Construction of a Knowledge Based Hybrid Simulation Environment

Myoung-Hee Kim*

Abstract

A knowledge based hybrid simulation environment has been established which supports the simulation procedure from the modeling step to the experimentation step by providing various tools, query, and advisory.

The knowledge base in the established environment uses the If-Then Rule to transfer the fact input from the user to the act for the execution of the simulation process. Simulation non-experts are the user target and non-specific area is the application target.

I. Introduction

As it is shown in terms of cognitive simulation, simulation based reasoning, the relationship between artificial intelligence and simulation can be classified as artificial intelligence applied to simulation[1] and simulation applied to artificial intelligence[2].

In the former case, the attempt of constructing expert systems in the simulation field has been active[3], but gradually the function of knowledge processing for simulation is being observed carefully [4] [5].

On the other hand, the tendency of the study on the simulation environment can be summarized into advanced modeling environment, advanced experimentation environment, mixed (hybrid) environment, and comprehensive environment[6]. This can be re-expressed by providing various tools, the reinforcement of the ability of query and advisory and intergrated simulation environment.

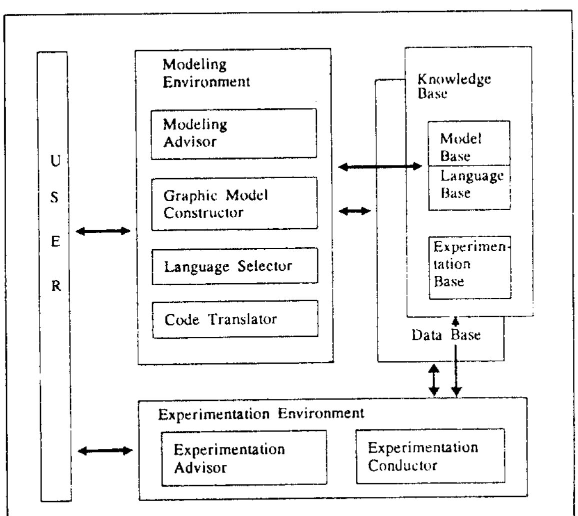
But, most of previous developed simulation environment provide either intergrated support for specific application area or partial support for general application area.

However, for the beginners of simulation, an environment providing intergrated support for non-

* Dept. of Computer Science, Ewha Womans University

specific application area and also various tools, methods and sufficient knowledge is needed. From this point of view, in this study, it is attempt to construct a simulation environment for non-expert and for general application area, and so the knowledge-based hybrid simulation environment which supports the simulation worksteps from the modeling to the experimentation has been designed and constructed using the user-interface known as query and advisory system and the knowledge-base.

The constructed simulation environment is divided into the modeling environment and the experimentation environment. Each has its own knowledge base for processing tasks. Figure 1 showes architectural overview. The whole process



〈Figure 1〉 Architectural Overview

may be performed by query and advisory. However, it can also be made through menu form and icon and make various contents of interaction possible by dividing the screen into windows.

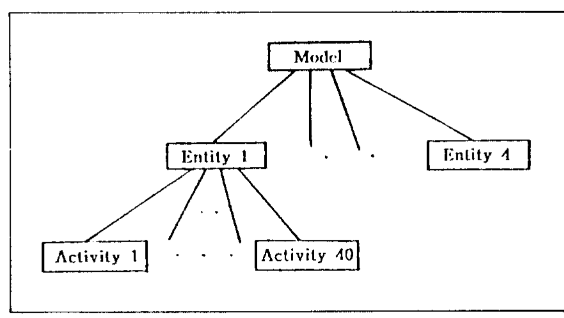
Ⅱ. Modeling Environment

The modeling environment receives the content of understanding of the target system as input from users who have no special knowledge or experience and takes charge of the task process from the construction of graphic model to the computer programming.

The process of model construction is divided into two steps, conceptual model specification and graphical model generation and is carried out by two task modules, modeling advisor and graphic model constructor.

First of all, the understanding of users about the simulation target system is specified by the response to the query provided by modeling advisor.

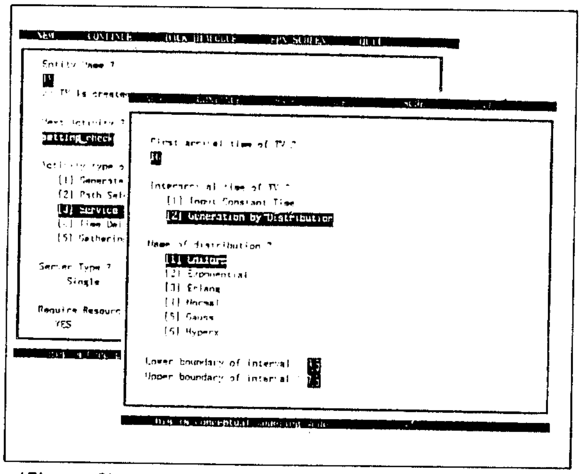
Query requires the contents of a model in a hierarchical structure as in the 〈figure 2〉.



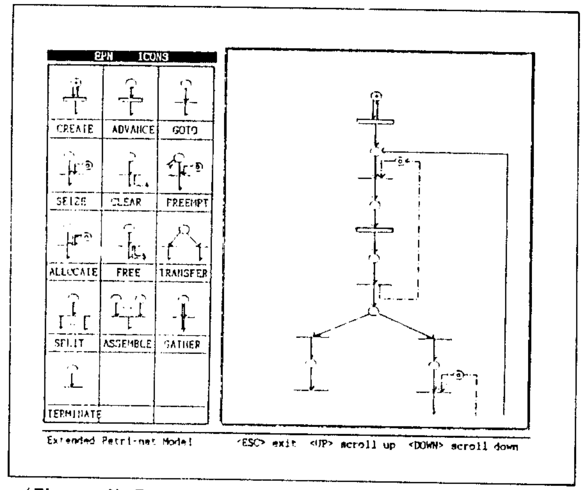
〈Figure 2〉 Hierarchical Structure of Model

For each entity, all the activities from creation to reduction are shown in order. The number of activities allowed for one entity is limited to 40. The number of types of activities are five as shown in 〈figure 3〉.

The example system implemented is inspection and adjustment stations on a TV production line where vertical control setting on the TV set is tested in the final stage of their production. If the setting is proper, then pass otherwise route to an adjustment station. After adjustment it is sended back to the inspection system. Uniform distribution is adopted for interarrival time of TV sets,



〈Figure 3〉 Example : Conceptual Model Construction



〈Figure 4〉 Example : Graphical Model Construction

inspection time, and adjustment time. Average adjustment rate is 0.15.

As a response to a query, the names and the types of entity and activities are defined or selected and the values of necessary attributes for model construction are input. The content of continuous queries are determined by the knowledge base and the response of the users. The names of entity and activity input by users are used as they are when the appropriate entity and activity are designated in the continuous query.

Next, in the step of graphical model construction, the user's understanding of target system, that is the conceptual model specified through query, is transformed and proposed into graphical model. By showing the activities of entities and the connection between them, the precise contents of the system can be confirmed to and modified by the users. The graphic tool used at this point is the EPN (Extended Petri Net)[7] which can express the additional functions of inhibitor arc, probablilistic path selection, and time delay in Petri-Net.

At this step as in 〈figure 4〉, icons of graphic elements of Extended Petri-Net are shown continuously at one side and the graphic model of the target system composed of these elements are shown at the other side. For modification, retrial has to be made from the conceptual model specification because the construction of graphic model is made automatically.

The constructed graphic model is transformed into the simulation program by Language Selector and Code Translator.

The simulation languages used in programming are SMPL and GPSS-Fortran. SMPL has simple functions and brief form of expression but GPSS-Fortran has the possibility of extending the function arbitraily in case of users accustomed to Fortran. Language selector analyzes the elements of predefined system and determines the programming languages appropriate for the systems.

Code translator uses the instructions of each language corresponding to each EPN graphical elements and the input data from the conceptual model specification step to generate programs. The parameters needed at this point are known to the users by query and the corresponding values are made to be keyed-in.

Ⅲ. Experimentation Environment

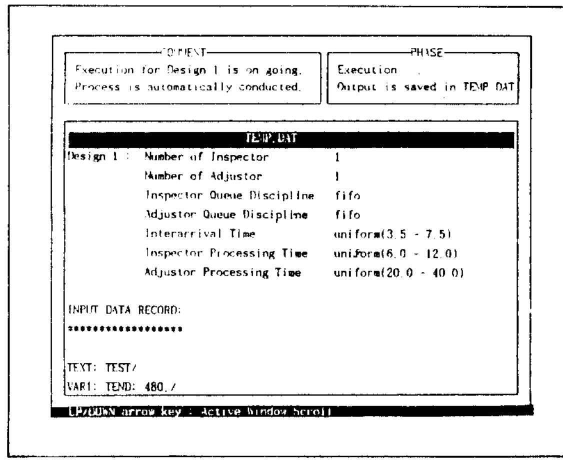
Experimentation environment supports the experimental design[8] and the experimentation. Each task process is carried out by the two task modules, experimentation advisor and experimentation conductor.

At first, for the experimental design, the selected control variable list from the whole variable list resulted from the previous task execution is presented. Then the value range of the selected control variables are input. Lastly the start/end condition of the simulation and the number of replication are determined.

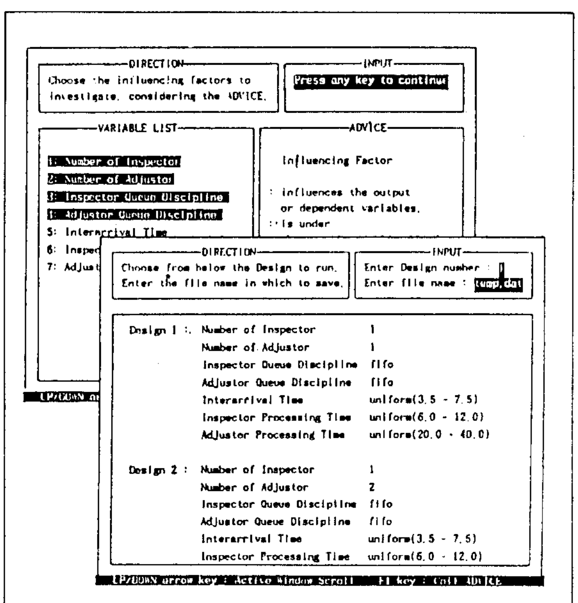
During the whole process, through four windows users get the direction of the work, necessory advice, menu for selecting the input details and the requests of input data which have to be keyed-in directly(see 〈figure 5〉).

Then, experimentation advisor displays the possible experimentations through factorial design and requires to select among these and to input the name of output file for it.

Finally, experimentation conductor presents in order the contents of the selected design, the list of the modified simulation program(see 〈figure 6〉), and the execution result.



〈Figure 6〉 Example : Experimentation



〈Figure 5〉 Example : Experimental Design

Ⅳ. Knowledge Base

The knowledge included in the knowledge base are divided into the knowledge about the execution order and the knowledge about the method and tools[9]. That is, the knowledge base is composed of the detailed knowledge about the execution process of model construction and experimentation, and the knowledge about functions. These functions include the required input parameters, EPN graphical model components, simulation language code corresponding to each type of activity, the instruction set of GPSS-Fortran[10] and SMPL[11], determination of control variables, value range, number of experimentation and the management of start/end conditions.

Because of these characteristics, the If-Then Rule is more functional in this aspect of compatibility with task modules and knowledge presentation and is used to construct the knowledge base.

The comparison between the result of the TMYCIN tried before this is in figure 7. The If-Then production rule takes the following form.

	If-Then Rule	TMYCIN
compatibility with task modules	compatible	less compatible
knowledge presentation	flexible	less flexible
distinction between knowledge and knowledge interpreter	less explicit	explicit
convenience of use	fair	good

〈Figure 7〉 Comparison With "If-Then Rule" and "TMYCIN"

The action part, as shown in 〈figure 8〉 is consisted of the dialogue management and the internal knowledge management. The dialogue management handles the knowledge related to the methods and tools and the internal knowledge management handles the knowledge related to the execution process.

These structures are all applied to each knowledge base for the model construction, for the program generation, and for the experimentation.

The six task modules for each simulation work steps mentioned above carry out the task with the knowledge obtained from the knowledge base and the interaction with the user. This is made through user interface. And the input data from every user and the generated data throughout the whole process are recorded in the database.

Ⅴ. Conclusion

A simulation environment using knowledge base which makes the performance of the continuous procedures for several task steps of a simulation

IF	THEN	
	DIALOGUE MANAG.	INTERNAL KNOWLEDGE MANAG.
· Start new model	−Set QNUM to QNEW	−Initialize entity & activity list −Initialize CUR ACTIVITY & PAR ACTIVITY
· QNUM is QNEW	−ask entity name −get entity name by user's input −set QNUM to QNEXT	−add activity of GENERATE type to activity list −set CUR ACTIVITY
· QNUM is QNEXTNULL and PAR ACTIVITY is not exist	−set QNUM to QTERM	−add activity of TERM type to activity list

〈Figure 8〉 Example : Rules for Model Construction

and the various types of interaction with users for task execution possible as well as providing necessary tools, methods, and knowledge for each task step has been constructed.

The construction task has been done by Borland C++ 3.0 and IBM PC 386 DX, VGA has been chosen as a target implementation system. The required memory size is about 300KB.

The functional features of the implemented system are the provision of graphic tool for modeling, automatic code generation, interactive experimental design and repetition of simulation execution.

On the other hand, there are limitations at the specification range of target system, the types of simulation language, and the supported experimental design methods.

Therefore, in the future studies, the range of system specification , tool, methods, and knowledge should be extended. Additionally, the function of validation should be included.

References

[1] Widman L.E. et al(Ed.), Artificial Intelligence, Similation & Modeling, Wiley, New York, 1989

[2] Elzas M.S. et al(Ed.), Modeling and Simulation Methodology in the Artificial Intelligence Era, North-Holland, 1986

[3] O'keefe, R.M., "Simulation and Expert Systems-A Taxanomy and some Exampls", SIMULATION vol.46 no.1, 1986, pp.10-16

[4] Norman, R.N., Knowledge based Simulation, Springer Verlag, Berlin, 1991

[5] Murray, K.J., "Knowledge-based simulation model specification", SIMULATION, vol.48 no.3, 1988, pp.112-119

[6] Ören, T.I., "Simulation Environments: Challenges for Advancement", Proc. of BICSS '92, vol.1, 1992, pp.8-12

[7] 김명희, " 가시적 모델링을 위한 GFN의 설계에 관한 연구", 한국생활과학 연구원 논총, vol.48, 1991, pp.11-20

[8] Wild, R.H., et al. "An Experimental Design Strategy for Designing Robust Systems using Discrete-event Simulation"

[9] Ören, T.I., "Knowledge bases for an advanced simulation environment", Proc. of Conf. on Intelligent Simulation Environment, 1986, pp. 16-22

[10] Schmidt, B., The Simulator GPSS-Fortran version 3, Springer Verlag, Berlin, 1987

[11] MacDougal, M.H., Simulating Computer Systems Techniques and Tools, MIT press, 1986

● 저자소개 ●

김명희(金明姬)

1972.2 이화여대 졸. 사회학전공(문학사)

1979.8 서울대학교 대학원졸, 전산학전공(이학석사)

1986.2 독일 Göttingen 대학교 대학원졸 전산학 전공(이학박사)

1986.3-1987.2 서울대. 이화여대 강사

1987년 3월 이후-현재 이화여대 전산과 교수