Knowledge-Based Approach for Computer-Aided Simulation Modeling

Young-Hae Lee* and Nam-Young Kim*

컴퓨터에 의해 수행되어지는 시뮬레이션 모델링을 위한 지식베이스 접근방법

이영해* · 김남영*

Abstract

A computer-aided simulation modeling system has been developed to allow the automatic construction of complete discrete simulation models for queueing systems. Three types of knowledge are used in the specification and construction of a simulation modeling: Knowledge of queueing system, simulation modeling, and a target simulation language. This knowledge has been incorporated into the underlying rule base in the form of extraction and construction rule, and implemented via the expert system building tool, OPS5. This paper suggested a knowledge based approach for automatic programming to enable a user who lacks modeling knowledge and simulation language expertize to quickly build executable models.

1. Introduction

Simulation as an analysis and design tool became popular after the advent of digital computer technology which allowed for high speed computation and large scale model programming. Simulation is the process of designing a model of a real world system and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies for the opera-

---

* Department of Industrial Engineering, Hanyang University

※ 본 논문은 1988년도 문교부 지원 학술진흥재단의 자유 공모과제 학술연구 조성비에 의하여 연구되었음

tion of the system[1]. The simulation life cycle is represented in Fig. 1.

In order for simulation to be valueable tool, the following knowledge are required from the simulationist[2];

1) Knowledge about the system to be simulated.

2) Knowledge about how to model the system.

3) Knowledge about the simulation language.

4) Knowledge about statistics for output analysis.

Despite the fact that simulation has its roots in the "science" of computer science, mathematics and statistics it is still considered to be an "art" and somewhat of an intuitive process. The recent surge of interest in artificial intelligence(AI) and expert system has led several simulationist to point out similarities between AI and Simulation thchniques and to suggest the potential for the use of AI and techniques in simulation[3, 4, 5, 6]. With particular regard to the model building aspect of the simulation life cycle, the work presented here consolidates earlier work in automatic programming with recent development toward expert system building tools.

Traditionally, a simulationist would build a model using a general purpose programming language or general purpose simulation language. That model would then be validated and executed to allow experimentation and the results analyzed and interpreted. Because of the difficulties in learning the details of current high level simulation language, some users employ the services of a simulation analyst to model and programs the system under their study. Various portion of this simulation life cycle can be automated by making use of current AI and expert system tools and techniques. Therefore, this paper represent a rule based

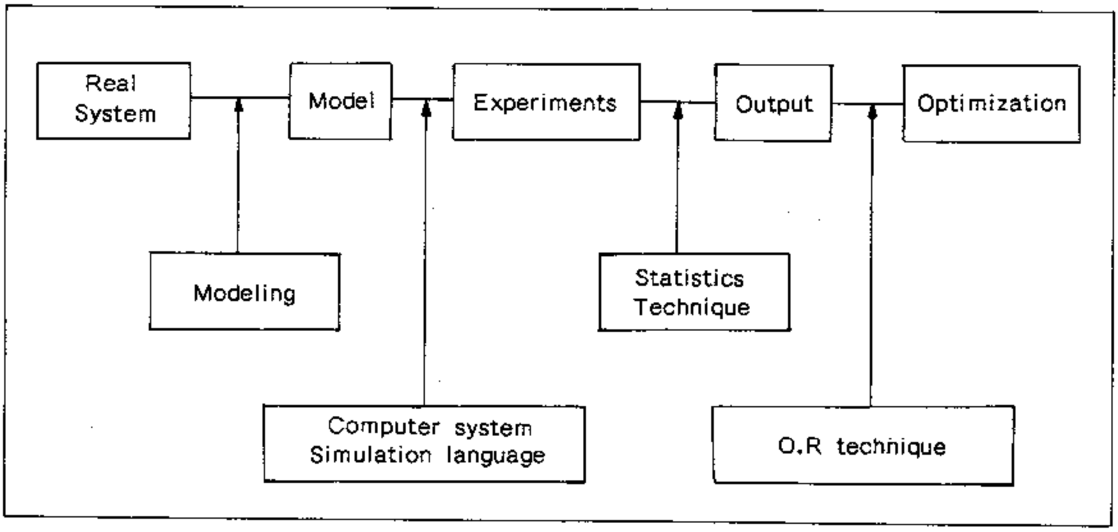

Fig. 1 Simulation Life Cycle.

system that aids the user with little or no knowledge of simulation model building to construct simulation model.

2. KBSM System Overview

The design of the knowledge-based system for the automatic construction of simulation model include four major components:

- The development of specification method
- The design of an internal representation method
- The identification of knowledge needed for model construction
- The development of rule base for the codification and implementation of that knowledge using an expert system building tool.

On overview of the KBSM system with implements this knowledge-based automatic programming approach to specification acquisition and discrete simulation model construction is in Fig. 2.

Three distinct types of knowledge are used as follows.

2-1. Domain knowledge

Domain knowledge for the KBSM system is knowledge of restricted subset of queueing system. Queueing system is described following components:

- arrival process
- service mechanism
- discipline of its queue
- balking or blocking

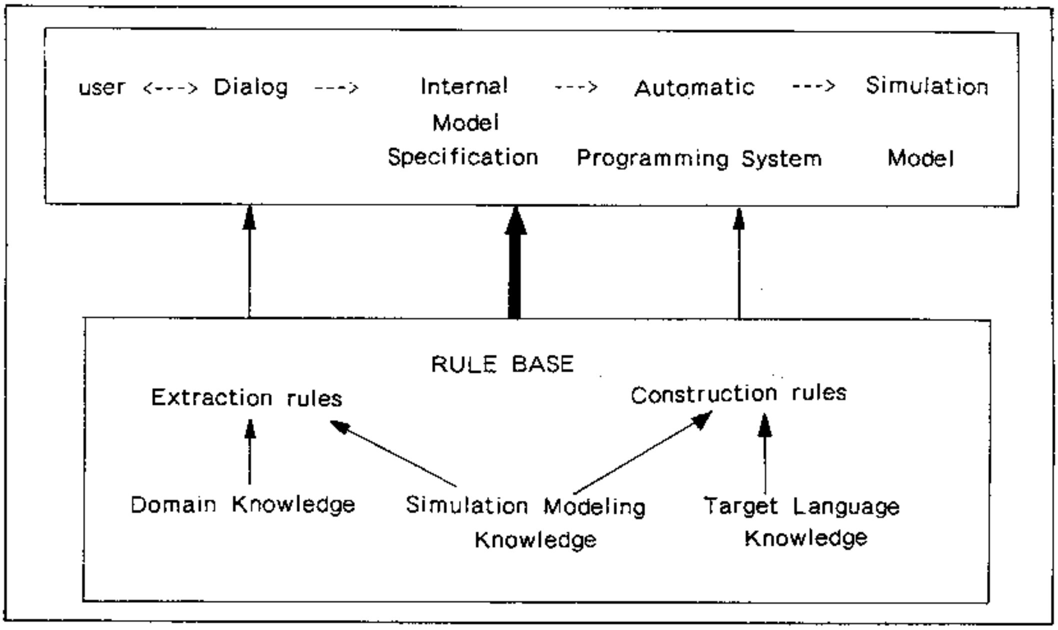

Fig. 2 KBSM system overview.

- looping[14]

2-2. Modeling knowledge

The general requirements for any model and the design decisions needed for the construction of a specific model. All models have several common characteristics. Models are composed of:

- temporary entities
- permanent entities
- entity flow and interactions
- statistics needed
- model termination conditions

2-3. Target language knowlege

Target language knowledge, SLAM II [15], includes an understanding of the semantics of the target language to allow the proper selection of construction for model implementation. Knowledge of the syntax is also necessary to produce an executable model.

This knowledge is incorporated into the KBSM system rule base in the form of condition-action rules. When a rules set of specified conditions are satisfied by the contents of working memory, the system fire the rules by executing the corresponding set of actions. The actions frequently change information within the system such that other conditions are then satisfied and the cycle is repeated.

In general, domain knowledge and modeling knowledge are combined to form specification extraction rules, and modeling and target language knowledge are combined to form model construction rule. The extraction rules guide the interactive specification session with the user. This process takes the form of a question and answer dialog or menu selection, to obtain an internal specification of a desired model. The construction rules then transforms the internal specification into an executable simulation model in the target language, SLAM. These transforms to and from Internal Model Specification(IMS) constitute the exetraction and construction processes as illustrated in Fig. 3.

3. Design of Internal Model Specification

The overall objective in the design of the IMS was to develop a set of language independent generic data structure that could

```
              extraction                          construction
User ----------------------> IMS --------------------> Model
```

Fig. 3 Transformation to and from IMS structure.

effectively represent the information provided in a queueing system and retain the information necessary for modeling that system. The first step in the development of the IMS structure was to evaluate several sample case studies within the restricted queueing system domain.

3-1. Evaluation of Sample Cases

The purpose of the case study evaluation was to identify the information provided in a queueing system problem description and to determined the role of that information in a model of the described system. The characteristics exemplified by the selected cases in SLAM II text book are summarized in Table 1.

3-2. Structure Design

IMS structure design allows specification

Table 1. Summary of cases used throughtout KBSM system design

Case	Characteristics
1. Bank	single tempoary entity type. multiple server, single service types. exponential, uniform distribution. termination : total customer.
2. Job shop	multiple temporary entity types. single server, single service types. exponential, normal distribution.
3. Maintenance	single server, multiple service types. triangular, uniform distribution. balking, blocking termination : subcontracted total entity.
4. Work Station	single temporary entity multiple service in series exponential distribution balking, blocking termination : total run time statistics : time in system
5. TV Inspection System	multiple server, multiple service types. statistics : time in system probabilistic branching looping termination : total run time

of both the static and dynamic component of a real queueing system and thus provides a precise description of what the model is to do rather than how it is to do it. The static component includes description of the temporary and permanent entities, the necessary distribution information, model termination characteristics. The dynamic components are those that describe the flow and interrelationship of entities as they progress through the model. Table 2. provides a key to these structures.

Table 2. IMS structures for static model component specification

Structure	Descroption and possible values
SYSTEM	
name	system description or model name
DATE	
day	form : mm/dd/yy
MODELER	
name	modeler name
TEMP_ENTITY_TYPE_INFO	
general_name	
number	number of types
TEMP_ENTITY	
name	if temp_entity_type_info. number = 1, temp_entity_type_info. general_name otherwise, name given by user
PERM_ENTITY_TYPE_INFO	
general_name	
number	number of types
PERM_ENTITY	
name	general_name if perm_entity_type_info. number = 1
number	number of server
service_id	service name
CREATE	
name	temp_entity. name
tbc	time between arrivals
tf	time for first creation of entity (default : 0)
mc	max. number of entities to be created (default : infinite)
m	max. number of branches to be taken (default : infinite)

Structure	Descroption and possible values
QUEUE_INFO	
name	perm_entity. name
iq	initial number of entities in queue (default : 0)
qc	queue capacity (default : infinite)
ARRIVAL_INFO	
temporary_id	temp_entity. name
distribution_id	entity type number for interarrival time
DISTRIBUTION	
id	arrival_info. distribution_id service_info. distribution_id
name	exponential, uniform, normal, triangular, constant
number 1	parameter dependent on distribution
number 2	parameter dependent on distribution
number 3	parameter dependent on distribution
DISPOSITION	
value	balking or blocking
BALKING	
name	node lable to be sent
value	termination or collect statistcs.
SERVICE_INFO	
name	service name
permanent_id	perm_entity. name
distribution_id	service type number
STATISTICS	
required_name	time in system or service time
required_name	time in system or service time or server utilization, etc.
histogram	
BALKING_TERMINATION	
number	number of balked entity for system termination.
MODEL_TERMINATION	
condition	run time or total entities
value	time or entity count
TEMP_ENTITY_FLOW_INFO	
name	temp_entity. name
previ_id	arrival or service_info. name
proportion	total proportion must be 100%
next_id	service_info. name or depart

4. Rule-Base Development

The transformation of user response to IMS structures and from IMS structures to final SLAM model and statements are achieved by the application of rules. Thus the development of the KBSM system rule base combines the specification, internal representation, and construction methods with the three types of knowledge.

4-1. Selection of Expert System Building tool

The KBSM system is a constructive system. A forward chaining inference mechanism would be desirable, meaning that the system would work from facts that describe an initial state to a final state representing conclusion. For the extraction process the initial state is that of having no information about the system to be simulated, that is, no specification, and the final state is a complete specification.

Likewise for the construction process, the initial state is that of having a specification and no SLAM model, and the final state is SLAM model that satisfies that specification requirements.

OPS5[16], as an expert system building tool was selected for the implementation of the system. It is an expert system language that a forward chaining inference mechanism and support both rule-based and imperative programming. The three components of OPS5 are working memory, production memory, and inference mechanism.

4-2. Formalization

Formalization is the process of organizing knowledge into English_like rules. The three kinds of knowledge are incorporated into the underlying rule base for the KBSM system as extraction and construction rules which take the form:

– condition ························=>action

The example of extraction rules are as follows.

– If arrival rate is unknown, ask the user to select the appropriate distribution to describe the time between arrivals.

– If the arrival distribution is normal, ask the user for the mean and standard deviation.

The extraction rules mentioned above give expression to OPS5 program, which is as follows.

```
(p arrival-rate-information)
   (temp-entity^name<B>)
   (arrival-info^temporary-id <B>
   ^distribution-id arrival-rate)
-(distribution)

⟶

(write)
  (crlf)
  | Which distribution best describe |
  |  the  arrival-rate?              |(crlf)
  |                                  |
  |        1-EXPONENTIAL             |(crlf)
```

```
      |  2-UNIFORM                  | (crlf)
      |  3-NORMAL                   | (crlf)
      |  4-CONSTNAT                 | (crlf)
      |  5-TRIANGULAR               | (crlf)
      |                             |
      | Enter the appropriate       |
      | distribution  number:       |
(make distribution^id arrival-rate
^name(accept))
(p normal-info)
  (distribution^name 3)
──→
(write)
  (crlf)
   |  What is the mean value?   | (crlf)
   |  Enter the value:          |
(bind<D> (accept))
(write)
  (crlf)
  |What is the standard deviation?| (crlf)
  |Enter the value:               |
(modify 1^name normal^number1<D>^number 2 (accept))
```

The execution result of extraction rule program, the distribution structure retained the following attribute values.
(distribution id arrival-rate name normal number 1 10 number 2 3)

The example of contruction rules as follows.

– If the arrival of temporary entity and parameter are accepted, generate a CREATE node structure.

– If time in system is needed, add a mark to the CREATE node using the first available attribute.

5. An Example

This section illustrates the use of process of KBSM system for the automatic construction of a serial work station model. The system to be simulated is described as follows.

The maintenance facility of a large manufacturer performs two operations. These operations must be performed in series; operation 2 must be follows operation 1. A proposed design leaves space for two units between the work stations, and space for four units before work station 1. Historical data indicates that the time interval between request for maintenance is exponentially distributed with a mean of 0.4 time units. Service times are also exponentially distributed with the first station requiring on the average 0.25 time units and the second station, 0.5 time units. If the queue of work station 2 is full, that is, if there are two units waiting for work station 2, the first station is blocked and a unit cannot leave the station. A blocked work station cannot serve other units.

To evaluate the proposed design, statistics on the following valuables are to be obtained over a period of 300 time units:

1. work station utilization.

2. time to process a units through the two work stations.

3. number of units/time units that are subcontracted.

4. number of units waiting for each work

station

5. fraction of time work station 1 is blocked.

The schematic diagram and SLAM II network illustrated in figure 4. and 5.

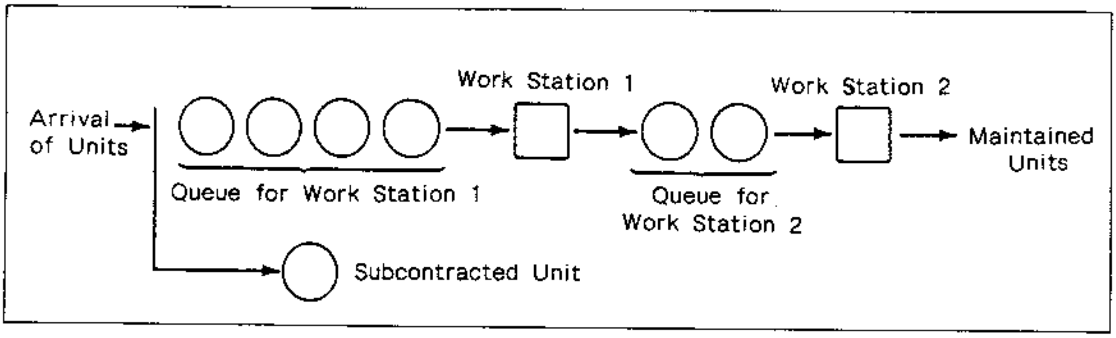

Fig. 4 Schematic diagram of work stations in series.

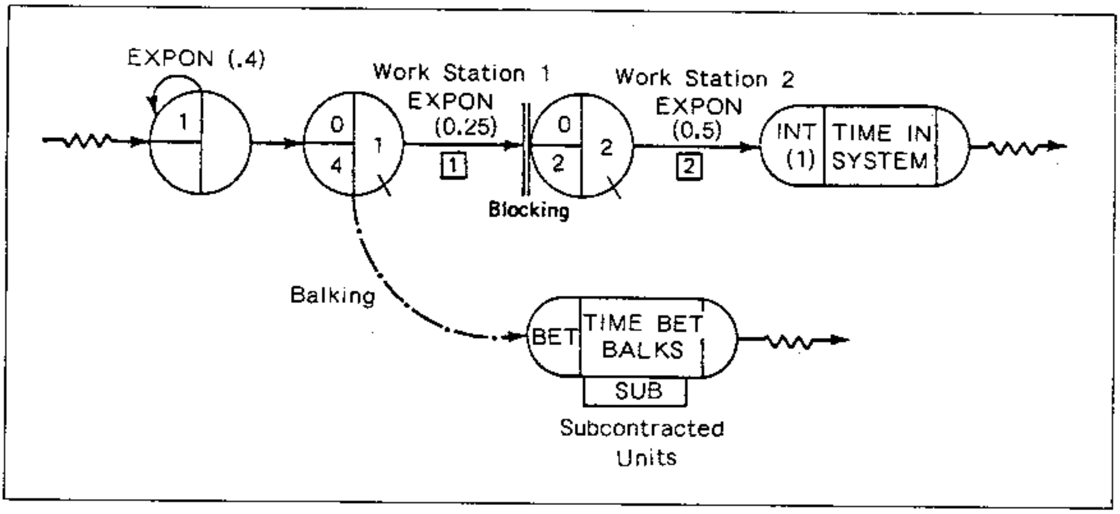

Fig. 5 SLAM II Network model of maintenance facility.

The KBSM system output for the example is represented in Fig. 6.

6. Conclusion

The KBSM system overcomes the traditional simulation modeling technique disadvantages and represents a natural next step toward automated software support for simulation and model building in a known simulation language. First, the structured interactive dialog provides an adequate user-system interface for the acquisition of a model specification. Second, the knowledge-based approach provides a framework for specification acquisit-

```
GEN, KIM, WORK STATION, 04/28/89, 1;
LIMIT, 2, 1, 50;
NETWORK;
        CREATE, EXPON(0,4), 1 ;
        QUEUE(1), 0, 4, BALK(SUB) ;
        ACT/1, EXPON(0.25) ;
        QUEUE(2), 0, 2, BLOCK ;
        ACT/2, EXPON(0.5) ;
        COLCT, INT(1), TIME IN SYSTEM,
        20/0/0.25 ;
        TERM ;
SUB     COLCT, BET, TIME BET. BALKS;
        TERM;
        END
INIT, 0, 300;
FIN;
```

Fig. 6 KBSM system output.

ion and model construction that eliminates the need for simulation language expertise on the part of the user. The user only required to be able to describe the structure and component of the system to be modeled and provide model parameter value. Finally the internal model specification representation, in conjunction with specification acquisition and model construction rules, provides software support for managing specification data and the development of a complete discrete simulation model.

The system developed represents one possible AI-based approach to computer assisted simulation modeling. Thus the KBSM system utilizes the knowledge representation methods and inference mechanism of an established expert system building tool to implement a knowledge-based approch to model specification and automatic model construction. For the development of the KBSM system capability and utility, future works are as follows.

1) Extension of rule base

2) Enhancement of specification capability

3) Extension of model construction capability

References

1. Shannon, R.E., systems simulation; the Art and the Science. Prentice-Hall, Inc., Englewood Cliffs, 1975.
2. Haddock, J., "An Expert system Framework Based on Simulation Generator", Simulation, 48, no.2, 45-53, 1987.
3. Arons, H.D.S., "Expert system in the Simulation Domain "Mathematics and Computers in simulation, North Holland pub. Co., Amsterdam, 25:1, 10-16, 1983.
4. Gains, B.D. and M.L.G. Shaw, "Expert systems and Simulation"., Proceedings of the 1985 Conference on artificial Intelligence, graphics, and simulation, SCS, San Diego, CA, 95-101, Jan. 1985.
5. Shannon, R.E., et al., "Expert system and Simulation", Simulation, 44:6, 275-284, June. 1985.
6. Shaw, M.L.G. and B.R. Gains, "A Framework for knowledge-based Systems Unifying Expert Systems and Simulation", Proceedings of the Conference on Intelligent Simulation Environments, SCS Simulation Series, 17:1, San Diego, CA, 38-43, Jan. 1986.

7. Ginsberg, A.S., et al., "Programming by Questionnaire", Rand Memorandom RM-4460-PR The Rand Corporation, Santa Monica, CA, April. 1965.
8. Stephen, C.M., "Simulation Program Generators". Simulation 23:6, 181-189, Dec. 1974.
9. George, E.H., "Natural Language Inputs to a Simulation Programming Systems", Technical Report NPS-55 HD 72101A Naval Postgraduate School, Monterey, CA, Oct. 1972.
10. Klahr, P., et al., "Rule-Oriented Simulation". Proceedings of the International Conference on Cybernetics and society IEEE. Boston, MA, 350-354, Oct. 1980.
11. Baskaran, V. and Reddy, Y.V., "An Introspective Environment for Knowledge Based Simulation". Proceedings of the 1984 Winter Simulation Conference, SCS, Dallas, TX, 645-651, Nov. 1984.
12. Adelsberger, H.H., et al., "Rule Based Object Oriented Simulation Systems". Proceedings of the Conference on Intelligent Simulation Environments, SCS Simulation Series, 17:1, San Diego, CA, 107-112, Jan, 1986.
13. Murray, K.J. and S. sheppard., "Knowledge based Simulation Model Specification", Simulation, 50:3, 112-119, Mar. 1988.
14. Banks, J. and J.S. Carson, II, Discrete Event System Simulation, Prentice Hall, Inc., Englewood Cliffs, NJ, 1984.
15. Pritsker, A.A.B., Introduction to Simulation and SLAM II., 3rd Ed., John Wiley and Sons, New York, NY, 1986.
16. Brownston, L., et al., Programming in Expert Systems In OPS5, Addison Wesley Co, 1985.