SPN을 이용한 DEVS 모델의 타당성 검사
Validation Test of DEVS Models using SPN

정 영 식*, 황 종 선*, 백 두 권*
Young-Sik Jeong*, Chong-Sun Hwang*, Doo-Kwon Baik*

Abstract

In this paper, we study validation test methods of DEVS(Discrete EVent system Specification) models using SPN(Stochastic Petri Net) models. We discuss conventional validation test methods of DEVS models and its problems. We propose a transformation method, by which DEVS models can be transformed to SPN models, by reviewing the features of DEVS models. Based on the model transformation method, we define a new homogeneous function for validation test and suggest a new validation test method of DEVS models using the property of SPN models and the new homogeneous function.

1. 서론

시뮬레이션은 실 시스템의 효과적이고 효율적인 운영을 도모하기 위하여 실 시스템의 동작을 이해하고 분석, 예측, 평가하는 과학적인 문제해결 접근방법이다[Law 82]. 시뮬레이션 수행단계는 실 시스템의 행위를 정확히 반영하도록 타당한 모델을 구축하는 모델링 단계와 모델이 의도하는 명령들을 컴퓨터 프로그램으로 작성하는 구현단계로 나누어진다[Law 82].

시뮬레이션 모델은 시간, 상태, 확률변수, 상호규칙등의 여러 측면에 따라 다양하게 존재하는데, DEVS(Discrete EVent system Specification) 모델은 연속적인 시간상에서 이산적으로 발생하는 사건에 따라 시스템의 상태를 분석할 수 있고 모델링 및 시뮬레이션 방법론의 형식화를 위한 견고한 이론적 기반을 제공하고 있다[Zeig 84]. 또한, DEVS 모델은 모듈적, 계층적 특성을 제공하고 집합론에 근거한 수학적 형식구조를 제공하여 실 시스템에 대한 체계적인 분석과정을 수행하게 되어 보다 현실적인 모델링을 가능하게 한다[Zeig 84]. 그러나 타당하지 못한 DEVS 모델이 구축되면 시뮬레이션을 통한 분석결과에 대하여 신뢰성이 없고 비경제적이다.

DEVS 모델의 타당성 검사에 대한 연구는 전무한 실정이며, 단지 Zeigler가 [Zeig 84]에서 제시하였다. [Zeig 84]에서 제시된 DEVS 모델의 타당성 검사방법은 많은 시간

* 고려대학교 전산과학과

과 노력이 요구되고, 반복적인 DEVS 모델링 과정으로 인한 전문적이고 경험적인 지식을 요구한다. 또한, 모델설계자에 의해 설정된 실험 프레임하에서 DEVS 모델의 구성요소에 속하는 상태전이함수, 시간 진행함수 및 출력함수에 대하여 commutative 성질의 보존성 검사가 어렵다는 문제점을 가지고 있다[Zeig 84].

이와같은 문제점들을 해결하기 위하여 DEVS 모델의 타당성 검사를 SPN 모델을 이용하는 새로운 방법을 제안한다. 본 논문의 제 2절에서는 DEVS 모델의 개념, 기존의 타당성 검사 방법을 고찰하고 그 문제점들을 자세히 설명한다. 제 3절에서는 DEVS 모델의 타당성 검사에 이용하는 SPN 모델에 대한 개념, 모델변환을 위한 관점을 재조명하고 DEVS 모델이 SPN 모델로 변환됨을 보인다. 마지막절에서는 모델 변환방법을 토대로 새로운 동질함수를 정의하고 SPN 모델의 특성과 동질함수를 이용하여 DEVS 모델에 대한 새로운 타당성 검사 방법을 제안한다.

2. DEVS 모델의 타당성 검사

2.1 DEVS 모델

DEVS 모델은 연속적인 시간상에서 이산적으로 발생하는 사건들에 대하여 시스템의 행위를 측정하는 것으로 다음과 같은 형식론에 의해 모델을 표현하는 형식모델이다[Zeig 84].

[정의 1] DEVS 구조

DEVS는 입력집합 X, 출력집합 Y, 상태집합 S, 시간진행함수 ta, 외부상태전이함수 δext, 내부상태전이함수 δint, 출력함수 λ로 구성된다.

$$COM = \langle Y, Y, S, ta, \delta ext, \delta int, \lambda \rangle$$

여기서, $ta : S \rightarrow R^{+}_{0\infty}$

$s \mapsto ta(s)$

단, $R^{+}_{0\infty}$는 음수를 제외한 실수집합

$\delta ext : S \times X \rightarrow S$

$(s, x) \mapsto \delta ext(s, x)$

$\delta iont : S \rightarrow S$

$s \mapsto \delta int(s)$

$\lambda : S \rightarrow Y$

$s \mapsto \lambda(s)$

입력집합은 시스템 외부에서 발생하는 사건들의 집합을 의미하고 출력집합은 출력변수들의 집합을 나타낸다. 상태집합은 상태변수들의 각 정의구역들의 곱집합을 의미하며 상태 $s(\in S)$는 시간진행에 따른 시스템의 순차적인 스냅샷(sequential snapshot) 상태를 의미한다. 시간진행함수는 시스템이 외부사건을 입력받지 않는 한, 상태 s에 머물수 있도록 허용한 시간으로 정의한다. 외부상태전이함수는 시스템 외부에서 발생한 사건에 의한 모델의 상태변화를 나타내는 함수로 정의하고, 내부상태전이함수는 외부사건이 없는 경우 시간진행에 따라 모델의 상태변화를 설명해 주는 함수로 정의한다. 출력함수는 상태 s에서 시스템의 출력으로 정의한다.

2.2 타당성 검사 및 문제점

DEVS 모델의 타당성 검사에 대한 전통적인 방법은 모델링된 실 시스템에 대하여 실험 프레임 (Experimental Frame : EF) 을 설정하고 각 EF를 고려한 DEVS 모델의 구성요소에 속하는 외부상태전이함수 δext, 내부상태전이함수 δint, 시간진행함수 ta, 출력함수 λ가 보존되면 타당하다[Zeig 84].

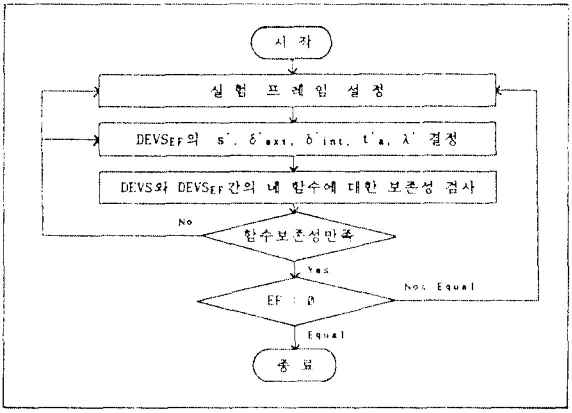
[정의 2] DEVS 모델의 타당성

"DEVS 모델이 타당하다." 는 다음과 동일한 의미를 갖는다.

① 모든 실험 프레임하에서 DEVS 모델이 타당하다.
(특정 EF하에서의 DEVS를 $DEVS_{EF}$로 표시한다.)

② 각 $DEVS_{EF}$의 δ'ext, δ'int, t'a, λ'가 DEVS의 δext, δint, ta, λ 성질을 보존한다.

기존의 DEVS 모델에 대한 타당성 검사를 위한 전체 수행절차는 〈그림 1〉과 같다.



〈그림 1〉 DEVS 모델의 타당성 검사 절차

각 단계에 대한 자세한 내용은 다음과 같다.

1) 실험 프레임 설정

실험 프레임은 시스템의 입출력 행위에 대한 한 부분으로써 반드시 모델설계자에 의해 관측되고 실험될 수 있는 환경들의 제한된 집합이다[Zeig 84]. 모델설계자는 실험 프레임을 출력목적에 적합한 서술변수로 부터 설정한다. 확률변수와 시간에 대한 countdown clock 변수들은 관측하지 못하는 변수들로 취급되어 이들을 출력변수로 하는 실험 프레임은 설정하지 않는다.

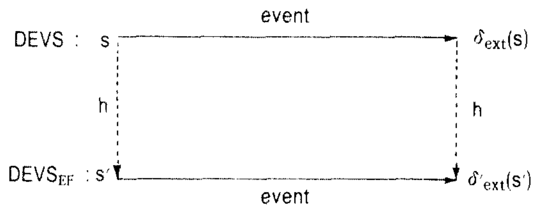
2) $DEVS_{EF}$의 s', $\delta'ext$, $\delta'int$, $t'a$, λ' 결정

설정된 실험 프레임하에서의 DEVS 모델링과정을 반복하여 상태집합, 외부상태전이함수, 내부상태전이함수, 시간진행함수, 출력함수를 새롭게 정의한다. 그러면, $DEVS_{EF}$에 대한 구조 $COM_{EF} = \langle X, Y, S', \delta'ext, \delta'int, t'a, \lambda' \rangle$를 정의할 수 있다.

3) DEVS와 $DEVS_{EF}$간의 네 함수 보존성 검사

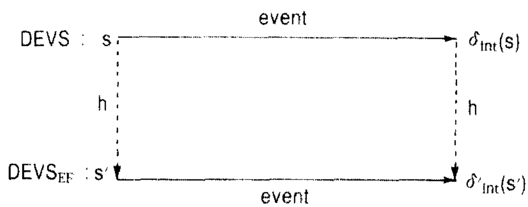
네 함수는 외부상태전이함수, 내부상태전이함수 δext $\delta' ext$, 내부상태전이함수 δint, $\delta' int$, 시간진행함수 ta, t'a, 출력함수 λ, λ'를 의미하고 DEVS의 상태집합 s와 $DEVS_{EF}$의 상태집합 s'를 각각 정의구역과 공변역으로 하는 함수 h를 정의한다. 정의된 함수 h를 이용하여 함수에 대한 보존성을 검사하는데 자세한 내용은 다음과 같다.

① 외부상태전이함수



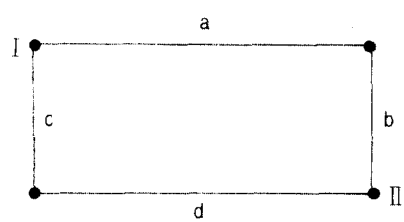
단, event : 경과시간 e에서 외부사건발생을 의미

② 내부상태전이함수



단, event : 스케쥴링된 내부사건발생을 의미

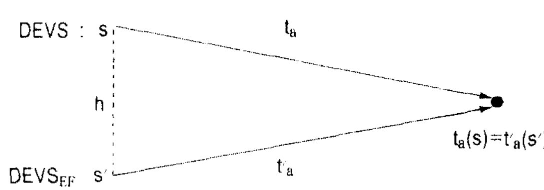
①과 ②의 함수 보존성은 〈그림 2〉의 commutative 다이어그램에 의해 Ⅰ에서 Ⅱ까지의 경로(path) ab에 대한 결과와 경로 cd에 대한 결과가 동일함을 증명해야 한다.



〈그림 2〉 commutative 다이어그램

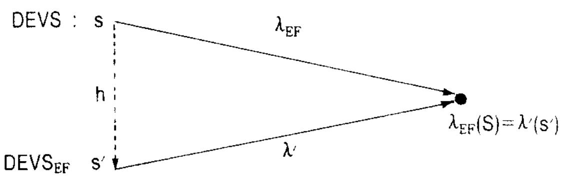
시간진행함수의 보존성은 DEVS와 $DEVS_{EF}$에 대한 각각의 시간진행함수 ta, t'a를 적용한 결과가 동일해야 한다.

③ 시간진행함수



출력함수의 보존성은 DEVS와 $DEVS_{EF}$에 대한 각각의 함수를 λ_{EF}, λ'라 하고 그 적용한 결과가 동일해야 한다. 이때, 함수 λ_{EF}는 DEVS에서 설정된 실험프레임들 중에서 특정한 EF에 대한 하나의 출력함수를 의미한다.

④ 출력함수



〈그림 1〉 과 같은 수행절차에 의한 DEVS 모델의 타당성 검사 방법은 다음과 같은 문제점을 갖는다.

1) 많은 시간과 노력을 요구한다.

실험 프레임 EF_1과 EF_2하에 $DEVS_{EF1}$이 타당하다고 해서 $DEVS_{EF2}$가 타당하다는 보장이 없기 때문에 모든 실험 프레임에 대한 타당성을 개별적으로 모두 검사해야 한다. 실험 프레임의 수만큼 타당성 검사 절차를 반복해야 하며, 각 절차에 필요한 지식과 경험이 상당히 요구된다.

2) 반복적인 DEVS 모델링 과정을 요구한다.

$DEVS_{EF}$의 타당성 검사에 상태집합, 외부상태전이함수, 내부상태전이함수, 시간진행함수 및 출력함수를 반복적으로 구성하여야 하는 번거러운 작업 수행과정이 요구된다.

3) 함수 보존성 검사가 어렵다.

모든 실험 프레임에 대하여 외부상태전이함수와 내부상태전이함수의 commutative 성질이 만족하는지를 검사하는 것과 시간진행함수와 출력 함수의 결과 비교를 별개로 수행하여야 한다. 이들은 함수 h의 정의와 이를 이용하는 함수적 성질에 의해 수행되기 때문에 절차가 복잡하고 난해하다.

3. DEVS의 SPN으로의 모델변환

3.1 SPN 모델

상태/사건모델로 간주되는 Petri Net은 순서도나 블럭 다이어그램과 같이 시각적으로 의소소통을 가능하게 해주는 도식적 도구일 뿐만아니라 견고한 이론을 갖는 수학적 모델링 도구이다[Moll 82, Mura 89, Pete 81].

Petri Net은 여러가지 목적에 따라 다양하게 수정, 확장되어 왔다. 본 연구에서는 transition 점화가 일반적인 확률분포함수에 의해 생성된 시간이 경과한 후 점화되고 inhibitor arc를 갖는 SPN(Stochastic Petri Net)에 대한 구조적 정의와 그래프를 소개하고 이용한다[Moll 82, Mura 89].

【정의 3】 SPN 구조

inhibitor arc를 갖는 SPN은 place의 유한집합 P, transition 유한집합 T, 입력함수 I, 출력함수 O, 억제입력함수 f, 점화대기시간함수 ρ, 초기마킹 μ_0로 구성된다.

$$N=\langle P,\ T,\ I,\ O,\ f,\ \rho,\ \mu_0\rangle$$

여기서,

$P=\{p_1,\ p_2,\ \cdots,\ p_n\}\ n\geq 0$

$T=\{t_1,\ t_2,\ \cdots,\ t_m\}\ m\geq 0,\ P\cap T=\phi$

$I=T\ \rightarrow\ P$

$t_i\mapsto I(t_i)$ 단, $I(t_i)$: t_i의 입력 place 집합

$O=T\ \rightarrow\ P$

$t_i\mapsto f(t_i)$ 단, $f(t_i)$의 inhibitor arc에 연결된 입력 place 집합

$\rho:T\rightarrow R^+$

$t_i\mapsto \rho(t_i)$ 단, $\rho(t_i)$: t_i의 점화대기시간

R^+:음수를 제외한 실수집합

μ_0 : 초기마킹

SPN 그래프에서 각 place(원으로 표시)는 상태를 나타내고 각 transition(선분으로 표시)은 사건을 의미한다. 입력함수 $I(t_i)=\phi$가 되는 t_i를 source transition 이라 정의하고 출력함수 $O(t_i)=\phi$인 t_i는 sink transition 이라 정의하고 이들은 집합 T에 포함된다. place에 토큰을 위치시킴으로써 상태가 유지됨을 표현하고 place와 transition, transition

과 place의 연결은 directed arc에 의해 구성된다. transition은 자신에게 입력되는 모든 place가 토큰을 가지고 있지 않을 때 점화가능한 것으로 분류한다. transition이 점화되면 각 출력 place에 토큰을 하나씩 첨가한다.

점화대기시간함수 ρ에 의해 transition ti 에 부여된 실수값은 ti가 점화 가능한 시점에서 점화될 때까지의 대기시간으로 정의하고 transition 점화는 원자적 운영(atomic operation)이다.

SPN에서 place에 분산된 토큰의 분포를 마킹 μ라 하고 마킹 μ로부터 도달 가능한 모든 마킹들을 고려할 때, SPN 구조 N과 마킹 μ에 대한 도달성 집합R(N, μ)는 다음과 같이 정의한다[Pete 81].

【정의 4】 도달성 집합

마킹 μ를 가진 SPN에서 도달성 집합(reachability set)은 다음과 같다.

① $\mu \in R(N, \mu)$

② if $\mu' \in R(N, \mu)$ & 임의의 transition ti에 대하여 $\mu'' = \delta(\mu'' \in ti)$이면 $\mu'' \in R(N, \mu)$

여기서, $\mu'' = \delta(\mu', ti)$: 마킹 μ''는 마킹 μ로부터 즉시 도달가능

【정의 5】 도달성 문제

도달성 문제(reachability problem)는 마킹 μ, μ'를 가진 SPN 구조 N이 주어지면 $\mu' \in R(N, \mu)$임을 검사하는 것이다.

단, R(N, μ)는 SPN 구조 N에 대한 마킹 μ로부터의 도달성 집합

【정의 6】 생존성

R(N, μ)에 속하는 모든 마킹에 대하여 transition ti가 잠재적으로 점화가능하면 SPN 구조 N에 포함되는 transition이 마킹 μ에서 live하다.

도달성, 생존성[Mura 89, Pete 81, 손 91]등의 SPN 속성과 도달성 집합R(N, μ)에 의해 개별적인 사건발생에 따른 시스템의 상태변화 및 도달가능한 상태가 어떤 것들이 존재하는지를 파악할 수 있다.

3.2 변환방법

이산사건 관점에서 시뮬레이션 모델을 구축하기 위한 월드 뷰(world view)는 모델구축 접근방법에 따라 사건 스케쥴링(event scheduling), 액티비티 조사(activity scanning) 및 프로세스 상호작용(process interaction)으로 분류한다[Law 82, Neel 87, Zeig 84].

각 월드 뷰는 고유한 특성을 가지고 있으며 그 중에서 사건 스케쥴링은 연속적인 시간상에서 이산적으로 발생하는 사건을 중심으로 시스템의 상태를 파악하고 비교적 독립적인 구성요소를 갖는 모델에 대하여 효과적인 문제해결 접근방법을 위한 월드 뷰이다. 이와같은 월드 뷰는 대화식 인터페이스, 그래픽등을 이용할 수 있어 응용범위가 넓고 소프트웨어 공학 방법론의 표준으로 응용될 수 있어 매우 융통성(flexibility)있는 관점을 제공한다. 이 때문에 사건 스케쥴링은 시뮬레이션 모델링에 많이 이용된다.

DEVS 모델의 상태집합 s($\in$S)는 일반적으로 모델링된 시스템의 상태를 나타내기 위한 상태변수들을 요소로 갖는 n-튜플(tuple) 구조이다. 상태집합 s를 $\langle s_1, s_2, \cdots, s_n \rangle$이라 가정하면, DEVS 모델은 입출력 개념을 명시하고 SPN 모델보다 추상적이고 포괄적인 모델을 구축하는 형식론이므로 상태변수 si들의 타입을 다음과 같은 다섯가지 타입으로 제한하고 사건 스케쥴링 월드 뷰로 제한한다[정 92].

i) R 타입 : $si \in \{r_j \mid j \in N\}$

ii) Q 타입 : $si \in \{(q_1, q_2, \cdots, qm) \mid m \in N\}$

iii) V 타입 : $si \in \{n_j \ \ n_j \in R^+, j \in N\}$

iv) E 타입 : $si \in \{(q_1, t_1), (q_2, t_2), \cdots, (q_n, t_n) \mid n \subset N\}$

v) T 타입 : $si \in \{\sigma_j \mid \sigma_j \in R^+, j \in N\}$

R 타입은 확률형 상태변수로서 시스템에서 발생하는 외부, 내부사건 발생을 나타내는 상태변수이다. Q 타입은 서술형 상태변수로서 엔티티가 구성요소에서 대기하는 상태를 리스트로 표현한 것이다. V 타입은 시스템의 구성 요소에 대한 엔티티의 수에 의미를 두고 있으며 상태변수의 값이 실수로 할당하여 그 구성요소의 상태를 나타내는 상태변수이다. 또한, 시스템의 구성요소를 상태에 대한 의미를 실수값으로 부여하는 것이 아니라 실수가 하나의 표현상 의미를 가지는 상태변수이기도 하다. 예를들면, 대학내

버스 운영 시스템에서 3개의 버스 정류장에 대하여 정수 1, 2, 3으로 표현하는 상태변수를 말한다. E 타입은 엔티티에 부여된 시간값을 나타내는 리스트 형태의 상태변수로서 시간값 $t_i(1 \leq i \leq n)$들은 임의의 확률분포함수를 이용하여 생성한다. T 타입은 시스템 구성요소 역할 시간을 나타내는 잔여시간 상태변수이고 이는 임의의 확률분포함수에 의해 생성되는 값을 갖는다.

【정의 7】 R-DEVS 구조

R-DEVS(Restricted-DEVS)는 입력집합 X, 출력집합 Y, 상태집합 Sr, 시간진행함수 ta, 외부상태전이함수 δext, 내부상태전이함수 δint, 출력함수 λ로 구성된다.

$$COM = \langle X, Y, Sr, ta, \delta ext, \delta int, \lambda \rangle$$

여기서, Sr:R, Q, V, E, T 타입의 곱집합으로 구성되는 상태집합

X, Y, ta, δext, δint, λ : 【정의 1】과 동일

R-DEVS 모델을 SPN 모델로 변환하는 과정은 SPN 구조 N에 대한 구성요소를 R-DEVS 모델의 구성요소로부터 생성하면 되는데 모두 여섯단계를 수행하여 구성한다.

<u>SPN으로의 변환방법</u>

단계 1. 상태집합 s로부터 SPN 모델의 모든 place들을 구성한다.

단계 2. 상태변수 영향도를 구성한다.

단계 3. 외부사건, 내부사건으로부터 SPN 모델의 모든 transition들을 구성한다.

단계 4. 상태변수 영향도를 이용하여 구성된 각 transition에 대한 입력 place 집합, 출력 place 집합 및 inhibitor arc에 연결된 place 집합을 구성한다.

단계 5. 각 transition 점화대기시간 생성을 위한 확률분포함수를 구성한다.

단계 6. 상태집합 s로부터 SPN에서 동일한 상태의 의미를 갖는 초기마킹 μ_0를 명세한다.

각 단계별로 자세한 내용은 다음과 같고 각 단계별 과정에 하나의 CPU와 QUEUE를 갖는 간단한 컴퓨터 시스템에 대하여 다음과 같이 R-DEVS로 구성된 모델을 적용한다.

<u>간단한 컴퓨터 시스템에 대한 R DEVS 모델</u>

1. 입력집합 $X=\{\emptyset, x\}$ 단, x:job이 QUEUE에 도착(QUEUE 도착사건)
2. 상태집합 $s=\langle r_1, n, r_2, \sigma \rangle$ 단, r_1, r_2:확률변수, σ:잔여시간변수
3. 시간진행 함수 $t_a(s)=\sigma$
4. 외부상태전이 함수 $\delta_{ext}(\langle r_1, n, r_2, \sigma \rangle, e, x) = \langle \Gamma_1(r_1), n+1, r_2, \sigma-e \rangle$ 단, e:경과시간변수
5. 내부상태전이 함수 $\delta_{int}(\langle r_1, n, r_2, \sigma \rangle) = \begin{cases} \langle r_1, n, r_2, \infty \rangle & \text{if } n \leq 1 \\ \langle r_1, n, \Gamma_2(r_2), SERVICE\_TIME(r_2) \rangle & \text{if } n \geq 2 \end{cases}$
6. 출력집합 $Y = \begin{cases} \{YES, NO\} & \text{if CPU UTILIZATION=BUSY} \\ R^+ & \text{if QUEUE STATE=LINE} \end{cases}$
7. 출력함수 $\lambda_1(s) = \begin{cases} YES \\ NO \end{cases}$ & $\lambda_2(s) = n(\in R^+)$

여기서, Γ_1:지수분포함수, Γ_2:정규분포함수, SERVICE_TIME: 정규분포함수

단계 1은 SPN의 place 집합 구성을 위한 과정이다. 상태집합 s을 구성하는 상태변수들 중에서 외부사건 발생을 의미하는 R 타입을 제외한 모든 상태변수들을 place로 정의하고 각 place에 대한 의미를 명세한다.

예 4.1 간단한 컴퓨터 시스템에서 상태집합 $s = \langle r_1, n, r_2, \sigma \rangle$임으로 상태변수 n은 QUEUE에 대기 중인 job의 수이고 이를 place p_1으로 명세한다. 상태변수 r_2, σ는 하나의 쌍이 되어 CPU에 대한 이용상태를 나타내므로 각각 place p_2, p_3로 명세한다.

단계 2의 상태변수 영향도는 시스템의 전체 운영과정을 파악하여 외부사건 발생에 대한 피영향자가 되는 상태변수 타입을 선택하고 스케쥴링된 각 내부사건에 대한 영향자와 피영향자에 해당되는 상태변수 타입을 선택한다. 시스템을 떠나는 내부사건이 발생하였을 경우 영향자가 되는 상태변수 타입과 피영향자가 되는 상태변수 타입을 결정한다.

예 4.2 간단한 컴퓨터 시스템에 대한 상태변수 영향도는 다음과 같다.

① 외부사건(QUEUE 도착사건)에 대한 영향도 : 상태변수 n이 피영향자가 된다.

② 내부사건(서비스 시작사건)에 대한 영향도 : 상태변수 n이 영향자, 상태변수 σ가 피영향자가 되며, 상태변수 r_2는 영향자와 피영향자 모두가 된다.

③ 내부사건(CPU 떠남사건)에 대한 영향도 : 상태변수 r_2, σ가 영향자이고 피영향자에 해당하는 상태변수는 존재하지 않는다.

단계 3에서는 SPN의 transition 집합 구성을 위한 과정으로 상태집합 s의 구성요소중 외부사건 발생을 의미하는 R 타입 상태변수와 내부사건 발생을 의미하는 E 타입, T

타입 상태변수로부터 구성한다. 외부사건 발생을 의미하는 R 타입 상태변수를 하나의 source transition으로 명세하고 E, T 타입에 의해 명세되는 내부사건들은 입력 place와 출력 place을 갖는 transition으로 명세한다. sink transition은 시스템의 전체 운영과정으로부터 시스템을 빠져나가는 엔티티가 존재하면 명세하고 그렇지 않으면 명세하지 않는다.

예 4.3 간단한 컴퓨터 시스템에서 상태변수 r_1은 외부사건(QUEUE 도착사건) 발생을 나타내므로 transition t_1으로 명세한다. 스케줄링된 내부사건은 job이 서비스를 받기 위해 QUEUE를 떠나는 사건(서비스 시작사건)이 존재하므로 이를 t_2로 명세한다. 또한, 서비스를 받은 job은 CPU를 빠져나가는 사건(CPU 떠남사건)이 존재하여 이를 t_3로 명세한다. 이때, t_1은 source transition이 되고 t_3는 sink transition이 된다.

단계 4에서는 각 transition에 대한 입력 place 집합을 단계 2의 상태변수 영향도에서 영향자에 해당되는 상태변수를 입력 place 집합의 원소로 결정하고, 출력 place 집합은 피영향자에 해당되는 상태변수를 출력 place 집합의 원소로 결정하고, 출력 place 집합은 피영향자에 해당되는 상태변수를 출력 place 집합의 우너소로 결정한다. 상태변수 영향도에서 R 타입과 T 타입의 쌍이 피영향자가 되는 경우 R 타입의 상태변수를 inhibitor arc에 의해 영결되는 place로 구성하여 억제입력함수 f의 치역으로 생성시킨다.

예 4.4 간단한 컴퓨터 시스템에 대한 입력함수 I, 출력함수 O, 억제입력함수 f는 다음과 같다.

① 입력함수 $I:I(t_1)=\phi(\because t_4:\text{source transition})$, $I(t_2)=\{p_1\}$, $I(t_3)=\{p_2, p_3\}$

② 출력함수 $O:O(t_1)=\{p_1\}$, $O(t_2)=\{p_2, p_3\}$, $O(t_3)=0$ ($\because t_4$: sink transition)

③ 억제입력함수 $f:f(t_2)=\{p_2\}$, $f(t_i)=\phi(i=1,3)$

단계 5에서는 각 transition이 점화가능한 상태에서 점화되기 위해서 대기하는 시간값을 생성하는 확률분포함수를 구성한다. 단계 3으로부터 각 transition이 R-DEVS 모델에서 각 사건발생을 나타내는 확률분포함수를 SPN 모델에서 일치하는 transition에 대한 확률분포함수는 sink transition의 입력 place pi에 대하여 place pi를 출력 place로 간주하는 transition ti의 확률분포함수와 동일하게 구성한다.

예 4.5 간단한 컴퓨터 시스템에 대한 각 transition들의 점화 대기시간 생성을 위한 확률분포함수는 다음과 같다.

R-DEVS		SPN	
사 건	확률분포함수	transition	확률분포함수
QUEUE 도착	지수분포	t_1	지수분포
서비스 시작	정규분포	t_2	정규분포
CPU 떠남	정규분포	t_3	정규분포

〈예 4.5〉 간단한 컴퓨터 시스템에 대한 각 transition들의 점화대기시간 생성을 위한 확률분포함수는 다음과 같다.

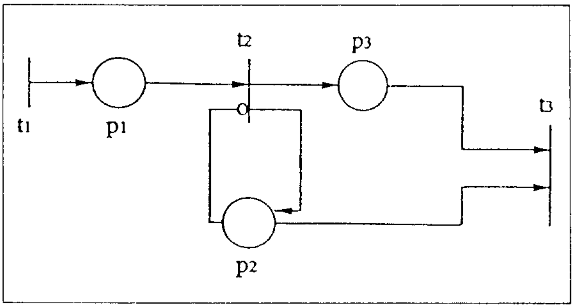
단계 6에서는 SPN 모델의 초기마킹을 생성하는 단계로서 변환하고자 하는 시간 t에서 상태집합 s에 의해 나타나는 시스템 전체상태의 의미를 SPN 모델의 초기마킹 μ_0로 구성한다.

예 4.6 간단한 컴퓨터 시스템에 대하여 기다리는 job이 없고 CPU가 idle 상태일때 변환된 SPN 모델의 초기마킹은 $\mu_0 = (0, 0, 0)$이다.

간단한 컴퓨터 시스템에 대하여 각 단계를 수행한 후 구축되는 SPN 구조와 SPN 그래프는 〈그림 3〉과 같다.

$P=\{p_1, p_2, p_3\}$, $T=\{t_1, t_2, t_3\}$, $I(t_1)=\phi$, $I(t_2)=\{p_1\}$, $I(t_3)=\{p_2, p_3\}$
$O(t_1)=\{p_1\}$, $O(t_2)=\{p_2, p_3\}$, $O(t_3)=\phi$, $f(t_2)=\{p_2\}$, $f(t_i)=\phi(i=1,3)$
t_1 : 지수분포함수, t_2 : 정규분포함수, t_3 : 정규분포함수
$\mu_0=(0, 0, 0)$

(a) 간단한 컴퓨터 시스템에 디한 SPN 구조



(b) 간단한 컴퓨터 시스템에 대한 SPN 그래프

〈그림 3〉 간단한 컴퓨터 시스템의 변환된 SPN 모델

4. SPN을 이용한 타당성 검사

기존의 타당성 검사 방법에 대한 개념을 확장하여 각 실험 프레임별 상태집합 s에서 s'으로의 함수 h에 대하여 본 연구에서는 상태집합 S를 정의구역으로 SPN 모델의 도달성 집합 R(N, μ)를 치역으로 하는 동질함수 h로 다음과 같이 새롭게 정의한다.

【정의 8】 동질함수

동질함수(homogeneous function) h는 다음과 같이 정의한다.

$h : S \rightarrow R(N, \mu)$

$s \rightarrow h(s)$

단, $\exists \mu i \in R(N, \mu)$, $h(s) = \mu i$

〈그림 4〉에서 동질함수 h는 변환하고자 하는 시간 t에서 시스템 상태를 나타내는 s에서 SPN의 마킹으로 사상되는 함수를 의미한다. 함수 δ는 R-DEVS의 외부상태전이함수와 내부상태전이함수를 의미하고 함수 t_f는 초기 마킹 μ_0에서 임의의 transition ti가 점화되어 도달되는 마킹으로의 함수를 의미한다.

$$\begin{tikzcd} \text{상태집합 : } s \arrow[r, dashed, "a"] \arrow[d, dashed, "h"'] & \delta(s) \arrow[d, dashed, "h"] \\ \text{도달성 집합 : } \mu_0 \arrow[r, "b"'] & t\_f(\mu_0) \end{tikzcd}$$

여기서, a:경과시간 e에서 사건발생, b:transition ti 점화,
δ:상태전이함수, t_f:transition 점화함수

〈그림 4〉 상태집합과 도달성 집합의 commutative 다이어그램

【정의 9】 모델의 행위적 동등성

동일한 시스템을 모델링한 모델 M과 모델 M'의 상태표현을 각 s_1, s_2라 가정하고 이들은 의미가 같은 상태를 나타내고 있다고 하자($s_1 \equiv s_2$로 표기), 이때, 의미가 같은 사건 e가 발생하여 시스템에 대한 상태를 모델 M과 모델 M'가 각 s_1',s_2'로 변화된 후 $s_1' \equiv s_2'$가 만족하면 모델 M, M' 행위적으로 동등($M_B \simeq M_B'$)하다.

〈그림 4〉의 commutative 성질과 SPN 모델을 통하여 시스템 상태집합 s에 대해 다음의 성질이 만족되면 모델링된 R-DEVS는 타당하다.

Ⅰ. $t\_f(h(s)) = h(\delta(s))$ 검사

Ⅱ. SPN의 도달성, 생존성 만족도 검사

$t_f(h(s)) = h(\delta(s))$에 대한 증명은 모델의 행위적 동등성(behaviorally equivalence)과 함께 다음 알고리즘에 따라 수행한다.

알고리즘 : commutative 성질 검사

1. 상태집합 s를 구성한다.;
2. 시간진행함수 ta에 의해 다음 발생할 내부사건 혹은 외부사건을 선택한다.; /* 선택된 사건을 e로 표시 */
3. 사건 e가 발생한 후 δ(s)를 명세한다.;
4. 동질함수 h를 정의한다.;
5. 함수 h를 적용하여 s에 대한 SPN의 마킹 μ를 결정한다.;
6. 함수 h를 적용하여 δ(s)에 대한 SPN의 마킹 μ'를 결정한다.;
7. 마킹 μ에서 사건 e와 동일한 의미를 갖는 transition t를 점화하여 t f(μ)로 변화시킨다.; /* 사건 e를 변환방법에 의해 동일한 의미를 가지는 transition t로 구성 */
8. if $\mu' \equiv t\_f(\mu)$ then R DEVS$_B \simeq$ SPN$_B$);
   else not(R DEVS$_B \simeq$ SPN$_B$);
9. if not(R-DEVS$_B \simeq$ SPN$_B$) then/*not satisfy commutative */
   begin
   9.1 R-DEVS 모델링 과정의 검토 및 반복적용;
   9.2 함수 h와 t_f의 재정의 및 재검토;
   9.3 GO TO 1;
   end;
10. STOP.

SPN 모델에 대한 도달성, 생존성에 대한 검사는 초기 마킹 μ_0에서 transition 점화조건에 따라 SPN 그래프에 속하는 임의의 transition ti가 항상 점화가능하여 다른 마킹으로 도달가능함을 보이면 충분하다.

동일한 관점에 의해 시스템 상태표현을 R-DEVS 모델에서는 상태집합, SPN 모델에서는 마킹으로 서로 같은 의미의 상태를 갖도록 변환하였다. 이와 더불어 모델의 행위적 동등성 정의를 이용하여 전통적인 DEVS 모델의 타당성 검사에 필요한 상태전이함수의 보존성 증명을 각 실험 프레임별로 수행할 필요가 없다. 또한, 시간진행함수와 출력함수에 대한 보존성 증명은 모델 변환방법에 의해 변환된 SPN 모델을 해석적으로 분석하는 방법에 내포되어 분석목적에 따라 SPN 모델의 도달성, 생존성이 만족하는지를 검사하면 된다.

5. 결 론

본 연구는 DEVS 모델에 대한 기존의 타당성 검사 방법이 갖는 많은 문제점들을 해결하기 위하여 SPN (stochastic Petri Net)모델을 이용하는 간단하고 효과적인 타당성 검사 방법을 제시하였다. SPN을 이용하기 위하여 DEVS 모델의 행위가 SPN과 동등하도록 모델에 대해 보는 관점을 재조명하여 DEVS 모델이 SPN 모델로 표현됨을 보이는 변환방법을 제안하였다. 모델 변환방법을 토대로 한 새로운 동질함수와 SPN 모델을 이용하여 DEVS 모델에 대한 타당성 검사 방법을 새롭게 제안하였다.

참 고 문 헌

[Law 82] A.M. Law & W.D. Kelton, *Simulation Modelling and Analysis*, McGraw-Hill Book Comp., 1982.

[Moll 82] M.K. Molloy, "Performance Analysis using Stochastic Petri Nets," IEEE Trans, Computer, Vol. C-31, No. 9, pp.913-917, Sep. 1982.

[Mura 89] T. Murata, "Petri Nets: Properties, Analysis and Applications," Proc. of The IEEE, Vol. 77, No. 4, pp. 541-580, Apr. 1989.

[Neel 87] F. Neelamkavil, *Computer Simulation and Modelling*, A Wiley Interscience Pub., 1987.

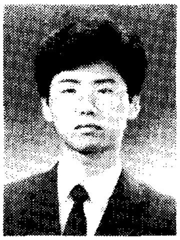
[Pete 81] T.L. Peterson, *Petri Net Theory and The Modeling of Systems*, Englewood Cliffs, NJ: Prentice hall, Inc., 1981.

[Zeig 84] B.P. Zeigler, *Theory of Modelling and Simulation*, A Wiley Interscience Pub., 1984.

[손 91] 손 진곤, "Petri Net과 DEVS 형식론을 이용한 컴퓨터 프로토콜의 모델링", 1991.

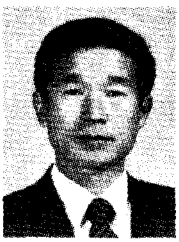
[정 92] 정 영식, 황 종선, 백 두권, "DEVS 모델의 SPN 모델로의 변환이론에 관한 연구", 고려대학교 이학논집, 제 32호, pp.139-149, 1992.

저자소개



정영식

1987년 고려대학교 수학과 졸업
1989년 고려대학교 대학원 전산학 전공(석사학위 취득)
1992년 고려대학교 대학원 전산학 전공(박사과정 수료)
관심분야: 모델링과 시뮬레이션, 분산 시스템, ICAI 등



황종선

1978년 Univ. of Georgia, Statistics & Computer Science 박사
1978년 South Carolina Londer 주립대학 조교수
1981년 한국표준연구소 전자계산실 실장
1982년~1990년 고려대학교 전자계산소 부소장
1986~1989년 한국정과학회 부회장
1982~현재 고려대학교 전산과학과 교수
관심분야: 인공지능, 알고리즘 이론, 분산 시스템, 프로그래밍 언어

백두권

1974년 고려대학교 수학과 졸업
1977년 고려대학교 산업공학과(석사학위 취득)
1977~1980년 대전조급대학 및 전남대학교 공과대학에서 전임강사로 근무
1981~1986년 미국 웨인 주립대학교 전산과학과에서 수학하여, 석사학위(1983) 및 박사학위(1986)를 취득
1986년~현재 고려대학교 전산과학과 조교수로 근무를 시작하여 현재 교수로 재직중
관심분야: 지식베이스, 모델링과 시뮬레이션, 분산 시스템, 소프트웨어 공학, 데이타베이스 등