

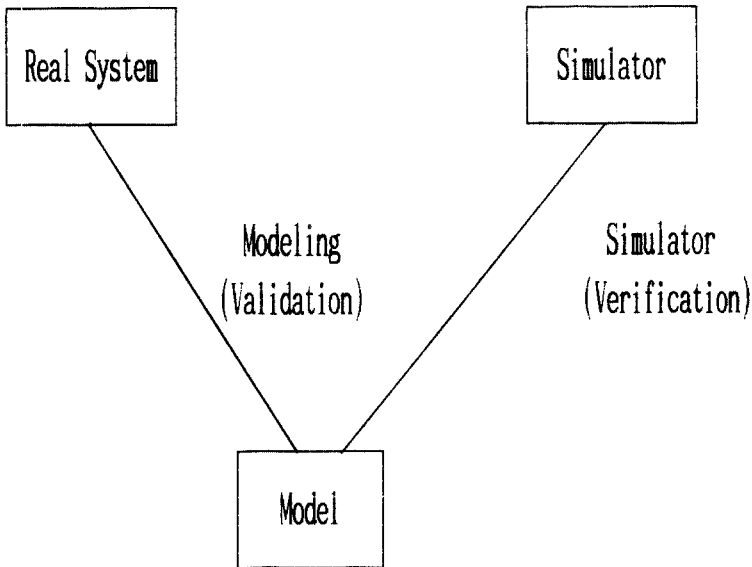
시뮬레이션 모델링 및 타당성 검사

백두권
고려대학교 전산과학과 교수

1. 서론
2. 모델링과 시뮬레이션의 기초
3. 모델링과 시뮬레이션의 이론
4. 모델링의 타당성 검토
5. 시뮬레이션 모델링 월드뷰
6. 결론

제 1 장 서 론

▶ 모델링과 시뮬레이션의 Framework(Zeigler)



- 3 Entities : Real System
Model
Simulator

- 2 Relations : Modeling (Abstraction)
Simulation (Experimentation)

- 2 Issues : Validation
Verificaton

제 2 장 모델링과 시뮬레이션(Modeling and Simulation)의 기초

2.1 모델링과 시뮬레이션의 필요성

- ① 시스템이나 문제를 정의하는 설명적 장치(explanatory devices)
- ② 한계적(critical) 구성요소를 결정하는 분석도구(analysis vehicles)
- ③ 제안된 해결방안을 종합하고 평가하는 설계평가도구(design assessors)
- ④ 미래의 개발계획을 예측하고 도와주는 예측적인 도구(predictors)

2.2 시스템이란

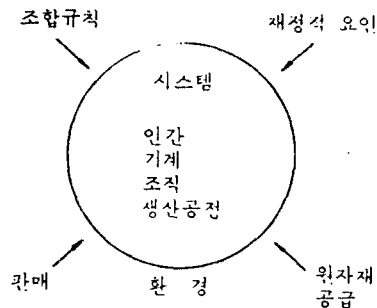
▶ 시스템의 정의

- ① 관심의 대상이 되는 실세계의 제한된 부분에 속해 있는 관련된 요소들의 집합
- ② 어떤 목적을 위해 하나 이상의 서로 관련있는 구성요소가 결합된 것

▶ 외적 요인(external factor)들이 시스템에 영향을 미칠 때는

- ① 시스템 정의를 확장시켜 외적 요인들을 포함한다.
- ② 외적 요인들을 무시한다.
- ③ 외적 요인들을 시스템의 입력으로 간주한다.

▶ 예



- 외적요인은 판매실적치나 예측치에 대한 통계적 서술등을 포함

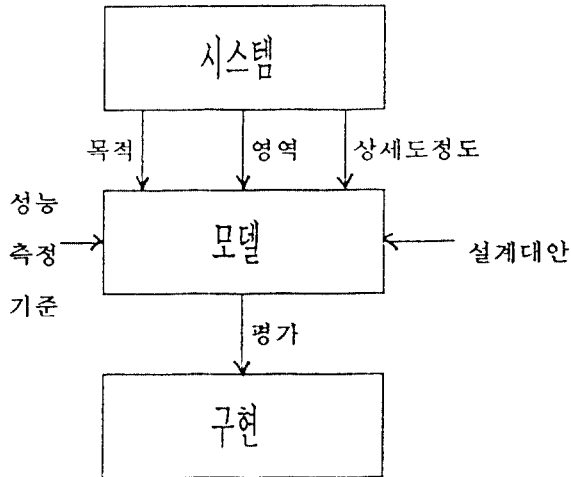
2.3 모델(Model) 과 모델링(Modeling)

- ▶ 모델 - 이론적인 원리에 근거한 실제 시스템의 서술
- ▶ 모델링 - 실제 시스템에 관한 모델의 구성
- ▶ 모델링이 쉬운 경우
 - ① 시스템에 부수되는 물리적인 규칙이 이용가능한 경우
 - ② 시스템에 대한 도형적 표현이 가능한 경우
 - ③ 시스템의 입력, 출력, 구성요소의 변화가 통제 가능한 경우
- ▶ 모델링이 어려운 경우(복잡한 대형 시스템)
 - ① 이용가능한 기본규칙이 거의 없다.
 - ② 기술하고 표현하기 어려운 많은 절차적인 요소가 포함되어 있다.
 - ③ 정량화하기 어려운 정책적인 입력들이 요구된다.
 - ④ 랜덤(random) 구성요소들이 중요한 원인이 된다.
 - ⑤ 인간의 의사결정이 시스템의 절대적인 부분을 차지한다.

=> 시뮬레이션을 이용하여 위의 어려움을 해결

2.4 모델링 과정(Modeling Process)

▶ 모델링 환경



▶ 모델링 단계

- ① 기술된 문제나 프로젝트 목적에 근거한 모델링 목적의 수립
- ② 목적에 근거한 시스템의 영역과 모델링의 상세한 정도가 작성
- ③ ①, ②에 의해 추상화된 모델이 생성
- ④ 모델에는 평가되어야 할 성능 평가 척도와 설계대안들이 포함
- ⑤ 명시된 성능 평가 척도로 표현된 설계대안에 대한 평가들을 모델의 출력으로 간주
- ⑥ 대안들의 평가를 통하여 시스템에 대한 권고안이 수립되면 구현단계가 시작
- ⑦ 구현은 구체적인 일련의 권고안을 이용하여 잘 정의된 환경에서 이루어짐

2.5 시뮬레이션의 정의와 활용

▶ 정의 : 실세계 시스템을 수학적이고 논리적인 모델로 설계하고,
컴퓨터를 이용하여 모델을 통해 시험하는 과정

▶ 활용

(1) 시뮬레이션을 통한 실세계 시스템의 추론

- ① 실시스템에 대한 실질적인 구축없이도 시뮬레이션을 이용하여 평가
- ② 비용이 많이 들고 실질적인 실험을 하기에 위험한 운영시스템인 경우
시뮬레이션을 이용해 쉽게 해결
- ③ 실험의 목적이 강도(stress)의 한계를 결정해야 하는 경우, 시뮬레이
션을 이용해 시스템 파괴의 위험을 배제할 수 있음.

(2) 시뮬레이션을 설계, 절차적 분석, 성능평가등에 이용

(3) 시뮬레이션은 시스템의 특성을 나타내는 시스템 상태 서술에 의해 표현된
모델을 시간 또는 운영규칙에 따라 어떤 상태에서 다른 상태로 이동하는
동적인 형태를 관찰하는 것임.

2.6 시뮬레이션의 예

▶ 단일 창구의 은행 문제

- 시스템의 상태 : 창구의 상태(분주 또는 유힬)와 고객수에 의해 정의
- 상태 변화의 경우
 - ① 고객이 은행에 도착
 - ② 고객이 서비스를 모두 받고 시스템을 떠남
- 시뮬레이션
 - : 고객이 시스템에 도착하는 사건과 시스템을 떠나는 사건을 순차적인 순서로 처리함으로써 시스템의 시간에 따른 상태를 결정
- 시간에 대한 각 상태변수들의 합

▶ 중요한 개념

- 시뮬레이션 시간 동안 어떤 순간에서라도 모델은 특별한 상태에 있음
- 사건이 발생하면 모델의 상태는 사건과 관련된 논리적이고 수학적 관련성에 의해 규정된 대로 변경
- 사건들은 모델의 동적인 구조를 정의하는 효과를 가짐
- 상태변화는 여러개의 관점(뷰)에서 볼 수 있음

예) 은행 시스템

- ① 고객이 서비스를 원할 때 고객이 부딪히는 프로세스(고객의 뷰)
- ② 창구의 상태를 변화시키는 사건들(은행의 뷰)

2.7 시뮬레이션 과정

▶ 성공적인 모델 개발 전략 :

비교적 간단한 모델을 이용하여 시작하되 문제해결 요구사항을 수용할 수 있도록 점진적으로 모델을 상세화해 나감

▶ 시뮬레이션 과정

- ① 문제의 공식화 : 문제 해결의 목적을 포함하여 연구하고자 하는 문제를 정의
- ② 모델 구축 : 문제의 형식화에 따라서 시스템을 수학적이고 논리적인 관계로 추상화함
- ③ 자료 획득 : 자료를 파악하고 명세화하고 수집함
- ④ 모델 변환 : 컴퓨터에서의 처리에 적절하게 모델을 준비함
- ⑤ 검증 : 의도한대로 컴퓨터 프로그램이 수행될 수 있도록 준비함
- ⑥ 타당성 검토 : 시뮬레이션 모델과 실시스템 간의 관계가 정확성이 있는가, 또는 일치하는가를 검토함
- ⑦ 실험 계획 수립 : 모델을 실험하기 위한 제반 계획을 수립함
- ⑧ 실험 : 실험 결과를 얻기 위하여 시뮬레이션 모델을 수행함
- ⑨ 결과 분석 : 문제 해결에 대한 추론을 얻고 권고안을 작성하기 위하여 시뮬레이션 실험 결과를 분석함
- ⑩ 구현과 문서화 : 시뮬레이션 결과로부터 도출되는 결정을 구현하고, 모델과 모델 이용에 대한 문서화를 함

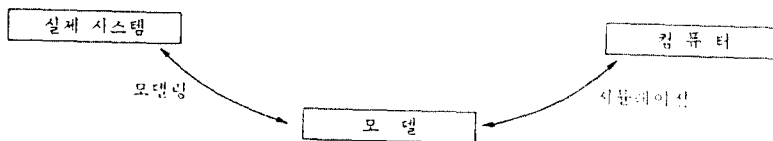
제 3 장 모델링과 시뮬레이션의 이론(Theory)

3.1 서론

▶ 모델링과 시뮬레이션

: 실제 시스템에 관한 모델을 구성하고, 이것들을 컴퓨터 상에서 실험하기 위한 제반 행위

▶ 모델링과 시뮬레이션의 기본 요소 및 상호 관계



- 실제 시스템 : 모델링의 대상이 되는 현실 세계의 시스템

- 모델 : 행위 데이터를 생성하는 명령어들의 집합

예) 오토마타 이론에 따른 표기법으로 표현,

이산 사건 형식론, 미분 방정식 등에 의해 표현

- 컴퓨터 : 프로그램으로 코드화된 모델-명령어들을 수행하는 디지털 컴퓨터
 => 사람이거나 기계이거나 간에 모델 명령어가 적당한 형태로 제공될 때,
 행위 데이터를 생성할 수 있는 하나의 계산 과정으로 파악해야 함

▶ 세가지 요소간의 관계

- 모델링 관계 : 모델의 타당성, 즉 모델이 실제 시스템을 얼마나 잘 나타내고 있는가와 관련되는 것으로 이는 실제 시스템과 모델이 생성한 데이터 사이의 일치도에 의하여 평가됨

▶ 시뮬레이션 관계 (simulation relation)

: 컴퓨터가 모델에 의해서 의도된 명령어들을 수행하는 “충실성”과 관련

▶ 모델을 사용하는 이유

- 현실 세계의 시스템을 직접 이용하여 실험시

-> 비용과 시간의 낭비

-> 현실적으로 불가능

- 시뮬레이션 : 반복 가능, 비파괴적, 분석 용이

. 과학 위주의 모델 : 구조적으로 타당한 모델에 초점

. 응용 위주의 모델 : 모델을 어떻게 만들 것인가에 관련

- 실제로 한 언어를 가지고 모델을 서술할 수 있어야 함

- 복잡한 모델을 자신의 여누에 적합한 모델로 단순화 시킬 수 있어야 함

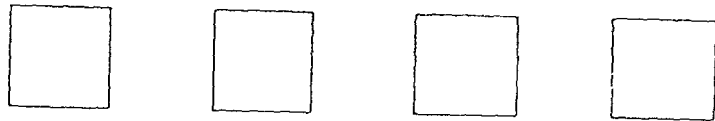
▶ 모델의 효율적 의사 소통시 포함되어야 할 사항

- 1) 모델에 관한 비형식적인 서술과 모델 구성을 위한 가정
- 2) 모델 구조의 형식적인 서술
- 3) 시뮬레이션을 수행하는 프로그램 제시
- 4) 시뮬레이션의 수행, 결과, 분석의 제시
- 5) 모델의 응용 범위와 그 타당성, 그리고 수행 비용에 관한 평가
- 6) 현재의 모델과 과거 혹은 미래의 다른 모델과의 관련성 검토

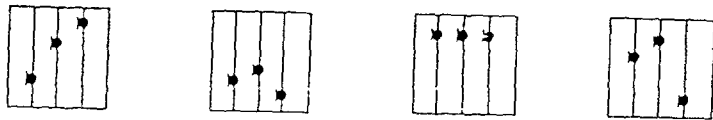
3.2 모델의 비형식적 서술

▶ 하나의 모델을 비형식적으로 제시하는 적절한 방법

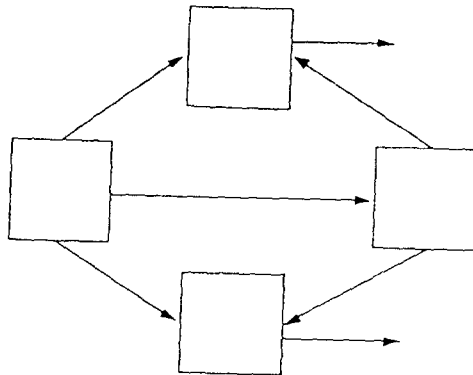
⇒ 모델의 구성 요소들과 구성 요소별 서술 변수 (descriptive variable),
 구성 요소들 사이의 상호 작용 (component interaction)을 서술



구성요소



서술변수



구성요소 상호작용

- 구성 요소 : 모델을 구성하고 있는 요소들을 의미
- 서술 변수 : 어느 한 시점에서의 구성 요소들의 상태를 나타내는 수단으로서 모델의 특성을 나타내는 매개변수를 포함
- 구성 요소들 사이의 상호 작용 ; 구성 요소들 간에 상호 영향을 미치도록 하는 규칙들의 의미

3.2.1 비형식적 서술의 예

[예제 3.1] 구성 - 5대의 터미널

1대의 CPU

운영방식 - round robin 방식

시계 방향의 순서로 사용자가 CPU에게 데이터를 보낸 후
응답을 받으면, 그 다음 사용자에게 순서가 넘어간다.

1) 구성요소

CPU, USER1, USER2, USER3, USER4, USER5

2) 서술변수

CPU

WHO.NOW - {1,2,3,4,5}의 범위를 갖음 ; WHO.NOW = i는 USER_i가 CPU의 서
비스를 받음을 의미

USER_i (i=1,2,3,4,5)

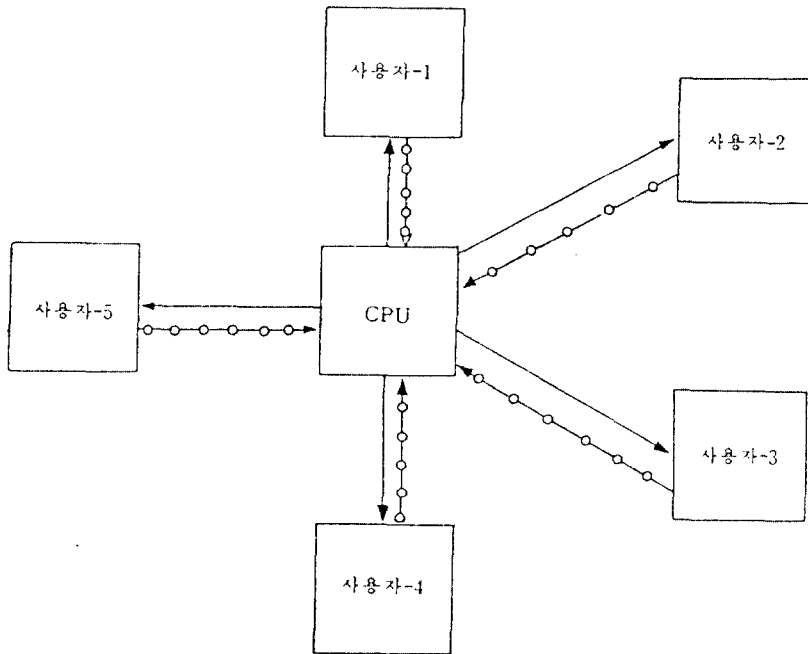
COMPLETION.STATE=[0,1] (0에서 1까지의 실수집합)의 범위를 갖음 ;

COMPLETION.STATE=S는 해당 USER가 자신의 프로그램을 S (0≤S≤1)만큼 완
성시켰음을 의미함 (0은 시작, ½은 반환성, 1은 완성을 각각 의미)

PARAMETER a_i=[0,1]의 범위를 가짐 ; USER i가 작업을 완성하는 속도 (비율)

3) 구성요소들 사이의 상호작용

- a. CPU는 일정한 속도로 차례차례 각 USER에게 서비스를 한다. 따라서 서술변수 WHO.NOW는 1, 2, 3, 4, 5, 1, 2, 등으로 순환하는 값을 갖는다.
- b. USER i 의 차례가 되면 (즉, WHO.NOW가 i 값을 가지면) USER i 는 나머지 일의 a_i 만큼을 완성시킨다. 따라서 USER i 의 COMPLETION-STATE가 s_i 였을 경우 s 에서 $s+a_i(1-s)$ 가 된다.



[예제 3.2]

- ▶ 모델 - 1) 국가 : 정부와 국민으로 구성
 - 2) 정부 : 정권을 정당에 의해 운영
 - 3) 정당 : 자유당이거나 부수당
 - 4) 정책 : 권력을 가진 정당이 결정 - 관대하거나 강제적임
 - 5) 시민쟁의 : 정부의 행위에 대한 국민의 반응 - HIGH 혹은 LOW

▶ 비형식적 서술

1) 구성요소

GOVERNMENT, PEOPLE

2) 서술변수

GOVERNMENT

PARTY : { CONSERVATIVE, LIBERAL } : 정부의 정치적 영향을 표현

POLICY : { PERMISSIVE, COERCIVE } : 정부가 따르는 정책 종류

PEOPLE

CIVIL.STRIFE : { LOW, HIGH } : 국민들의 불만 상태 표현

3) 구성요소의 상호작용

P1 : COERCIVE 정책은 다음 해에 반드시 CIVIL.STRIFE를 HIGH가 되게 함

P2 : PERMISSIVE 정책은 항상 시민 불만도를 LOW로 유지

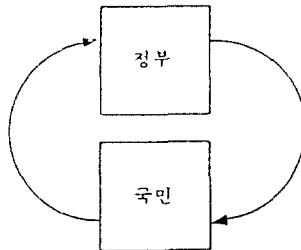
P3 : PARTY는 CIVIL, STRIFE가 LOW이면 계속 권력 유지,
HIGH이면 바뀜

P4 : CONSERVATIVE 정부는 일단 권력을 갖게 되면 결코 POLICY를 바꾸지
않음. 이전의 정당이 취한 POLICY도 바꾸지 않음

P5 : LIBERAL 정부는 PERMISSIVE 법안을 통과시켜 HIGH CIVIL, STRIFE에
대응. 평온한 상태가 된 후에는 COERCIVE한 태도로 변함

* 공리 : 정부 정책의 변화에 어떻게 반응하는가

정부가 국민의 반응에 따라서 어떻게 정책을 결정하는가



3.2.2 비형식적 서술의 일반화

1) 구성요소

COMPONENT, A

COMPONENT, B

.

.

.

COMPONENT, Z

실제 시스템의 개념적 요소에 구성요소를 연결시키는
일반적 기술

2) 서술변수

COMPONENT, A

VARIABLE, A1

VARIABLE, A2

.

.

VARIABLE, An

COMPONENT, B

VARIABLE, B1

각 변수의 범위 : 이 범위(집합) 중 임의의 원소를
나타내는 기호는 변수가 각각에 연결된 구성원소를

간략하게 서술하는 역할을 함

VARIABLE, Bn

COMPONENT, Z

VARIABLE, Z1

.

.

VARIABLE, Zn

PARAMETERS

PARAMETER, 1

PARAMETER, 2 각 매개변수의 범위 ; 이 범위(집합)중 임의의 원소를 나타내는 기호는 각 매개변수가 모델의 구조를 기술하는 역할을 함

.

.

PARAMETER, n

3) 구성요소의 상호작용

{ 규칙과 공리 및 법칙에 의해 비형식적으로 서술되는 구성요소 간의 효과, 영향, 행동 및 의사소통 }

4) 도형

{ 구성요소는 사각형으로 표시된다. 구성요소 B에 대한 구성요소 A의 영향은 사각형 A에서 사각형 B로 연결시키는 화살표에 의해 표시된다. }

3.3 모델의 형식적 서술

▶ 비형식적 서술의 문제점 : 불완전, 불일치, 모호함

- 불완전성 : 설계자가 현실 시스템의 제반 사항들을 완전히 서술하지 못함으로써 기인
- 불일치성 : 비록 설계자가 모든 사항들을 고려하였다고 하여도, 하나 이상의 규칙들을 동일한 상황에 적용하는 경우
- 모호성 : 하나의 특정한 상황에서 여러가지 가능성이 제시됨으로써 설계자가 어느 것을 의도했는지가 불분명하게 될 때

3.3.1 모델의 분류

- ▶ 시간적인 측면
 - 연속시간 모델 (continuous time model)
 - 이산시간 모델 (discrete time model)

- ▶ 서술변수의 범위 측면
 - 이산상태 모델 (discrete state model)
 - 연속상태 모델 (continuous state model)
 - 혼합상태 모델 (mixed state model)

- ▶ 확률변수의 구체화 방법에 따라
 - 확정모델 (deterministic model)
 - 확률모델 (probabilistic model) 혹은 추계모델 (stochastic model)

- ▶ 현실 세계의 시스템이 자신의 주변환경과 상호작용하는 측면
 - 자율모델 (autonomous model)
 - 비자율모델 (nonautonomous model)

- ▶ 상호작용규칙이 시간에 종속되는지의 여부에 따라
 - 시간 독립 모델 (time invariant model)
 - 시간 종속 모델 (time varying model)

3.3.2 시스템 명세로서의 모델

▶ 시스템 명세

- 특정한 프로그래밍 언어와 무관하게 모델의 명령어들을 표현하는 방법
- 시스템의 요소들을 간단히 서술
- 비형식적 서술 -> 시스템 명세 : 시스템들이 상호 연결된 네트워크 형태를
 취함
 - . 정적 구조 : 구성요소, 서술변수
 - . 동적 구조 : 구성요소의 상호작용과 영향 표시도 (influence diagram)

3.4 모델링과 시뮬레이션을 위한 구현요소

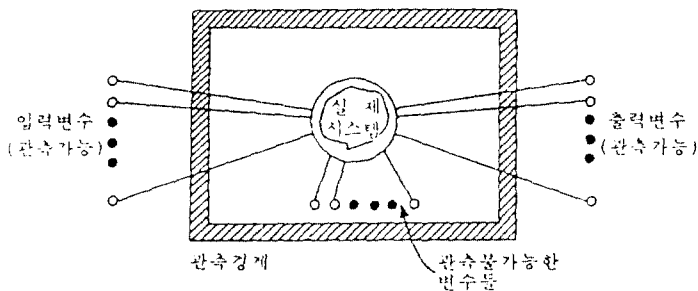
▶ 모델링과 시뮬레이션의 구현요소

- . 실제 시스템 (real system)
- . 실험 프레임 (experimental frame)
- . 기본 모델 (base model)
- . 총괄 모델 (lumped model)
- . 컴퓨터 (computer)

3.4.1 실제 시스템 : 행위 (behavior)

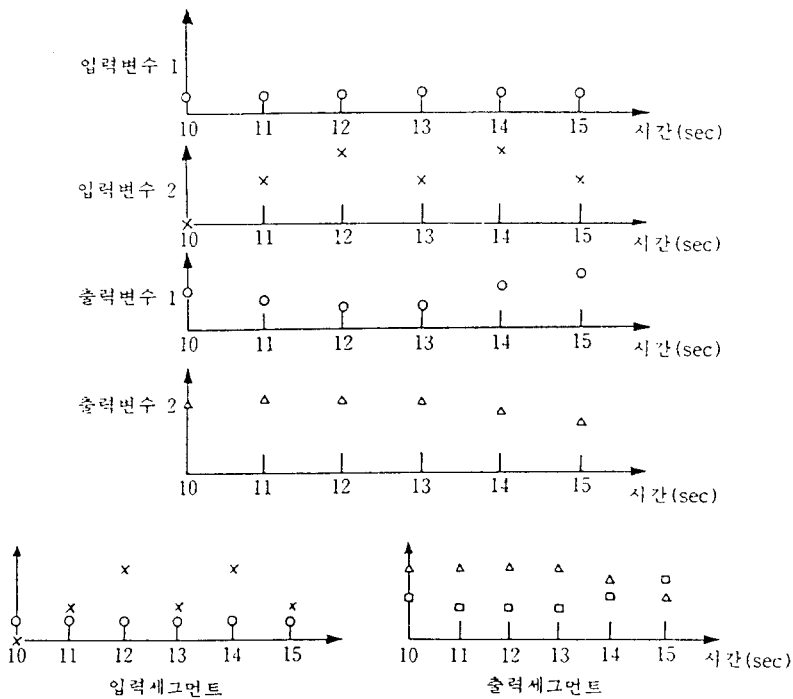
▶ 실제 시스템

- 관측 가능한 현실 세계의 자료에 지나지 않음
- 종류 : 자연적인 것, 인위적인 것, 혼합 형태



▶ 관측가능한 변수

- 입력변수 : 시스템의 외부로부터 영향을 주는 것으로 실험자는 어느 한 시점에서 이 변수들을 제어 가능
- 출력 변수 : 입력 변수에 대응되는 시스템의 결과



- 입력 세그먼트 (input segment), 입력 궤적 (input trajectory)
: 모든 입력 변수도표들의 합
- 출력 세그먼트 (output segment) : 모든 출력 변수 도표들의 합
- 입출력 행위 (input-output behavior)
: 실험으로 얻을 수 있는 모든 가능한 입출력 세그먼트의 쌍

3.4.2 실험-프레임 : 타당성 (validity)

- 관측되거나 실험되어질 환경들의 제한된 집합
- 실제 시스템의 입출력 행위의 한 부분
- 상대적으로 간단한 모델로 하여금 입출력 쌍을 생성하도록 하되,
그것들이 비교의 기준이 허용되는 범위내에서 프레임의 입출력 쌍과 일치하도록 만들 수 있다.
- 한 모델에 대한 타당성은 하나의 실험 프레임과 입출력 쌍이 일치하는가를 측정하는 비교기준과 관련됨

3.4.3 기본 모델 : 가정의 완전 설명

- 실제 시스템의 입출력 행위를 설명하는 모델
- 기본 모델은 모든 가능한 실험-프레임 위에서 타당해야 함

3.4.4 총괄 모델 : 단순화 (simplification)

- 실제 시스템 내의 기본 모델등으로부터 적절히 구성요소들을 결합하고 상호작용들을 단순화하여 얻는다.

3.4.5 컴퓨터 : 복잡성 (complexity)

- 총괄 모델의 입출력 쌍들이 생성되도록 돕는 계산 장치
- 시뮬레이션 : 입출력 쌍들의 궤적이나 그 성질들은 어느 가상된 시간에서 다음 시간 사이에 생성되는 과정
- 컴퓨터는 총괄 모델에서 제공되는 명령어들을 수행하는데, 이 명령어들은 프로그램을 구성
- 컴퓨터와 인간 모두에게 이용가능한 자원들에 대한 제한
 - 1) 하나의 단계를 시뮬레이션하는데 걸리는 시간
 - 2) 모델의 구조를 표현하고 상태변수의 값을 저장하기 위하여 필요한 저장 공간
 - 3) 모델을 프로그램으로 표현하거나, 컴파일 또는 어셈블링하여 기억장소로 옮기는데 따르는 부하
 - 4) 프로그램의 오류 수정, 매개변수의 조절 그리고 프로그램(컴퓨터)과 실험-프레임(실제 시스템)에 대한 모델의 타당성 조사에 포함되는 시간 및 노력

3.4.6 그 밖의 중요한 사항들

1) 실제 시스템과 기본 모델의 차이점

- . 실제 시스템 : 단지 관측 가능한 자료들의 출처
- . 실제 시스템의 기본 모델 : 실제 시스템의 구조적 특성을 구성요소 및 이들
의 상호작용과 연관시킴

2) 기본 모델과 총괄 모델의 차이점

- . 총괄 모델 : 모델 제작자가 구조를 완전히 알고 있음
- . 기본 모델 : 모델 제작자가 구조를 아주 부분적으로만 알고 있음

3) 서로 다른 프로그래밍 언어는 같은 모델을 다른 방법으로 프로그래밍 한다.

3.4.7 예제 : 엘리베이터 시스템 설계

▶ 실제 시스템

- 1) 건물, 엘리베이터 설계도, 평면도, 사무실의 크기 등의 구조적 자료
- 2) 각 사무실에서 수행되는 업무들의 성격에 관한 자료
- 3) 사무실 간의 정보 (메모, 전화 등) 흐름에 관한 자료
- 4) 사무원과 고객에 관한 개별적 특성 자료
- 5) 고객이 사무원을 만나기 위해 대기하는 시간, 사무원과 함께 보내는 시간, 다음에 어느 사무실로 갈 것인가에 대한 선택, 엘리베이터를 기다리는 시간, 엘리베이터 내에서 소비하는 시간등에 관한 자료

▶ 실험-프레임

: 관심의 대상이 되는 행위 => 엘리베이터 서비스

- 예). 각 층에서 기다리는 사람들의 최대 숫자
 . 각 층 사이의 왕래 횟수

▶ 기본 모델

- OFFICE

. 각 OFFICE는 RECEPTIONIST와 OFFICERS로 구성

. 0,1,2,...,N으로 번호가 붙은 FLOORS라는 서로 다른 집합들로 분리

- INPUT, SET

- . PERSON으로 구성
- . FILE : PERSON의 개인적 특성을 서술,
가야할 첫번째 OFFICE를 지정
- . 0번의 FLOOR에 들어갈 PERSON은 지정한 OFFICE를 가기 위해 필요하다면
ELEVATOR를 호출하고 ELEVATOR에 오른다.
- . OFFICE에서 PERSON은 RECEPTIONIST에 의해 FILE을 토대로 OFFICER를 지정받
고 OFFICER를 기다림
- . OFFICER를 만나서 얼마동안의 시간을 보낸 후 건물을 나가든지, 다른 사무
실로 가든지 하는데, 이는 FILE과 OFFICER에 의해 결정

- ELEVATOR

- . ELEVATOR는 항상 왕복운행
- . 타거나 내리고자 하는 PERSON이 있을 때 멈춤

▶ 총괄 모델

- 기본 모델의 FLOOR와 일치하는 구성요소들을 포함
- 각 FLOOR는 SERVER를 갖고 있음
- INPUT, SET
 - . PERSON : FILE을 갖지 않음

=> 맨 아래의 FLOOR에 들어가 확률변수로 결정되는 FLOOR로 향함

=> ELEVATOR를 통과한 뒤 해당 FLOOR에 기다리고 그의 차례가 오면 또 다른 확률 변수가 SERVER로 부터 서비스받는 시간을 결정하고 또 다른 확률 변수에 의하여 다음 목적지가 결정

- ELEVATOR : 기본 모델과 동일

※ 단순화의 종류

- 1) 구성요소들(OFFICERS)을 총괄하여 더 크고 합성된 구성요소(FLOOR)로 만듦
- 2) 전체적으로 한 서술변수를 제거하여 확률적 방법으로 해결

3.4.8 단순화 과정(Simplification Process)

▶ 단순화를 위한 기본 연산

- 하나 이상의 구성 요소, 서술변수, 상호작용 법칙의 제거
- 하나 이상의 확정변수를 확률변수로 대체
- 하나 이상의 서술변수의 범위 집합을 조정
- 구성요소들을 블럭단위로 분류, 통합하고 블럭 내에 있는 서술변수들을 집약 (aggregation)

1) 구성요소, 서술변수, 상호작용 법칙의 제거

: 모델 제작자는 결과를 얻는데 있어서 별 영향이 없는 부영향자들을 무시할 수 있다.

. 주영향자 (first-order effect)

: 결과를 얻는데 중요한 몇 가지 요소들

. 부영향자 (second-order effect)

: 결과를 얻는데 중요하지 않은 다수의 요소들

- 구성요소 제거 => 그에 따른 서술 변수 제거 => 관련된 상호작용 법칙 제거

[예제 3.3] 서수변수의 제거

▶ 예제 3.2의 국가 모델에서

- 가정 : GOVERNMENT의 정치적 경향(PARTY)은 불안스런 시민의 반응과 무관

=> 서술변수 PARTY 제거

=> P4, P5 수정, P3 제거

. P4' : CIVIL STRIFE가 HIGH

=> 정부는 다음 해에는 PERMISSIVE POLICY 채택

. P5' : CIVIL STRIFE가 LOW

=> 정부는 POLICY를 변경 안함

[예제 3.4] Taylor 급수

$$y = a_1x_1 + a_2x_2 + b_1x_1^2 + b_2x_2^2 + b_3x_1x_2 + c_1x_1^3 + \dots$$

가정 : 언제나 $x_1, x_2 < 1$ 이고

$b_1, b_2, b_3, c_1, \dots$ 등이 a_1, a_2 보다 작다

=> x_1^n, x_2^n ($n \geq 2$) 항 생략 가능

$$\therefore y = a_1x_1 + a_2x_2$$

2) 확정변수의 확률변수

- 확률적 총괄 모델 : 상호작용 규칙의 일부를 확률 메커니즘에 의해 유도해내는데, 이는 해당 변수들을 확률변수로 바꿔줌으로써 가능

3) 서술변수의 범위 조정

- 서술변수의 범위를 조정하는 일은 총괄 모델의 구성요소와 서술변수의 범위집합이 기본 모델의 범위집합보다 더 작도록 조정함으로써, 총괄 모델의 변수가 기본 모델의 구성요소들과 관련된 조건보다 더 적은 수의 조건들을 기술하도록 하는 것이다.

[예제 3.5] 반올림

▶ 가정 : 은행 모델

기본 모델 - 계정 잔고를 1원 단위로 1전 단위로 기록

총괄 모델 - 백원단위만 사용

예) 기본 모델 - $d_1 \dots d_6.00d_1 \dots d_6.02 \dots d_6.99$

총괄 모델 - $d_1 \dots d_6$

4) 구성요소들의 그룹화 및 서술변수들의 집약

▶ 집약 방법

- 먼저 구성요소들에 대한 변수들의 성질을 알려지 않도록 합성
- => 결과적으로 얻어진 변수들에 대한 범위집합을 조정

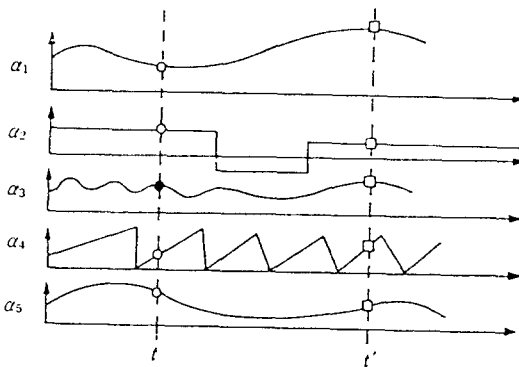
3.5 시뮬레이션과 형식적 모델 명세

▶ 형식적 모델 서술

- 모델을 수학적으로, 즉 집합론의 언어로 서술
- 프로그램으로 표현되어지는 구현적 형식에 비해서 보다 “순수한” 형태를 취함

3.5.1 상태 (state)의 개념

▶ 상태변수 : 미래의 값을 계산하기 위하여 필요한 일부 서술변수



- t, t' : 모델 시각

$t - t'$: 모델 경과 시간

- 상태변수 집합 : 시각 t' 에서의 모든 서술변수들의 값을 유일하게 결정해주는 시각 t 에서의 몇개의 서술변수

3.5.2 상태 변수의 특성

▶ 상태와 관련된 사항

- 프로그램의 초기화
- 반복 실행
- 프로그램의 인터럽트 및 재실행
- 프로그램의 복구

▶ 시간 t 에서 상태변수값 y_1, \dots, y_n 이 주어졌을 때, t' 에서 y_1', \dots, y_n' 의 서술 변수 값들을 계산한다고 가정하고 이 때 $t_1, t_2, \dots, t_i, t_{i+1}, \dots$ 을 모델 시각이라 하자.

- 계산 혹은 시뮬레이션

: 만일 시간 t_i 에서 y_1^i, \dots, y_n^i 가 주어졌을 때 시뮬레이션 프로그램으로 t_{i+1} 에서 $y_1^{i+1}, \dots, y_n^{i+1}$ 을 얻을 때 t_i 에서 t_{i+1} 까지의 모델 전이 (model transition)

- 계산순간들 (computation instants) : $\{ t_1, t_2, \dots \}$

- 이산 사건 시뮬레이션 : 연속적인 다수의 단계구간 (구간의 크기는 $t_{i+1} - t_i = h$)들로 이루어지는 경우

- 시간 독립 모델 (time invariant model)

: 상호작용 규칙이 시간이 아닌 상태값 (y_1, \dots, y_n)에만 의존할 때

3.5.3 시간 독립 모델에 대한 이산사건 시뮬레이션의 절차

▶ 시각 t 에서 상태 변수값 y_1, y_2, \dots, y_n 이 주어졌을 때 t' 에서의 서술변수값들을 계산하기 위해서, 이산사건, 시간 독립인 경우에, 프로그램이 반복적으로 수행하는 절차

▶ 가정 : $t = t_M$ 이고 $t' = t_{M+N}$ 이며

$t_M, t_{M+1}, \dots, t_{M+N}$ 은 t 와 t' 사이의 계산 시간

▶ 시뮬레이션 절차

단계 1) y_1, \dots, y_n 을 값으로 하는 상태변수 $\alpha_1, \dots, \alpha_n$ 을 초기화

단계 2) 시계를 t_M 으로 초기화 (모델 시각은 T 에 저장)

단계 3) 새로운 $\alpha_1, \dots, \alpha_m, \alpha_{m+1}, \dots, \alpha_n$ 을 산출을 위하여 $\alpha_1, \dots, \alpha_m$ 을 상호 작용규칙에 적용

단계 4) 시계를 현재 시각에 h 만큼 증가

단계 5) 만일 시계가 $t_M + Nh$ 보다 큰 시각을 가리키면 멈추고,
아니면 3)으로 감

- 증명 : 절차가 하나의 모델을 정확히 시뮬레이션하고 있는가를 보여줌

비형식적 증명

형식적 증명

※ 비형식적 증명

- 1)에서 시각 t 일때 정확한 상태값 y_1, \dots, y_n 이 위치 $\alpha_1, \dots, \alpha_n$ 에 놓여지고
 - 2)에서 모델 시각 t 를 위치 T 에 저장하고
 - 3)에서는 상호작용 규칙이 y_1, \dots, y_m 에 적용되어 $y_1', \dots, y_m', y_{m+1}', \dots, y_n'$ 이 산출된다. 이 값들은 $t+h$ 에서의 정확한 모델값들이며 상호작용 규칙에 의하여 전이 $(0, h)$ 가 일어난다. 즉 모델은 시간 독립적이고 컴퓨터는 정확히 이 규칙들을 수행한다.
 - 4)에서는 모델 시간을 h 만큼 증가시키고
 - 5)에서는 반복 수행 여부를 결정한다.
- ∴ 이러한 절차는 이산시간을 시뮬레이션에 관한 본질적인 특성을 표현하는 표본적 모델

3.5.4 모델의 형식적 명세

▶ 3.5.3의 단계 3)

: 입력 (상태변수값) → 출력 (서술변수값을 산출) : 부프로그램

∴ 부프로그램 : 함수 f

- 정의구역 : 상태변수값들의 집합

- 치역 : 서술변수값들의 집합

$$f (y_1, \dots, y_m) = (y_1', \dots, y_m', y_{m+1}', \dots, y_n')$$

이 때 함수 f는 δ, λ 의 두 가지 함수로 이루어졌다고 할 수 있다.

$$\delta (y_1, \dots, y_m) = (y_1', \dots, y_m')$$

$$\lambda ((y_1, \dots, y_m), (y_1', \dots, y_m')) = (y_1', \dots, y_m', y_{m+1}', \dots, y_n')$$

$$\therefore f (y_1, \dots, y_m) = \lambda ((y_1, \dots, y_m), \delta (y_1, \dots, y_m))$$

⇒ 이를 그림으로 표현하면

$$y_1, \dots, y_m \rightarrow f \rightarrow y_1', \dots, y_m', y_{m+1}', \dots, y_n'$$

$$y_1, \dots, y_m \rightarrow \delta \rightarrow y_1', \dots, y_m' \rightarrow \lambda \rightarrow y_1', \dots, y_n'$$

- 때로는 함수 λ 는 y_1, \dots, y_m 에 독립적으로

$$\delta (y_1, \dots, y_m) = (y_1', \dots, y_m')$$

$$\lambda (y_1', \dots, y_m') = (y_1', \dots, y_n')$$

$$f(y_1, \dots, y_m) = \lambda(\delta(y_1, \dots, y_m))$$

이를 그림으로 표현하면

$$y_1, \dots, y_m \rightarrow \delta \rightarrow y_1', \dots, y_m' \rightarrow \lambda \rightarrow y_1', \dots, y_n'$$

▶ 정규형 : 이렇게 단순화시킨 형태

δ : 상태전이함수 (state transtion function)

- 시각 t_i 에서 상태변수값들을 취해서 시각 t_{i+1} 에서의 상태 변수값들을 생성

λ : 출력함수

- 현재의 계산 시각에 모델 자체가 구한 상태 변수값들을 취하여 현재의 나머지 서술변수값들을 산출

▶ 모델의 상태, 출력, 전이함수

- DESCRIPTIVE, VARIABLES : 모델 서술변수들의 집합

OUTPUT, VARIABLE : 출력변수들의 집합

STATE, VARIABLE : 상태변수들의 집합

RANGE, β : β 가 취하는 값의 범위 집합 ($\forall \beta \in \text{STATE, VARIABLE}$)

RANGE, γ : γ 가 취하는 값의 범위 집합 ($\forall \gamma \in \text{OUTPUT, VARIABLE}$)

$\beta_1, \beta_2, \dots, \beta_i \in \text{STATE, VARIABLES}$

$\gamma_1, \gamma_2, \dots, \gamma_j \in \text{OUTPUT, VARIABLES}$

. STATE VARIABLE

- 모델의 한 STATE란 상태변수들의 값을 갖는 것
즉, STATE VARIABLE의 원소들에게 각각 하나의 값을 할당시킨 것
- 치역 : CROSS PRODUCT

$$\text{CROSS PRODUCT} = \prod_{\beta_i \in \text{STATE VARIABLES}} \beta_i$$

or

$$\text{RANGE } \beta_1 \times \text{RANGE } \beta_2 \times \dots \times \text{RANGE } \beta_i \times \dots$$

- 모델은 모든 상태 변수값을 다 지정할 필요는 없고, 관련된 것들만 지정
=> STATES $\subseteq \prod_{\beta_i \in \text{STATE VARIABLES}}$

$$\beta_i$$

. OUTPUT VARIABLE

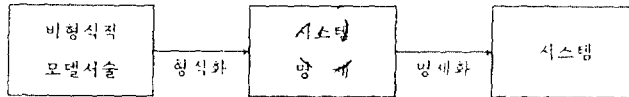
$$\Rightarrow \text{OUTPUTS} \subseteq \prod_{\gamma_j \in \text{OUTPUT VARIABLES}} \gamma_j$$

$$\gamma_j$$

- 시간 독립적인 구간이 h인 이산시간 모델에서의 전이함수와 출력함수
 $\delta(\text{STATE}) =$ 시각 $t_i + h$ 에서의 모델 상태
 $\lambda(\text{STATE}) =$ 시각 t_i 에서의 모델 출력

3.5.5 구조와 행위

- ▶ 이산 사건 시스템 명세
: $\langle \text{STATES, OUTPUT, } \delta_h, \lambda \rangle$
- ▶ 형식화 과정



- ▶ 시스템 명세 (system specification)
: 모델을 어떻게 시뮬레이션할 것인가를 제시하는 형식적이고 정확한 방법
예) 한 모델에 대해 시각 t 에서 t' 까지 시뮬레이션

- 관측구간 : 구간 $[t, t']$
- 계산시각 : $t_M, \dots, t_M + Nh$
- 다음과 같은 $STATE_q$ 를 가짐

=> 각각의 시간 $t_M, \dots, t_M + Nh$ 에 대하여

$STATES_{q_M, q_{M+1}, \dots, q_{M+N}}$ 이 대응

$$\therefore q_{M+i} = \delta (q_{M+i-1}) \quad 63$$

▶ 상태궤적 (state trajectory) 함수 $\text{STRAJ}_q, [t, t']$

$\text{STRAJ}_q, [t, t'] : \{ t_M, t_M+h, \dots, t_M+Nh \} \rightarrow \text{STATES}$ such that

$$\text{STRAJ}_q, [t, t'](t_M+ih) = \lambda(q_{M+i}) \text{ for } 0 \leq i \leq N$$

3.5.6 비자율 모델 (Nonautonomous model)

▶ 예제

- . DESCRIPTIVE VARIABLE : 서술변수의 집합
- . INPUT VARIABLES의 부분 집합 : 외부적으로 값이 결정되어지는 입력 변수
- . NON INPUT VARIABLE = DESCRIPTIVE VARIABLES-INPUT VARIABLES

- 다음의 필요충분 조건 가정

NON INPUT VARIABLES의 부분 집합은 상태변수들의 집합

$\langle \Rightarrow \rangle$ t에서 이들 변수들의 값과 t와 t+h에서의 INPUT VARIABLES의 값들이 t+h에서 NON INPUT VARIABLE의 값들을 유일하게 결정해준다.

\Rightarrow 정규형

\Rightarrow 순차기계 (sequential machine) : $\langle \text{INPUT, STATES, OUTPUTS, } \delta_h, \lambda \rangle$

. 전이함수 δ_h : STATES X INPUTS \rightarrow STATES

. 출력함수 λ : STATES X INPUTS \rightarrow OUTPUTS

. STATE_q와 INPUT 수열 X_M, \dots, X_{M+N} 에 관련된 상태수열

: q_{M+1}, \dots, q_{M+N} 과 $q_{M+i+1} = \delta_h (q_{M+i}, X_i)$

. 출력의 경우

STATE_q와 INPUT 수열 X_M, \dots, X_{M+N} 에 관련된 출력수열

: 수열 y_M, \dots, y_{M+N} 이고 $y_{M+i} = \lambda (q_{M+i}, X_{M+i})$

- 상태행위 (state behavior) : 입력 상태 궤적 쌍들의 집합

- 입출력행위 (input-output behavior) : 입출력 궤적 쌍들의 집합

3.5.7 난수 생성

▶ 선형 합동 난수 생성기 (linear congruential random number generator)

: 다음 규칙에 따라 난수 생성

가) S_0 : 정수이고 $s \in [0, p]$

나) $S_{i+1} = (a s_i + b) \bmod p$ for $i = 0, 2, 3, \dots$

S_i

다) $r_i = \frac{S_i}{p}$

p

※ 이 때 S_i : seed, S_0 : 초기 seed

- 모델링 : 자율적 순차기계 $M = \langle Q, Y, \delta, \lambda \rangle$

Y : 출력집합 $[0, 1]$

δ : 전이함수

λ : 출력함수

Q : 상태집합 $[0, p]$

즉, $\delta : Q \rightarrow Q, \delta(q) = (aq+b) \bmod p$ for all $q \in Q$

$\lambda : Q \rightarrow Y, \lambda(q) = q/p$

3.5.8 지체변수 (lagged variable)들의 상태 표현

▶ 난수 발생에 대한 또다른 방법

: 이전에 사용한 seed를 이용하여 다음의 seed를 구함

가) $S_{-5}, S_{-4}, S_{-3}, S_{-2}, S_{-1} \in [0, p)$: 정수

나) S_{i-3}, S_{i-5} ($i = 0, 1, 2, \dots$) 가 주어졌을 때 규칙

$$S_{i+1} = (aS_{i-3} + bS_{i-5}) \bmod p$$

$$S_i$$

$$r_i = \frac{S_i}{p}$$

$$p$$

- 다음의 seed 생성을 위하여 5개의 seed가 준비되어야 함

3.5.9 임의성 (Randomness)

▶ 임의성에 관한 문제 : 난수의 특성

- 앞에서 서술한 난수 발생 공식은 엄밀한 의미에서의 난수의 성격에 위배
 (∵ 어느 순간이 지난 후부터는 값이 반복)

- 임의성을 위한 실용적인 판단 기준

: 적당하다고 생각되는 검사를 어떤 수열이 통과했을 때 그 수열을 난수로 인정

▶ 난수 판단 검사 방법의 예

- 난수가 $[0,1]$ 사이에서 균등하게 분배되어지는가

- 두 난수 사이에 순차적 상관관계가 있는지 측정

제 4 장 모델의 타당성 검토(Model Validation)

4.1 개요

- ▶ 시뮬레이터(simulator)의 당면 문제
 - : 시뮬레이션 모델의 검증(verification)과 타당성 검토(validation)
- 검증 : 하나의 시뮬레이션 모델이 컴퓨터 프로그램 상에서 의도하는 대로 수행하는지를 검사하는 것
- 타당성 검토 : 시뮬레이션 모델이 고찰하는 현 세계의 시스템을 정확히 표현하고 있는지 확인하는 것
- ▶ 타당성 정도
 - 모사적 타당 (replicatively valid)
 - : 모델이 실제 시스템으로부터 이미 얻은 데이터와 부합
 - 예보적 타당 (predictively valid)
 - : 실제 시스템으로부터 데이터를 얻기 전에 데이터와 부합
 - 구조적 타당 (structurally valid)
 - : 모델의 구조와 실제 시스템의 내부 동작들 사이의 관계에 관한 것으로 관측된 실제 시스템의 행위를 재현할 뿐만 아니라 이러한 행위를 생성하는 방법

4.2 시뮬레이션 모델의 검증(Model Verification)

▶ 시뮬레이션 모델의 검증 기법

- 기법 1 : 프로그램을 여러 개의 모듈이나 부프로그램 단위로 작성하여 오류 수정
- 기법 2 : 프로그램의 정확성 평가에 프로그래머 이외(시스템 분석가)의 사람을 포함
- 기법 3 : 추적(trace) : 가상된 시스템 상태, 즉 사건의 리스트의 내용, 상태 변수, 통계 자료 수집 등이 의도한 대로 동작하는 가를 보기 위하여 각 사건이 발생한 후 바로 인쇄
- 기법 4 : 간단하고 쉬운 가정하에 시뮬레이션을 수행
- 기법 5 : 시뮬레이션 결과를 그래픽 터미널 상에 표시

4.3 모델의 타당성 검토시 고려 사항

1) 타당성 검토의 목적 :

한 모델이 실제의 시스템 자체를 가지고 실행하는 것이 가능한 경우에 내려질 결론과 똑같은 결론을 얻기 위해 결정자가 실제로 사용할 수 있도록 모델이 개발되었는가를 확인

2) 시뮬레이션 모델은 유효성과 경제성을 갖추어야 함

3) 특정 목적을 위해 개발되어야 함

4) 지정된 일련의 평가 기준에 대하여 유효하도록 하여야 함

5) 모델의 개발과 타당성 검토는 시뮬레이션 연구의 전 과정에서 상호 협력하여야 함

6) 타당성 검토를 위한 통계적 방법의 이용은 시뮬레이션 환경을 고려한 가정들이 포함되어야 함

4.4 타당성 검토를 위한 3단계 방법론

- ▶ 단계 1. 고도의 표면적인 타당성을 갖는 모델의 개발
 - : 시뮬레이션 작성자는 다음을 포함한 기존의 모든 정보들을 이용
 - 전문가들과의 대화
 - 기존 이론의 사용
 - 시스템의 관측
 - 일반 지식의 이용
 - 직관 (intuition)

- ▶ 단계 2. 모델을 위한 가설의 경험적인 검증
 - 시뮬레이션 모델의 입력 = 이론적 확률 분포
 - > chi-square나 kolomogrorov-Smirnor 검정
 - 둘 이상의 자료의 집합들을 합병
 - > 동질성에 관한 Mann-Whitney 혹은 Kruskal-Wallis 검정

- ▶ 단계 3. 시뮬레이션 출력 자료의 정확성 확립
 - 고찰중인 것과 유사한 시스템이 존재
 - > 기존 시스템의 모델을 개발하여 이의 출력 자료를 실제 시스템의 자료와 비교

4.5 실제 관측치와 시뮬레이션 출력 데이터의 비교 방식

▶ 검사 방식 (inspection approach)

: 현세계 관측치에 대한 통계량과 출력 자료의 통계량들을 계산하여 모델 통계량과 적절한 통계적 방식의 사용없이 단순히 시스템으로부터 얻은 통계량을 비교

▶ 신뢰 구간 방식 (confidence interval approach)

: 모델과 시스템으로부터 많은 양의 자료를 수집하는 것이 가능한 상황에서 모델과 시스템을 비교하는 것 보다 신뢰성있는 방식

▶ 시계열 방식 (time-series approach)

: 모델과 시스템의 출력 자료를 사용하여 각 자료의 자기 상관 (autocorrelation) 구조를 파악하여 분석에 이용

제 5 장 시뮬레이션 모델링 월드뷰

5.1 개요

- ▶ 모델의 기본구조는 월드뷰(world view)에 의해 결정됨

- ▶ 시뮬레이션 모델링 월드 뷰
 - 이산형 시뮬레이션 모델링
 - 연속형 시뮬레이션 모델링
 - 혼합형 시뮬레이션 모델링

5.2 모델링 월드뷰(world view)

▶ 시뮬레이션에서 시간은 중요한 독립 변수

=> 다른 변수들은 시간의 함수로 표현되며 종속적인 변수가 됨

▶ 월드뷰는 종속 변수의 형태에 관련됨

▶ 이산형 시뮬레이션

- 종속 변수가 사건 발생 시간이라고 불리는 특정한 시간에 이산적으로 변화할 경우

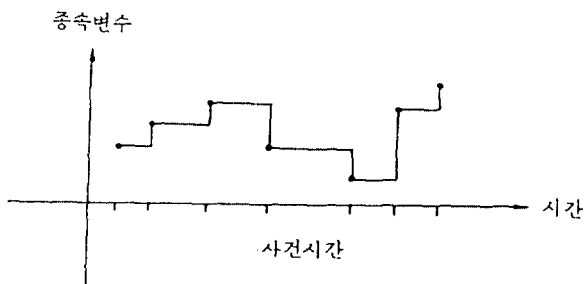
- 예 : 은행출납원 문제

. 종속 변수 : 대기열 내의 고객 수

. 사건 발생 시간 : 고객이 도착하는 시간,

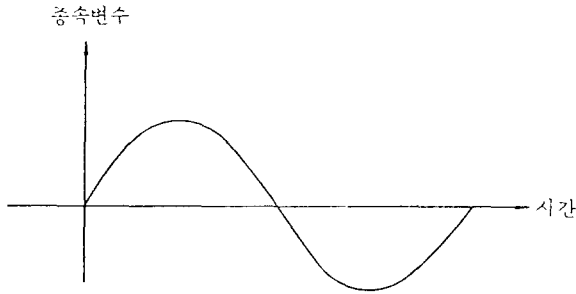
고객이 떠나는 시간

- 종속 변수 측정치에 대한 그래프



▶ 연속형 시뮬레이션

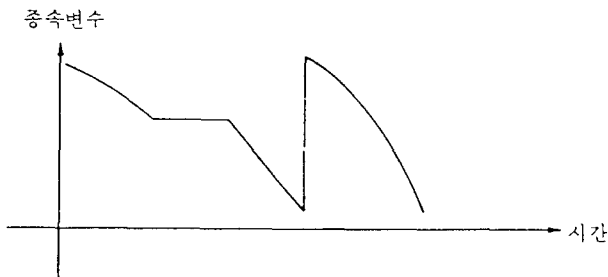
- 모델의 종속변수가 시뮬레이션 시간동안 연속적으로 변화



- 예 : 화학공정에서 반응물질의 농도, 우주선의 위치와 속도

▶ 혼합형 시뮬레이션

- 종속변수가 이산적으로 변할 수도 있고, 연속적으로 변할 수도 있음



- 이산적으로 변하는 변수들과 연속적으로 변하는 변수들간의 상호작용으로부터 발생

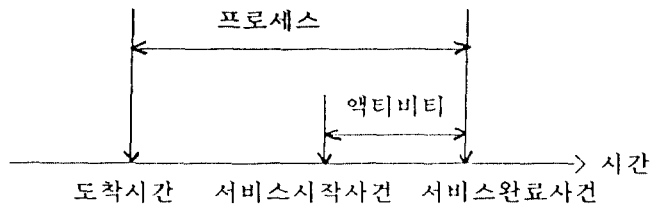
5.3 이산형 시뮬레이션 모델링

- ▶ 모델링 : 엔티티들이 포함된 액티비티들을 재생하여 시스템의 잠재적 형태와 성능에 대한 것들을 배우는 것
 - : 시스템의 상태들을 정의하고, 시스템을 어떤 상태에서 다른 상태로 옮기는 액티비티들을 구축함으로써 달성됨

- ▶ 시스템 상태
 - : 사건 발생 시간에만 변함 (다음 사건 진행방법 사용)

- ▶ 모델의 공식화
 - ① 각 사건발생시에 일어나는 상태의 변화
 - ② 시스템의 엔티티가 사용되는 액티비티
 - ③ 시스템의 엔티티가 흐르는 프로세스

- ▶ 액티비티, 프로세스, 개념간의 관계



- 프로세스 : 시간에 따라 정렬된 사건들의 순서

▶ 이산형 시뮬레이션 모델링의 월드뷰

- 사건 중심 방법
- 액티비티 관찰방법
- 프로세스 중심 방법

5.3.1 사건 중심 진행 방법 (event orientation)

- ▶ 사건 발생 시간에 일어나는 변화를 정의함으로써 시스템을 모델화
 - => 모델 작성자는 시스템의 상태를 변화시키는 사건을 결정하고, 각 사건의 형태와 관련된 논리를 개발
- ▶ 예 : 은행 문제
 - 시스템의 상태 : 출납원의 상태와 대기중인 고객수에 의해 정의
 - 사건 중심 모델은 고객의 도착 시간과 서비스 완료시간에 무엇이 발생하는지를 기술
 - 사건 목록 : 사건 발생 시간들이 미리 기록됨
 - 사건 기록 일지 : 상태 변화가 일어날 수 있는 미래의 시간들을 나타냄

5.3.2 액티비티 관찰 중심 방법 (activity scanning)

- ▶ 시스템의 엔터티들과 관계되는 액티비티들을 기술하고,
액티비티의 시작과 종료를 유발하는 조건들을 규정
- ▶ 각 액티비티에 대한 처리가 보장되기 위해서는 매 시간의 진행마다 액티비티 전체 집합을 관찰할 필요가 있음
- ▶ 액티비티 기간이 무한정하고, 규정된 조건을 만족하는 시스템 상태에 의해 액티비티 기간이 결정되는 경우에 적합
- ▶ 이산형 사건 중심 진행 방법보다 비효율적

5.3.3 프로세스 중심 진행 방법(process interaction)

- ▶ 시스템을 통과하는 엔터티의 흐름을 모델링하기 위하여 사건들의 순서를 정의하는 문장들로 표현
- ▶ 사건들의 순서는 엔터티들이 프로세스를 통하여 이동할 때, 시뮬레이션 언어에 의하여 자동적으로 처리됨
- ▶ 프로세스를 통한 엔터티들의 흐름에 대한 기술을 제공
- ▶ 사건 중심 진행 방법보다 융통성이 떨어진

5.4 연속형 시뮬레이션 모델링

- ▶ 시스템상태는 시간에 따라 변하는 종속변수에 의해 표현됨
- ▶ 연속적으로 변화하는 일련의 상태변수들에 대한 방정식을 정의함으로써 모델이 구축
- ▶ 연속형 시스템들의 모델들은 보통 상태 변수들의 도함수로 표현됨
- ▶ 미적분으로 구축된 모델을 수행하는 컴퓨터
 - ┌ 아날로그 컴퓨터 : 논리적 제어기능과 자료저장능력이 떨어짐
 - └ 디지털 컴퓨터 : 빠른 속도와 정확성 가짐
- ▶ 연속형 시스템은 증가치(difference) 방정식을 이용하여 모델링하기도 함
 - : 시간 k 에서의 상태 변수값을 이용하여 $k+1$ 시간의 상태 변수값을 계산
 - : 예
$$S_{k+1} = S_k + r\Delta t$$
- ▶ 시뮬레이션 언어는 블록 중심 처리나 방정식 중심 처리를 채택함
 - 블록 중심 처리 : 아날로그 컴퓨터의 회로 요소를 흉내내는 일련의 블록들을 이용
 - 방정식 중심 처리 : 미분 방정식이나 증가치 방정식 형태로 명시적으로 코딩

5.5 혼합형 모델링 (combined discrete-continuous modeling)

- ▶ 종속 변수들이 이산적으로 변하거나 연속적으로 변함
- ▶ 시스템이 엔터티, 엔터티들과 관련된 속성, 상태 변수로 기술됨
- ▶ 이산형 변수와 연속형 변수간에 발생할 수 있는 기본적인 상호작용
 - ① 이산적인 값의 변화가 연속형 변수에 일어남
 - 예 : 새로운 발전소를 완공하는 경우 총에너지가 순간적으로 증가
 - : 호수에 화학약품을 살포하는 경우 특수한 종류의 수가 순간적으로 감소
 - ② 임계값(threshold value)을 획득한 연속형 상태 변수를 포함한 사건이 다른 사건의 발생을 유발
 - 예 : 규정된 농축수준에 도달할 때 화학공정은 완료되나 이때 세척과 유지보수 액티비티를 위하여 공정이 중지
 - ③ 연속형 변수의 기능적 서술이 이산적 시간의 순간에 변화
 - 예 : 생태계로 공해물질이 방출되면 서식하는 개체군에 영향을 주는 성장률을 변화
 - : 우주선이 결합을 완료하면 우주선의 비행을 시뮬레이션하는 방정식이 바뀌어야 함

▶ 발생할 수 있는 사건

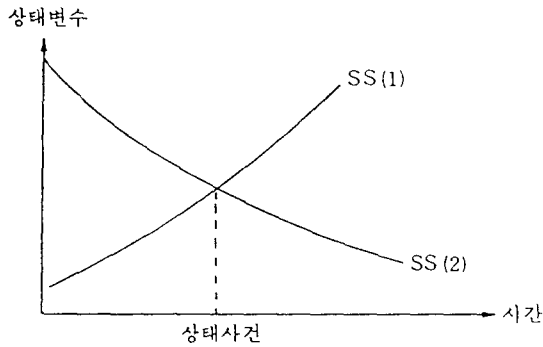
① 시간사건(time-event)

: 명시된 시간에 사건이 발생

② 상태사건(state-event)

: 사건이 계획되지 않고 특별한 상태에 도달하면 사건이 발생

: 예



- 상태 변수 SS(1)이 SS(2)보다 커지는 순간에 상태 사건이 발생

제 6 장 결론

▶ Simulation Modeling과 Validation

