

運營研究道具로서의 Simulation

朴 景 洙*

1. Simulation의 개념

1.1 서론

Simulation이란 「模擬실험」이라는 말로서 “주어진 system의 모형(model)을 이용해서 실험을 하는 것”을 말한다. 이렇게 말하면 Simulation의 범위는 대단히 넓어서 「모의국회」로부터 「군사기동훈련」 또는 자동차의 안전도 검사에 manikin을 태워서 충돌시키는 실험등 모든 것이 포함되는 것이다. 그러나 이러한 實體모형(scale model)에 의한 실험이 번거로울 뿐만 아니라 비용이 많이 소요되고(기동훈련을 한번 실시하기 위한 예산을 생각해볼 것), 또 system의 성격이 복잡해지면 기술적, 경제적으로 대단히 힘들어지게 된다. 그래서 과학자들은 실제 system의 특성을 나타내는 數式 Model을 만들어서 풀려고 하였다. 예를 들어 M軍과 N軍이 발사속도 r인 무기로 p의 사살확율을 가지고 전투를 할 때 시간이 지남에 따른 잔여병력이

$$\begin{cases} \frac{dm}{dt} = -p_n r_n n \\ \frac{dn}{dt} = -p_m r_m m \end{cases}$$

로 나타난다는 Lanchester model이나, Poisson (λ) input과 exponential(μ) service time으로 묘사되는 single server queuing system에서 待機行列(queue)에 평균치가

$$Lq = \lambda^2 / \mu(\mu - \lambda)$$

로 표시되는 것들이다.

이러한 방법의 강점은 모형실험을 할 필요없이 간편하게 풀 수 있고, 주어진 문제의 특성을 추출하여 system內的 인과관계들을 통찰할 수 있게 하는 것이다. 다시 말하면, 위의 예에서 발사속도와 사살확율의 상호관계, 또는 service를 빨리 할 때의 待機行列의 변화 등이다. 그러므로 수학적 model을 만들어서 이것이 실제 system을 잘 나타내고, 풀 수 있

을 때에는 일반적으로 Simulation을 하는 것보다 훨씬 더 효율적이다. 그러나 주어진 문제가 복잡해짐에 따라 수학적인 해법을 발견하는 데는 한계가 있는 것이다. 이러한 필요에 의해서 구상된 것이 소위 좁은 의미의 “Simulation”이며 이는 數式이나 법칙(logic)들을 사용하여서, 어떤 특성을 가진 추상적인 “대상물”이 Simulation에 포함된 모든 數式이나 logic을 거쳐갈 때 결과적으로 system의 성능을 나타내는 어떤 특성에 주는 영향을 관찰하기 위하여 기록정리해 주는 것이다. 쉽게 말하자면 Simulation은 “유령회사의 금전출납부”와 같다고 비유될 수 있을 것 같다. 더구나 요새는 계산속도가 빠른 computer의 출현과 함께 computer를 사용해서 Simulation을 할 수 있어서 큰 문제들을 분석하는 데 없어서는 안될 도구가 되었다. 이렇게 computer가 일반화 되고, Simulation이 보편화되면서 Simulation하면 通常 “Computer Simulation”을 말하는 것 같으며 좁은 의미로 정의하자면 “불확실하거나 복잡한 조건에서 실제 system을 운용하기 위한 의사결정을 할 수 있도록 computer화된 수학적 model을 가지고 사실과 가정에 기초를 둔 代案들을 평가하기 위해서 이용되는 模擬실험 方法”이다. [1] 흔히 Simulation을 하기 위해서는 수학적인 냄새가 풍기는 model을 사용한다. 그리고 system 전체의 작동을 묘사하기 보다는 simulation model은 system 개개 부분의 작동을 묘사하여 이것을 종합하는 입장을 취한다. 다시 말하여 System을 여러개의 간단한 부분으로 나누는데, 각 부분들의 작동을 다 예측할 수 있을 때까지 (힘들면 확률적으로나마) 나누어 나간다. 그리고 이런 작은 부분들끼리의 상호관계가 묘사 되어서 Simulation model에 포함된다. 그래서 Simulation을 하면 거대한 model설정 작업을 여러개의 작은 부분으로 나누고, 이런 부분들을 자연스러운 순서로 연결지어 주어서 computer가 이들 상호간의 작용을 계산할 수 있게된다.

Simulation model이 설정되면, system의 실제 운

*한국과학원 산업공학과

용을 模擬하기 위하여 input data를 집어넣어 돌려 보고 system 전체의 작동결과가 기록 된다. 이런 模擬 실험을 여러번하고 또 model에 사용되는 媒介 변수(parameter)의 값이나 각 부분 간의 상호연결을 조금씩 바꾸어 주면 결국은 여러가지 代案 중에서 가장 좋은 것을 선택할 수 있게 된다.

1.2 Simulation의 목적

먼저 말한 바와 같이 사실은 우리가 관심을 가지고 있는 system 자체를 실제 운용해보는 것이 가장 정확한 방법이지만, 이 방법이 불편하고, 비싸고, 또 어떤 때에는 장시간을 요하기 때문에 우리는 實體 모형도 사용하고, 수학적 model로 풀 수 있는 것은 풀고, system이 복잡하여 이런 방법들이 다 불가능할 때는 Computer Simulation model을 설정하여 Simulation을 한다.

따라서 Simulation이란 대개가 system의 추상적인 model을 이용한 실험방법이고 system의 성격에 따라서 이 실험이라는 것이 표본추출(sampling)을 할 때가 많다.

Simulation을 하면 그 model 설정 과정에서의 부분적인 생략, 또 parameter들을 예측 할때의 오차, 또 제한된 량의 표본추출을 하는 데서 오는 통계적 오차들 때문에 가장 좋은 Simulation 결과를 가져온 代案이 실제 system에 적용되어 꼭 같은 결과를 가져 오리라고 보증하기는 힘들다, Simulation이 적절히 수행되었다면 발견된 代案이 아주 좋은 것이라는 것은 기대해도 좋다.

한가지 부연할 것은 Simulation이 대개 computer에 의해서 수행되지만 이것은 단지 많은 량의 중간 계산이나 기록을 해야한다는 것이고 꼭 어떤 實體의 인 관계가 있는 것은 아니다.

1.3 Computer Simulation의 장점

① 실제 system으로 실험하는 것이 아니라, 그것의 model을 사용하여 실험할 수 있어서 싸고 안전하다.

② system의 model이 한번 설정되면 여러가지 경우를 분석하기 위하여 계속 사용될 수 있다.

③ Simulation 방법은 앞으로 계획된 system을 분석하기에 편하며 특히 계획된 system에 대한 정보가 없을 때 편하다.

④ system 운용에 대한 문제점들이 값싸게 얻어진다.

⑤ 수학적 배경이 약한 사람들도 이용할 수 있다. [2]

1.4 Computer Simulation의 단점

① Simulation의 실행도는 model과 input data의 실행도에 달려있다. 엉터리 model은 이익은 커녕 해만 주게 된다. computer사용자들 간에 흔히 쓰이는 “쓰레기가 들어가면 쓰레기가 나온다”라는 표현이 Simulation에도 적용된다.

② Simulation model을 만드는 일이 복잡하며 변화하기가 힘들다. 다른 system에 대해서는 다른 program을 써야 하고, GPSS등과 같은 Simulation language들이 있으나 사용절차가 복잡하다.

③ Simulation model을 돌리는 computer사용료가 비싸다.

④ Simulation이 필요이상으로 보편화되어, 더 좋은 방법이 있는데도 불구하고 많은 사람들은 Simulation방법을 사용한다. [2]

2. system의 성격과 model의 종류

2.1 system의 성격

우리가 어떤 system을 연구하든지 간에 우리가 관심을 갖는 system의 특성에 따라서 대체로 2가지로 분류할 수가 있다. [2]

(1) 첫째는 continuous system으로 우리가 관심을 갖는 system의 특성은 continuous variable이다. (整數가 아니고 水量이라든가 기온과 같이 小數로 나타나는 변수)

continuous system의 쉬운 예는 화학공장에서 찾아볼 수 있고 또 인공위성의 발사체도라든가 회전체도 또는 수력발전소 dam의 水量들을 들 수 있겠다. 이런 continuous system을 연구하자면 미분방정식을 포함하는 수학적 model을 사용해야 되고 원칙적으로 Analog computer를 사용하여 Analog Simulation을 해야 한다. (註: 보통 우리가 말하는 computer는 Digital computer이고 Digital computer로도 Analog Simulation을 흉내낼 수 있다).

(2) 둘째는 discrete system으로 우리가 관심을 갖는 event(사건, 事象)이 간헐적(discrete)으로 일어나며 이것은 한개 두개 셀 수 있는 discrete variable이다. discrete system의 쉬운 예는 서울역 대합실에서 기차를 기다리는 사람수라든가 전투상황에서의 전투원의 증감 또는 창고에 쌓인 물품의 증감등 산업 전반에 걸쳐서 찾아볼 수 있다.

2.2 Simulation model의 종류

우리의 관심에 따라서 system을 어떻게 분류하는냐에 따라서 system의 model도 결정된다.

(1) continuous system을 분석하자면 continuous Simulation을 하여야 하는데, 이 경우에는 Simulation

model자체를 확률적으로 표시하기에는 기술적으로 너무 힘들다. 따라서 通常 실제 system을 model로 추출하는 과정에서 통계적인 방법에 의해서 parameter등을 구한 후에는 Simulation 자체는 model에 포함되어 있는 각 부분들간의 상호작용을 계산하고 기록하는 정도에 그친다(deterministic).

(2) 이와는 달리 시간적으로 볼 때 간헐적으로 일어나는 event에 관심이 있을 때는 discrete simulation을 하는데, 문제의 성질에 따라서 deterministic한 입장을 택할 수도 있으나 이런 것은 Simulation이라기 보다는 “계산”이라고 불러야 하겠고, event의 확률적인 발생순간을 모의하기 위해서 “Monte Carlo” 방법을 이용해서 Simulation을 한다.

Monte Carlo방법이란 표본추출(statistical sampling)에 의한 Simulation을 말하는 것으로, 실제 system의 확률적 특성을 묘사하는 확률분포곡선을 이용해서 亂數(random number)로 부터 실제의 실험과 똑같은 확률분포특성을 갖는 一連의 event들이 발생되어 model에 포함된 logic들을 통과하면서 실제 system의 운용을 simulate하는 것이다.

(3) 승부놀이(games): 원칙적으로는 전에 말한 continuous Simulation이나 discrete Simulation이나 또는 어떤 종류의 program이든지를 이용해서 여러 team들이 각각 다른 政策을 사용하든가 parameter들을 사용해서 Simulation을 하여 그 결과를 가지고 승부를 결정하는 것을 말한다. 그러나 보편적으로 널리 이용되는 interactive computer game의 형태는 주로 business management game과 war game등을 들 수 있다. 이들 game의 특징은 “회화적”(interactive) 이어서 단계적인(sequential) 의사결정을 요한다. 과거의 정보에 기초를 둔 결정은 새로운 환경을 형성하고, 따라서 계속되는 결정에 영향을 미치게 된다. 따라서 game의 특징은 sequential decision, 攄환(feed back) 그리고 새로운 반응에 있다.

3. 간단한 예를 통한 Simulation절차

3.1 continuous Simulation

먼저 말한 바와 같이 continous Simulation을 하자면 Analog computer를 써야 하겠으나, 기술적으로 힘들고 또 digital computer의 발달로 Analog computer는 경영자와 공학도들의 손에서 멀어져 가고 있는 느낌이다. 여기서는 보통(digital) computer를 써서 간단한 문제를 simulate하는 예를 들겠다.

문제: 어떤 동네의 상수도계통은 수원지와 저수지로 구성되어 있다. 수원지에서 流出量은 수원지 水量÷4 이며 평균적으로 물이 수원지를 통과하여 저수지까지 흘러드는 데는 4일이 걸린다. (통계적으로 결정 됨). 이 상수도계통이 정상상태에 있다가 어느날 갑자기 하루에 100ton씩 비가 와서 2일을 계속하였다. 저수지 水位의 변화는(그림 1참조).

해: 이 system의 정상상태를 0으로 잡고, 오늘을 t , 내일을 $(t+1)$ 로 표시하면 이 system의 model은
수원지 수량 $(t+1)$ =수원지 수량 (t) +유입량 (t)

表 1. Simulation과정

日(t)	流入量	수원지	流出量	저수지	소비량
0	0	0	0	0	0
1	100	0	0	0	0
2	100	100	25	0	0
3	0	175	34.7	25	0
4	0	131.3	32.8	68.7	0
5	0	98.5	24.6	101.5	0
6	0	73.9	18.5	126.1	0
7	0	55.4	13.9	144.6	0
8	0	41.5	10.4	158.5	0
9	0	31.1	7.8	168.9	0
10	0	23.3	5.8	176.7	0

註: 정상상태물 0으로 놓았음.

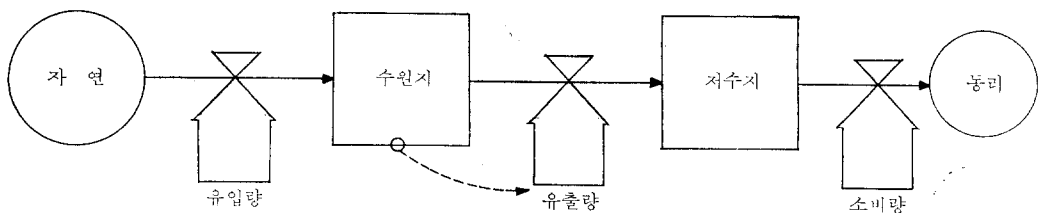


그림 1. 상수도 계통

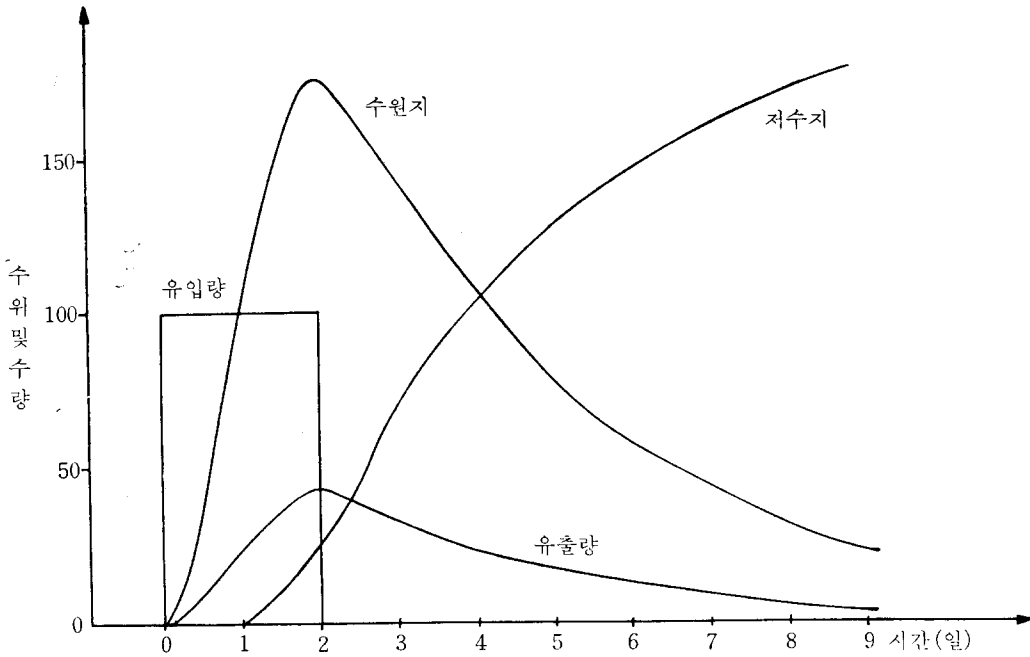


그림 2. 저수지 Simulation결과

-유출량(t)

$$\text{유출량}(t) = \text{수원지 수량}(t) \div 4$$

$$\text{저수지 수량}(t+1)$$

$$= \text{저수지 수량}(t) + \text{유출량}(t) - \text{소비량}(t)$$

이 Model에 t를 0에서부터 대입하여 축차적으로 계산을 하면 表 1 과 같고 저수지의水位 등이 그림 2 에 표시되었다.

3.2 discrete Simulation

discrete Simulation에서 중요한 것은 event의 확률적 발생을 simulate하는 것으로 필요한 절차를 통해 설명하겠다.

문제 : Poisson 분포를 갖고 毎分 1명꼴로 도착하는 고객을 service하는데 30초 내지 1분이 걸린다.

(uniform distribution). 대기행렬의 평균치는

해 : (1) 亂數(random number) 발생

주어진 문제가 확률적이기 때문에 주어진 확률분포로부터 표본추출을 해야 하는데 이때 사용되는 것이 亂數이며 가장 간단한 방법은 0에서 9까지의 숫자가 쓰여진 10면체의 주사위를 상상할 수 있고 이 주사위를 3번 굴려서 얻은 3자리의 수치를 소수점 이하의 수치로 생각하여 이런 실험을 계속하면 0.001부터 0.999까지 변화하는 亂數를 얻을 수 있다. 실제 computer에서는 Multiplicative Congruential Method 즉

$$\text{다음亂數} = \text{먼저亂數} \times a \text{를 } b \text{로 나눈 나머지}$$

를 쓰며 a로는 5자리 이상의(0이나 1이 계속되지 않는) 숫자, b로는 2를 computer기억장치의 1단어의 bit수로 자승한 수가 통상 쓰인다. 또한 대부분의 산업공학 분야의 서적 끝에 부록으로 나와 있는 亂數表를 사용해도 된다. 예를들어 104, 465, 225, 616, 061, 594, 711, 568, 253, 178...라는 亂數들을 얻었다. 가정하자.

(2) Monte Carlo方式에 의한 분포곡선으로부터의 표본추출

① 도착시각

고객들의 도착시각이 Poisson 분포를 갖는다면 고객도착시간 간격(t)는 negative exponential분포를 갖게 되니까 $e^{-1 \times t} = \text{亂數}$ 의 관계에서 $t = -L_n(\text{亂數})$ 로 표본을 추출할 수 있다.

도착 순서	亂數	도착시각 간격 = t	도착시각 = Σt
1	.104	2.26	2.26
2	.465	.77	3.03
3	.225	1.5	4.53
4	.616	.48	5.01
5	.061	2.8	7.81

② service시간

service시간(τ)는 나머지 亂數들을 이용하여 $\tau = \text{亂數}/2 + 0.5$ 의 관계를 사용하면

service순서	亂數	service시간 = τ
1	.594	.797
2	.711	.855
3	.568	.784
4	.253	.627
5	.178	.589

③ 결과계산

이렇게 亂數로 부터 정해진 도착시각과 service 시간이 실제로 운영된 실예중의 한 표본이라 생각하고 그림으로 표시하면 그림 3 과 같이 되고 여기서 system內的 평균 고객수는 graph의 총면적 3.983을 Simulation시간 8.339로 나눈 0.47人 이고, 대기 고객수는 graph에서 1과2 사이의 부분의 총면적 0.331을 Simulation 시간으로 나눈 0.04人이다. 이런 방법을 사용하여 service가 2배로 저하되었을 때의 영향등 많은 사항을 분석할 수가 있다. 한가지 부연할 것은, 실제 Simulation에서는 확률적 실험빈도를 높이기 위해서 훨씬 많은 표본을 추출해야 한다.

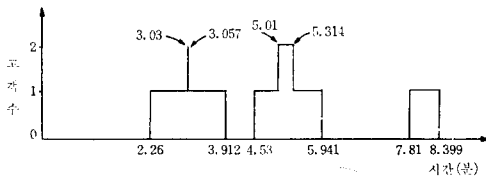


그림 3. 대기행렬 Simulation 결과

3.3 games

원칙적으로 어떤 Simulation에도 game으로 사용할 수 있으나 간단한 예를 들자면 일정한 量의 초기 자본을 가진 사람이 증권투자를 하는 game을 들 수 있겠다. 이런 game에서는

① 일정한 量의 초기자본을 가상적으로 할당받은 사람이 computer terminal을 사용해서

② Monte Carlo 方式으로 亂數를 기초로 하여 미리 정해진 확률분포에 의해서 그날의 주가가격이 확률적으로 결정되는 Simulation Program에 의해서 computer terminal에 적혀 나올 때

③ 그날의 가격과 과거의 가격을 기초로 하여 그 추세를 판단하여 사든지, 팔든지 등의 결정을 terminal을 이용해서 적어넣는 것을 계속하면

④ 주어진 시간 동안의 증권투자 행위를 simulate 할 수 있고, 그 결과를 다른 사람과 비교할 수도 있다.

4. Simulation language

최근 Computer Simulation이 일반화 되어감에 따라 미리 program된 Simulation언어가 개발되고 있다. 이들 Simulation언어의 목적은 Simulation model을 설계하기 위한 일반적 構造를 제공하고 Simulation model을 Computer Program으로 신속히 전환할 수 있게 하는데 있다. 또한 여러가지 相異한 Simulation언어가 있는 것은 이용자의 편의를 위하여 각각의 언어가 특정한 System에 적용될 수 있게 함으로서 Simulation절차를 보다 더 자동화하려는 것이다.

예를 들면 DYNAMO와 SIMPATE는 본래 一連의 수학 방정식을 포함하는 計量경제 model로서 형성된 대규모 經濟體係를 모의하기 위해서 만들어진 continuous Simulation program이며 GPSS와 SIMSCRIPT는 日程計劃과 待機行列 문제를 풀기 위해서 만들어진 discrete Simulation program이다.

이들 두가지 역할을 다 할 수 있는 것이 GASP이며 FORTRAN으로 작성되어 있기 때문에 사용자가 편리하게 다시 확장하여 program 할 수 있는 이점이 있다.

game으로 쓰여진 program중에는 여러 종류의 business game과 ATLAS(A Tactical, Logistical, and Air Simulation), BALFRAM(Balanced Force Requirements Analysis Methodology), TCM(Theater Combat Model), TWSP(Tactical Warfare Simulation Program) 등의 war game이 있고 PROSIM(Production System Simulator)이라는 discrete Simulation program은 일반적인 생산관리 system總體의 운영을 simulate 할 수 있는 program으로 경영 game으로 쓰여지기도 한다.

4.1 DYNAMO

이는 M.I.T. 계산소에서 미분방정식을 포함한 수학적 model을 simulate하기 위해서 개발된 것이며 數表와 圖表가 同時에 作圖된다. 여기에는 Simulation에 특유한 논리적 모순에 관하여 광범위한 오류의 검토·방법이 포함되고 있으며, 상이한 조건下에 있는 model을 검토하기 위하여 여러가지 편리한 細部節次(subroutines)가 포함되어 있다.

그림 4 와 5는 DYNAMO를 이용하여 생산-재고-계통을 simulate한 예인데 생산량과 재고량을 계획하는 특정한 경영정책下에서 고객의 주문량이 10% 증가했을 때 각 부분간의 정보지연 때문에 빚어지는 파급효과를 일목요연하게 알 수 있다[3]. 여기서 정보지연을 줄인다거나 생산량과 재고량 계획

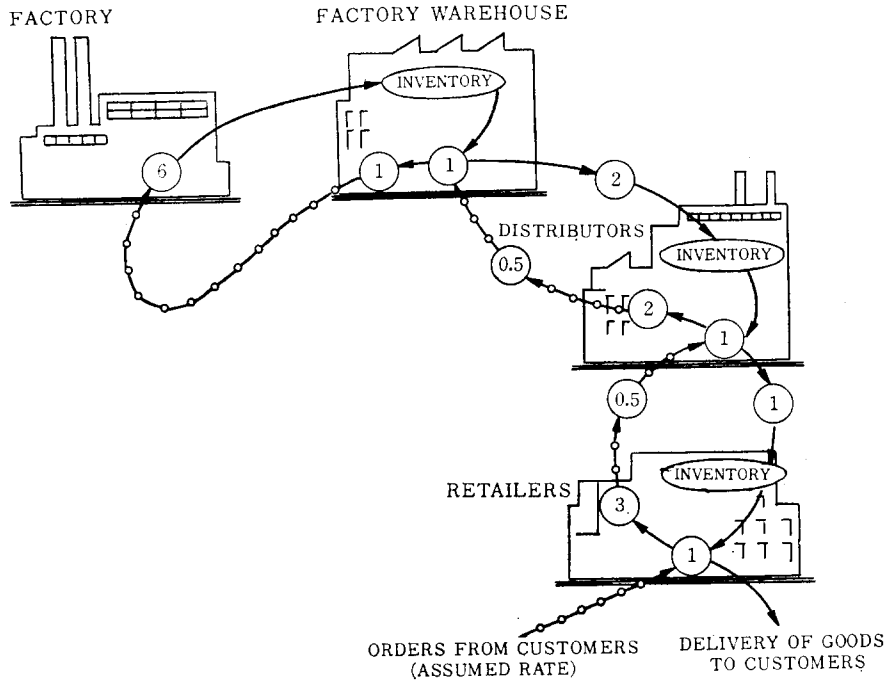


그림 4. 생산분배 계통도

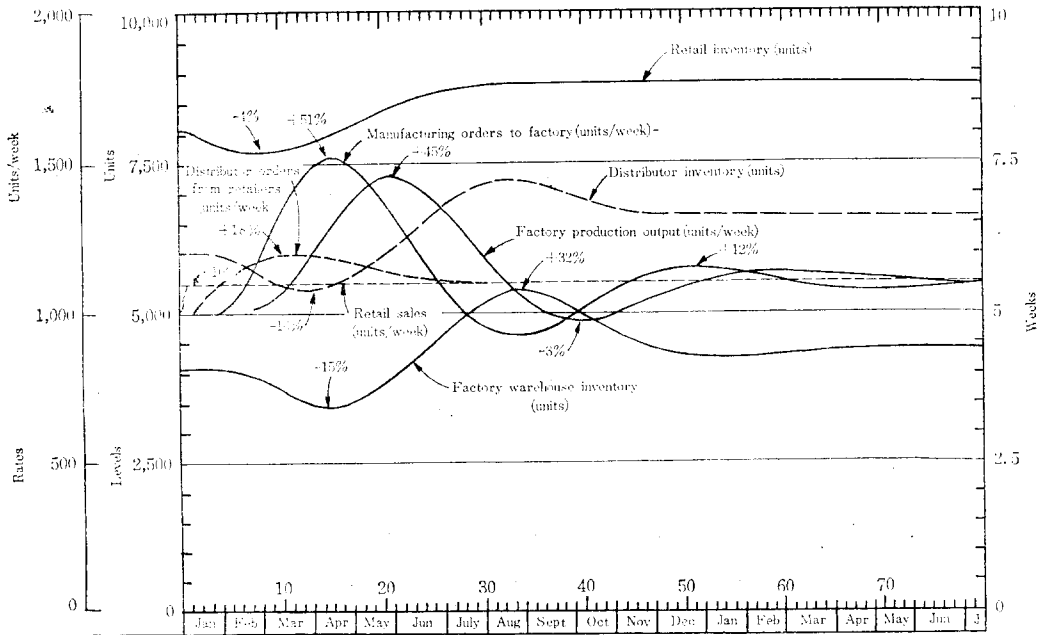


그림 5. 주문량이 10% 증가했을 때 생산 분배 계통의 반응

表 2.

GPSS

Storage	Capacity	Average Contents	Average Utilization	Entries	Average Time/Tran	Current Contents	Maximum
Men	3	2.185	.728	1924	7.087	—	3
Nowon	50	49.182	.983	1974	155.471	50	50

을 바꾸어 주면서 여러번 simulate하면 가장 적합한 정책들이 결정될 것이다.

4.2 GPSS(General Purpose Simulation System)

이는 자동화된 Simulation을 실시하기 위하여 비교적 고정된 절차를 요구하는 program이지만, 광범위한 각종 system에 적용되는 program이다.

그림 6과 7은 GPSS를 이용하여서, 재봉틀 50대가 필요한 피복공장에서 3대를 더 외부에서 빌리고 3명의 수선공이 있을 때의 가동율을 모의한 것이며 그 결과가 表 2에 나와있다. [4] 이들 정책을 변경시켜 가며 여러번 simulate하면 몇대를 더 빌리고 수선공은 몇명이 경제적으로 가장 적당한가가 밝혀질 것이다.

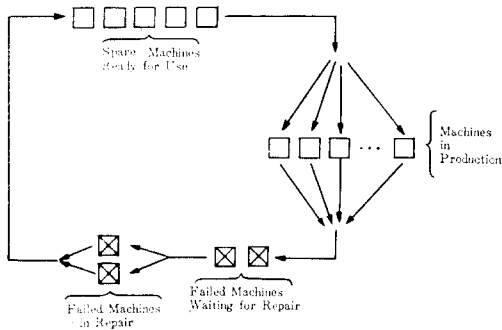


그림 6. 피복공장의 장비 순환도

5. Simulation응용례

그동안 Simulation은 산업, 군사등 많은 분야에 넓게 이용되어 왔다. 그중에 몇개를 예로 들어 본다 면 [5]:

- ① 많은 항공회사가 어떤 큰 비행장을 같이 쓰고 있을 때, 각 항공회사의 운영 상태를 simulate하여 適正 정비시설과 능력, 착륙시설, 예비 비행기數, 등 政策을 결정.
- ② 네거리의 車輛통행을 Simulate하여 自動신호등의 適正 時間간격을 결정.
- ③ 정비업무를 simulate하여 適正 정비人員을 결정.

④ 한 國家의 經濟를 simulate하여 政策결정의 영향을 豫測.

⑤ 大規模 戰鬪를 simulate하여 攻防 무기체계를 평가.

⑥ 大規模 在庫 및 分配體系를 simulate하여 체제 구조를 改善

⑦ 한 기업의 전반업무를 simulate하여 政策이나 운영변경의 영향을 평가하고, 경영진을 훈련하기 위하여 business game으로 사용.

⑧ 通信體系를 simulate하여 저렴한 비용으로 주어진 需要를 충족시키기 위한 각 部品の 容量결정.

6. 결 론

Simulation은 진실로 대단히 편리한 도구이다. 그러나 절대로 이것이 “만병통치”는 되지 못한다. Simulation은 원래 통제학적인 근사치를 구하는 방법이고 適正정책이 직접적인 해로 나오는 것이 아니라 단지 몇개의 代案을 비교하는 데 그친다. 또한 Simulation이란 어디 까지나 計量的인 기법이기 때문에 모든것이 數値로 나타나고, 숫자로 나타나지 않는 인과관계나, 또는 modeling단계에서 빠뜨린 부분에 대한 영향은 알 수가 없다.

그럼에도 불구하고 Simulation은 運營연구에 있어서 대단히 편리한 도구이고 다른 方法으로 해결할 수가 없는 많은 運營문제에 있어, 없어서는 안될 중요한 기법이다.

REFERENCES

1. Thieraurt, R.J. & R.A. Grosse, *Decisimn Making Trough Operations Research*, John Willey & Sons, Inc. (1971)
2. Schmidt, J.W. & R.E. Taylor, *Simulation and Analysis of Industrial Systems*, Richard D. Irwin, Inc, (1970)
3. Forrester, J., *Industrial Dyamics*, M.I.T. Press, Cambridge, Mass. (1961)
4. Schriber, T.J., *Simulation Using GPSS*, John Wiley and Sons, (1974)
5. Hillier, F.S. and G.J. Lieberman, *Introduction to Operations Research*, Holden-Day Inc. (1963)

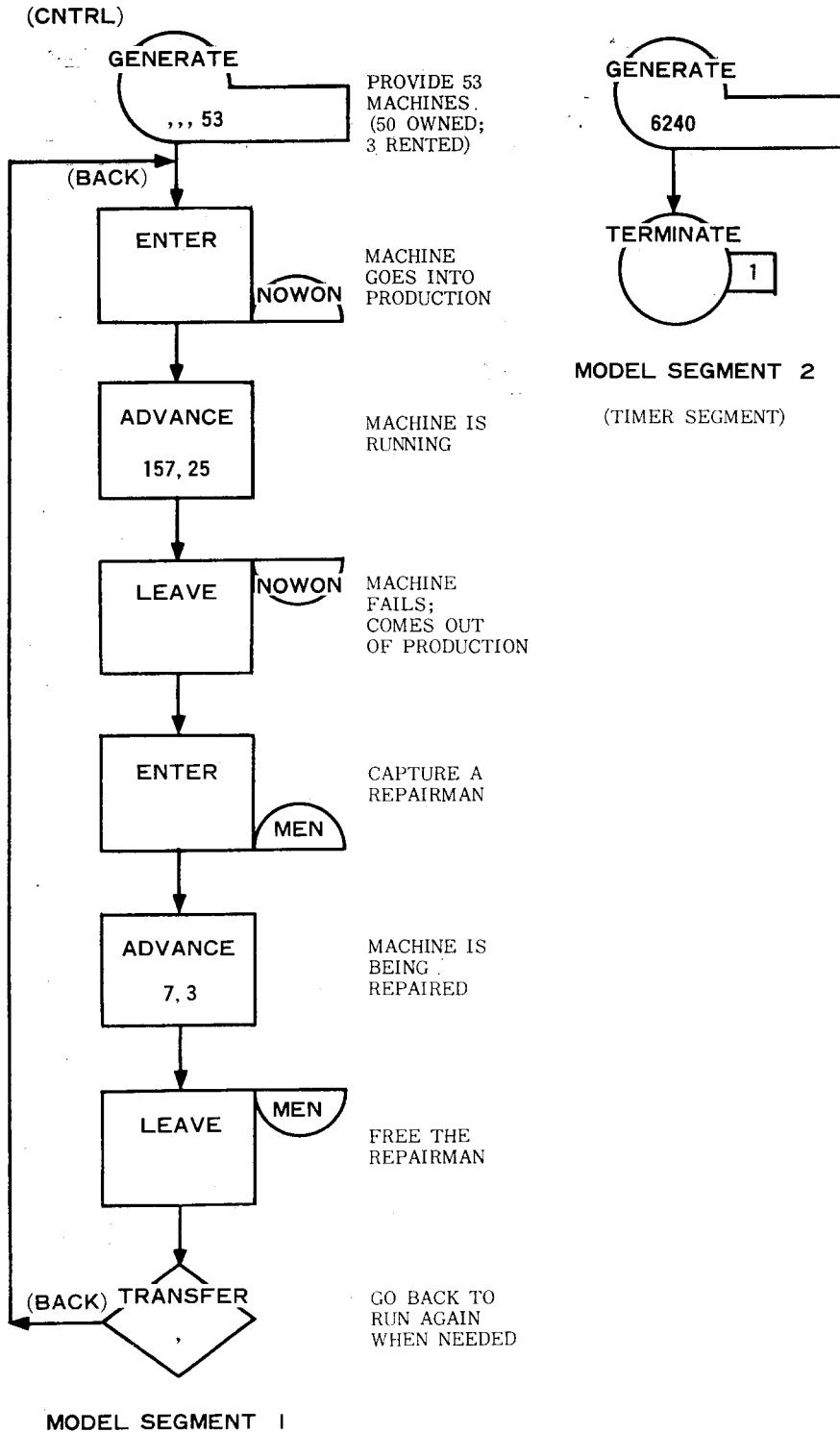


그림 7. GPSS Block Diagram