

형식 도구를 이용한 이산사건 시뮬레이션의 모델 형식화 방법론

정영식*·황종선*·백두권*

A Model Formalization Methodology of Discrete Event Simulation with Formal Tools

Young-Sik Jeong*, Chong-Sun Hwang* and Doo-Kwon Baik*

Abstract

The DEVS(Discrete EVent system Specification) formal model for discrete event simulation is a hierarchical, modular model. Because the DEVS formal model has a mathematical structure, it provides a theoretic background of discrete event simulation model. However, the DEVS formal model is difficult to understand because of its mathematical structure. Also, since the DEVS formal model is often constructed by heuristic, subjective method of model designer from the informal model, a systematic model built-in methodology does not exist.

In this paper, we propose the model formalization methodology from an informal model to the DEVS formal model. For this formalization methodology, we introduce formal tools for model construction based on the DEVS from an informal model : Event Dependency Graph(EDG) for the event analysis and State Representation Graph(SRG) for the system state analysis.

1. 서 론

시뮬레이션은 실 시스템의 효율적인 운영을
도모하기 위하여 실 시스템의 동작을 이해하고

분석, 예측, 평가하는 문제해결 방법이다[5,6,
7.]. 시뮬레이션은 해석적 방법의 적용이 어려
운 문제에 대해서도 적용이 가능하고, 시간이
나 비용측면에서 실현 불가능한 시스템의 결과
예측 및 동적인 시스템 분석을 위한 효과적인

* 고려대학교 전산학과

접근방법이다[4,6,7,12]. 시뮬레이션의 수행단계는 크게 실 시스템의 행위를 정확하게 반영할 수 있도록 타당한 모델을 구축하는 모델링 단계와 이 모델에 의해 의도되는 명령들을 정확히 수행하는 컴퓨터 프로그램을 작성하는 시뮬레이션 단계로 세분화된다[6,7,12].

시뮬레이션 모델은 시간 중심, 상태 중심, 확률변수 중심, 출력범위 중심, 환경영향 중심, 상호작용 규칙 중심 및 모델링 형식론 등의 여러 측면에 따라 분류한다[4,12]. 모델링 형식론에 의한 시뮬레이션 형식모델의 예로는 고정 시뮬레이션의 기본이 되는 미분방정식(differential equation), 일반적으로 컴퓨터 시뮬레이션에 이용되는 이산시간(discrete time), 이산사건(discrete event) 모델 등이 있다. 이산시간 모델은 이산적인 시간상에서 시스템의 행위를 측정하는 모델이고 이산사건 모델은 연속적인 시간상에서 이산적으로 발생하는 사건을 중심으로 시스템의 행위를 분석하는 모델이다[4,6,7,12].

이산시간 모델은 고정된 시간진행으로 인해 그 구간내에 사건이 발생하지 않아도 불필요하게 모델의 행위가 측정되고 특정 시간구간내에서 여러번의 사건이 발생하여도 이를 한 사건이 발생한 것으로 간주함으로써 모든 사건들에 대한 모델의 행위를 측정하지 못하는 단점을 갖는다[1]. 그러나 이산사건 모델은 연속적인 시간스펙트럼상에서 이산적으로 발생하는 사건에 따라 시스템의 행위를 측정함으로써 이산시간 모델링의 문제점을 해결하여 보다 현실적인 모델링을 가능하게 한다. 이산사건 시뮬레이션 모델링은 대기행렬 문제, 불확실적 요소를 갖는 시장예측 문제, 교통체계의 운영문제 및 컴퓨터 시스템 성능평가 문제 등의 많은 응용분

야에서 사용되고 있다[1,6,7].

이러한 이산사건 시뮬레이션 모델 구축에 대한 이론 연구는 Shannon, Zeigler, Oren 등에 의해 진행되어 왔다[2,4,7,11,12]. 특히 Zeigler는 이산사건 시뮬레이션 모델링에 대한 DEVS(Discrete Event system Specification) 형식모델을 제시하여 모델링 및 시뮬레이션 방법론의 형식화를 위한 기반을 제공하였다[11, 12]. DEVS 형식모델¹⁾은 모듈적, 계층적 특성을 가지며, 집합론에 근거한 수학적 형식구조를 제공한다[2,11,12].

일반적으로 이산사건 모델을 손쉽게 구축하기 위해서 이용되는 그래픽 도구는 크게 세 범주로 나누어진다. 첫째, 시뮬레이션 전용언어의 변환을 위한 도구로 Block Diagram, Activity Network 등이 있다[4,6]. 둘째, 시스템을 구성하는 기본요소들을 그래픽 요소들로 대응시키는 도구들로 Activity Cycle Diagram, Entity-Connection Approach, Queueing Model Generator 등이 있다[4,10]. 셋째, 기존의 시스템 분석을 위한 모델링 도구로서 여기에는 상태전이 모델, 논리 모델, 대수 모델, 언어 모델 등이 이용된다. 특히, 시스템의 상태분석은 상태전이 모델로 분류되는 Finite State Machine(FSM), PetriNet(PN), Program Evaluation and Review Technique(PERT), Hierarchy Input Processing Output(HIPO) 및 Data Flow Diagram(DFD) 등의 그래픽 도구가 활용된다[3,8,9,10].

첫번째 범주는 일반 프로그래밍 언어를 위한 순서도(flow chart)의 역할을 벗어나지 못한 문제점을 갖고 있고, 두번째 범주는 시스템의 주요 구성요소에 대한 그래픽 표현이 생략되거나, 특별한 구성요소들 사이의 관계에만 치중

1) 간단히 'DEVS'라고도 한다.

하고 시스템 표현내용이 지나치게 단순하다는 단점을 갖는다. 세번째 범주는 사건이 발생할 때마다 시스템내의 어떠한 구성요소가 상태변화를 일으키고, 상태변화가 발생한 구성요소가 다른 구성요소에 어떠한 영향을 미치는가에 대한 명확한 관계를 도식적으로 나타내지 못한다. 또한 상태분석과 함께 시스템의 상태변화에 영향을 주는 사건과 이 사건들 상호간의 논리적인 관계분석이 요구된다[6,12]. 그러나 이러한 사건분석을 쉽고 용이하게 수행할 수 있는 그래픽 도구가 거의 전무한 실정이다.

한편, DEVS는 그 구조의 수학적 특성 때문에 모델설계자가 이해하기 어렵고, DEVS의 구축을 위해서는 상당한 지식과 경험의 축적을 필요로 한다. 특히 DEVS 모델링에서 이용할 수 있는 상태표현을 위한 그래픽 도구가 개발되어 있지 못하고 시스템의 개별적 구성요소와 관련시켜서 상태를 분석하여야 하기 때문에 기존의 상태표현 도구들을 이용하기가 부적합하다. DEVS의 사건분석은 모델설계자가 특별한 지원도구없이 수학적 표현방법에 의해 수행되었다. 수학적 표현방법을 이용하기 때문에 모델설계자가 사건분석을 수행하기가 상당히 난해하고, 사건분석을 위한 도식적 도구가 없기 때문에 사건분석 결과에 대한 타당성 문제가 야기된다.

DEVS는 실 시스템에 대한 비형식적 서술(비형식적모델)을 토대로 개발되는데, 지금까지의 이러한 비형식모델에서 DEVS로의 변환은 상태분석과 사건분석을 위한 도구가 없기 때문에 모델설계자의 경험적, 주관적 방식에 의해 수행됨으로써 DEVS의 구축 방법은 일관적이고 체계적이지 못하다는 중대한 문제점을 내포하고 있다. DEVS 구축을 위한 체계적인 모델 형식화 방법론의 기여는, 실시스템과 비형식모

델로부터 DEVS를 위한 정보를 명시적으로 추출하지 못하는 데에 기인한다.

따라서, 본 논문에서는 먼저 실 시스템과 비형식적모델로부터 DEVS에 필요한 사건분석과 상태분석을 사건종속 그래프(Event Dependency Graph : EDG)와 상태표현 그래프(State Representation Graph : SRG)에 의해 도식화하고, 이를 토대로 DEVS로 변환하는 단계적이고 체계적인 방법론을 제안한다. 먼저, 제 2장에서는 DEVS와 전통적인 모델 구축방법을 설명하고 제 3장에서는 DEVS를 위한 정보의 명시적 추출을 위해 DEVS에 앞서 구성되어야 하는 EDG와 SRG를 설명한다. 제 4장에서는 EDG와 SRG를 이용하는 단계적이고 체계적인 DEVS구축을 위한 모델 형식화 방법론을 제안, 설명한다.

2. DEVS의 전통적 구축방법

DEVS는 연속적인 시간상에서 이산적으로 발생하는 사건들에 대하여 시스템의 행위를 측정하는 것으로 다음과 같은 형식론에 의해 모델을 표현한다.

[정의 1] DEVS 형식모델

DEVS는 입력집합 X , 출력집합 Y , 상태집합 S , 시간진행함수 t_a , 외부상태전이함수 δ_{ext} , 내부상태전이함수 δ_{int} , 출력함수 λ 로 구성된다.

$$COM = \langle X, Y, S, t_a, \delta_{ext}, \delta_{int}, \lambda \rangle$$

여기서, $t_a : S \rightarrow R^+ \cup \infty$

$$s \rightarrow t_a(s)$$

단, $R^{+0,\infty}$ 는 음수를 제외한 실수집합

$$\delta_{ext} : S \times X \rightarrow S$$

$$(s, x) \rightarrow \delta_{ext}(s, x)$$

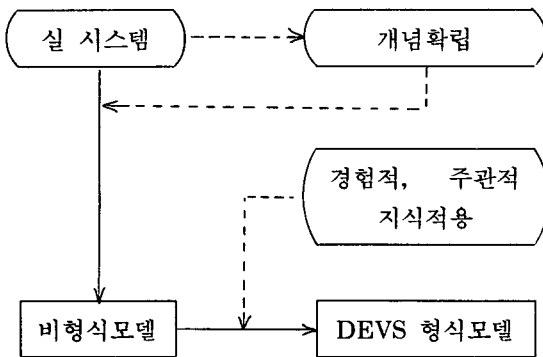
$$\delta_{int} : S \rightarrow S$$

$$s \rightarrow \delta_{int}(s)$$

$$\lambda : S \rightarrow Y$$

$$s \rightarrow \lambda(s)$$

입력집합 X 는 시스템 외부에서 발생하는 사건들의 집합을 의미하고 출력집합 Y 는 출력변수들의 집합을 나타낸다. 상태집합 S 는 상태변수들의 각 정의구역들의 곱집합을 의미하며 상태 $s(s \in S)$ 는 시간진행에 따른 시스템의 순차적인 스냅샷(sequential snapshot) 상태를 의미한다. 시간진행함수 $t_a(s)$ 는 시스템이 외부사건을 입력받지 않는 한, 상태 s 에 머물 수 있도록 허용한 시간으로 정의한다. 외부상태전이함수 δ_{ext} 는 시스템 외부에서 발생한 사건에 의한 모델의 상태변화를 나타내는 함수로 정의하고, 내부상태전이함수 δ_{int} 는 외부사건이 없는 경우 시간진행에 따라 모델의 상태변화를 설명해 주는 함수로 정의한다. 출력함수 $\lambda(s)$ 는 상태 s 에서 시스템의 출력으로 정의한다.



[그림 1] DEVS의 전통적인 구축방법

DEVS를 구축하는 단계는 일반적으로 [그림 1]과 같다[12].

비형식모델(informal model)은 실 시스템으로부터 구성하게 되는데, 시스템의 구성요소(components), 각 구성요소별 서술변수(descriptive variables) 그리고 구성요소간의 상호작용(component interactions)의 세가지로 구성된다[12]. 각 서술변수는 그 정의구역 및 의미가 함께 기술되고, 구성요소 상호작용 동작, 영향, 통신들과 함께 그것들에 관련된 가정, 규칙 등이 비형식적으로 서술한다[12]. [그림 2]는 하나의 QUEUE와 하나의 SERVER를 가진 단일 서버 시스템에 대한 비형식모델의 예를 보여준다.

비형식모델은 설계가 용이한 반면에 일반적으로 불완전하고 일관적이지 못하며, 모호한 특성을 갖기 때문에 이를 직접 컴퓨터 프로그램으로 변환하기에는 어렵다. 이러한 비형식모델은 [정의 1]과 같이 보다 명확한 형식모델로 변환되어야 한다.

DEVS는 전제상태집합 $Q = \{(s,e) | s \in S, 0 \leq t_a(s)\}$ (e : 경과시간)와 시간진행함수 t_a 를 이용하여 시스템에 대한 동적인 표현능력을 갖고 연속적인 시간 스펙트럼상에서 이산적으로 발생하는 사건에 따라 시스템의 행위를 측정함으로써 보다 현실적인 모델링을 가능하게 하는 장점을 갖는다. 또한, DEVS모델은 모듈적, 계층적 특성을 가지며, 집합론에 근거한 수학적 형식구조를 제공하는 것이 장점이다[2,11,12].

한편, DEVS는 그 구조의 수학적 특성 때문에 모델설계자가 이해하기 어렵고, DEVS 모델 구축을 위해서는 상당한 지식과 경험의 축적을 필요로 한다. 비형식모델을 구축한 후 모델설계자의 경험적, 주관적 방식에 의해 수행됨으로써 일관적이고 체계적이지 못하다는 중대한

1. Componets

QUEUE, SERVER

2. Descriptive Variables

QUEUE Descriptive

LINE - with range $R^+ : LINE=n$ means that unumber of jobs in the system.

SERVER Descriptive

SERVICE.TIME-with range $R^+ : a$ random variable assigning time it will take job now first in line to processed.

SERVICE. TIME. LEFT-with range $R^+ : SERVER.TIME.LEFT=\sigma$ means that job being processed will leave SERVER in σ form now.

BUSY-with range {YES, NO} : indicator whether SERVER is serving a job.

BYE-with range $\{\phi, a, b, \dots\} : BYE=\phi$ means no job, $BYE=x$ means job x is leaving.

3. Component Interactions

A job in system at clock time t join the back of the QUEUE.LINE. As prior jobs in LINE are processed, it advances to the fornt of LINE. When it is first in LINE, it samples SERVICE.TIME to get a time σ , and SERVICE.TIME.LEFT is set to σ . The job wait at the head of the LINE until SERVICE.TIME.LEFT becomes 0 : then its departure.

[그림 2] 단일 서버 시스템의 비형식모델

문제점을 내포하고 있다. 체계적인 모델구축 방법론의 결여는 시스템과 비형식모델로부터 DEVS의 정보를 명시적으로 추출하지 못하는 데에 기인한다. DEVS가 보다 널리 이용되기 위해서는 비형식모델에서 DEVS로의 체계적인 구축방법론이 제공되어야 한다.

세되는 이산사건들과 그들 사이의 논리적인 관계에 대한 분석이 요구된다. 그리고 각 사건에 따른 시스템의 전체상태와 각 구성요소의 상태 표현방법과 구성요소들 상호간의 영향에 대한 분석이 필요하다. 본 절에서는 사건분석을 위한 도식적 표현도구로 활용 가능한 그래픽 도구와 사건발생시 시스템의 각 구성요소와 함께 상태표현을 위한 그래픽 도구를 정의한다.

3. 사건분석과 상태분석 도구

실 시스템과 비형식모델로부터 DEVS를 구축하기 위해서는 비형식모델에서 모호하게 명

3.1 사건종속 그래프 : EDG

사건종속 그래프(Event Dependency Graph:

EDG)는 시스템과 비형식모델로부터 추출할 수 있는 사건들과 이러한 사건사이의 관계를 나타내는 방향성 그래프(direct graph)이다.

[정의 2] EDG

사건종속 그래프 G는 사건집합 N_G , 간선집합 E_G , 시간집합 T, 조건집합 C, 사건종속함수 f로 구성된다.

$$G = \langle N_G, E_G, T, C, f \rangle$$

여기서, $N_G = \{n_1, n_2, \dots\} \quad m \geq 0$

$$E_G = \{e_1, e_2, \dots, e_n\} \quad n \geq 0, \quad N_G \cap E_G = \emptyset$$

단, $e_i = \langle n_j, n_k \rangle \quad 1 \leq i \leq m, \quad 1 \leq j, k \leq m$

$$f : E_G \rightarrow T \times C$$

$$e_i \rightarrow f(e_i) = \langle t_k, c_j \rangle$$


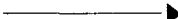
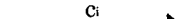
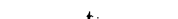

단, $\langle t_k, c_j \rangle$

$$= \begin{cases} \langle t_k, c_j \rangle & \text{if } t_k \neq 0 \ \& \ c_j \neq 0 \\ c_j & \text{if } t_k = 0 \\ t_k & \text{if } c_j = 0 \\ 0 & \text{if } c_j = t_k = 0 \ (j, k \in N) \end{cases}$$

사건종속 그래프 G에서 N_G 는 공집합이 아닌 노드들의 유한집합이고 각 노드는 시스템에서 발생가능한 사건들을 나타낸다. E_G 는 노드들의 쌍 즉, 간선들의 집합을 나타낸다. T는 시간값을 갖는 원소들의 집합이다. 시간값들을 비형식모델의 서술변수로부터 명세되는 확률분포함수에 의해 생성된다. C는 시간값을 제외한 사건발생을 위한 조건들의 집합이다. 비형식모델의 서술변수로부터 각 사건발생시의 상황설정 에 부합되는 정의구역의 값이 조건 c_i 로 결정된다. f는 시스템에서 발생하는 사건사이의 관계를 명세하고 $f(e_i)$ 는 사건종속 그래프 G내에서 간선들의 라벨에 부여된다. 사건종속 그래프 G

의 주요 구성기호는 <표 1>과 같다.

<표 1> EDG의 주요 구성기호

구 성 기 호	의 미
	사건 노드
	무조건 간선
	조건 간선
	시간지연 무조건 간선
	시간지연 조건 간선

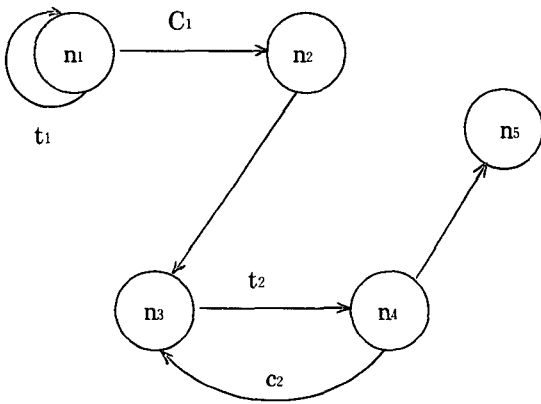
EDG의 무조건 간선은 두 사건사이에서 전체 조건이 없는 것을 나타내며, 사건이 연속적으로 발생함을 의미한다. 모델설계자가 사건명세를 논리적으로 세분화시킴으로써 표현된다. 조건 간선은 한 사건(시작노드)이 발생하고 난 후 다음 사건(도착노드)이 발생하기 위해서는 시간조건을 제외한 어떤 특정한 조건이 만족하여야 함을 나타낸다. 시간지연 무조건 간선은 두 사건사이의 논리적인 관계가 시간값과 관련되는 것으로 한 사건이 발생하고 난 후 다음 사건이 발생하려는 일정한 시간이 지나야 함을 나타낸다. 시간지연 조건 간선은 한 사건이 발생한 후 다음 사건이 발생하려면 일정한 시간이 지나고 주어진 조건이 만족되어야 함을 나타낸다.

DEVS 모델설계자는 DEVS 개발에 앞서, EDG에 의해 시스템에서 발생 가능한 사건들, 사건간 논리적인 관계 및 전체적인 사건 순서(event ordering)를 명시적으로 나타낼 수 있다. 또한 EDG는 방향성 그래프이기 때문에 간선 $\langle n_i, n_j \rangle$ 와 $\langle n_j, n_i \rangle$ 는 서로 다른 간선을 나

타내고, 간선 $\langle n_i, n_j \rangle$ 는 그 간선의 시작노드와 도착노드가 같은 의미를 갖는 자기간선(self edge)을 말한다.

EDG는 모델설계자가 비형식모델로부터 DEVS 형식모델을 개발하고자 할 때 상태분석 도구와 함께 사용된다. 즉, DEVS의 입력집합과 상태집합을 도출하는 도구로 활용된다.

[그림 3]은 단일 서버 시스템에 대한 EDG를 나타낸다. 비형식모델로부터 EDG를 구성하는 방법은 4.2절에서 자세히 설명한다.



여기서, n_1 : QUEUE 도착사건,
 n_2 : QUEUE 떠남사건,
 n_3 : 서비스 시작사건,
 n_4 : 서비스 완료사건,
 n_5 : SERVER 떠남사건
 $e_1 = \langle n_1, n_1 \rangle, e_2 = \langle n_1, n_2 \rangle, e_3 = \langle n_2, n_3 \rangle$
 t_1 : job 도착간격 시간경과, t_2 : SERVER의 서비스 시간경과
 c_1 : BUSY = No, c_2 : $n \geq 2$ (n : QUEUE에 대기중인 job의 수)
 $f(e_1) = t_1, f(e_2) = c_1, f(e_3) = \phi, f(e_4) = t_2, f(e_5) = c_1, f(e_6) = \phi$

[그림 3] 단일 서버 시스템의 EDG

3.2 상태표현 그래프 : SRG

상태표현 그래프(State Representation Graph : SRG)는 개별적인 사건발생에 따라 시스템의 어떠한 구성요소가 상태변화를 일으켜서 시스템내 다른 구성요소에 어떻게 영향을 주는가를 명세하는 그래프이다.

[정의 3] SRG

상태표현 그래프 G 는 구성요소집합 COM_G , 유한상태집합 S_G , 사건집합 N_G , 간선집합 E_G , 입력함수 I , 출력함수 O , 상태관계 함수 f 로 구성된다.

$$G = \langle COM_G, S_G, N_G, E_G, I, O, f \rangle$$

여기서, $COM_G = \{com_1, com_2, \dots, com_p\} \quad p \geq 0$

$$S_G = \{s_1, s_2, \dots, s_q\} \quad q \geq 0$$

$$N_G = \{n_1, n_2, \dots, n_m\} \quad m \geq 0$$

$$E_G = \{e_1, e_2, \dots, e_k\} \quad k \geq 0, S_G \cap N_G = \phi,$$

$$S_G \cap E_G = \phi, E_G \cap N_G = \phi$$

$$\text{단, } e_i = \begin{cases} \langle s_i, s_j \rangle & \text{if } \exists s_i \ \& \ \exists s_j \\ \langle \phi, s_j \rangle & \text{if } \exists s_j \\ \langle s_i, \phi \rangle & \text{if } \exists s_i \ (1 \leq i, j \leq k) \end{cases}$$

$$I : S \rightarrow N_G$$

$$s \mapsto I(s)$$

$$O : S \rightarrow N_G$$

$$s \mapsto O(s)$$


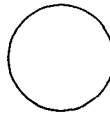

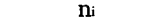
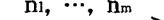
$$f : E_G \rightarrow \prod_{i=1}^n N_i \ (\text{단, } N_i = N_G)$$

$$e_i \mapsto f(e_i)$$

상태표현 그래프 G 에서 COM_G 는 시스템의 구성요소집합으로 정의한다. S_G 는 시스템의 각 구성요소에서 나타날 수 있는 상태노드 집합이고, N_G 는 시스템에서 발생가능한 사건집합이

다. E_G 는 상태노드의 쌍 즉, 간선들의 집합을 나타낸다. I 는 시스템의 구성요소내 상태노드에 입력되는 사건으로의 함수로 정의하고 O 는 시스템의 구성요소내 상태노드에서 출력되는 사건으로의 함수이다. $f(\langle s_i, s_j \rangle)$ 는 간선에 어떤 사건이 관련되는가를 나타내기 위한 함수로써 해당 간선에 라벨링한다. 상태표현 그래프 G 의 주요 구성기호는 <표 2>와 같다.

<표 2> SRG의 주요 구성기호

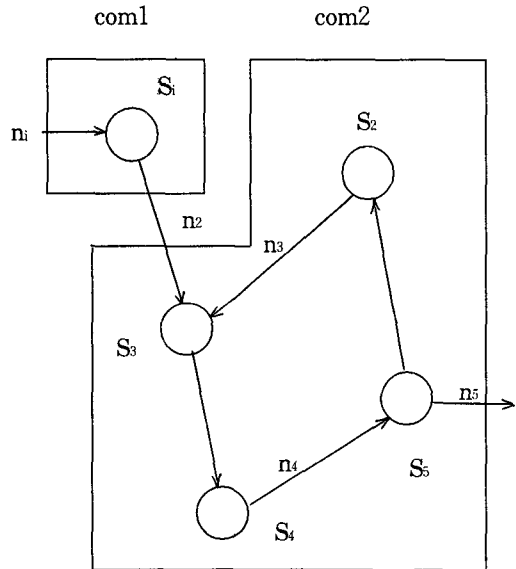
구 성 기 호	의 미
COM	
	구성요소
STATE	
	상태노드
	공사건 발생 간선
	사건 n_i 의 발생 간선
	다중사건 발생 간선

SRG에서 시스템의 구성요소는 사각형으로 나타내고 각 구성요소내 상태는 원으로 나타내며, 상태들간의 연결은 간선으로 나타내는데 이는 각 구성요소내에 관련되는 사건발생을 의미한다. 공사건 발생 간선은 각 구성요소내의 상태변화가 사건발생에 의한 것이 아니라 논리적으로 상태변화가 발생될 때 상태노드간 연결시 사용되는 간선이다. 다중사건 발생 간선은 여러 사건이 동시에 관련될 때 사용되는 간선이다.

EDG와 SRG가 같은 비모형식모델로부터 구

성된다면, 두 그래프에서 명세되는 사건들의 의미와 표현방법은 동일해야만 한다. <그림 3>에서 n_1, n_2, n_3, n_4, n_5 의 사건들은 SRG에서 상태전이에 영향을 주는 사건들로 서로 동일한 의미와 표기법으로 사용된다. SRG는 EDG함께 DEVS 모델의 입력집합과 상태집합의 요소들을 결정하는데 직접적인 지원도구로 활용된다.

[그림 4]는 단일 서버 시스템에 대한 SRG를 나타낸다. 비형식모델로부터 SRG를 구성하는 방법은 4.2절에서 자세히 설명한다.



여기서, $com_1, com_2 = \text{SERVER}$

$S_1 = \text{대기상태}, S_2 = \text{SERVER idle상태},$

$S_3 = \text{서비스 시작상태},$

$S_4 = \text{SERVER busy 상태},$

$S_5 = \text{서비스 완료상태}$

$e_1 = \langle \emptyset, s_1 \rangle, e_2 = \langle s_1, s_3 \rangle, e_3 = \langle s_2, s_3 \rangle,$

$e_4 = \langle s_3, s_4 \rangle, e_5 = \langle s_4, s_5 \rangle, e_6 = \langle s_5, s_2 \rangle,$

$e_7 = \langle s_5, \emptyset \rangle$

$I(s_1) = n_1, I(s_3) = n_2, I(s_i) = \emptyset (i=2,4,5)$

$O(s_1) = n_2, O(s_5) = n_5, O(s_i) = \emptyset (i=2,3,4)$

$f(e_1) = n_1, f(e_2) = n_2, f(e_3) = n_3, f(e_4) =,$

$\emptyset, f(e_5) = n_4, f(e_6) = \emptyset, f(e_7) = n_5$

[그림 4] 단일 서버 시스템의 SRG

4. DEVS 모델의 형식화 방법론

4.1 방법론 개요

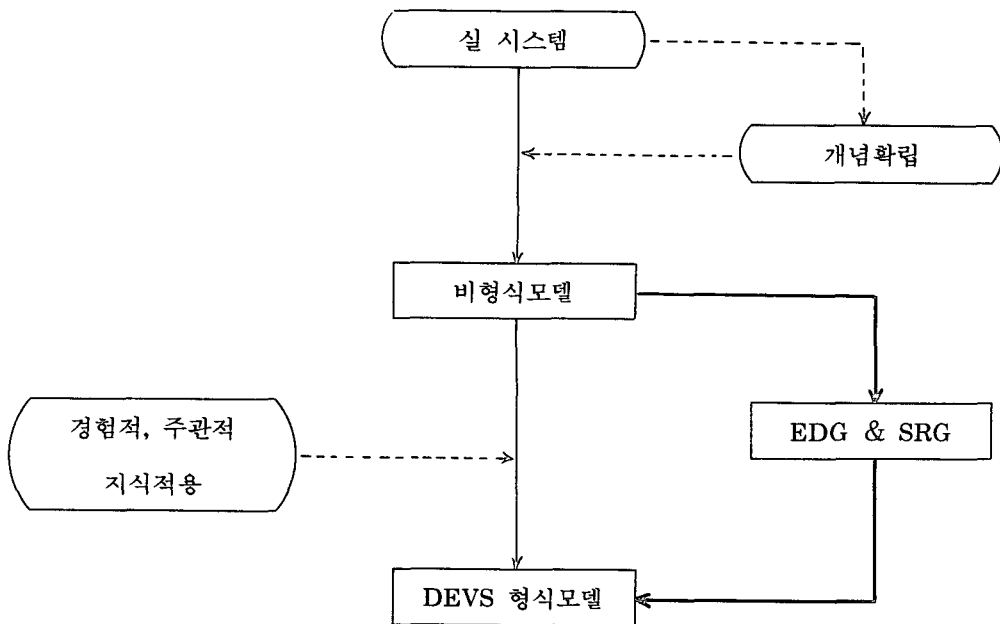
기존의 DEVS 구축방법은 실 시스템으로부터 비형식모델을 구축하고 비형식모델로부터 직접 DEVS를 구축한다. 그러나 본 논문에서는 비형식모델로부터 DEVS를 구성하기에 앞서, 먼저 3장에서 설명한 그래픽 도구를 이용하여 DEVS 정보를 명시적으로 도출한 후, 이로부터 DEVS 구성을 체계적으로 모델링한다. [그림 5]는 본 논문의 접근방법론과 기존의 접근방법론을 비교하고 있다. 체계적인 DEVS 구축을 위한 접근방법론은 다음의 세 과정으로 구성된다.

과정 1: 실시스템으로부터 비형식모델을 구축한다.

과정 2: 비형식모델로부터 사건종속 그래프와 상태표현 그래프를 구성한다.

과정 3: 사건종속 그래프와 상태표현 그래프로부터 DEVS를 구성한다.

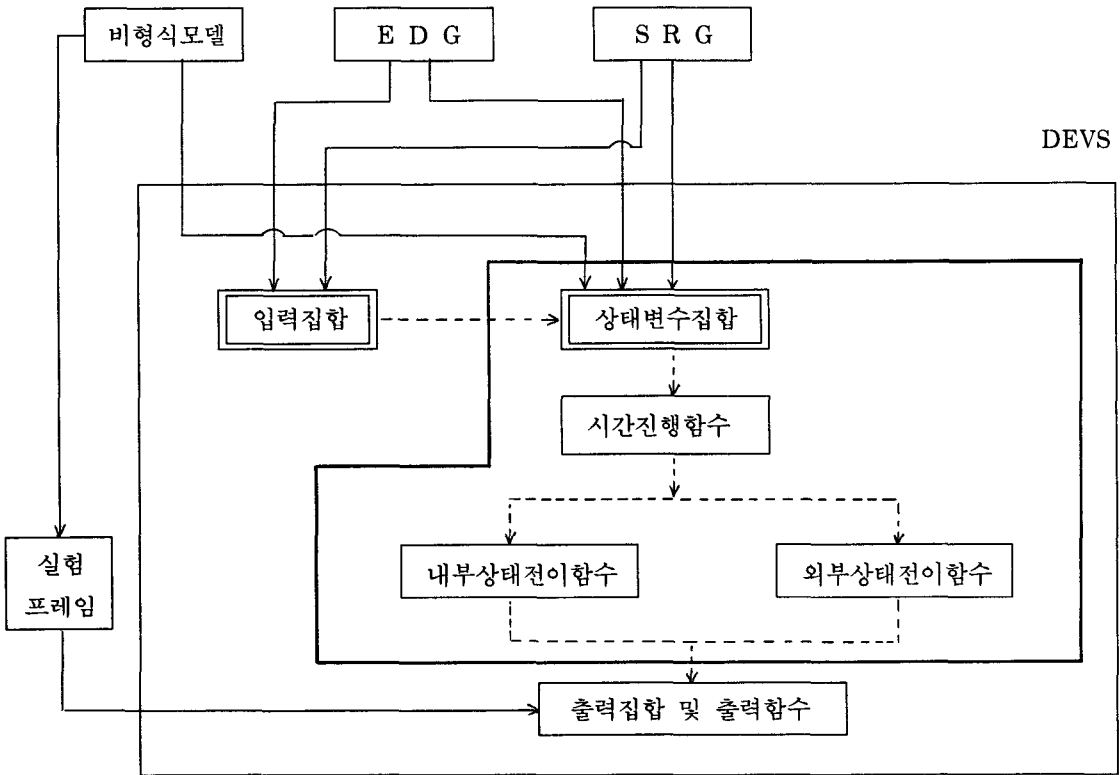
단계 1에서는 2절에서 살펴보았듯이 DEVS 형식모델의 전통적인 구축 방법과 동일하게 비형식모델을 구축한다. 단계 2에서는 DEVS 구축을 위해 필요한 정보를 명시적으로 표현하고 비형식모델로부터 그 정보를 추출하여 사건종속 그래프와 상태표현 그래프를 구성한다. 단계 3에서는 비형식모델로부터 도출된 EDG와 SRG로부터 체계적이고 도식적인 방법에 의해 DEVS를 구축한다.



[그림 5] 형식도구에 의한 DEVS 형식화 방법론

비형식모델, EDG, SRG 및 DEVS 구성요소 사이의 관련성은 [그림 6]과 같고 비형식모델의 서술변수, 구성요소 상호작용과 EDG, SRG

에 명시적으로 표현된 정보들을 이용하여 DEVS를 체계적으로 구축한다.



→ : 지원도구, --->: DEVS 요소관계

[그림 6] EDG와 SRG에 의한 DEVS 모델링

4.2 EDG와 SRG 구성방법

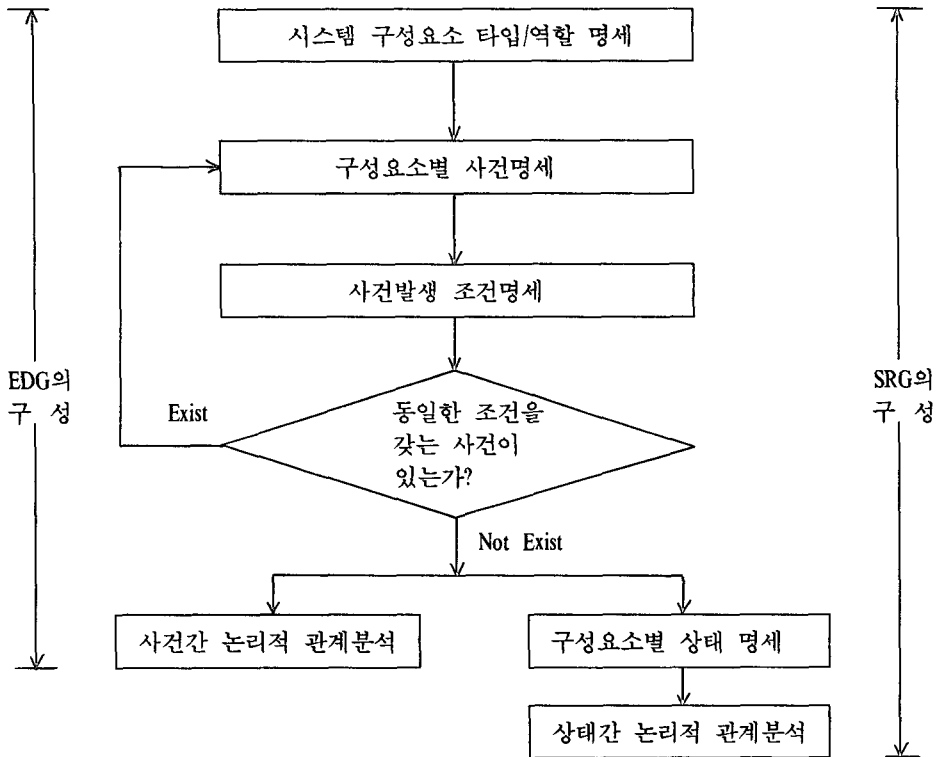
[그림 7]과 같은 단계적인 분석과정에 의해 EDG와 SRG를 도출한다. 시스템 구성요소 타입/역할 명세, 구성요소별 사건명세 및 사건발생 조건명세는 EDG와 SRG 구성을 위한 공통적인 단계로서 다음에 설명하고, 그 이후의 단계들은 (가) EDG 구성과 (나) SRG 구성에

서 설명한다.

단계 1: 시스템 구성요소 타입/역할 명세

이 단계는 시스템의 구성요소들을 능동타입(active type)과 수동타입(passive type)으로 분류하고 시스템에서 각 구성요소들의 역할을 명세한다.

시스템의 구성요소들을 활동능력(acting po-



[그림 7] EDG와 SRG 구성을 위한 분석과정

wer)의 유무에 따라 능동타입과 수동타입으로 분류한다. 능동타입은 구성요소 자신의 역할을 수행하기 위한 시간과 구성요소내에서 시스템에 입력되는 엔티티가 필요한 시간동안 활동하고 이웃하는 구성요소에 상태변화를 야기시키고 활동할 수 있는 원인을 제공하는 타입이다. 수동타입은 외부사건 혹은 이웃하는 구성요소들에 의해 자신의 상태가 변화되는 타입으로, 자신의 역할 수행을 위한 어떠한 활동도 수행하지 않는다.

단일 서버 시스템으로 구성요소 QUEUE는 job이 서비스를 받기 위해서 시스템으로 입력될 경우 자신의 상태 즉, 기다리는 job의 수가 하나 증가되고 QUEUE 자신의 역할 수행

을 위한 활동은 하지 않기 때문에 수동타입이다. 구성요소 SERVER는 시스템에 입력되는 job을 처리하는 요소이며, 자신의 역할 수행을 위한 시간동안 활동함으로써 능동타입이다.

단계 2 : 구성요소별 사건명세

이 단계는 시스템의 각 구성요소와 관계되는 사건을 명세한다.

사건명세는 각 사건들의 의미로부터 외부 및 내부사건을 결정한다. 비형식모델로부터 단지 시스템에 엔티티를 받아들이고 빠져나가는 중간매개 역할만을 수행하는 시스템의 입구, 출구와 같은 구성요소에 대한 사건명세는 하지 않는다. 명세된 각 사건들을 EDG의 사건노드

가 되며 SRG에서는 상태전이의 원인이 되는 것으로서 간선의 라벨이 된다. 엔티티들이 시스템에 입력되어 어떤 구성요소들 사이의 상호작용을 통하여 출력되는지에 대한 분석을 토대로 관련되는 사건들은 순차적으로 인덱싱하여 표현한다.

단일 서버 시스템에서 수동타입인 QUEUE와 관련되는 사건은 QUEUE 도착사건(n_1) QUEUE 떠남사건(n_2)이고 능동타입인 SERVER에 관련되는 사건은 서비스 시작사건(n_3), 서비스 완료사건(n_4), SERVER 떠남사건(n_5)이다. 사건 n_1 만 외부사건이고 이외의 모든 사건들은 내부사건이다.

단계 3 : 사건발생 조건명세

이 단계는 명세된 각 사건들의 사건발생 조건을 명세한다.

이 단계에서는 각 구성요소에 명세된 각 사건이 어떤 경우에 발생하는지를 분석하여 각 사건이 발생하기 위한 조건을 명세한다. 외부사건으로 정의되는 사건은 구성요소의 서술변수와 시스템의 구성요소 타입/역할로부터 특정한 시간 t 에서 사건발생 상황이 설정되는데, 그 상황을 나타내는 서술변수내 정의구역의 값을 사건발생 조건으로 명세한다. 내부사건의 조건은 일반적으로 음이 아닌 양의 실수값을 갖는 시간지연 조건과 사건이 발생할 수 있는 상황설정으로 부터 구성된다. 각 사건발생 조건이 동일한 사건이 존재하면 같은 의미의 사건을 뜻하기 때문에 하나의 사건으로 통합시키고 같은 조건이 없는 사건만이 명세될 때까지 단계 2부터 반복수행한다.

단일 서버 시스템에 대한 사건발생 조건은 n_1 이 job 도착간격 시간경과, n_2 가 BUSY : NO, n_3 가 $n \geq 2$, n_4 가 SERVER의 서비스 시간경과

이고, n_5 의 조건은 존재하지 않는다.

(가) EDG 구성

비형식모델로부터 EDG의 도출을 [그림 7]에서 보는 바와같이, 앞서 설명한 세 단계 외에 사건간 논리적인 관계분석을 필요로 한다.

단계 4 : 사건간 논리적 관계분석

이 단계는 시스템에서 발생하는 사건사이의 논리적인 관계를 명세한다.

이 단계는 두 사건간 관계에 대한 존재 유무를 시스템의 구성요소 상호작용과 사건명세로부터 파악하고 단계 3의 사건발생 조건명세와 관련시켜서 EDG 간선의 라벨에 부여한다.

두 사건간 관계성이 존재하는 갯수는 사건발생 조건의 수보다는 항상 크거나 같고 사건의 인덱싱이 단계 2에서 순차적으로 결정되어 기본적으로 연속적인 사건간에는 관계가 존재한다. 사건사이의 관계성이 존재하면서 사건발생 조건이 없는 경우(\emptyset)는 연속적으로 사건이 발생됨을 의미한다. 두 사건간 관계분석의 횟수는 최악의 경우 명세된 사건 수의 제곱수만큼 수행하게 된다.

단일 서버 시스템에 대한 사건간 관계분석은 사건 n_1 은 사건 n_1 이 발생한 후 job 도착간격 시간이 경과하면 발생하고 사건 n_2 는 사건 n_1 이 발생한 후 SERVER의 상태가 idle일 때 발생한다. 사건 n_3 는 사건 n_2 가 발생한 후 즉시 발생하고 사건 n_4 가 발생한 후 조건 $n \geq 2$ 가 만족하면 발생한다. 사건 n_4 는 사건 n_3 가 발생한 후 SERVER의 서비스시간이 경과하면 발생한다. 사건 n_5 는 사건 n_4 가 발생한 후 조건없이 발생한다.

구성요소별 사건명세로부터 EDG의 노드를 도출하고 두 사건간 관계가 존재하고 한 사건

이 발생한 후 다음 사건이 어떤 조건하에서 발생하는지를 단계 3과 단계 4로부터 간선과 라벨을 도출한다. EDG는 네단계의 분석과정을 수행함으로써 구성되며, [그림 3]은 본 구성단계를 통하여 작성된 그래프이다.

단일 서버 시스템에 대한 EDG의 노드는 단계 1,2로부터 n_1, n_2, n_3, n_4, n_5 가 되고 간선은 단계 4로부터 $e_1 = \langle n_1, n_1 \rangle, e_2 = \langle n_1, n_2 \rangle, e_3 = \langle n_2, n_3 \rangle, e_4 = \langle n_3, n_4 \rangle, e_5 = \langle n_4, n_3 \rangle, e_6 = \langle n_4, n_5 \rangle$ 가 된다. 각 간선의 라벨은 단계 3과 단계 4로부터 e_1 에는 job 도착간격 시간경과, e_2 에는 BUSY=NO, e_4 에는 SERVER의 서비스 시간경과, e_5 에는 $n \geq 2$ 가 되고 e_3, e_6 는 연속적인 사건발생을 나타내기 때문에 라벨을 부여하지 않는다.

(나) SRG 구성

비형식모델로부터 SRG의 도출은 시스템 구성요소 타입/역할 명세, 구성요소별 사건명세, 사건발생 조건명세의 세 단계외에 구성요소별 상태명세와 상태간 논리적 관계분석을 필요로 한다.

단계 4 : 구성요소별 상태명세

이 단계는 각 구성요소에서 나타나는 상태를 명세한다.

구성요소별 상태명세는 비형식모델의 시스템 구성요소 역할과 구성요소 상호작용으로부터 각 구성요소에서 나타날 수 있는 상태를 명세한다. 단계 2로부터 사건명세가 있지 않은 구성요소에 대해서는 상태를 명세하지 않는다.

단일 서버 시스템의 구성요소 QUEUE는 대기상태, SERVER는 SERVER idle상태, 서비스 시작상태, SERVER busy상태, 서비스 완료상태가 명세된다.

단계 5 : 상태간 논리적 관계분석

이 단계는 기본적으로 단계 4에서 명세된 임의의 한 상태가 어떤 사건과 관련되어 다음 상태로 변화되는가를 파악하는 단계로서 상태전이 명세와 상태전이에 관련되는 사건을 결정하는 과정으로 세분화한다. 상태전이는 비형식모델의 구성요소 상호작용과 단계 1로부터 구성요소의 역할과 각 구성요소에 나타나는 상태가 어떠한 순서로 전이되는가를 파악한다. 상태전이에 관련되는 사건의 결정은 구성요소별 사건 명세와 사건발생 조건명세로부터 어떤 사건(입력사건)이 발생함으로써 각 구성요소별로 명세된 상태중 어떤 것이 되고, 각 구성요소별로 명세된 상태중에서 어떤 상태에서 어떠한 사건(출력사건)이 발생하는지를 결정한다.

단일 서버 시스템에서는 구성요소 QUEUE를 거쳐서 SERVER에서 서비스를 받고 시스템을 벗어나고 상태전이는 QUEUE 대기상태에서 SERVER의 서비스 시작상태, SERVER busy상태, 서비스 완료상태, SERVER idle상태로 전이되고 SERVER는 다시 QUEUE 떠남 사건이 발생함에 따라 서비스 시작상태에서 같은 순서에 의해 상태가 반복 전이된다. <표 3>은 단일 서버 시스템에 대한 입력, 출력사건을 명세한 것이다.

<표 3> 단일 서버 시스템에 대한 입출력 사건명세

상 태	입력사 건	출력사 건
대기상태	n_1	n_2
SERVER idle상태	ϕ	n_3
서비스 시작상태	n_2, n_3	ϕ
SERVER busy상태	ϕ	n_4
서비스 완료상태	n_4	n_5

〈표 3〉에서 사건 n_2, n_3 가 모두 발생하여야 서비스 시작상태가 된다. 시스템의 구성요소들로부터 SRG의 사각형을 도출하고 명세된 상태들을 SRG의 각 구성요소내 상태노드로 구성한다. 상태전이 명세를 이용하여 상태간 연결을 간선으로 나타내며, 각 간선에 대한 라벨은 상태전이에 관련되는 입출력사건을 부여한다. 사건발생에 의한 상태전이가 아니고 단순한 상태변화만을 의미하면 간선의 라벨을 부여하지 않는다.

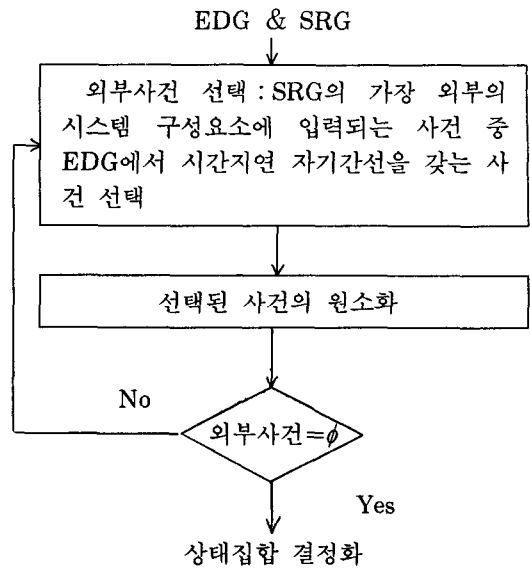
SRG는 다섯단계의 분석과정을 수행함으로써 구성되며, [그림 4]는 본 구성단계를 통하여 작성된 그래프이다. 단일 서버 시스템에 대한 SRG의 사각형은 QUEUE와 SERVER가 되고 단계 4로부터 QUEUE내에서는 대기상태, SERVER내에서는 서비스 시작상태, SERVER busy상태, 서비스 완료상태, SERVER idle 상태가 상태노드가 구성된다. 단계 5의 상태전이 명세에 의해 상태간 연결이 간선으로 구성되고 입출력 사건명세에 의해 n_1, n_2, n_3, n_4, n_5 가 간선의 라벨로 부여된다.

4.3 DEVS로서의 변환방법

[그림 6]으로부터 DEVS의 각 요소를 결정하기 위한 방법은 다음과 같다. 각 단계별 분석과정에 단일 서버 시스템을 적용하여 DEVS가 구축됨을 보인다.

4.3.1 입력집합 결정방법

EDG와 SRG로부터 외부사건들을 원소로 하는 입력집합은 [그림 8]과 같은 분석과정을 통하여 결정한다. 입력집합에는 공사건 \emptyset 을 항상 포함한다.



[그림 8] 입력집합 분석과정

단일 서버 시스템에서는 [그림 3]의 EDG와 [그림 4]와 SRG로부터 외부사건선택 단계에서 QUEUE 도착사건 n_1 이 선택되고 입력집합 X의 원소로 확정한다. n_1 이외의 외부사건이 더 이상 존재하지 않고 시간간격에 의해 계속 같은 의미의 사건이 발생함으로써 외부사건 n_1 이 입력집합을 대표하는 원소로 설정되어 입력집합 $X = \{\emptyset, n_1\}$ 이 된다.

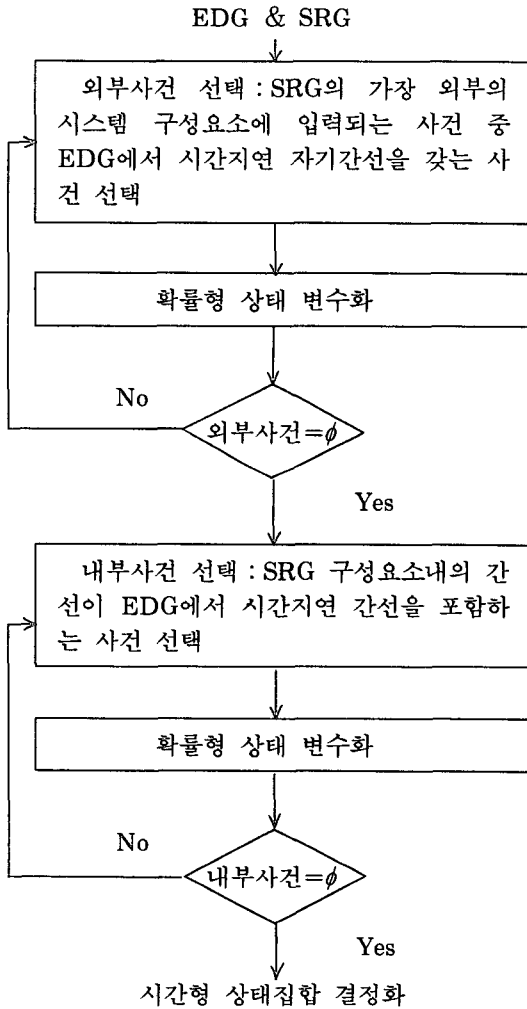
4.3.2 상태집합 결정방법

상태집합은 상태변수들의 각 정의구역들의 곱집합을 의미한다[12]. 상태변수들은 일반적으로 다음의 세가지 타입으로 분류한다[12].

- 1) 확률형 상태변수 : 시스템에서 발생하는 모든 사건들의 발생을 나타내기 위한 변수
- 2) 시간형 상태변수 : 시스템의 구성요소 자신의 활동시간에 대한 잔여시간 변수 (time left variables)와 시스템 전체상태 경과에 대한 경과시간 변수(elapse time)

variables)

- 3) 서술형 상태변수 : 특정 시간 t 에서 시스템의 각 구성요소에 대한 상황(status)을 나타내는 변수



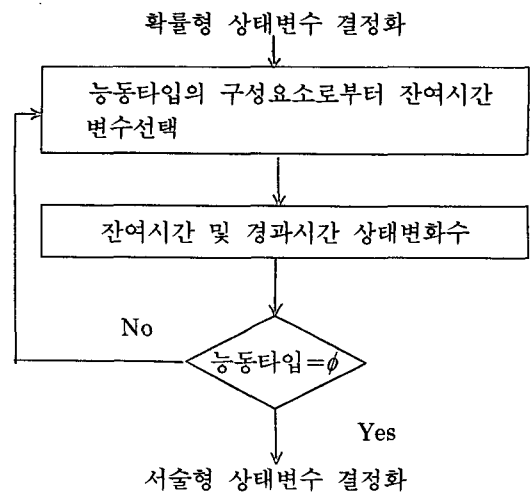
[그림 9] 확률형 상태변수 분석과정

시스템에서 시간의 흐름을 나타내는 변수 중에서 시간형 상태변수에는 잔여시간 변수와 경과시간 변수만을 포함하고 시스템의 구성요소에 입력되는 엔티티에 부여된 시간진행을 나타

내는 변수는 서술형 상태변수에 포함한다[16]. 예를 들면, 슈퍼마켓 시스템에서 슈퍼마켓의 쇼핑지역에 소비자가 쇼핑하는 시간을 쇼핑지역에 도착하는 사건발생때 부여받은 것을 나타내는 서술변수는 $\{(q_1, \tau_1), (q_2, \tau_2), \dots, (q_n, \tau_n)\}$ 이다. 이때, q_i 는 각 엔티티들이고 τ_i 는 해당 소비자에게 쇼핑시간이 부여된 시간을 나타내는 확률변수이다.

확률형 상태변수는 [그림 9]에서와 같이 EDG와 SRG로부터 외부사건을 선택하여 확률형 상태변수를 정의하고 SRG 구성요소내의 간선이 EDG에서 시간지연 간선을 포함하는 경우 그 시간이 구성요소 자신의 활동 시간이고 시작노드에 해당하는 내부사건을 선택하여 확률형 상태변수를 정의한다.

단일 서버 시스템에서 외부사건 선택으로부터 job 도착사건 발생을 나타내는 외부사건을 r_1 으로 명세한다. 내부사건 선택으로부터 SERVER에서 job이 서비스를 받는 사건발생을 나타내는 서비스 시작사건 m_3 가 선택되고 다음 과정에서 r_2 로 확률형 상태변수화 된다.



[그림 10] 시간형 상태변수 분석과정

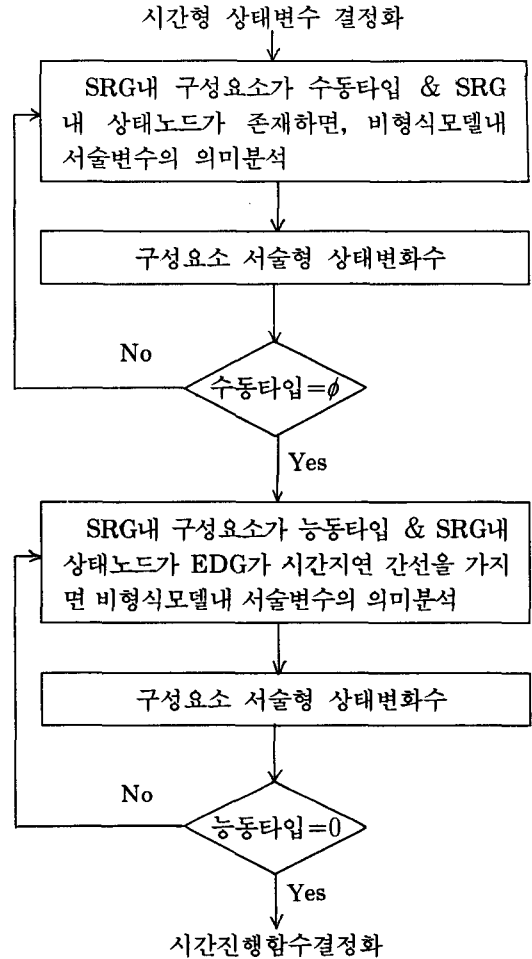
[그림 10]에서 능동타입의 구성요소로부터 잔여시간 변수는 SRG 구성요소내의 간선이 EDG에서 시간지연 간선을 포함하고 그 시간이 구성요소 자신의 활동 시간인 것을 선택한다. 잔여시간 변수의 기호는 일반적으로 σ 를 사용하고 σ 의 수는 외부사건발생을 나타내는 확률형 상태변수의 수를 제외한 내부사건 발생을 위한 확률형 상태변수의 수와 일치시킨다.

단일 서버 시스템에서 SRG의 구성요소내 EDG의 시간지연 간선이 구성요소 자신의 활동 시간을 위한 것이고 그것의 수가 하나이기 때문에 시간형 상태변수는 σ 로 결정한다. 시스템 전체에 대한 경과시간 변수는 대개 e 로 표현한다.

[그림 11]의 수동타입에 대한 서술변수의 의미분석은 비형식모델의 구성요소 서술변수에서 모델설계자의 관점에 따라 약간 상이한 의미와 정의구역을 사용한다. 예를 들면, 일반적인 QUEUE에 대한 치역을 $n \in \mathbb{Z}, \{\emptyset, a, b, \dots\}$ 로 각각 정의하고 이들의 의미는 각 큐내 대기중인 엔티티 수, 큐내 대기중인 엔티티 리스트의 의미를 갖는다. 능동타입에 대한 서술변수의 의미분석은 시간형 상태변수를 제외한 시간변수를 갖는 구성요소에 대한 의미파악을 수행한다. 각 타입에 적합한 서술형 상태변수화는 구성요소 서술변수의 의미로부터 특정 시간 t 에서 정의구역의 값을 상태변수화시킨다.

단일 서버 시스템에서의 서술형 상태변수는 QUEUE가 수동타입이고 시간 t 에서의 상태서술이 QUEUE에 대기중인 job의 수로 나타나기 때문에 $n(\in \mathbb{R}^+)$ 이 된다. 능동타입에 대한 서술형 상태변수는 존재하지 않는다.

[그림 9], [그림 10], [그림 11]의 분석과정으로부터 단일 서버 시스템의 상태변수는 r_1, n, r_2, σ 이 된다. 각 상태변수의 정의구역들의



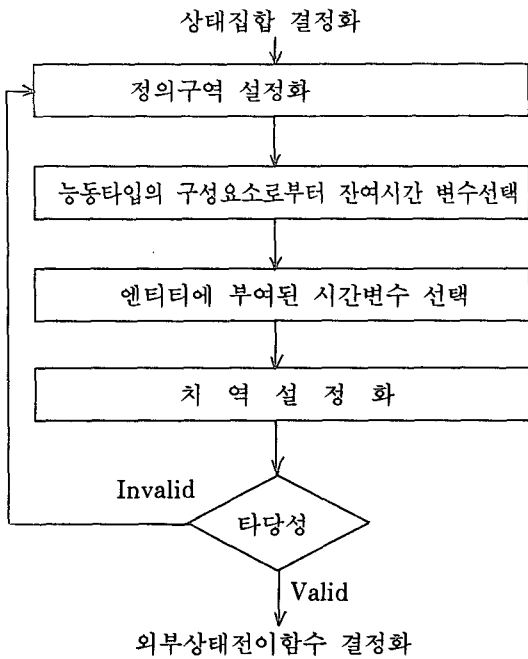
[그림 11] 서술형 상태변수 분석과정

곱집합이 상태집합임으로 상태집합 s 는 $\langle r_1, n, r_2, \sigma \rangle$ 이 된다.

4.3.3 시간진행함수의 결정방법

상태집합 s 에서 상태유지 시간으로의 시간진행함수 t_s 를 결정하기 위한 분석과정은 [그림 12]와 같다.

엔티티에 부여된 시간변수는 SRG 구성요소내의 간선이 구성요소내의 간선이 EDG에서 시간지연 간선을 갖고 그 시간이 엔티티에 부



[그림 12] 시간진행함수 분석과정

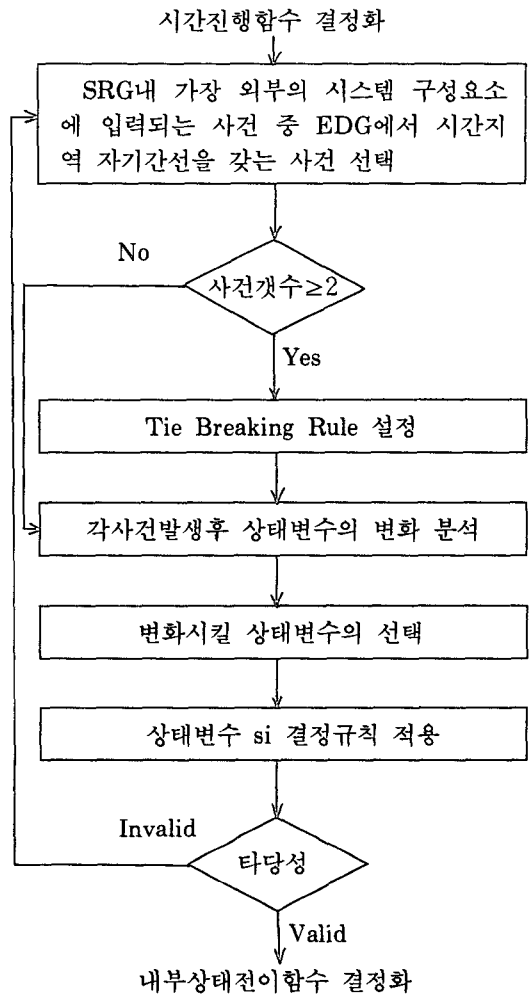
여된 시간인 경우의 시간변수를 선택한다. 상태집합이 t_a 의 정의구역이 되고 잔여시간 변수들과 엔티티에 부여된 시간변수들의 집합을 치역으로 설정한다. 일반적으로 치역은 $t_a(s) = \min\{\sigma_i, t_i | \sigma_i > 0, t_i > 0\}$ 이 된다.

단일 서버 시스템에서의 시간진행함수는 상태집합 $s = \langle r_1, n_1, r_2, \sigma \rangle$ 가 정의구역이 되고 잔여시간 변수는 하나 존재하고 엔티티에 부여된 시간변수는 존재하지 않아 치역은 $t_a(s) = \sigma$ 이 된다.

4.3.4 외부상태전이함수의 결정방법

외부상태이함수는 상태집합을 정의구역으로 상태변수의 정의구역들의 곱집합을 치역으로 하는 함수이다. δ_{ext} 의 결정을 위한 분석과정은 [그림 13]과 같다.

의미가 다른 외부사건이 2개 이상이면 사건이 동시발생될 경우를 고려한다. 이때, tie



[그림 13] 외부상태전이함수 분석과정

breaking rule은 각 사건발생 후의 상태가 유일하게 결정되어야 함으로 사건발생의 우선순위를 결정하여 다음 분석과정 단계로 진행한다. 상태변수의 변화분석은 각 외부사건 발생마다 정의된 상태변수들중에서 어떤 상태변수가 변화되는가를 파악한다. 변화시킬 상태변수의 선택은 상태집합 $s = \langle s_1, s_2, \dots, s_n \rangle$ 라 가정하면 경과시간이 e만큼 지난 후 외부사건 x가 발생하였을 경우 어떤 s_i 가 변화되는지를 명세하고 s_i 가 변화되는 일반적인 규칙은 다음과

같다.

$$\delta_{ext}(\langle s_1, s_2, \dots, s_n \rangle, e, x) = \langle s'_1, s'_2, \dots, s'_n \rangle$$

- 1) s_i 가 서술형 상태변수인 경우 s'_i 은 서술형 상태변수이다.
- 2) s_i 가 활률형 상태변수인 경우 s'_i 은 다음 중 어느 하나가 된다.
 - ① s_i 와 동일한 활률형 상태변수
 - ② 임의의 함수 $\Gamma: [0,1] \rightarrow [0,1]$ 에 의한 활률형 상태변수
 - ③ s_i 가 잔여시간 변수와 엔티티에 부여된 시간변수인 경우 s'_i 은 $s_i - e$ 가 된다.

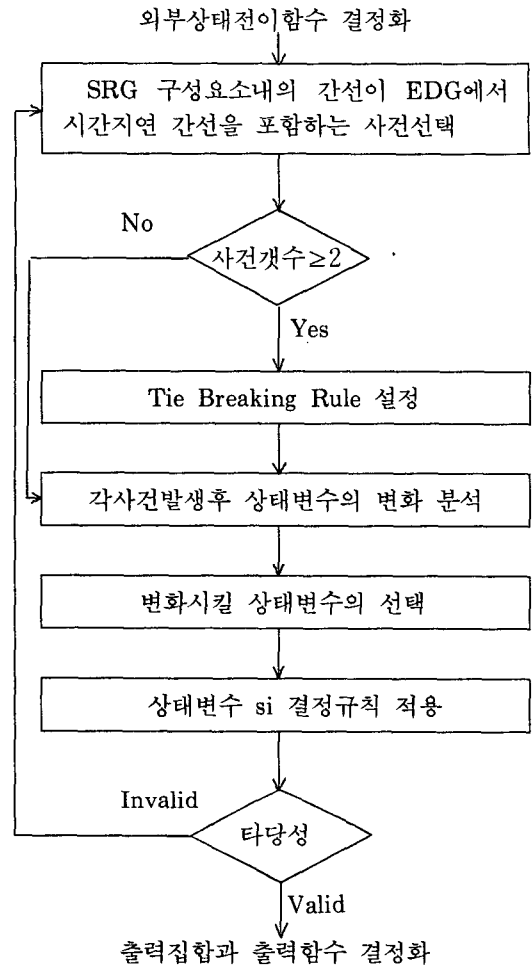
단일 서버 시스템에서 외부상태전이함수는 외부사건 r_1 이 발생하였을 때, 상태변수 r_1, n 과 σ 가 변화되는 것이고 s' 결정규칙에 의해 다음과 같이 된다.

$$\begin{aligned} \delta_{ext}(\langle r_1, n, r_2, \sigma \rangle, e, n_1) \\ = \langle \Gamma(r_1), n+1, r_2, \sigma - e \rangle \end{aligned}$$

4.3.5 내부상태전이함수의 결정방법

내부상태전이함수는 상태집합을 정의구역으로 스케줄링된 내부사건이 발생한 후 상태변수들의 변화를 결정하여 변화된 상태변수와 변화되지 않은 상태변수의 정의구역들의 곱집합을 치역으로 하는 함수이다. δ_{int} 의 결정을 위한 분석과정은 [그림 14]와 같다.

내부사건 선택과정에서 SRG 구성요소내의 간선이 EDG에서 시간지연 간선을 포함하는 경우 그 시간이 구성요소 자신의 활동 시간을 위한 것일 때는 시작노드에 해당하는 사건, 엔티티에 부여된 시간인 것일 때는 도착노드에 해당하는 사건을 선택한다. 이와 같은 내부사건이 2개 이상이면, 사건이 동시발생될 경우를



[그림 14] 내부상태전이함수 분석과정

고려한다. 이때 tie breaking rule은 각 사건발생 후의 상태는 유일하게 결정되어야 함으로 구성요소별 혹은 다른 기준을 두고 우선순위를 결정하여 다음 분석과정 단계로 진행한다. 상태변수의 변화분석은 내부사건 발생마다 정의된 상태변수중에서 어떤 상태변수가 변화되는가를 그 사건이 발생하는 조건에 따라 파악한다. 변화시킬 상태변수의 선택은 상태집합 $s = \langle s_1, s_2, \dots, s_n \rangle$ 라 가정하면, 각 사건발생에 의해 어떤 s_i 가 내부사건 발생조건에 따라 어떤

상태변수로 바뀌는지를 나타내는 규칙으로서 일반적으로 다음을 따른다. 3)의 ∞ 는 내부사건이 스케줄링되지 않았음을 의미한다.

$$\delta_{int}(\langle s_1, s_2, \dots, s_n \rangle) = \langle s_1', s_2', \dots, s_n' \rangle$$

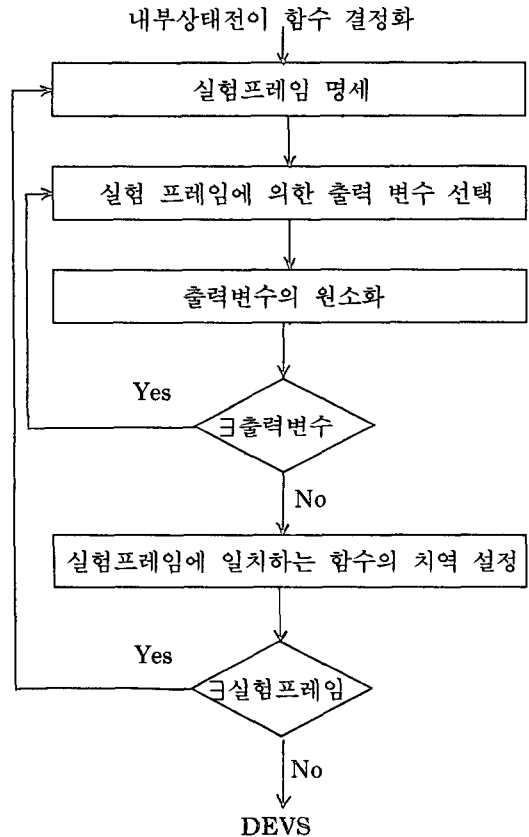
- 1) s_i 가 서술형 상태변수인 경우 s_i' 은 서술형 상태변수이다.
- 2) s_i 가 확률형 상태변수인 경우 s_i' 은 다음 중 어느 하나가 된다.
 - ① s_i 와 동일한 확률형 상태변수
 - ② 임의의 함수 $\Gamma : [0,1] \rightarrow [0,1]$ 에 의한 확률형 상태변수
- 3) s_i 가 잔여시간 변수인 경우 s_i' 은 다음 중 어느 하나가 된다.
 - ① ∞
 - ② 임의의 함수 $MEAN.TIME : [0,1] \rightarrow R^+$ 에 의한 잔여시간 변수
- 4) s_i 가 엔티티에 부여된 시간변수인 경우 s_i' 은 임의의 함수 $MEAN.TIME : [0,1] \rightarrow R^+$ 에 의한 서술형 상태변수

단일 서버 시스템의 경우 SRG 구성요소내 간선이 EDG에서 시간지연 간선을 포함하는 시작노드가 n_3 하나만 존재함으로 n_3 가 발생한 후 상태변수 변화분석을 통하여 조건 $n=1$ 과 조건 $n \geq 2$ 인 경우로 나누어서 변화되는 상태변수를 선택하면, 전자인 경우 σ 가, 후자인 경우 σ 와 r_2 가 된다. 상태변수 s_i' 결정규칙에 의해 내부상태전이함수는 다음과 같다.

$$\delta_{int}(\langle r_1, n, r_2, \sigma \rangle) = \begin{cases} \langle r_1', n, r_2, \infty \rangle & \text{if } n=1 \\ \langle r_1, n, \Gamma(r_2), SERVICE.TIME(r_2) \rangle & \text{if } n \geq 2 \end{cases}$$

4.3.6 출력집합과 출력함수의 결정방법

실험 프레임에 따라 비형식모델에서 관계된 출력변수들은 출력집합으로 결정한다. 출력집합과 상태집합에서 출력집합으로의 출력함수를 결정하는 분석과정은 [그림 15]와 같다.



[그림 15] 출력집합과 출력함수 분석과정

실험 프레임은 모델설계자가 시스템의 관심을 갖는 부분을 명세한다. 실험 프레임에 의한 출력변수 선택은 비형식모델에서 서술변수의 정의구역내 변수들을 출력변수로 설정한다. 설정된 출력변수들을 포함하는 정의구역이 출력집합이 된다. 출력함수 s 를 위한 치역은 실험 프레임에 일치하는 출력변수로 설정하고 출력집합과 출력함수의 개수는 실험 프레임의 개수

와 일치한다.

단일 서버 시스템에서 첫번째 실험 프레임은 비형식모델로부터 SERVER UTILIZATION으로 명세하면, 이에 부합되는 서술변수 BUSY는 출력변수, YES, NO갈 설정되어 출력집합 $Y = \{YES, NO\}$ 가 결정된다. 두번째 실험 프레임은 QUEUE STATE로 명세하면, 서술변수 LINE의 출력변수 n 이 설정되어 출력집합 $Y = R^+$ 가 된다. 출력함수는 각 실험 프레임에 일치하는 출력집합내 출력변수들을 결정한다. 확률변수 r_1 은 지수분포함수에 의해 생성하고 잔여시간 변수 σ 는 정규 분포함수에 의해 생성한다고 가정한다.

$$1) Y = \begin{cases} \{YES, NO\} & \text{if SERVER UTILI-} \\ & \text{ZATION=BUSY} \\ R^+ & \text{if QUEUE STATE=LINE} \end{cases}$$

$$2) \lambda_1(s) = \begin{cases} YES & \& \lambda_2(s) = n(\in R^+) \\ NO \end{cases}$$

5. 결 론

DEVS에 의한 이산사건 시뮬레이션 모델의 구축을 위한 전통적인 방법은 DEVS에 필요한 정보를 명시적으로 추출하지 못하므로 단계적, 체계적인 모델구축 수행절차가 존재하지 않아서 모델설계자의 경험적, 주관적 지식에 전적으로 의존하여 모델링한다.

본 논문에서는 이산사건 발생에 따른 실 시스템의 행위, 분석, 예측을 위하여 계층적, 모듈적 특성을 갖는 DEVS를 위한 명시적이고 형식화된 수행절차를 갖는 모델구축 방법론을 제안하였다. DEVS 요소 결정을 위한 명시적 정보를 추출하는 사건분석 도구 EDG와 상태

분석 도구 SRG에 대한 구조적 정의 및 개념을 제시하였다. 또한, 비형식모델로부터 EDG와 SRG를 구성하는 단계적인 구성방법을 제안했고 EDG와 SRG로부터 DEVS로 변환시키는 수행절차를 형식화하였다. DEVS 구축을 위한 일관적이고 체계적인 모델 형식화 방법론의 실례를 단일 서버 시스템을 가지고 세부적인 단계에 모두 적용됨을 보였다.

본 논문의 단계적이고 체계적인 모델 형식화 방법론에 의해 이전보다 객관적인 DEVS를 구축할 수 있어서 은행 시스템, 슈퍼마켓 시스템 등의 사건 스케줄링 관점에서 실 시스템 분석을 위한 시뮬레이션에 폭넓게 응용될 수 있다. 또한, DEVS가 보다 용이하게 사용되고 DEVS에 대한 이해를 향상시키는 토대를 제공하였다.

참 고 문 헌

- [1] Christensen, E.R., *DEVS-SCHEME User's Manual*, Department of Electrical and Computer Engineering Univ. of Arizona, Tucson, Sep. 1986.
- [2] Concepcion, A.I. and B.P. Zeigler, "DEVS Formalism: A Framework for Hierarchical Model Development," *IEEE Tran. Soft. Eng.*, Vol.14, No.12(1988), pp.228-241.
- [3] Denning P.J., Jack B. Dennis, and Joseph E. Qualitz, *Machine, Languages, and Computation*, Prentice-Hall, Inc., Englewood cliffs, 1978.
- [4] Elzas, M. S., T.I. Oren, B.P. Zeigler,

Modelling and Simulation Methodology in the Artificial Intelligence Era, Elsevier Science Pub., 1986.

- [5] Garzia, R. F. and M. R. Garzia, "Discrete-event simulation," *IEEE Spectrum*, Vol.23, No.12(1986), pp.32-36.
- [6] Hoover, S.V. and R.F. Perry, *Simulation a Problem-Solving Approach*, Addison-Wesley Pub. Co., Inc., 1989.
- [7] Law, A.M. and W. D. Kelton, *Simulation Modelling and Analysis*, McGraw-Hill Book Co., 1982.
- [8] Peterson, T.L., *Petri Net Theory and The Modeling of Systems*, Prentice-Hall, Inc., Englewood cliffs, N.J., 1981.
- [9] Pressman, R.S., *Software Engineering a Practitioner's Approach*, McGraw-Hill Book Co., 1987.
- [10] Zeigler C.A., *Programming System Methodologies*, Prentice-Hall, Inc., Englewood cliffs, N.J., 1983.
- [11] Zeigler, B. P. "System-Theoretic Representation of Simulation Models," *IIE TRANSACTIONS Industrial Engineering Research & Development*, Vol. 16, No.1(1984), pp.19-34.
- [12] Zeigler, B. P., *Theory of Modelling and Simulation*, A Wiley Interscience Pub.. 1984.