□ 기술해설 □

이산사건 모델링과 시뮬레이션

고려대학교 김 영 실* · 백 두 권**

목 차

1. 서 론
2. 모델링과 시뮬레이션
2.1 모델링 월드뷰
2.2 시뮬레이션 모델의 분류
3. 이산사건 모델링
3.1 이산사건 모델의 분류
3.2 이산사건의 시간진행 방법
3.3 이산사건 모델의 구성요소
3.4 이산사건 모델링 형식론
4. 이산사건 시뮬레이션
4.1 이산사건 시뮬레이션의 개념
4.2 이산 사건 시뮬레이션의 예
4.3 이산 사건 시뮬레이션 언어
5. 결 론

1. 서 론

우리 주변의 많은 일들은 시작하여 완성하기까지에 여러 가지 위험들을 내포하고 있다. 이러한 위험 요소를 제거하고 좀 더 합리적이고 적절한 해결책을 찾기 위하여 컴퓨터를 이용하여 실험하는 것이 컴퓨터 시뮬레이션(computer simulation)이다. 컴퓨터 시뮬레이션이란 넓은 의미로 실 세계 시스템을 수학적이고 논리적인 모델로 설계하고, 컴퓨터를 이용하여 이 모델을 실험하는 과정이다. 따라서 시뮬레이션은 모델에 대한 실험 계획과 실험뿐 만 아니라 모델구성 과정을 포함한다. 이 실험 또는 시뮬레이션을 통하여 실 시스템(real system)에 대한 추론(inference)을 유추할 수 있으며, 이로 인해 실 시스템에 대한 실질적인 구축 없이도 시뮬레이션을 이용하여 평가할 수 있다. 또, 비용이 많이 들고 실질적인 실험을 하기에 위험한 운영시스템인 경우나 실험의 목적이 파괴적 강도(stress)의 한계를 결정해야 하는 경우에도, 시뮬레이션을 이용해 최소의 비용으로 시스템 파괴의 위험을 배제할 수 있다.

그러므로 시뮬레이션 모델은 실 시스템의 설계, 절차적 분석, 성능평가 등에 이용될 수 있다. 한편 시뮬레이션 모델링에서는 어떤 시스템이 컴퓨터 시스템에 부합되게 서술될 수 있다고 가정하는 경우도 있다. 이때 가장 중요한 개념이 시스템 상태(system states) 서술에 관한 개념이다. 일련의 변수들이 시스템의 특성을 나타낼 수 있고 또한 변수 값들의 각 조합이 시스템의 고유한 상태와 조건을 표현하는 경우, 변수 값들의 처리는 결국 시스템의 상태 변화(어떤 상태에서 다른 상태로의 이동)를 나타내는 것이다. 시뮬레이션 실험은 모델에 잘 정의되어 설계된 운영규칙에 따라서 모델이 어떤 상태에서 다른 상태로 이동하는 동적인 변화를 관측하는 것을 포함한다.

시스템의 상태 변화는 시간에 따라 연속적으로 일어날 수도 있고 이산적으로 일어날 수도 있다. 시간의 이산적인 순간은 모델의 입력 성질에 따라 결정적 또는 확률적으로 생성될 수 있다. 이산적인 변화 모델과 연속적인 변화 모

*정회원
**종신회원

델의 동적인 행태를 기술하는 절차는 다르지만, 시간에 따른 시스템 상태의 변화를 기술하여 시스템을 시뮬레이션하는 기본적인 개념은 동일하다.

2. 모델링과 시뮬레이션

시뮬레이션이란 실 시스템에서 모델을 만들고 그 모델을 컴퓨터로 실험을 하는데 관련되는 제반활동들을 의미한다. 따라서 시뮬레이션은 실 시스템(real world), 모델(model), 시뮬레이터(simulator)의 3가지 엔티티(entity)들에 관련되며, 이들간의 관계는 〈그림 1〉과 같다. 여기에서 실 시스템은 데이타의 출력이고 모델은 데이타를 생성하는 일련의 명령들의 집합이며, 시뮬레이터는 모델의 명령들을 실행할 수 있는 장치로서 컴퓨터를 의미한다. 모델링(modelling)은 실시스템과 모델사이의 관계성을 나타낸것이다. 그리고 시뮬레이션(simulation)은 모델과 시뮬레이터 사이의 관계성(relationship)을 나타내는 것으로, 시뮬레이터가 모델에 의해 의도되는 명령들을 정확히 수행하고 있는가를 살펴보는 것이 모델 검증(verification)의 문제를 다룬다.

시뮬레이션은 이처럼 모델에 큰 영향을 받으므로 모델의 형태에 따라서 시뮬레이션의 형태가 구분된다. 시뮬레이션 모델은 분석측면에 따라서 다양하게 정의되는데 크게 결정적(deterministic) 모델과 확률적(probablistic) 모델, 정적(static) 모델과 동적(dynamic) 모델, 이산형(discrete) 모델과 연속형(continuous) 모델, 물리적(physical) 모델과 수리적(mathmetical) 모델 등의 4가지로 분류할 수 있다[32].

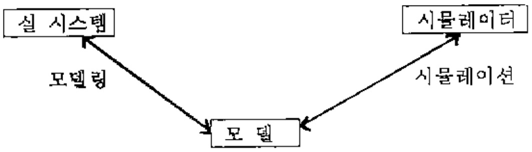


그림 1 모델링과 시뮬레이션의 기본 구조

시뮬레이션을 위한 모델 개발에 있어서 분석가는 모델 대상이 되는 시스템을 기술하기 위해 개념적인 기본구조를 갖고 있어야 한다. 이 기본구조는 시스템의 기능적인 관계를 잘 이해해서 그것을 표현할 수 있는 전체적인 시각인 모델링 월드뷰(world view)에 의해서 결정된다. 만일 모델 작성자가 SLAM, SIMSCRIPT 등의 시뮬레이션 언어를 사용한다면 기본구조는 언어에 의해 자연히 암시적으로 표현된다. 그러나 모델 작성자가 Fortran, PL/1, C등과 같은 범용언어를 사용하는 경우에는 모델 작성자가 시스템 기술을 조직화하는 기본구조를 작성해야 한다. 어떤 경우이든 간에 모델작성자에 의해 사용되는 기본구조는 시스템 기술을 명확히 해주는 개념적인 도구가 된다[24].

2.1 모델링 월드뷰

어떤 시스템의 모델은 이산적인 변화나 연속적인 변화로 분류될 수 있다. 이 분류는 모델에 의한 분류이며, 실 시스템에 대한 분류가 아님을 유념해야 한다. 사실 똑같은 시스템을 이산형 모델로 모델링할 수도 있고 연속형 모델로 모델링할 수도 있다. 대부분의 시뮬레이션에서는 시간이 중요한 독립변수이다. 시뮬레이션에 포함되는 다른 변수들은 시간의 함수로 표현되며 종속적인 변수가 된다. 시뮬레이션에서 이와 같은 이산적이냐, 연속적이냐 하는 구분은 종속변수의 형태에 따른 것이다.

시뮬레이션의 형식모델의 분류를 계층적으로 나타내면 〈그림 2〉과 같다. 이 구조를 보면 이산사건 모델은 연속적인 시간 모델에서 사건이 이산적으로 발생하는 모델을 의미한다 [34].

2.2 시뮬레이션 모델의 분류

시뮬레이션 모델 중에서 시스템 상태의 변화를 기준으로 모델을 분류하는 경우, 이산형 시뮬레이션 모델, 연속형 시뮬레이션 모델, 혼합형 시뮬레이션 모델로 분류된다[24].

(1) 이산형 시뮬레이션 모델(Discrete Simulation Model)

이산형 시뮬레이션은 상태변화가 구분 가능한 시점에서 일어나는 시스템을 시뮬레이션할 때 사용한다. 일반적으로 이산형 모델의 종속변수 값은 사건발생시간 사이에는 변하지 않는다.

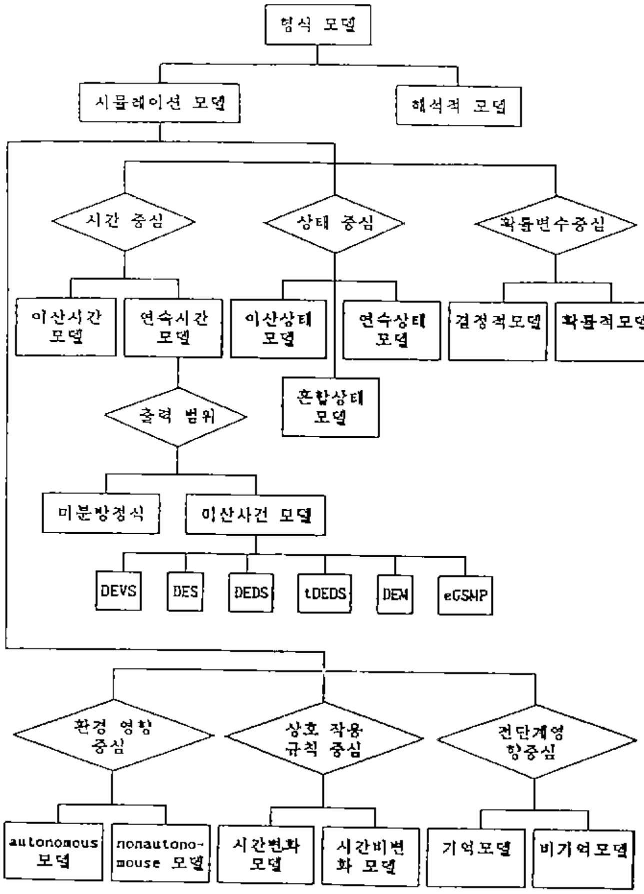


그림 2 시뮬레이션 모델의 분류

이산형 시뮬레이션에서 종속변수의 값의 변화를 나타낸 것이 〈그림 3〉이다.

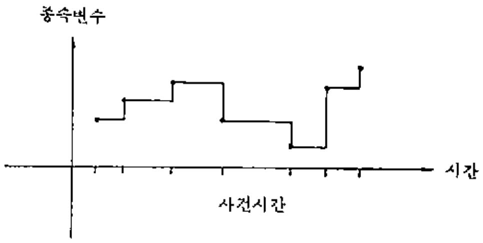


그림 3 이산형 시뮬레이션에서 도출된 종속변수 측정치

(2) 연속형 시뮬레이션 모델(Continuous Simulaton Model).

연속형 시뮬레이션에서 모델의 종속변수는 시뮬레이션 시간동안 연속적으로 변화할 수 있다. 그러나 연속적인 모델도 시간의 측면에서는 연속적일 수도 있고 이산적일 수도 있다.

이것은 시뮬레이션 중에서 종속변수의 값이 어떤 순간에도 존재하면 연속형이고, 특정한 순간에만 가능하면 이산형이 된다. 연속형 시뮬레이션에서 종속변수 값의 변화를 나타낸 것이 〈그림 4〉이다.

전형적인 연속형 시뮬레이션 모델은 시간에 대한 상태 변수의 변화율을 나타내는 하나 이상의 미분방정식들을 포함한다. 만일 미분방정식들이 아주 간단하다면 시간 0에서 상태 변수

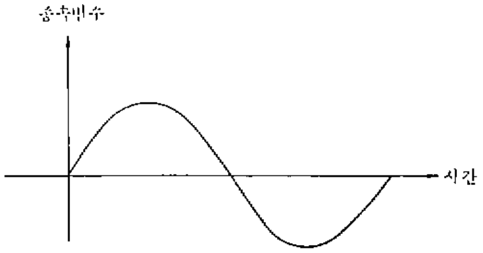


그림 4 연속형 시뮬레이션에서 도출된 종속변수 측정치

의 값에 대한 함수로 모든 시간에 대한 값을 산출하기 위하여 분석적으로 풀 수 있다. 대부분의 연속모델에 있어 분석적 해결방법(analytical solution)은 불가능하다. 그러나 수치해석적 방법들(예를 들어, Runge-Kutta 적분법)이 시간 0에서 상태 변수의 주어진 특정한 값에 대하여 미분방정식들을 수치적으로 적분하기 위해 사용된다.

이러한 연속형 시뮬레이션 모델의 설계를 위해 많은 언어들이 개발되어 왔다. 특히 CSMP III는 IBM에서 개발한 방정식중심 언어(equation-oriented language)이다.

(3) 혼합형 시뮬레이션 모델(Combined Simulation Model)

실 세계 대부분의 시스템들은 전적으로 이산적이거나 전적으로 연속적인 것이 아닌 경우가 많다. 이 경우 연속형과 이산형을 혼합한 모델을 구성하여 시뮬레이션을 수행해야 한다. 이러한 시뮬레이션을 혼합형 시뮬레이션(combined simulation)이라 한다. Pritsker와 Pegden [24]은 이산적으로 변화하고 연속으로 변화하는 상태 변수들 사이에서 일어날 수 있는 3가지 형태의 기본적인 상호작용(interaction)을 다음과 같이 기술하고 있다.

① 이산사건(discrete event)은 연속상태변수의 값을 이산적으로 변화시킬 수도 있다.

② 이산사건은 어떤 특정시점에서 연속상태변수를 변화시키는 요인을 발생시킬 수도 있다.

③ 임계치(threshold value)에 도달하는 연속상태변수는 이 때 이산사건을 발생시킬 수도 있다.

혼합형 시뮬레이션의 가장 중요한 측면은 이산적으로 변하는 변수들과 연속적으로 변하는 변수들 간의 상호작용으로부터 발생한다. 예를 들어 화학공장에서 반응물질의 농도가 규정된 수준에 도달되면 공정은 끝날 수 있다. 혼합된 시뮬레이션 언어는 이러한 조건의 발생 탐지와 이러한 순서의 모델링에 대한 지원이 가능해야 한다. 혼합형 시뮬레이션 모델에서 도출된 종속변수의 측정치 궤적은 〈그림 5〉와 같다.

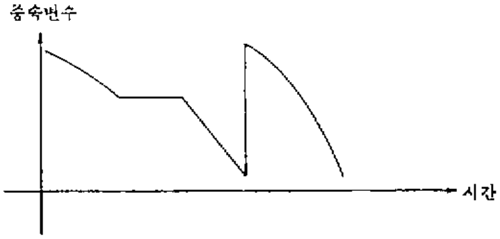


그림 5 혼합형 시뮬레이션에서 도출된 종속변수 측정치

3. 이산 사건 모델링(Discrete Event Modeling)

사람, 설비, 주문, 원자재 등과 같은 이산형 시스템 영역내의 대상을 엔티티(entity)라 부른다. 엔티티에는 많은 종류가 있으며, 각 종류는 여러 가지 특성이나 속성을 갖는다. 엔티티들은 함께 집단화되어야 하는 공통의 속성을 가질 수 있다.

이산 사건 모델링의 목적은 엔티티들이 포함된 액티비티(activity)들을 재생하여 시스템의 잠재적 행태와 성능에 대한 것들을 연구하고자 하는 것이다. 이러한 목적은 시스템의 상태들을 정의하고, 시스템을 어떤 상태에서 다른 상태로 옮기는 액티비티들을 구축함으로써 달성된다.

시스템의 상태는 엔터티들의 속성에 할당된 수치의 측면에서 정의가 된다. 모든 엔터티들이 어떤 상태에 대하여 정의된 속성값의 범위에 있을때 시스템은 그 상태에 있다고 말한다. 따라서 이산사건 시뮬레이션은 시간에 따른 시스템 상태의 동적인 묘사이다.

이산사건 시뮬레이션 모델은 다음과 같은 것들에 의하여 공식화 될 수 있다.

①각 사건발생시에 일어나는 상태의 변화
②시스템의 엔터티가 사용되는 액티비티
③시스템의 엔터티가 흐르는 프로세스

3.1 이산사건 모델의 분류

이산사건 모델링에서 사건과 액티비티간의 관계는 〈그림 6〉과 같으며, 액티비티가 시작하거나 끝나는 결정이 이루어지는 시점에 사건이 발생하고 있다. 프로세스(process)는 시간에 따라 정렬된 사건들의 순서이며, 여러 액티비티들을 포함할 수 있다. 이러한 개념들은 이산사건 모델링에 대한 3가지 월드뷰 대안을 자동적으로 유도한다. 즉 사건중심 진행방법, 액티비티 관찰중심 방법, 프로세스중심 진행방법이 있다.

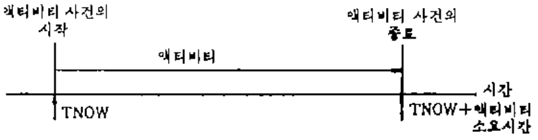


그림 6 사건과 액티비티간의 관계

(1) 사건중심 진행방법

사건 중심의 월드뷰에서는 사건 발생시간에 일어나는 변화를 정의함으로써 시스템을 모델화한다. 모델 작성자는 시스템의 상태를 변화시키는 사건을 결정하고, 각 사건의 형태와 관련된 논리를 개발하여야 한다. 시간에 따른 순서대로 각 사건과 관련된 논리를 실행시킴으로써 시스템에 대한 시뮬레이션이 된다.

사건 중심 진행방법을 설명하기 위해 은행 서비스 문제를 고려해 보자. 고객들이 시스템에 도착하고, 아마도 잠시 기다리다가, 은행원에 의한 서비스를 받고서 시스템을 떠난다. 시스템의 상태는 은행원의 상태와 대기중인 고객수에 의해서 정의된다. 고객이 시스템에 도착하거나 시스템을 떠나는 경우를 제외하면 시스템의 상태는 변하지 않는다. 따라서, 이 시스템에 대한 사건중심 모델은 고객의 도착시간과 서비스 완료시간에 무엇이 발생하는지를 기술하면 된다. 시스템 상태는 이와 같이 사건의 발생 시간에만 변하기 때문에 시스템의 동적 구조를 완벽히 기술하기 위하여 고객의 도착사건과 서비스 완료사건이 이용된다.

(2) 액티비티 관찰중심 방법

액티비티 관찰중심 방법에서는 모델 작성자는 시스템의 엔터티들과 관계되는 액티비티들을 기술하고, 액티비티의 시작과 종료를 유발하는 조건들을 규정한다. 액티비티를 시작시키거나 종료시키는 사건들은 모델 작성자에 의해서 계획되지 않고, 액티비티에 의해서 명시되는 조건들로부터 유발된다. 시뮬레이션 시간이 진행됨에 따라 액티비티를 시작시키거나 종료시키는 조건들이 관찰된다. 만약 규정된 조건들이 만족되면 액티비티에 대한 적절한 조치가 취해진다. 각 액티비티에 대한 처리가 보장되기 위해서는 매 시간의 진행마다 액티비티 전체 집합을 관찰할 필요가 있다.

어떤 형태의 문제에 대하여 액티비티 관찰중심 방법은 적절한 모델링의 기본구조를 제공할 수 있다. 예컨데 액티비티 기간이 무한정하고, 규정된 조건을 만족하는 시스템 상태에 의하여 액티비티 기간이 결정되는 경우에 이 접근방법이 매우 적합하다. 그러나 매 시간이 진행될 때마다 각 액티비티들을 관찰해야 하기 때문에 이 접근방법은 사건중심 진행방법과 비교해 보면 상대적으로 비효율적이다. 결과적으로 액티비티 관찰중심 방법은 이산사건 시뮬레이션에 대한 모델링의 기본구조로써 널리 채택되지 않았다.

(3) 프로세스중심 진행방법

많은 시뮬레이션 모델들은 정의된 형태에서 발생하는 요소들의 순서를 포함한다. 예를 들어 서비스 프로세스를 기다리는 엔티티들의 대기열은 이러한 요소의 순서로 볼 수 있다. 이러한 사건들의 순서와 관련된 논리는 일반화될 수 있고, 하나의 문장에 의하여 정의될 수 있다. 따라서 시뮬레이션 언어는 이러한 문장들을 사건들의 적절한 순서로 변화할 수 있다. 프로세스 중심의 시뮬레이션 언어에서는 시스템을 통과하는 엔티티들의 흐름을 모델링하기 위하여 이러한 문장들을 채택하고 있다. 이와 같은 문장들은 사건들의 순서를 정의하고, 사건들의 순서는 엔티티들의 프로세스를 통하여 이동할 때 시뮬레이션 언어에 의하여 자동적으로 처리된다.

프로세스중심 진행방법에서는 프로세스를 통한 엔티티들의 흐름에 대한 서술을 제공한다. 프로세스중심 진행방법에서 이러한 서술이 단순한 이유는 문장과 관련된 사건 논리가 시뮬레이션 언어가 제공하는 표준화된 일련의 기호들에 관심이 있기 때문에 오히려 모델링의 융통성은 사건중심 진행방법보다 떨어진다.

3.2 이산사건의 시간 진행 방법

이산사건 시뮬레이션은 시간상 구분 가능한 시점에서 단지 상태 변수의 변화를 이용하여 시스템을 모델링한다. 이 시간상 구분가능한 시점들은 사건(event)이 발생하는 시각이며, 사건이란 시스템의 순간적인 상태변화를 의미한다. 비록 이산사건 시뮬레이션을 개념상으로 손작업에 의하여 수행할 수 있지만, 상당한 양의 자료를 저장하고 조작하여야 하므로 컴퓨터상에서 수행하는 것이 일반적이다.

두 형태의 사건(예를 들면, 은행에서 고객의 도착과 서비스의 완료사건)은 시스템의 상태를 변화시킨다. 그러나 일부 이산사건 시뮬레이션에서 사건은 실제로 그러한 변화를 일으키지 않을 목적으로도 이용된다. 예를들면 정해진 시각에 시뮬레이션 수행을 중지시키거나 시스템의 동작에 관한 결정을 스케줄링하기 위하여 사건을 이용할 수 있는데 실제로 시스템은 상태를 변화시키지 않을 수도 있다.

이산사건 모델의 동적인 성질 때문에, 시뮬레이션을 진행함에 따라 시뮬레이션된 시간의 현재값을 유지할 필요가 있으며, 시간을 진행시키기 위한 메카니즘이 필요하다. 시간의 현재값을 가지는 시뮬레이션 모델의 변수를 시뮬레이션 시계(simulation clock)라 한다. 이 시계의 단위는 Fortran과 같은 범용언어로 모델을 작성할 때 명확히 표현되지 않으며, 입력 매개변수(input parameter)와 같은 시간단위로 가정된다. 또한 시뮬레이션 시간과 컴퓨터상으로 시뮬레이션을 수행하는 데 걸리는 실제 시간 사이에는 아무런 관계도 존재하지 않는다.

시뮬레이션 시계를 진행시키기 위하여 2가지 방식, 즉 다음사건 시간 진행법(next-event time advance)과 고정증가 시간 진행법(fixed-increment time advance)이 제안되어 왔다. 첫번째 방식은 대다수의 시뮬레이션 언어와 범용언어로 모델을 작성하는 사람들이 주로 사용하며, 두 번째 방식은 첫번째 방식의 특수한 부류에 속한다. 따라서 여기에서 논의되는 모든 이산사건 모델은 다음사건 시간 진행 방식을 사용한 것이다.

다음사건 시간 진행 방식에서 시뮬레이션 시계는 0으로 초기화되고 다음 사건들의 발생 시각이 결정된다. 이 때의 시계는 이 사건들 중 최초로 사건이 발생한 시각으로 맞추어 진다. 이 시점에서 하나의 사건이 발생했다는 사실을 나타내기 위해 시스템의 상태가 변경되며 다음에 발생할 사건들에 대한 시각이 또한 변경된다. 시계는 다시 이 사건들 중 최초의 발생 시각으로 맞추어지며 시스템의 상태가 변경되고 다음에 발생한 사건들에 대한 시각이 변경된다. 시계를 진행시키는 과정은 예정된 중단조건이 만족될 때까지 계속된다. 이산사건 시뮬레이션 모델에 대한 모든 상태는 사건의 발생 시각에 따라 변화하며 시스템의 비활동 기간은 사건시각에서 다음의 사건시각으로 시간을 증가시킴으로써 생략된다. 고정증가 시간 진행 방식은 고정된 시간만큼씩 시계를 진행시키므로 비활동 기간의 생략이 불가능하며 상당한 양의 컴퓨터 시간을 소모한다.

3.3 이산사건 모델의 구성요소

비록 시뮬레이션이 아주 다양한 현실 세계의 시스템들에 적용되어 왔을지라도 이산사건 모델들은 많은 공통적인 구성요소들을 공유하며 프로그램을 코딩하고 오류수정을 하며 변형시키는 하나의 논리적 구조가 존재한다. 〈그림 7〉의 구성요소들은 다음사건 시간 진행 방식을 사용하는 이산사건 시뮬레이션 모델에서 사용된다[32].

①시스템 상태 : 특정시각에 있어서의 시스템을 기술하기 위해 필요한 상태 변수(state variable)들의 모임.

②시뮬레이션 시계 : 시뮬레이션 시각에 대한 값을 제공하는 변수.

③사건 리스트 : 각 형태의 사건이 발생하는 시각을 포함하는 리스트.

④통계용 계수기 : 시스템 수행(system performance)에 관한 통계정보를 저장하기 위한 변수.

⑤초기화 루틴 : 시간 0에서 시뮬레이션 모델을 초기화 시키는 서브루틴(subroutine).

⑥시간 루틴 : 사건 리스트로부터 다음에 발생할 사건을 결정하고 시뮬레이션 시계를 진행시키는 서브루틴

⑦사건 루틴(Event Routine) : 특정한 형태의 사건이 발생할 때 시스템 상태를 새롭게 하는 서브루틴.

⑧레포트 발생기 : 통계용 계수기로부터 시스템 수행의 측정값에 대한 추정량을 계산, 시뮬레이션이 끝날 때 결과를 인쇄하는 서브루틴.

⑨주프로그램 : 다음에 발생할 사건을 결정하기 위하여 시간 루틴(timing routine)을 호

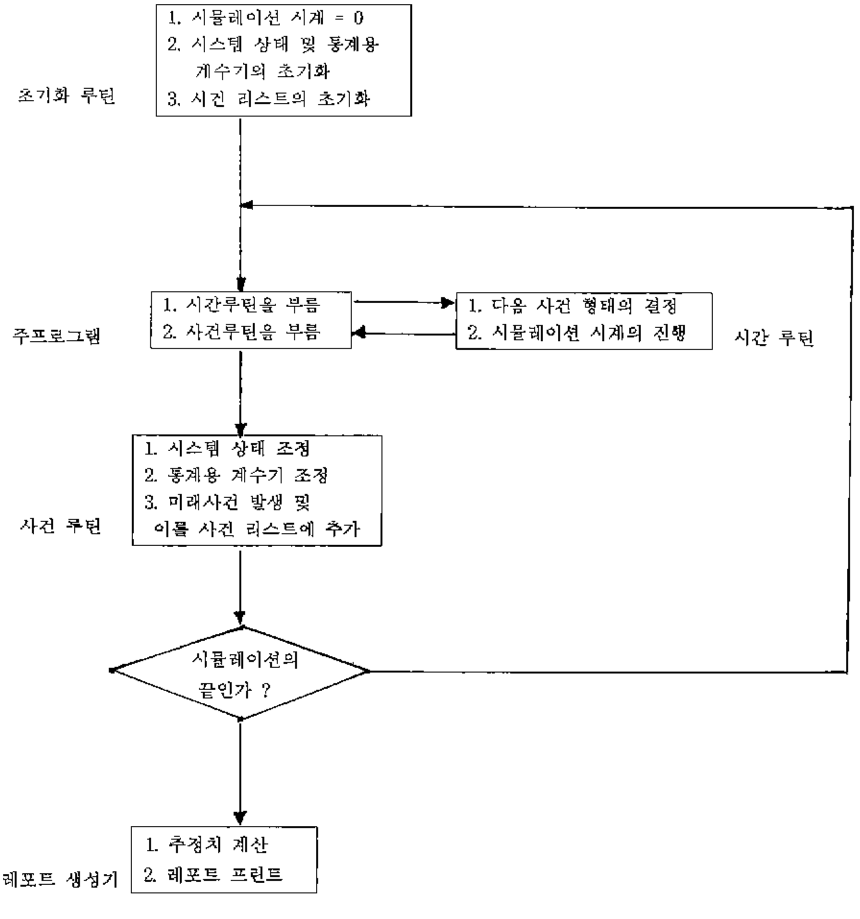


그림 7 구성요소들간의 논리적 관계

출하고 시스템 상태의 변경을 위하여 해당 루틴으로 제어를 옮기는 프로그램.

3.4 이산사건 모델링 형식론

이산사건 모델링을 위한 대표적인 형식론인 DEVS(*Discrete EVent System Specification*) 모델[20, 21, 22, 23]은 연속적인 시간상에서 이산적으로 발생하는 사건들에 대하여 시스템의 행위를 측정하는 것으로 다음과 같은 형식론에 의해 모델을 표현한다.

(1) DEVS 모델링 형식론

DEVS는 입력집합 X, 출력집합 Y, 상태집합 S, 시간진행함수 ta, 외부상태전이함수 δ_{ext}, 내부상태전이함수 δ_{int}, 출력함수 λ 등의 7개의 구성요소로 이루어진다. 즉,

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

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

$s \rightarrow ta(s)$

단 $\mathbf{R}^{+}_{0,\infty}$는 음수를 제외한 실수의 집합

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

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

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

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

$\lambda : S \rightarrow Y$

$s \rightarrow \lambda(s)$

입력집합 X는 시스템외부에서 발생하는 사건들의 집합을 의미하고 출력 집합 Y는 출력변수들의 집합을 나타낸다. 상태집합 S는 상태변수들의 각 정의 구역들의 곱집합을 의미하며 상태 $s(\in S)$는 시간 진행에 따른 시스템의 순차적인 스냅샷 상태를 의미한다. 사건진행함수 ta(s)는 시스템이 외부사건을 입력받지 않는 한 상태 s에 머물 수 있도록 허용한 시간으로 정의한다. 외부상태 전이함수 δ_{ext}는 시스템 외부에서 발생한 사건에 의한 모델의 상태변화를 나타내는 함수로 정의하고 내부상태 전이함수 δ_{int}는 외부의 사건이 없는 경우 시간진행에 따라 모델의 상태변화를 설명해 주는 함수로 정의한다. 출력함수 $\lambda(s)$는 상태 s에서 시스템의 출력으로 정의한다.

(2) DEVS 모델링의 특성

DEVS 모델링을 하므로써 Petri net이나 유한상태 기계(finite state machine)에서 모델링할 수 없었던 시간 개념을 유효적절하게 도입하여 모델링 할 수 있다. DEVS 형식론은 이산사건 시스템 모델이 동적표현 능력을 갖게해 주며, 컴퓨터 네트워크와 같은 분산환경에서의 시뮬레이션을 위한 연구에 적합하며, DEVS 모델을 이용한 모듈화 설계[2]가 가능하도록 하는 장점을 가지고 있다. 또한 DEVS 모델은 모듈적 계층적 특성을 가지며 집합론에 근거한 수학적 형식구조를 제공하고 있다[2, 19, 21].

DEVS 모델링은 그 구조의 수학적 특성 때문에 모델설계자가 이해하기 어렵고, DEVS 모델구축을 위해서는 상당한 지식을 필요로 한다. 또한 비형식적인 모델을 구축한 후 모델설계자의 경험적, 주관적 방식에 의해 수행됨으로써 일관적이고 체계적이지 못한 문제점을 가지고 있다. 이러한 단점으로 인하여 보편적인 모델링 방법으로 DEVS를 사용하는데 제한점이 되고 있다. 따라서 해석적 분석 능력을 DEVS 형식론과 연결해 주는 방법에 대한 연구[34]도 있다. 또 모델링의 가식화를 통하여 모델을 쉽게 구축해 주는 연구[15, 32]가 있다.

4. 이산사건 시뮬레이션

이산사건 시뮬레이션은 사건이 발생하는 시간에 시스템의 상태를 표현하는데, 대부분의 시스템에서 이산 사건중심방법이 시뮬레이션 모델링에 유용한 방법이 된다.

4.1 이산사건 시뮬레이션의 개념

이산사건 모델은 발생할 수 있는 사건 유형을 정의하고 각 사건 유형과 관련된 논리를 모델링 함으로써 구축된다. 시간에 따른 순서에 의하여 각 사건의 논리에 따라 상태의 변화를 추적함으로써 시스템의 동적인 모습이 생성된다.

이산사건 모델에서의 시스템은 속성들을 갖고 상태변수들과 엔티티들에 의해 표현된다. 모델의 상태는 시뮬레이션에서 사용된 변수들의 초기값을 명시하고 초기 엔티티들을 생성하고,

사건들을 스케쥴링함으로써 초기화된다. 시뮬레이션 수행 동안에 엔티티들이 액티비티에 관여함에 따라 모델은 어떤 상태에서 다른 상태로 이동하게 된다. 이산사건 시뮬레이션에서 시스템 상태는 어떤 것이 완료되는 액티비티 종료 때에만 변한다. 사건은 액티비티의 시작과 종료를 나타내는 데 사용된다. 어느 시점의 순간에 발생하여 액티비티를 시작시키거나 종료시키는 사건의 개념은 아주 중요하다. 이 관계는 〈그림 6〉과 같다.

이산사건 시뮬레이션에서는 사건과 사건 사이에서 시간은 증가하지 않으며, 시스템 형태는 사건이 발생함에 따라 일어나는 상태변화에 의하여 시뮬레이션 된다. 사건이 발생할 때, 모델의 상태는 4가지 방법으로 변할 수 있다. 첫번째 방법은 하나 이상의 변수 값이 바뀌는 것이며, 두번째 방법은 현재의 엔티티 개수가 바뀌는 것이며, 세번째 방법은 엔티티의 속성에 부여된 값이 1개 이상 바뀌는 것이며, 네번째 방법은 사건 리스트를 통하여 엔티티간에 존재하는 관계를 변화시키는 것이다.

사건들은 시뮬레이션 동안 규정된 시간에 발생하도록 스케쥴된다. 사건들은 속성들을 갖으며 사건 목록에 발생시간 순서에 따라 기록 유지된다. 예를들어 서비스 완료사건을 스케쥴할 때 고객의 서비스에 관련된 속성들은 사건의 일부가 되며 서비스 완료사건의 처리시에 이 속성들이 이용된다. 따라서 만약 엔티티가 일련의 액티비티를 거친다면(각 액티비티는 사건으로 표현되는 액티비티의 완료개념을 가짐), 엔티티의 속성들은 처리될 각 사건과 연관을 가질 것이다.

4.2 이산사건 시뮬레이션의 예

은행에서 고객도착사건과 관련된 논리를 생각해 보자. 먼저 취해져야 할 동작은 다음 고객의 도착을 계획하는 것이다. 이것은 고객의 도착순서에 의하여 계산된다. 즉 첫번째 도착이 일단 계획되면, 계속되는 도착사건들이 계획될 수 있다. 현재의 고객도착사건은 고객도착시간에서 시스템 상태에 달려 있다. 만약 은행원이 바쁘다면, 대기 고객수를 증가시켜야 하며 시스템의 상태가 바뀐다. 그렇지 않은 경우는 고객은 즉시 서비스를 받을 수 있다. 이 경우 시스템의 상태는 은행원이 서비스하는 데 소요되는 시간을 더하여 이 고객에 대한 서비스 완료사건을 계획하여야 한다.

이제는 서비스 완료사건을 처리하는 논리를 살펴보자. 이 사건에서는 고객과 은행원의 처리사항이 명시되어야 한다. 이 시스템에서는 서비스를 완료받은 고객은 은행을 떠난다고 가정하고 있다. 현재의 고객에 대해 서비스를 완료한 은행원에 대해서는 먼저 은행원의 서비스를 받기 위하여 대기열에 기다리는 고객이 있는지를 확인하여야 한다. 만약 고객이 기다리고 있다면 우리는 대기열의 고객수를 1만큼 줄이고, 첫번째 대기고객에대한 서비스 완료사건을 계획하여야 한다. 기다리는 고객이 없는 경우에는 은행원은 유휴상태로 들어간다.

사건중심 진행방법을 이용한 은행원 문제에 대한 시뮬레이션을 생성하기 위하여 시뮬레이션 동안의 기록일지에 따른 시점에 이러한 사건들이 발생하도록 한다. 사건기록일지는 처음에는 첫번째 고객도착사건에 해당하는 사건을 포함한다. 시뮬레이션이 진행됨에 따라 추가적인 도착사건과 서비스 완료사건이 사건과 관련되어 앞에서 설명된 논리에 따라 사건기록일지에 계획이 된다. 시뮬레이션 시간은 사건의 발생에 따라 증가되면서, 각 사건은 시간적인 순서에 따라 실행이 된다. 시뮬레이션에 대한 이와 같은 사건중심 진행방법은 모델링 노력을 크게 완화시켜 주는 것이다. 모델의 동적인 행태를 얻기 위해서는 각 사건시간에서의 시간변수들이 검토될 필요가 있다.

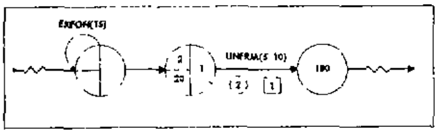
만약 이산 사건 모델을 프로그래밍하기 위하여 모델 작성자가 Fortran과 같은 범용언어를 사용한다면, 사건기록일지를 개발하고 시간적인 순서에 따른 사건처리를 위한 시간장치를 개발하는 데 많은 노력이 소요될 것이다. 그러나 이와 같은 기능은 모든 이산형 사건의 모델에 공통적으로 필요하기 때문에 SLAM II나 SIMAN 등을 포함한 여러 시뮬레이션 언어들이 개발되어 왔다. 이러한 시뮬레이션 언어들은 이산 사건 모델에서 공통적으로 봉착하는 기능뿐만이 아니라 사건 계획을 위한 특수한 기능도 제공해 준다.

이산사건 시뮬레이션의 프로그래밍을 위하여 Fortran, GASP, SIMSCRIPT, GPSS, SLAM 등 여러가지 프로그래밍 언어를 사용하고 있는데 이중에서 SALM 네트워크 모델을 이용하여 은행원이 고객에게 서비스를 제공하는 시스템을 예를 들어 *Informal Model Description*, SLAM Network Model, SLAM program을 만들어 보면 다음과 같다.

(1) Informal Model Description

① 2명의 행원이 있는 은행 모델
② 두 행원이 동시에 고객에게 서비스
③ 대기열의 용량은 20명
④ 시뮬레이션을 시작할 때 2명의 고객이 대기열에서 서버를 기다린다.
⑤ 고객들의 도착 간격 시간은 평균이 15인 지수분포
⑥ 서버의 서비스 시간은 5와 10사이의 일양분포
⑦ 100명의 고객이 시스템을 떠나면 시뮬레이션을 종료

(2) SLAM Network Model



(3) SLAM Program

```
GEN,JEONPIL,TWO PARALLEL SERVERS,3/29/95;
LIMITS,1,1,50;
NETWORK;

 CREATE,EXPON(15);            CREATE BANK CUSTOMERS
 QUEUE(1),2,20;          WAIT FOR AVAILABLE BANK TELLER
 ACTIVITY(2)/1,UNFRM(5,10); TWO TELLERS IN PARALLEL
 TERM,100;                    END SIMULATION WHEN 100

 CUSTOMERS HAVE BEEN SERVED
  ENDNETWORK;
FIN;
```

(4) 출력 결과 및 분석

```
1]                    **INTERMEDIATE RESULTS**
1              S L A M  I I  S U M M A R Y  R E P O R T
               SIMULATION PROJECT TWO PARALLEL SERVERS     BY JEONPIL
               DATE 3/29/1996                       RUN NUMBER  1 OF  1
                CURRENT TIME  0.5628E+03
                STATISTICAL ARRAYS CLEARED AT TIME  0.0000E+00
                  **FILE STATISTICS**
FILE                 AVERAGE     STANDARD    MAXIMUM   CURRENT    AVERAGE
NUMBER  LABEL/TYPE      LENGTH      DEVIATION    LENGTH    LENGTH     WAITING
TIME
  1     QUEUE        0.2086      0.5165       3         0        1.8369
  2     CALENDAR      2.0530      0.7514       4         1        7.8729
                     **SERVICE ACTIVITY STATISTICS**
ACTIVITY   START NODE OR    SERVER      AVERAGE          STANDARD      CURRENT
AVERAGE   MAXIMUM IDLE   MAXIMUM BUSY   ENTITY
INDEX        ACTIVITY LABEL    CAPACITY    UTILIZATION     DEVIATION     UTILIZATION
BLOCKAGE   TIME/SERVERS,  TIME/SERVERS   COUNT
  2      TWO TELLERS          2          1.0530       0.7514         0          0.0000        2.0000
  2.0000     100
```

표 1 시뮬레이션 언어의 비교

비교기준	시뮬레이션 언어				
	Fortran	GASP	SIMSCRIPT Ⅱ.5	GPSS Ⅴ	SLAM
문제의 개념화 정도	나쁨	보통	좋음	우수함	우수함
System oriented toward	None	전부	전부	큐잉	모두
Modelling approach					
Event-scheduling		가능	가능	불가능	있음
Process-interaction		불가능	가능	가능	있음
Continuous		가능	가능	불가능	있음
제공되는 루틴					
Random sampling built in	없음	있음	있음	없음	있음
Statistics-gathering capability	빈약	우수함	우수함	좋음	우수함
List-processing capability	빈약	좋음	우수함	보통	좋음
Ease of getting stanaard report	빈약	우수함	보통	우수함	우수함
Ease of designing special report	보통	좋음	우수함	빈약	좋음
Debugging aids	보통	좋음	우수함	보통	좋음
Computer runtime	우수함	좋음	좋음	빈약	좋음
Documentation for learning language and for renference	매우좋음	매우좋음	보통	매우좋음	매우좋음
Self-documenting code	빈약	좋음	좋음	우수함	좋음
Cost	저가	저가	고가	저가	보통

위의 프로그램을 수행한 결과 서버를 기다리는 고객의 숫자는 평균이 0.2086이고 표준 편차는 0.6165이다. 그리고 서버를 기다리는 고객의 최대 숫자는 3명이다. 고객이 서버를 기다리는 평균 시간은 1.8369이다. 또한 서버는 2명이며 고객의 서버에 대한 평균 이용도는 1.0530이다. 이 값이 1이 넘는 이유는 각각의 서버의 이용도를 합했기 때문이다.

서버당 최대 유휴 시간은 2이며 최대 바쁜 시간은 100이다. 이 상태로 보아 서버들은 너무 바쁜 시간이 많기 때문에 서버 한명을 더 고용하는 것이 좋다는 것을 시뮬레이션 결과로 알 수 있다.

이러한 예제를 통해 알 수 있듯이 이산사건 시뮬레이션은 실제 회사의 업무를 효과적으로 하기 위한 인원의 배치 문제에 적용하거나 컴퓨터 시스템의 구성이나 성능 평가, 분산환경에서 분산 시스템의 설계 등에 이용할 수 있다.

4.3 이산 사건 시뮬레이션 언어

컴퓨터 시뮬레이션 언어는 복잡한 실세계 시스템에 대한 개발과 시뮬레이션 실행을 도와주는 역할을 한다. 컴퓨터에서 시뮬레이션을 수행하기 위해서는 주어진 문제의 성격과 시뮬레이션을 수행하는 환경을 고려하여 적절한 시뮬레이션 언어를 선택하여야 한다. 시뮬레이션 언어에는 시뮬레이션 분류와 마찬가지로 연속형 시뮬레이션 언어(continuous simulation language)와 이산사건 시뮬레이션 언어(discrete event simulation language), 그리고 혼합형 시뮬레이션 언어(combined simulation language)로 구분된다.

연속형 시뮬레이션은 시스템이 연속적으로 변화되는 것을 시뮬레이션하는 것으로 시뮬레이션 방법은 주로 미분 방정식을 이용하므로 ASCL및 CSSL-Ⅳ는 미분방정식의 기술 및 수치 계산 방법으로 미분방정식의 해를 구하는 과정을 용이하게 하기위해서 개발되었다.

특히 이산사건 시뮬레이션을 위한 언어는 월드 뷰인 사건중심 방법이나 프로세스중심 방법에 따라 다르게 처리한다. Fortran은 과학용 언어로 시뮬레이션에서 사용하기 위해서 설계된 언어는 아니다. 그러나 대부분의 사건 중심 시스템의 분석가들은 Fortran을 사용한다. GPSS는 트랜잭션 중심방법을 사용하는 고수준의 구조화된 시뮬레이션 언어이다. 또한 이것은 일반적인 프로세스 중심 방법의 특별한 경우에도 사용되며, 특히 job shop과 같은 대기열 시스템을 설계하는데 사용된다. Fortran과 GPSS는 이산사건 시뮬레이션 모델링을 하는데 널리 사용되고 있다.

SIMSCRIPT과 SLAM은 고수준의 시뮬레이션 언어로 모델을 쉽게 만들도록 해 준다. SIMSCRIPT Ⅱ.5과 SLAM은 Fortran과 달리 앞으로 일어날 사건의 리스트 관리나 Built-in Random Variate Generator, Built-in Statistics Gathering Routine 등을 지원해 준다. 또 GPSS와 다르게 복잡한 계산을 쉽게 해 준다. SLAM은 연속형 시뮬레이션이 가능하도록 지원해 준다. SLAM은 Fortran을 기반으로 하며 Fortran에서 사건 중심 시뮬레이션을 쉽게 하도록하는 GASP을 포함한다.

또한, 이산 사건 시뮬레이션언어인 DEVS-Scheme은 Texas Instrument의 Scheme 기능을 포함한 SCCOPS(SCheme Object-Oriented Porgramming System)에서 DEVS 형식론을 구현한 LISP 계열의 시뮬레이션 언어이다. 또 외부사건을 병렬화하는 방법을 통하여 DEVS Abstract Simulator가 병렬 구현되었으며 Christesen은 Ada 언어환경하에서 DEVS abstract simulator를 개발하였고, Time Wrap을 결합하여 병렬화 시켰다. 객체지향형 환경하에서의 DEVS 형식론인 DEVSIM++의 환경도 개발되었다[11, 23, 29]. 이러한 시뮬레이션 언어를 비교한 표를 참조하면 〈표 1〉과 같다[5].

5. 결 론

시뮬레이션을 통해 미리 시스템을 평가 분석하므로써 많은 이점을 가질 수 있다. 이러한 시뮬레이션의 개념은 컴퓨터 시스템의 성능 평가뿐만아니라 시스템의 구성, 새로운 환경 구축시에도 활용될 수 있다. 특히나 앞으로의 모든 분야는 서비스와 관련되어 발전되어 간다. 통신망의 서비스 기능의 적용이나 기존의 데이타를

이용하여 고객에게 새로운 서비스를 제공하거나 기존의 서비스를 개선하고자하는 영역에서 시뮬레이션을 이용하므로써 더 타당한 방법을 선택하게 해 준다.

시뮬레이션은 과학적인 방법으로 문제를 해결하는 하나의 방법이다. 이러한 시뮬레이션의 많은 방법중에서도 이산사건 시뮬레이션 방법은 가장 많이 활용되고 있는 방법이기도 하다. 이러한 이산사건 시뮬레이션은 다양한 분야에서 이용되고 있다. 특히 이 기법은 물류 분야나 산업의 생산 영역에서 많이 활용되어져 왔던 방법이기도 하다[13, 17, 26]. 통신 프로토콜 모델링의 대표적인 방법으로 수학적 기초를 갖는 Petri-Net을 이용하고 있으나, 이것은 상태 폭발 문제와 모듈 설계의 문제, 그리고 동적 표현 능력의 부족이라는 문제점들을 내포하고 있다. 시간 개념을 도입하여 동적 표현 능력을 갖도록 하기 위해 DEVS와 결합하는 방법에 관한 연구[30]가 있으며, 형식 모델 사이의 상호관련성에 대한 Hass의 연구는 이산사건 시뮬레이션을 위하여 지수분포에 의한 확률변수를 갖는 SPN 모델로의 변환 방법인데, 이 연구는 너무 수학적 이론에 치우쳐 있어 모델링하기 어렵다는 단점이 있다. 이와 유사한 연구로 시뮬레이션적인 방법과 해석적 방법간의 상호 관계를 연구[34]한 경우도 있다.

시뮬레이션 모델링의 형식론 기반을 제공한 DEVS 모델에 관련된 연구를 토대로한 [2, 19, 20, 21]와 이러한 모델을 병렬 분산환경에 적용한 연구[29], 네트워크 환경하에서 패킷망의 성능 분석에 이용한 연구[27]가 있다. 또, 이산 사건 시뮬레이션이 많이 적용되는 교통문제 영역의 응용 연구[6, 10]이 있다.

분산환경에서 데이타를 분할하는 기법에 대한 시뮬레이션 연구[7], 멀티버젼을 이용한 트랜잭션 처리방법에 대한 시뮬레이션으로 SLAM과 Fortran을 이용한 연구[33]가 있다. 그외에도 시뮬레이션을 활용한 각종 연구[9, 4, 13, 17]가 있으며 인공지능에서 개발된 기술이나 지식 베이스 결합 영역도 활발한 연구로[11, 25] 등이 있다.

최근에는 컴퓨터의 빠른 처리 속도를 요구하는 병렬처리가 필수적인 요소로 대두되고 있는데, 시뮬레이션에서도 이러한 병렬처리 연구[8, 29]가 진행되고 있으며, 인공지능 기법을 도입한 연구로는[17, 12, 4, 3] 등이 있으며 인공지능 기법 중에 최근 많은 관심이 있는 Genetic Algorithm과의 결합에 관한 연구가 진행중이다.

시뮬레이션의 발전은 시뮬레이션 자체에 대한 연구와 더불어 다른 분야와 결합된 내용의 연구가 활발히 진행 되고있으며, 이러한 방법들로 인하여 많은 분야에서 효과적인 전략을 세우는데 시뮬레이션 기법을 이용하게 될 것이다. 또, 시뮬레이션 방법은 미래 사회에서 어떤 것을 계획하거나 설계, 실시하기 전에 반드시 수행해야하는 과제이며, 이러한 시뮬레이션을 통해 나온 결과들을 분석하고 적용하므로써 많은 도움을 받을 수 있을 것이다.

참 고 문 헌

[1] Doo Kwon Baik, "Performance Evaluation of Hierachical Simulator : Distributed Model Transformation and Mappings," Ph. D. Dissertation, Wayne State Univ., 1986.

[2] A. I. Concepcion, B. P. Zeigler, "DEVS Formalism : A Framework for Hierachical Model Development," IEEE Tran. on Soft. Eng., Vol. 14, No. 12, pp. 228-241, Feb., 1988.

[3] Gerald T. Mackulak, Jeffery K. Cochram and Paul A. Savory, "Ascertaining Important Features For Industrial Simulation Environments," SIMULATION Vol. 63 No. 4, pp.211-222, October, 1994.

[4] Hideo Fujimoto and Kazuhiko Yasuda, "Experimental Investigation of Decision Rule Sets for Multilevel Jop Shop Scheduling Using a Computer Simulation," New Directions in Simulation for Manufacturing and Communications, pp.475-481, 1994.

[5] Jerry Banks and John S. Carson, Discrete-Event System Simulation, Prentice Hall, 1984.

[6] J. M. ten have, "The development of the NLRATC Research Simulator (Narsim) Design Philosophy and Potetial for ATM research," Simulation Practice and Theory

1, pp.31-39, 1993.

[7] Seongjin Park, Daeha Park, and Doo Kwon Baik, "Design of a Data Partitioning Model in Distributed Database System," Proceeding of thr First international Conferences on Computer Technology and Applications, pp.92-97, 1994.

[8] R. Ayani, Y. Ismailov, M. Lijenstam, A. Popescu, H. Rajaei, and R. Ronngren, "Modelling and Simulation of High Spped LAN," SIMULATION, Vol. 64 No. 1, January, 1995.

[9] Susu Morito and Keun Hyun Lee, "An Application of Simulated Anealing for an FMS Dispathcing Priority Problem," New Directions in Simulation for Manufacturing and Communications, pp.197-203, 1994.

[10] Tatsuo Nakamura and Yoichi Hibino, "An Anaysis on Pedestrian Traffic Flow using Discrete Event Simulation," New Directions in Simulation for Manufacturing and Communications, pp.188-192, 1994.

[11] Tag-Gon Kim, and B. P. Zeigler, "The DEVS-Scheme Simulation and Modeling Environment," Chapter 2 of the Book Knowledge Based Simulation : Methodology and Application (eds : Paul A. Fishwick and Richard B. Modjeski) Springer Verlag, Inc., pp. 20-35, 1990.

[12] Ubirajara Ferreria and Harpal Maini, "On Simulating Parallel Computation : An Application for Combinational Optimization Problems", New Directions in Simulation for Manufacturing and Communications, pp.439-445, 1994.

[13] Wayne J. Davis, "Evolving Simulation Requirement with the Coordination of Flexible Manufacturing System," New Directions in Simulation for Manufacturing and Communications, pp.482-488, 1994.

[14] Wolfgang Kuehn, "Selecting Discrete Event Simulation Software for Manufacturing," New Directions in Simulation for Manufacturing and Communications, pp.428-433, 1994.

[15] Y. Yamamoto, and M. Lenngren, "Graphical Model Building System," Proc. the *16th Annual Simulation Symposium, pp.* 161-176, 1983.

[16] Yasuhiro Hirakawa, "A Simulation Technique for Analysing the Performance of Flexible Manufacturing System," New Directions in Simulation for Manufacturing and Communications, pp.489-492, 1994.

[17] Yoshiro Fukuda, Hironri Hibino, Keiji Mitsuyuki and Fumino Kojima, "Integrated Environment for Production System Simulations," New Directions in Simulation for Manufacturing and Communications, pp.434-438, 1994.

[18] C. A. Ziegler, Programming System Methodologies, Englewood Cliffs, NJ : Prentice Hall, Inc., 1983.

[19] B. P. Zeigler, "System-Theoritic Representation of Simulation Models," IIE TRANSACTIONS Industrial Engineering Research & Development, Vol. 16, No. 1, pp. 19-34, Mar., 1984.

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

[21] B. P. Zeigler, Multifacetted Modelling and Discrete Event Simulation, Academic Press Inc. LTD. Pub., 1984.

[22] B. P. Zeigler, "DEVS Representation of Dynamical Systems : Event_Based Intelligent Control," Proc. of the IEEE, Vol. 77, No. 1, pp. 72-80, Jan., 1989.

[23] B. P. Zeigler, and Gouquing Zhang, "Modelling Hierachical Discrete Event Models to Multiprocessor System : Concepts, Alogorithm, and Simulation," Journal of Parallel and Distrbuted Computing, Vol. 9, pp. 271-281, 1990.

[24] 김강현, 백두권, 시뮬레이션, 한국방송통신대학, 1992

[25] 김명희, "지식기반형 하이브리드 시뮬레이션 환경의 구축," 한국시뮬레이션학회논문지, 제1권 1호, pp. 25-30, 1992.

[26] 박진우, 우상복, "FMS 생산계획에서 대기 네트워크 모델의 적용 가능성에 관한 연구," 한국시뮬레이션학회논문지, 제1권 1호, pp.48-54, 1992.

[27] 박상희, 이왕재, 이석필, 지승도, "DEVS 시뮬레이션을 이용한 패킷망 모델링의 성능 분석," 한국시뮬레이션학회논문지, 제3권 1호, pp.75-88, 1994.

[28] 시스템 시뮬레이션 이론 및 실제, 한국시뮬레이션 학회, 1992.

[29] 성여락, 정 성훈, 김탁곤, 박규호, "병렬분산 환경하에서의 DEVS 형식론의 구현," 한국시뮬레이션학회논문지, 제1권 1호, pp.64-76, 1992.

[30] 손진곤, "Petri Net과 DEVS 형식론을 이용한 컴퓨터 통신 프로토콜 모델링," 박사학위논문, 고려대학교, 1990.

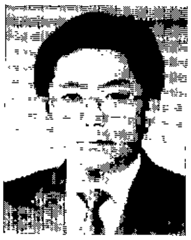
[31] 송기순, "이산사건 시뮬레이션을 위한 그래픽 모델링 도구의 설계와 구현," 석사학위논문, 고려대학교, 1992.

[32] 이영해, 백두권, 시스템 시뮬레이션, 경문사, 1991.

[33] 유인관, "다중버전에 기반한 실시간 트랜잭션 모델," 석사학위논문, 고려대학교, 1994.

[34] 정영식, "DEVS와 SPN을 이용한 통합적 시스템 분석 모델링," 박사학위논문, 고려대학교, 1993

백 두 권



1974년 고려대학교 수학과 졸업
1977년 고려대학교 산업공학과에서 석사학위 취득
1977년~1980년 대전초급대학 및 전남대학교 공과대학에서 전임강사로 근무
1981년~1986년 미국 웨인 주립대학교 컴퓨터과학과에서 수학하여 석사학위(1983) 및 박사학위(1986)를 취득
1986년~현재 고려대학교 전산과학과 조교수로 근무를 시작하여 부교수를 거쳐 현재 교수로 재직중
관심분야 : 모델링 및 시뮬레이션, 소프트웨어 공학, 지식공학, 분산시스템, 멀티미디어데이타베이스

김 영 실

1989년 고려대학교 전산과학과 졸업
1991년 고려대학교 석사학위취득
1993년~현재 박사과정에 재학중
관심분야 : 모델링 및 시뮬레이션, 지식베이스, 객체지향 데이타베이스, 전문가 시스템, 멀티미디어 데이타베이스

● 정보과학회 영문 논문지 논문모집 ●

- 제출기한 : 창간호 — 1995년 7월 31일(월)
- 제　　출 : 한국정보과학회 사무국 한영진
- 주　　소 : 서울특별시 서초구 방배3동 984-1(머리재빌딩 401호)
㊝137-063　T. 02-588-9246　F. 02-521-1352