M_{t_i} \text{ for each } i \in D$$
$$I_1 = \{2,3,5\}$$
$$I_2 = \{1,3\}$$
$$I_3 = \{1,4\}$$
$$I_4 = \{1,3\}$$
$$I_5 = \{1,2\}$$

$$Z_{1,2} = (M_1.t_1, M_2.t_1)$$
$$Z_{1,3} = (M_1.t_1, M_3.t_1)$$
$$Z_{1,5} = (M_1.t_1, M_5.t_1)$$
$$Z_{2,1} = (M_2.t_2, M_1.t_2)$$
$$Z_{2,3} = (M_2.t_2, M_3.t_2)$$
$$Z_{3,1} = (M_3.t_3, M_1.t_3)$$
$$Z_{3,4} = (M_3.t_3, M_4.t_3)$$
$$Z_{4,1} = (M_4.t_4, M_1.t_4)$$
$$Z_{4,3} = (M_4.t_4, M_3.t_4)$$
$$Z_{5,1} = (M_5.t_5, M_1.t_5)$$
$$Z_{5,2} = (M_5.t_5, M_2.t_5)$$

여기서 select함수는 동시에 enable된 여러 개의 트랜지션 중에서 무작위(random)하게 선택하는 함수로 구현하면, 페트리네트의 동적 모델과 동일하게 된다.

5. 결론 및 향후 연구 방향

한 개의 시스템을 시간 페트리네트와 DEVS 형식론으로 혼용해서 모델링 했을 경우, 전체 시스템을 시뮬레이션 하기 위한 모델 변환 기법을 제시 하였다. DEVS 형식론 [5]은 이산사건 모델을 계층적으로 모듈화하여 명세할 수 있는 수학적 틀을 제공한다. 반면에 시간 페트리네트는 시스템을 계층적으로 모듈화하여 표현할 수 없다. 따라서 시간 페트리네트를 DEVS 형식론으로 변환하는 것은 통합 시뮬레이션을 할 수 있는 기반을 제공한다. 이를 위해 모듈러 페트리네트와 의사 모듈러 페트리네틀 개념을 정의하고, 이를 사용하여 시간 페트리네트를 DEVS 형식론으로 변환하는 알고리즘을 제시하였다. 또한 예를 통해 이 알고리즘이 동작을 설명하였다.

향후 연구과제는 다음과 같은 것들이 남아 있다. 첫째, 이 논문에서 시간 페트리네트를 DEVS로 변환하는 알고리즘을 제시하였다. 하지만, 이 연구에서는 제시한 알고리즘이 올바른 것을 예를 통해서 간접적으로 보여 주었지만 수학적(formal)으로 증명하는 것이 남아있다. 둘째, 제안한 알고리즘을 기반으로 하여 시간 페트리네트와 DEVS모델을 지원하는 시뮬레이션 환경(framework)을 구축하는 것이 필요하다. 이러한 시뮬레이션 환경이 제공될 때 이론적 모델에서 벗어나 실생활에서 활용이 되는 모델로 발전할 수 있을 것이다.

참고문헌

[1] J. L. Peterson, *Petri Net Theory and The Modeling of Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[2] T. Murata, "Petri Nets: Properties, Analysis and Applications," *Proceeding of the IEEE*, Vol 77, No. 4, pp. 541-580, Apr. 1989.

[3] C. Ramchandani, *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*, Phd Thesis, MIT, 1973

[4] M. A. Holliday and M. K. Vernon, "A Generalized Timed Petri Net Model for Performance Analysis," IEEE Trans. on Software Engineering, Vol. SE-13, No. 12, Dec., 1987.

[5] C. G. Cassandras, *Discrete Event Systems: Modeling and Performance Analysis*, Aksen Associates, Inc., 1993.

[6] B. P. Zeigler, H. Praehoff, T.G. Kim *Theory of Modelling and Simulation(2nd Ed.)*, Academic Press, San Diego, 2000.

[7] B. P. Zeigler, *Multifacettedf Modelling and Discrete Event Simulation*, Academic Press, London, 1984.

[8] 안 명수, 박 성봉, 김 탁곤, "DEVSim++: 의미론에 기반한 이산사건 시스템의 객체지향 모델링 및 시뮬레이션 환경, 한국정보과학회 논문지, 제 21권, 제 9호, 페이지 1652-1664, 1994년 9월

● 저자소개 ●

김영찬
1985: 아주대학교 공과대학 전자공학과 학사
1987: 한국과학기술원 전기및전자공학과 석사
1995: 한국과학기술원 전기및전자공학과 박사
1984 - 1990: 삼성전자 종합연구소 연구원
1995 - 1995: 한국과학기술원 위촉연구원
1997 - 1998: 시스템공학연구소 및 전자통신연구원 연구원
1998 - 현재: 한밭대학교 정보통신-컴퓨터공학부 조교수
관심분야: 시스템 검증, 형식론(Petri Net, DEVS등), 데이터베이스, XML

김탁곤
1988: 아리조나대학교 전자/컴퓨터공학과 공학박사
1980 - 1983: 국립수산대학(현: 부경대학교) 통신공학과 전임강사
1987 - 1989: 아리조나 환경연구소 연구 엔지니어
1989 - 1991: 캔사스 대학교, 전자전산학과, 조교수
1991 - 현재: 한국과학기술원, 전자전산학과, 조교수/부교수/교수
2001 - 현재: 미국 시뮬레이션 기술사
2000 - 현재: SIMULATION: Trans of SCS 편집위원장
모델링/시뮬레이션 방법론 및 환경 개발, 시뮬레이션에 기반한 시스템 분석 등에 관하여 국제적으로 활발한 연구 활동을 하고 있으며 국제학술지/국제학술발표대회에 100 편 이상의 논문을 개재 하였음. 시뮬레이션 모델링에 관련된 다수의 국제학술지 편집을 담당하고 있다. 저서(공저자: B.P. Zeigler, H. Praehofer)로 2000 년 미국 Academic Press에서 발간한 모델링 시뮬레이션 전문 교재인 *Theory of Modeling and Simulation(2nd edition)* 이 있음. 한국시뮬레이션 학회 초대 편집위원장을 역임하였고 현재 학술부 회장 임. IEEE(국제 전기전자 학회) 및 SCS(국제 시뮬레이션 학회)의 Senior Member 이며 ACM(미국전산학회) 및 Eta Kappa Nu 의 Regular Member 임.