

## 시뮬레이션과 그 소프트웨어

정 종 훈\*

### I. 緒 言

시뮬레이션(simulation)은 自然科學이나 社會科學에서 많이 이용되고 있는 Operations Research의 한 技法으로서, 컴퓨터와 그 언어의 개발과 향상 그리고 설계 및 분석하고자 하는 시스템의 복잡성으로 말미암아 그 중요성이 더해가고 있는 추세이다. 이에 본 지면을 통하여 시뮬레이션의 本質과 長短點 그리고 그 應用 분야를 간단히 살펴본 후, 현재 많이 사용하고 있는 시뮬레이션용 소프트웨어에 대해서 요약하여 紹介하고자 한다. 즉 시뮬레이션 言語의 種類와 特性 그리고 言語의 選擇基準을 살펴보고, 시뮬레이션을 效率的 및 綜合的으로 遂行할 수 있도록 개발된 시뮬레이션 시스템의 하나인 슬램시스템(SLAMSYSTEM)에 대해서 紹介하고자 한다.

### II. 本 論

#### 1. 시뮬레이션 概要

##### 가. 시뮬레이션의 本質

모델링과 시뮬레이션(modeling and simulation)이란 실제 시스템(real system)에 관한 모델(model)을 構成하고, 이것들을 컴퓨터상에서 實驗하기 위한 諸般 行爲를 말한다. 이 정의에 따르면 모델링과 시뮬레이션은 세 가지 주요 요소, 즉 실제시스템(real system), 모델(model), 컴퓨터(computer)와 관련된다. 또한 그림1에서 보는 바와 같이 모델링과 시뮬레이션 過程은 위의 세 가지 요소들뿐만 아니라 그들 상호간의 관계(relation)에도 관심을 두어야 한다. 모델링과

정에서는 실제 시스템과 모델 사이의 관계를, 그리고 시뮬레이션 과정에서는 컴퓨터와 모델 사이의 관계를 주로 다룬다.

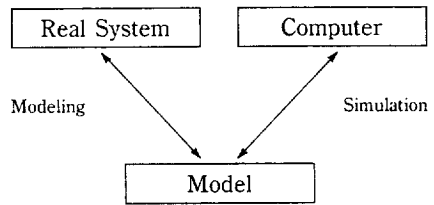


Figure 1. Relationship of modeling and simulation

시뮬레이션은 運營科學(operations research : OR)과 經營科學(management science)에서 가장 널리 이용되는 技法들 중의 하나로 그 이용이 증가일로에 있다. 그러나 시뮬레이션의 보다 넓은 受容과 利用에는 여러 障礙物이 있어왔다. 첫째, 대형시스템을 研究하기 위한 모델들은 너무 복잡하여 모델을 시뮬레이션하기 위한 프로그램을 작성하는 것은 정말로 어려운 일이다. 이 문제는 최근에 시뮬레이션 모델을 코드화시키는 데 필요한 여러 特性들을 提供하는 특수 목적용 컴퓨터 언어(special-purpose computer language)의 開發로 간편해졌다. 복잡한 시스템의 시뮬레이션에 대한 두번째 문제는 상당히 긴 컴퓨터 시간(computer time)이 소요된다는 점이다. 그러나 이 문제도 계산에 드는 費用을 줄임으로써 완화시킬 수 있다. 마지막으로 시뮬레이션은 그 복잡성에도 불구하고 컴퓨터 프로그래밍의 연습에 불과하다는 인상을 남긴다. 결과적으로 많은 시뮬레이션 研究가 경험적인 모델의 설계(heuristic model building), 코딩, 결과를

\* 전남대학교 공과대학 교수

언기 위한 프로그램의 遂行으로 구성되어 왔다. 특히, 현실 세계의 설비(facility)나 프로세스(process)의 다양한 運營(operation)을 模倣하거나 시뮬레이션을 하기 위하여 컴퓨터를 이용하고 있다.

이 設備 혹은 프로세스를 시스템이라 하는데, 시스템을 과학적으로 연구하기 위하여 어떻게 동작하는가에 관한 일련의 假說들을 세워야 한다. 이 가설들은 수학적 혹은 논리적 聯關性을 가지며 시스템의 동작방식을 이해하기 위하여 사용되는 모델의 구성요소가 된다. 만일 모델을 구성하는 관계가 아주 간단하다면, 대수학, 미적분, 확률론 등과 같은 수학적 방법을 이용하여 정확한 정보를 얻을 수 있다. 이를 분석적 해결방법(analytic solution)이라 한다. 그러나 대부분의 현실 세계의 시스템들이 분석적으로 평가할 수 있는 실제의 모델을 구성하기엔 너무 복잡하다. 따라서 시뮬레이션에 의한 방법으로 고찰하여야 한다. 시뮬레이션에서는 시간 週期마다 수치적으로 모델을 평가하기 위하여 컴퓨터를 사용하며, 모델의 특성을 推定하기 위하여 자료를 收集한다.

시뮬레이션 이용의 한 예로, 공장을 좀더 확장시키려 하고 있는 제조회사를 고찰하자. 생산된 소득으로 가설 비용을 충당할 수 있을지는 의문이지만 공장을 확장하였다가 큰 소득이 없어 다시 철거하는 것은 비경제적이다. 그러나 면밀한 시뮬레이션 연구를 통하여 현재(확장하지 않은 상태)와 확장한 경우의 운영을 시뮬레이션함으로써 문제해결의 실마리를 찾을 수 있다.

이러한 시뮬레이션을 손쉽게 效率의으로 수행하기 위해서는 시뮬레이션 言語와 이에 관련된 소프트웨어의 使用이 絶對的으로 필요하다.

#### 나. 시뮬레이션의 長點과 短點

다음은 시뮬레이션의 長點에 관한 事項들이다.

1) 확률적 요소들을 갖는 대부분의 복잡한 현실세계 시스템들은 分析的으로 評價가 가능한

수학적 모델로 정확히 기술할 수 없다. 그러므로 시뮬레이션을 통해 이를 해결할 수 있다.

2) 시뮬레이션을 이용하면 어떤 運營 環境에 있어서도 현존 시스템의 效率를 추정할 수 있다.

3) 시뮬레이션을 통해서 특수 요구에 맞는 시스템의 設計(혹은 새로운 運營 方針) 방법들을 비교할 수 있다.

4) 시뮬레이션은 실제의 시스템을 가지고 실험할 때보다 실험을 위한 環境의 調節이 더 쉽다.

5) 시뮬레이션은 경제 시스템처럼 오랜 시간이 걸리는 시스템을 研究할 수 있도록 하며, 반대로 아주 짧은 시간에 동작하는 시스템에 있어서는 시간을 확장시켜 연구할 수 있도록 해준다.

시뮬레이션도 그 자체의 短點이 없는 것은 아니다. 다음은 그 短點들을 열거한 것이다.

1) 시뮬레이션 모델은 개발상 많은 費用과 時間을 요한다.

2) 확률 시뮬레이션 모델의 遂行은 특정의 입력 매개변수의 集束에 대하여 단지 모델의 특성에 대한 추정량을 산출한다. 그러므로 각 입력 매개변수에 대하여 여러 번의 독립적인 수행이 필요하다. 이러한 이유로 시뮬레이션 모델은 고정된 수의 시스템 설계 방법들을 비교할 때는 좋으나 일반적으로 시스템의 最適化(optimization)를 위해서는 좋지 않다. 만일 분석적 모델(예를 들면 대기행렬 이론에 의한 모델)이 적합하다면, 여러 입력에 대한 실제의 시스템 특성을 쉽게 算出할 수 있다. 그러므로 타당한 분석모델이 이용 가능하거나 쉽게 개발된다면 시뮬레이션 모델을 이용하는 것보다 더 바람직하다.

3) 시뮬레이션에 의해 산출되는 많은 양의 수치들은 연구 결과에 대하여 과잉의 신뢰를 주는 경향이 있다. 만일 모델이 타당하지 않다면 시뮬레이션 結果들은 그들이 어떻게 나타났든간에 실제의 시스템과 관련하여 유용하지 못한 情報를 제공하게 될 것이다.

시뮬레이션 연구가 주어진 狀況에 적용이 가능한지의 여부를 결정할 때, 이들의 장단점과 특

수 상황에 적합한 모든 면을 고려해야 한다. 끝으로 시뮬레이션 모델과 분석모델 모두가 유용할 수도 있다. 특히 시뮬레이션은 분석모델이 필요로 하는 가설들의 妥當性을 검사하기 위하여 사용될 수 있다. 한편, 분석모델은 시뮬레이션 연구에 있어 올바른 方法을 제시해 줄 수도 있다.

다. 시뮬레이션의 使用 分野

컴퓨터 技能의 향상과 시뮬레이션 전용 言語의 開發로 시뮬레이션의 사용은 더욱 촉진되고 있다. 또한, 시뮬레이션의 原理는 모든 분야에 적용될 수 있어서 여러 조직에서의 사용 頻度가 높다.

다음 표 1은 1971년도에 미국의 1,000개 大企業을 상대로 해서 얻은 여러 OR技法들의 사용 頻度を 나타낸다.

표 1 미국의 大企業에서 사용되는 OR技法들의 頻度

技 法	頻 度	%
시뮬레이션 (simulation)	60	29
선형계획법 (linear programming)	43	21
네트워크분석 (PERT/CPM 포함)	28	14
재고관리론 (inventory theory)	24	12
비선형계획법 (nonlinear programming)	16	8
동적계획법 (dynamic programming)	8	4
정수계획법 (integer programming)	7	3
대기이론 (queueing programming)	7	3
기 타	12	6
합 계	205	100

표 1 에서와 같이 시뮬레이션이 미국의 대기업에서 가장 많이 사용되는 OR技法인 것을 알 수 있다. 이 이유는 시뮬레이션의 原理가 概念的으로 간단하고 쉬워서 기업의 관리자가 쉽게 사용할 수 있고 광범위한 분야에 사용될 수 있기 때문이다. 시뮬레이션이 특히 많이 이용되는 分野를 열거하면 다음과 같다.

- (1) 항공회사의 공항 運營 문제
- (2) 交叉路에서 交通 신호 操作
- (3) 고장보수팀의 크기 결정 문제
- (4) 放射能 차단벽을 통과하는 입자수의 研究
- (5) 경제정책이 미치는 影響 調査
- (6) 대규모 군사작전의 연구
- (7) 在庫 및 分配(distribution)
- (8) 기업정책이 기업에 미치는 影響의 연구
- (9) 通信施設의 규모결정 문제

- (10) 댐, 수력발전소 등의 規模決定 문제
  - (11) 生産施設 능력 및 位置 決定 문제
  - (12) 공장의 工程시스템 分析
  - (13) 산업 공장의 適正 設計
- 위 예들은 시뮬레이션이 실제로 많이 쓰이는 분야를 열거한 것이다

다음은 시뮬레이션이 특히 效果的으로 쓰일 수 있는 경우를 열거한다.

- (1) 복잡한 시스템내에서 시스템 구성 요소간의 相互作用을 연구한다.
- (2) 조직에 變化(내부 및 외부에서의 變化)가 생겼을 때 발생하는 결과를 관찰한다.
- (3) 시뮬레이션 모델을 만들 때 얻은 지식은 시스템 運用시 유용하게 사용될 수 있다.
- (4) 시뮬레이션은 教育用으로 사용될 수 있다.
- (5) 모델의 入力 및 出力을 관찰함으로써 시

시스템에 중요한 영향을 미치는 요소를 찾아낸다.

(6) 중요방침(policies)을 시행하기 전에 시뮬레이션을 함으로써 결과를 예측할 수 있다.

## 2. 시뮬레이션용 소프트웨어

모델링 이론과 시뮬레이션에 대한 기본을 認知했으므로 이제 시뮬레이션 모델의 구성과 분석을 용이하게 하기 위하여 현재 이용할 수 있는 시뮬레이션용 소프트웨어에 대해서 살펴보기로 한다. 먼저 대략적으로 그 범위를 조사해 보고 뒤에 소프트웨어 공학적 관점에서 직접적으로 關心이 있는 도구를 고찰할 것이다.

### 가. 언어, 패키지, 시스템의 分類

시뮬레이션, 소프트웨어에 의해 현재 제공되는 支援設備의 기본적인 분류를 시도하는 것도 유용한 일이 될 것이다. 편의상 설비를 다음과 같이 분류할 수 있다.

1) 언어의 演算(Linguistic Operations) : (가) 모델 기술과 프로그램 明細를 위한 구문과 의미, (나) 誤謬檢出, (다) 실행 가능한 형태의 프로그램으로의 翻譯 및 컴파일을 위한 設備

2) 시뮬레이션 實行과 統計 : 실행의 초기화, 종결, 그리고 사용자가 열거한 데이터를 가지고 사용자가 열거한 통제하에서의 시뮬레이션 실험의 재실행

3) 데이터 分析과 表示 : 통계적 분석, 최적화, 그래픽 등을 위한 기능 설비

4) 모델, 데이터, 프로그램의 貯藏과 檢索 : 화일, 데이터, 라이브러리 그리고 OS수준보다는 시뮬레이션 소프트웨어 내에서 접근할 수 있는 기타 같은 부류들에 대한 設備

이러한 분류를 토대로 다음 소프트웨어 用語들에 의한 設備를 기술할 수가 있다.

a. 시뮬레이션 言語 (Simulation Language) : 정의에 의해 1번 形態의 설비 언어 연산을 제공하고, 2번 形態의 설비 (실험통제)와 3번 形態의 設備 (분석/설비)를 소량 제공한다.

b. 시뮬레이션 패키지(Simulation Package) :

2, 3 形態의, 그리고 보다 최근에는 4번 形態의 일부를 위한 루틴(routines)을 提供하는데, 대개 어떤 제한된 영역내에서의 폭넓은 支援을 提供하고 있다.

c. 시뮬레이션 시스템(Simulation System) : 1,2,3,4 形態에 대한 통합적인 접근과 사용을 제공한다.

이러한 용어는 앞에서 대략적으로 言及된 모델링과 시뮬레이션 活動들의 전 變域에 대한 포괄적이고 통합된 지원을 위한 시뮬레이션 소프트웨어 기술 수준의 발전에 중점을 두고 있다. 이제 이러한 발전의 증거를 보여주고 있는 시스템을 논의할 것이다. 이러한 추세들은 대부분 언어, 패키지, 또는 시스템들과 聯關되어 분류된다.

### 나. 시뮬레이션 言語 特性

프로시슈어 및 요구되는 機械 明細로부터 독립적으로 모델을 構築하려는 趨勢 : 離散(discrete)變化 영역에 있어서, 기계어에서 FORTRAN으로, GASP로, GPSS, SIMSCRIPT 그리고 (ALGOL) SIMULA, SLAM으로의 발전은 기계 수준의 고려에서 보다 높은 수준의 모델중심의 구성으로의 移動을 분명하게 해주고 있다. CSSL (Continuous System Simulation Language) 정의에 의해 標準화된 連續 시스템 시뮬레이션 언어들에서도 똑같은 발전이 일어났다. CSMP, CSSL3, ASCI 등은 數值積分과 관련된 많은 계산상의 결정 판단으로부터 사용자를 보호하려는 函數的 構文(functional constructs)들을 포함하고 있다. 모델 明細가 사용될 하드웨어의 관점에서 不變이라는 원칙은, 이상적으로 어떤 모델 명세가 동등한 아날로그, 디지털, 혹은 하이브리드 프로그램들로 轉換될 수 있는 HLI에서와 같이 하이브리드 컴파일리 設計의 기본이 된다. 프로그램 작성으로부터 모델 기술을 분리시키는 방법에 여전히 많은 것이 남아 있다. 이 방향에서의 대부분의 정확한 이동은 질의에 의해 구성되는 사용자 입력으로부터 시뮬레이션 프로그램을 만들어 내는 GASP와 DRAFT와 같은 언어들에 의해 표현될 것이다. 開發되는 시뮬레이션 言語의

趨勢는 다음과 같다.

a. 시물레이션 프로그래밍에 있어서 函數的 요소들의 포함 및 분리에 대한 趨勢: 위에서 암시했듯이, 현재의 시물레이션 言語들은 사실은 시물레이션 패키지이다. 즉, 이들은 모델 구조에 대한 기술상의 規則을 넘어서 特徵들을 제공한다. 프로그램들은 實行制御와 分析/表示 형태의 작용을 實行하기 위한 모델 기술문(statement)들과 명령들의 혼합으로 복잡해지고 있는 趨勢이다. CSSL 정의는 적어도 약간의 분할(segmentation)(INITIAL, DYNAMIC, TERMINAL으로 의)을 제공하지만, 그 이상도 가능하다.

b. 모델링 형식화에 있어서 融通性(flexibility)에 대한 趨勢: 兩立할 수 없는 월드-뷰(world-view) - '連續' 대 '離散事件' 그리고 후자의 경우, '事件' 대 '活動' 대 '네트워크(network)' - 를 표현하는 의사전달이 되지 않는 형식화인 것으로 일단 간주되는 것들도 SIMSC-RIPT, GASP, 그리고 SIMULA와 標準言語들로 되어가고 있는 중이다.

#### 다. 시물레이션 패키지

시물레이션과 연관된 활동들의 광범위한 지원 趨勢를 보기위해 다음의 예를 살펴보자. 모델 操作(model manipulation) 패키지인 DYMOLA는 사용자나 그 環境에 의해 열거된 입력/출력 방향을 부과하기 위하여 모델 방정식들을 運營하는 프리프로세서(preprocessor)이다. FORSIM은 사용자가 편미분 방정식들의 집합으로서 明細된 모델을 일반 미분 방정식들의 시스템으로 쉽게 변경시켜 이 시스템을 시물레이션할 수 있도록 한다. 예를 들면 FORMAC과 같은 공식 조작 道具들은 미분방정식들로 이루어진 어떤 시스템을 그에 聯關된 派生 시스템을 만들기 위하여 기호적으로 작용하는데 사용될 수 있다. 그래픽 수단을 통해, 신호흐름 그래픽(signal flow graphs)을 정의하고 조작한다. 이러한 잠재적 膨창 모델 조작 範疇에 지원의 예가 있다.

a. 實行制御(Execution Control): WISSIM과 같은 패키지는 사용자가 CSMP에 포함된 것과 유사한 명령을 사용하는 FORTRAN 프로그램의

초기화, 실행, 종결을 쉽게 나열할 수 있도록 한다. APT는 정책검사를 위한 보다 폭넓은 제어를 제공한다.

b. 統計的處理(Statistical Processing): SAS, SPSS, AUTOGRP와 같은 패키지들은 사용자가 統計量, 相關關係, 信賴水準 등을 계산하기 위하여 데이터들에 대한 통계적 처리를 가능하게 한다.

c. 그래픽 디스플레이(Graphical Display): SIMPAK과 같은 패키지들은 시물레이션된(실변수) 軌跡들에 대한 간편한 디스플레이를 제공한다. 반면에 그 모델 상태에 대한 'moving picture'라는 SIMULA 외부집합(External Class)을 만들어낼 수 있다.

d. 最適化(Optimization): 線形 計劃(LP: Linear Programming)패키지는 LP문제들의 간편한 기술과 해를 위한 도구를 제공한다. ISML과 NAG와 같은 라이브러리는 여러 직접 탐색루틴들을 포함한다. 그러한 루틴들의 보다 통합된 패키지는 CSSL에 근거한 언어들로부터 이용된다.

e. 貯藏과 檢索(Storage and Retrieval): 시물레이션 출력을 위해 특별히 고안된 최초의 일반적인 데이터베이스는 SDL/1이다. 농업과 운송 모델과 같은 특별한 영역의 있는 모델들의 라이브러리가 開發되고 있는 중이다.

#### 라. 技能의 共有趨勢와 相互活用

DYNASTAT와 같은 패키지들은 통계처리 및 최적화를 위한 능력과 더불어 시물레이션 言語들의 기술적 技能들을 共有한다. SDL/1은 처음부터 SLAM내에서 호출될 수 있도록 考案되었다. 따라서 데이터 檢索과 貯藏에 직접 시물레이션을 연결시킨다. Cellular Space Simulation Package들은 모델 明細, 實行制御, 그리고 그래픽 디스플레이 능력을 제공한다.

사람과 기계 사이의 상호작용에 대한 지원은 선택할 수 있는 도구영역을 제공하는 패키지들 때문에 중요하게 되었다. 위에 언급된 패키지들과 언어들은 相互作用적이거나 최근에 완성된 대화형 버전(interactive version)을 處理한다.

마. 시뮬레이션 모델

統合과 包括性的의 趨勢는 앞에서 언급했듯이, 시스템과 패키지들이 設備들의 포괄적이고 통합된 技能을 제공하는 정도에 의하여 구별한다. 그러한 시스템은 지금도 상당히 드물지만, 이미 일부의 상당한 발전이 있었다. SCENARIO는 OSAF의 Avionic Simulation Laboratory에 있는 비행기, 制御 시스템, 感知 장치 등의 모델들이 공유되고 시뮬레이션될 수 있고, 따라서 표시 및 분석될 수 있는 상호 작용 기능을 제공한다. 工學者는 그의 문제나 관심에 따라 구성 요소들의 여러 細分化된 수준에서 모델들을 선택할 것이다.

計量 經濟學的의 모델링은 소프트웨어 시스템이 개발되고 있는 한 분야로서, 데이터 經營, 모델 構成, 모델들의 整列, 시뮬레이션 및 變數 測定을 위한 道具를 제공한다.

예를 들면 MBS(Model Bank System)는 시뮬레이션 패키지인 DYNAMO의 접근과 같은 여러 방법들 하에서 개발된 사회-경제학적 모델들을 연결하기 위한 여러 가지 편리한 기능을 제공하는 시스템이다. 보다 커다란 내용으로의 合併 趨勢는 모델링과 시뮬레이션 활동들은 경영정보 시스템, 법인계획 시스템, 그리고 금융계획 시스템들과 같은 보다 거대한 컴퓨터에 근거한 시스템으로 合併되고 있다.

3. 시뮬레이션 言語

가. 序 言

시뮬레이션을 效率的으로 遂行하기 위해서 여러가지 시뮬레이션 언어가 개발되었으며, 그 시뮬레이션의 特性에 알맞는 언어가 선택되어 널리 이용되고 있다. 오늘날 사용하는 대부분의 시뮬레이션 言語는 세가지 모델링 방식 중 하나 또는 그 이상을 이용하고 있는데 그 방식들은 네트워크(network) 모델링, 離散(discrete) 모델링, 連續(continuous) 모델링 그리고 連續과 離散의 混合 모델링이다. 그리고 이산 모델링은 두 방식 즉 사건계획방식(event-scheduling

approach) 또는 프로세스-상호작용방식(process-interaction approach)을 이용하고 있다. 현재 미국에서 널리 사용되는 시뮬레이션 언어인 GASP IV, SIMSCRIPT II.5, GPSS에 대한 간단히 비교 설명한 후 GASP IV에 뒤이은 새로운 시뮬레이션 언어인 SLAM을 소개 하고자 한다. 그리고 어느 시뮬레이션 언어를 구입 혹은 契約할 것인지 또한 특정 연구과제에 사용할 시뮬레이션 언어를 決定하기 위한 기준을 제시하고자 한다.

나. 汎用言語와 시뮬레이션 言語의 比較

시뮬레이션 연구의 遂行에 있어 가장 중요한 결정 중의 하나는 시뮬레이션 언어의 選擇이다. 부적절한 선택은 시뮬레이션이 제때에 완성될 수 없어 시뮬레이션 프로젝트를 실패로 끝나게 할 우려가 있다. 다음은 FORTRAN, PL/1, BASIC, C 같은 汎用言語와 비교한 시뮬레이션 언어로 프로그래밍하는 데 관련된 長點들이다.

- 1) 시뮬레이션 言語는 자동으로, 시뮬레이션 모델을 프로그래밍하는 데 필요한 대부분의 특징들을 제공한다. 결과적으로 프로그래밍하는 데 걸리는 시간을 상당히 감소시킨다.
  - 2) 시뮬레이션 언어들은 시뮬레이션 모델링을 위한 자연스러운 骨格을 提供한다. 이들의 기본적인 블럭(building block)들은 FORTRAN과 같은 범용언어의 블럭보다 시뮬레이션에 더 適合하다.
  - 3) 시뮬레이션 모델들은 시뮬레이션 言語로 작성되었을 때 일반적으로 필요시 보다 修正하기 쉽다.
  - 4) 대부분의 시뮬레이션 言語는 시뮬레이션 遂行동안 動的의 記憶場所 割當(dynamic storage allocation)을 가능하게 한다. FORTRAN의 경우에는 그렇지 못하다.
  - 5) 시뮬레이션 언어에서는 여러 가지 형태의 가능한 誤謬들이 구별되어 자동으로 검사되기 때문에 誤謬의 檢出이 쉽다. 보다 적은 양의 프로그램(code line)이 쓰여지므로 誤謬를 犯할 가능성이 더 적다.
- 한편 많은 시뮬레이션 모델들이 아직 汎用言

語로 작성되고 있다. 이러한 범용언어의 선택은 다음과 같은 장점을 갖는다.

1) 대부분의 사용자들은 汎用言語를 알고 있으나 시뮬레이션 언어는 그렇지 못하다.

2) FORTRAN이나 BASIC은 거의 모든 컴퓨터에서 이용이 가능하다. 그러나 특정 시뮬레이션 언어는 분석가가 사용하려 하는 컴퓨터상에서 이용이 불가능할 수도 있다.

3) 효과적으로 작성된 FORTRAN 프로그램은 시뮬레이션 言語로 작성된 프로그램보다 적은 수행시간이 걸릴 수 있다. 이는 시뮬레이션 언어가 一連의 블럭(building block) 들을 가지고 다양한 시스템들을 모델화하기 위해 設計된 반면에, FORTRAN 프로그램은 특정 應用에 맞추어 設計되었기 때문이다.

4) 汎用言語는 어떤 시뮬레이션 언어보다 많은 融通性이 프로그래밍에 허용된다. 예를 들면 복잡한 수치계산 등은 GPSS에서 쉽지 않다.

비록 두 형태의 언어를 사용하는 데는 명백히 각각의 장점이 있지만 사용자는 시뮬레이션 언어의 사용에 신중해야 한다. 만일 이러한 決定이 있어야 한다면 다음에 논의되는 언어의 選擇 基準들은 특정 시뮬레이션 언어를 결정하는 데 도움이 될 수도 있다.

#### 다. 시뮬레이션 言語의 種類

시뮬레이션 言語로 널리 사용되고 있는 GASP IV, SIMSCRIPT II.5, GPSS, SLAM에 대해서 그 特性을 살펴보기로 한다.

##### 1) GASP IV

GASP IV(Pristsker)는 30개 이상의 FORTRAN 서브루틴과 함수(function)들로 구성되는 사건중심(event-oriented) 시뮬레이션 언어로 각 부프로그램은 필요한 시뮬레이션 活動(activity)을 遂行한다. GASP IV는 FORTRAN으로 쓰여졌기 때문에 배우기 쉽고 FORTRAN 컴파일러를 가진 거의 모든 컴퓨터상의 이용이 가능하다 (대략 20,000단어의 기억장소를 요함). GASP IV와 SLAM은 가장 문서화된 언어이며 離散事件 시뮬레이션은 물론 연속 및 혼합된 離

散-連續 시뮬레이션(combined discrete-continuous simulation)을 수행할 수 있는 언어이다.

GASP IV는 FORTRAN의 확장이므로, 시뮬레이션을 위해 특별히 고안된 언어보다는 세련되지 못하며, 강력하지 못하다. 그러나 적은 費用으로 이용이 가능하여 많은 조직체에 경제적인 선택이 되어 왔다.

##### 2) SIMSCRIPT II.5

SIMSCRIPT II.5는 오늘날 강력한 시뮬레이션 言語로 많은 사람들이 생각하고 있는 사건중심 혹은 프로세스중심 시뮬레이션 言語이다. 1960년대초 Rand 회사(Rand Corporation)에 의해 개발되어온 이래로, SIMSCRIPT는 수많은 version들을 통해 발전되어 왔는데 가장 최근의 version인 SIMSCRIPT II.5는 CACI, Inc.의 특허제품이다. SIMSCRIPT II.5는 IBM, UNIVAC, CDC, Honeywell, CDC 등에서 만든 대다수의 컴퓨터에 사용이 가능하며, 또한 時分割 네트워크(time-sharing network)에서도 사용 가능하다. SIMSCRIPT에서는 이용 가능한 명령문의 다양성으로 인하여 일반적인 프로그래밍 작업들이 FORTRAN에서 보다 效果的으로 이루어질 수 있다. 더우기 영어와 유사하며 자유로운 형태의 構文(syntax)은 SIMSCRIPT 프로그램을 읽기 쉽고, 거의 문서화(self documenting)할 수 있도록 해준다. SIMSCRIPT II.5는 시뮬레이션 출력 결과에 대한 통계적인 분석을 遂行하는 패키지를 가진 중요 시뮬레이션 言語 중의 하나이다. SIMSCRIPT II.5의 단점은 다른 시뮬레이션 言語와 비교해 볼 때, 教材가 없다는 것이다. 이 어려움은 가까운 시일 내에 解決될 것으로, 현재 CACI에서 마련중에 있다. GPSS/H를 제외하고 다른 言語와 비교할 때, SIMSCRIPT II.5의 또 하나의 短點은 구입비용이 많이 든다는 점이다. 결과적으로 SIMSCRIPT II.5의 프로그래밍 能力이 최우선인 거대한 시뮬레이션 프로젝트나 시뮬레이션을 많이 사용하는 조직단체에서 사용되는 경향이 있다.

##### 3) GPSS

GPSS(General-Purpose Simulation Sys-

tem)는 대기 시스템에 아주 적합한 프로세스중심 시뮬레이션 言語이다. 본래 1960년대초 IBM 회사에서 개발하였으며, 여러 version들을 통해 발전되어 왔다. 최근의 두 version은 GPSS/360과 GPSS V이다. 역사적으로 GPSS는 가장 인기 있는 시뮬레이션 언어로 많은 시뮬레이션의 큐잉특성과 컴퓨터 산업에 있어 IBM의 축적된 영향에 기인한다. GPSS-360/V는 IBM 360,370 컴퓨터상의 이용이 가능하며 다른 회사가 개발한 version들은 대부분의 주요 컴퓨터와 時分割 네트워크(time-sharing network)에서의 이용이 가능하다(이들 version들은 GPSS 360/V와 완전히 일치하지 않는다). 가장 흥미로운 새로운 구현(implementation)은 James O.Henriksen이 개발한 GPSS/H이다. GPSS/H는 GPSS 360/V의 해석방식(interpretive approach)과 비교해 볼 때, 컴파일러 언어이다.

GPSS의 주요 魅力은 편리함과 시뮬레이션 모델 구성의 속도이다. 많은 시뮬레이션 프로젝트들이 시간 제약을 받기 때문에 프로그래밍 능력은 중요한 고려사항이 될 수 있다. 그렇지만 다른 시뮬레이션 言語처럼 GPSS도 자체의 단점을 가지고 있다. GPSS에서 시뮬레이션 時計는 정수 값을 가지며 시뮬레이터는 默示的으로 모든 活動(activity) 시간을 표현하는 기본 시간 단위를 선택해야 한다. GPSS 360/V는 지수분포나 정규분포 등의 표준 확률 분포로부터 確率變數를 생성하는 자동 메카니즘을 가지고 있지 않다. 대신 사용자는 일련의 직선을 사용하여 원하는 확률 변수의 分布函數를 역으로 근사치를 구한다. GPSS는 프로그래밍 柔軟性(flexibility) 면에서 GASP IV나 SIMSCRIPT II.5 보다 못하다. 복잡한 수치적분을 하거나, GPSS를 사용하여 특별한 출력 레포트들 얻으려는 분석가는 서브루틴(예를 들어 FORTRAN)을 작성하여 GPSS HELP 명령문에 의하여 자신의 프로그램과 접속시켜야 한다. 끝으로 GPSS 360/V 난수열(random number stream)발생기(generator)는 缺點을 가지고 있다. 예를 들면 8개의 모든 난수열이 동일하다. 그러나 어떤 GPSS version들은 다른 난수발생기를 사용하고 있기도 하다.

GPSS언어는 40개의 표준 명령문들로 구성되며, 각 명령문은 수행 동작을 암시해 주는 블럭이라는 圖式的인 표현을 갖는다. GPSS모델을 구성한다는 것은 시스템을 통해 고객들이 거치는 과정들을 나타내는 블럭 圖表로 표준 블럭들을 결합시키는 것으로 생각할 수 있다. 블럭도표 모델(block-diagram model)이 구성되면, 컴퓨터에서 수행할 GPSS 명령문들로 번역된다. 그리고 블럭 圖表 자체가 프로그래밍 言語에 익숙하지 않은 관리자에게 모델의 特性을 설명해주는 데 유용하다.

시스템의 서비스를 요하는 고객 혹은 일시적인 個體(entity)를 GPSS에서는 transactions이라 한다. transaction이 요하는 서비스를 제공하는 서버 혹은 영구 개체들을 facilities 혹은 storages라 하며 각각 단일 서버 혹은 서버들의 그룹에 해당한다.

#### 4) SLAM

SLAM(Simulation Language for Alternative Modeling)은 Pegden과 Pritsker에 의해 개발된 시뮬레이션 언어로서 네트워크 모델링, 離散 모델링, 連續 모델링, 混合 모델링이 가능하다. SLAM의 이산 모델링에서는 사건중심(event-oriented)과 프로세스중심(process-oriented) 방식을 동시에 사용하고 있다. SLAM의 사건중심방식은 GASP IV와 유사하다. 프로세스중심 방식에 있어 사용자는 노드(node)라 불리는 표준기호들과 가지(branch)들을 사용하여 시스템을 상호연결된 네트워크 구조로 표현한다. 그러므로 고객이 시스템을 거쳐가는 過程(process)에 대한 도식적인 표현으로서 네트워크를 생각할 수 있다. 시스템에 대한 네트워크 모델이 만들어진 후, 컴퓨터상으로 수행 가능한 일련의 SLAM 프로그램 문장들로 번역된다.

SLAM의 실제 매력은 SLAM이 주는 모델링 방법의 多樣性으로 나타난다. 이용자들은 사건이나 프로세스(혹은 둘다)를 이용 離散事件 모델을 구성할 수 있으며, 미분방정식을 포함하는 연속모델을 구성할 수도 있다. 또한 상기의 모든 요소들을 사용하여 혼합모델(combined model)



을 구성할 수도 있다. 離散 모델링과 連續 모델링을 위한 SLAM의 프로그램 구조는 각각 그림 2와 그림 3과 같다. 특히 SLAM은 GASP IV와 같은 장점들을 가지고 있고 모델링의 柔軟性과 그 능력을 증대시키므로, 마치 GASP IV를 대신한 것처럼 보인다.

SLAM의 프로세스-상호작용방식은 사건계획방식에 비하여 여러 가지 잇점을 가지고 있다. 첫째로, 많은 형태의 시스템에 있어 이 방식은 個體(entity)가 시스템을 거쳐가는 동안 개체의 전체과정(experience)은 단일 루틴으로 기술되기 때문에 보다 자연스런 構造(frame-work)를 제공한다. 더우기 이 방식이 시뮬레이션 언어로 구현될 때 그 언어는 강력한 용어(macrostatement)를 제공하는데, 시뮬레이션 모델에서 흔히 발생하는 狀況을 (예를 들면 대기 시스템에서 고객의 도착) 사건 논리(event logic)로 자동적으로 바꿔게 한다. 결과적으로 이 방식을 이용하여 작성한 모델은 사건계획방식을 이용하여 작성한 모델보다 더 적은 프로그램 라인으로 요약할 수도 있다.

한편, 프로세스-상호작용을 이용한 시뮬레이션 言語는 단지 주어지는 표현용어(macrostatement)들로 제한하기 때문에 융통성이 결여될 수도 있다. 프로세스-상호작용방식은 SLAM뿐만 아니라 SIMCRIPT II.5와 GPSS에서도 허용된다.

라. 시뮬레이션 言語의 選擇基準

汎用言語가 아닌 시뮬레이션 言語를 사용하기로 결정을 했다는 가정하에 시뮬레이션 언어를 선택하는 데 있어 도움이 되는 기준들을 보이기로 한다. Shannon이 제안한 것처럼 시뮬레이션 언어에 관한 決定을 내리는 데는 두가지 단계가 존재한다. 첫번째 단계에서 사용할 조직단체는 언어를 구입할 것인지 혹은 임대할 것인지를 결정해야한다. 한 언어가 다른 언어보다 주어졌던 應用에 더 適合할 수도 있기 때문에 한가지 언어만 선택할 필요는 없다. 두번째 단계에서 분석가는 어떤 언어를 시뮬레이션 연구에이용할 것인지를 결정해야 한다.

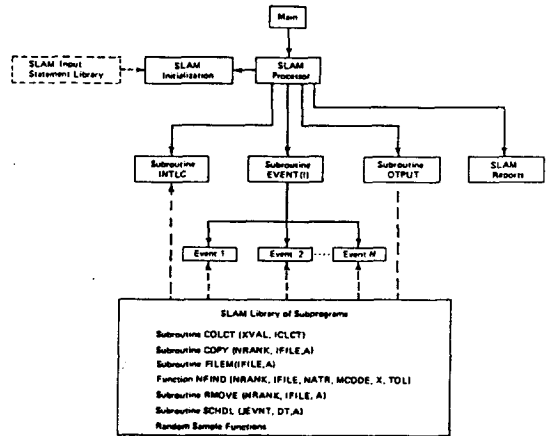


Figure 2. SLAM II organization for discrete event modeling.

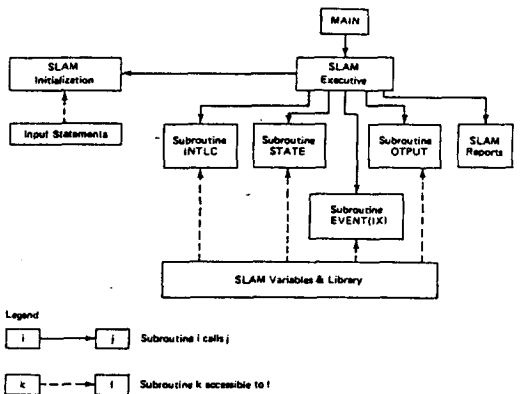


Figure 3. SLAM II organization for continuous modeling

다음은 조직단체가 시뮬레이션 言語의 選擇에 있어 고려해야 할 몇 가지 기준에 관한 사항들이다.

- 1) 회사의 컴퓨터 시스템상에서 言語의 利用可能性
- 2) 言語를 設置, 유지하는 데 드는 費用
- 3) 遂行할 시뮬레이션 研究 횟수
- 4) 시뮬레이션될 시스템의 形態
- 5) 言語의 文書化 程度

- 6) 言語의 배우기 쉬운 程度
  - 7) 言語의 컴퓨터 記憶場所 요구량
  - 8) 言語의 컴퓨터 遂行시간의 효율성
  - 9) 言語의 柔軟性(flexibility) 및 能力(power)
- 사용자가 특정한 應用에 필요한 시뮬레이션 言語를 選擇해야 할 때, 위의 기준들 중의 대다수가 關聯性이 있다. 그러나 추가로 다음의 요인들 또한 중요하다.

- 1) 회사에서 혹은 時分割 네트워크상 言語의 利用 可能性
- 2) 해결해야 할 問題의 性質
- 3) 사용자가 알고 있는 言語
- 4) 시뮬레이션 研究에 이용 가능한 시간의 總量 및 言語가 요하는 프로그래밍 努力
- 5) 言語로 작성한 모델의 携帶를 위한 簡便性

(흔히 정부조직 및 군사조직에 중요)

6) 모델의 특성을 프로그래머 이외의 다른 사람, 즉 모델 사용자가 관리자에게 전달하는 言語의 能力

시뮬레이션 언어를 選擇하는 데 있어서 추가로 몇 가지 안내를 제공하기 위해 표 2에 14가지 選擇基準에 대한 GASP IV, SIMSCRIPT II.5, GPSS, SLAM의 비교를 보였다. 이 기준에 대한 리스트는 완전한 것이 아니다. 특히 컴퓨터 수행 시간 效率를 포함시키지 않고 선택하였는데 이는 수행시간 效率가 주로 사용된 언어, 프로그래머의 숙련도, 응용에 종속되는 까닭이다. 평가된 많은 기준들은 應用에 從屬의이며 이 비교는 일반적으로 언어들에 이 평가기준에 상대적으로 어떠한지를 보여주기 위한 것이다.

표 2. GASP IV, SIMSCRIPT II.5, GPSS, SLAM의 比較

區 分	言 語			
	GASP IV	SIMSCRIPT II.5	GPSS	SLAM
이용가능한 시스템	모든 시스템	모든 시스템	대기행렬시스템	모든 시스템
가격	낮음	높음	GPSS V, 중간 ; GPSS/H, 높음	낮음
사건계획방식	예	예	아니오	예
프로세스-상호작용방식	아니오	예	예	예
혼합된 離散형-連續형 시뮬레이션	예	예	아니오	예
확률변수 발생의 容易性	예	예	아니오	예
결과분석 패키지	한정적	예	한정적	예
시뮬레이션 모델링	보통	좋음	매우 좋음	매우 좋음
학습의 容易性	매우 좋음	보통	좋음	매우 좋음
컴퓨터에의 互換性	매우 좋다	좋음	보통	매우 좋음
문서화 정도	매우 좋음	보통	매우 좋음	매우 좋음
柔軟性	매우 좋음	매우 좋음	보통	매우 좋음
프로그램 내의 의사 전달 능력	보통	매우 좋음	매우 좋음	매우 좋음

4. 시뮬레이션 시스템

시뮬레이션 모델 開發에 관한 研究는 보통 資

料 收集, 모델 設定 및 實行, 代替 모델 提示, 結果 分析, 그리고 結果에 따른 要約 및 提案 등의 순으로 전개된다. 시뮬레이션 언어로 구성된 소프

트웨어들이 이러한 시뮬레이션 과정들을 지원해 줌으로써 사용자들이 손쉽게도 效率的으로 産業 및 應用 工學的인 시스템들을 計劃하고 分析하여 설계할 수 있도록 하여 준다. 이같은 시뮬레이션에 의한 模擬 實驗을 통하여 어떤 시스템을 실제로 設計 제작하지 않고서 分析을 할수 있기 때문에 더욱 더 이상적이며 또한 기존의 工程이나 工場들의 시스템들을 分析함으로써 문제점들을 해결 보완할 수 있다. 따라서 이처럼 시뮬레이션 言語를 기초로 해서 시스템의 분석을 쉽게 하고 학습을 용이하게끔 개발된 종합적인 시뮬레이션 소프트웨어가 바로 시뮬레이션 시스템이다. 시뮬레이션 시스템의 하나인 슬램시스템에 대해서만 說明하기로 한다.

가. 슬램시스템 (SLAMSYSTEM)

슬램시스템은 SLAM을 기초로 開發된 하나의 시뮬레이션 시스템으로서 기존 기타 소프트웨어에 비해서 사용하기가 쉽고 그 기능 면에서도 뛰어난 소프트웨어이다. 이 시스템은 그림 4와 같이, 포트란 言語에 기초를 둔 슬램 언어(SLAMII)를 이용하면서 모델을 구축하고 분석하며 시뮬레이션 결과들을 자동적으로 제시하여 준다. 그리고 슬램시스템은 시뮬레이션의 결과를 그래픽으로 나타내어 그 시스템의 동적 변화를 화면에 표시할 수 있는 애니메이션 技能을 갖고 있다. 슬램시스템은 連續 및 離散 모델들을 개발하는데 사용될 수 있을 뿐 만 아니라 복합적인 混合 모델(combined model) 개발시에도 매우 편리하게 이용될 수 있다. 이에 슬램시스템의 特徵 및 長點들을 요약하면 다음과 같다.

1. 특정 시스템을 모델링 할 경우 일일이 汎用 言語로 프로그램을 짜지 않고서도 슬램 언어의 記號들을 이용하여 네트워크를 형성함으로써 손쉽게 시뮬레이션을 할 수 있다. 즉 소프트웨어 자체내에서 네트워크의 각 기호들을 슬램 언어로 자동적으로 옮겨 실행함으로써 불필요한 프로그래밍을 요하지 않는다. 물론 어떤 현상을 슬램 기호로 나타낼 수 없을 경우이나 連續인 모델링인 경우에는 특별히 FORTRAN 언어로 프로그램을 추가시킬 수도 있다.

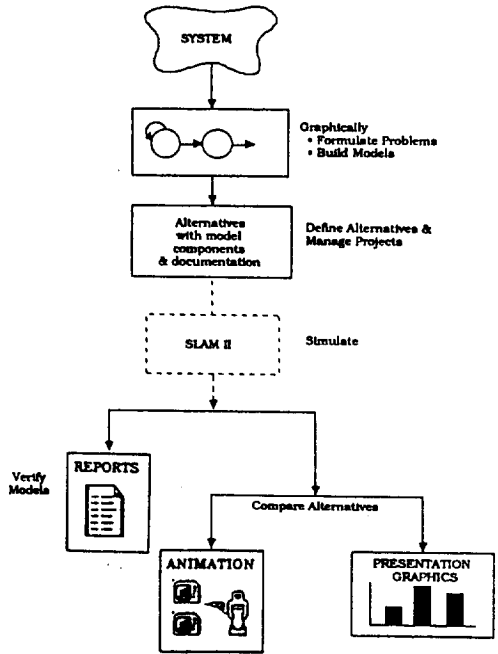


Figure 4. SLAMSYSTEM support environment

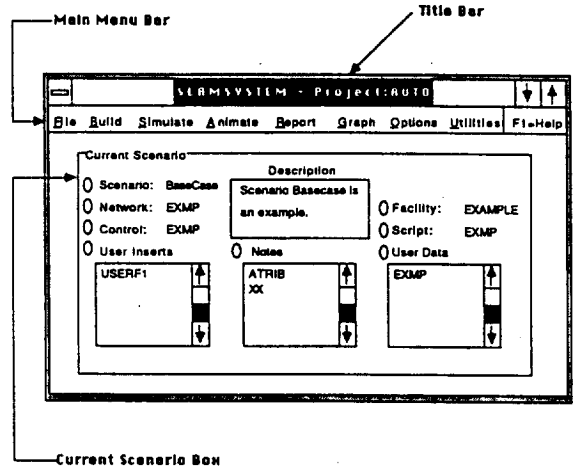


Figure 5. SLAMSYSTEM executive window

2. 슬램시스템은 Microsoft Windows (ver.3.0 이상)을 基礎로 하여 開發되어 졌기 때문에 pull-down 메뉴 방식(그림 5)으로 運轉되어 손쉽게 사용할 수 있을 뿐만 아니라 여러 작업을 동시에 할 수 있기 때문에 效率的인 시뮬레이션을 할 수 있다. 또한 각 작업시 마다 메뉴에서

자동적으로 필요한 사항과 잘못 기입된 것들을 지적해 줌으로써 보통 시뮬레이션 모델링에서 발생하기 쉬운 犯失을 最少化할 수 있다.

3. 슬램시스템은 하나의 연구과제(project)에 대해서 여러개의 代替 모델(alternative model)들을 提示할 뿐만 아니라, 代替 모델간의 比較 分析을 통하여 적정 모델을 찾을 수 있도록 하였다.

4. 슬램시스템은 시뮬레이션에 사용된 모든 變數들에 대해서 自動적으로 가능한 모든 것들을 分析한 후 結果들을 막대, 선, 파이 등의 그래프들과 報告書 식으로 提示하여 준다.

5. 슬램시스템은 애니메이션(animation) 기능을 갖고 있기 때문에 시뮬레이션 결과를 즉 分析하고자 하는 시스템의 構成 因子들의 動的 變化를 그래픽으로 나타낼 수 있다.

### III. 結 言

시뮬레이션 言語와 그에 관련된 소프트웨어들은 시뮬레이션을 效率적으로 수행하기 위해서 개발되었으며 현재도 개발되고 있다. 이같은 소프트웨어의 중요성에도 불구하고 어떤 시스템을 分析하고 設計하기 위해 시뮬레이션을 할 때 汎用言語를 사용하는 경우가 많이 있다. 범용언어를 사용하여 시뮬레이션을 하면 프로그램 작성에 많은 노력과 시간이 소요되고 誤謬를 범할 확률도 높으며 結果 分析에도 많은 어려움이 따

른다. 따라서 시뮬레이션을 하고자하는 경우 시뮬레이션 언어에 기초를 둔 소프트웨어를 사용하는 것이 매우 效果的이며, 또한 分析하고자 하는 시스템의 特性에 알맞은 시뮬레이션 소프트웨어를 選定하는 것이 바람직하다.

특히, 여러가지의 소프트웨어중에서도 슬램시스템은 社會科學 및 自然科學에 관련된 모든 시뮬레이션을 손쉽고 빠르며 效率적으로 할 수 있도록 도와주는 강력한 시뮬레이션 소프트웨어이다. 이같은 시스템을 익혀서 우리의 農業分野에 있어서의 시뮬레이션 모델링에 適用한다면 매우 고무적일 것 같다. 그리고 다음에 機會가 있으면 슬램시스템에 대해서 더욱 자세히 紹介하고자 한다.

### 參 考 文 獻

1. 이영해, 백두권, 1990. 시스템 시뮬레이션, 경문사.
2. A. Alan B. Pritsker, 1986. Introduction to Simulation and SLAM II, Third edition, A Halsted Press Book, John Wiley & Sons, NY.
3. A. A. Pritsker, C. E. Sigal and R. D. Jack Hammesfahr, 1989. SLAM II Network Models For Decision Support, Prentice-Hall, Inc.
4. SLAM II Quick Reference Manual, 1990. Pritsker Corporation, IN.
5. SLAMSYSTEM : Total Simulation Project Support, 1990. Version 2.1, Pritsker Corporation, IN.