Proceedings of the 1991 Winter Simulation Conference
Barry L. Nelson, W. David Kelton, Gordon M. Clark (eds.)

SUPER SIMULATION SHELL FOR PLANTS

Zen-ichirou Muro
Industrial Engineering Section Planning Department
Kawasaki Steel Corporation Chiba Works
1 Kawasaki-cho Chiba, Chiba 260 JAPAN

Norihiko Yanagihara
Industrial Enigneering Section Planning Department
Kawasaki Steel Corporation Mizushima Works
1 Kawasaki-dori Mizushima, Kurashiki 712 JAPAN

Hideo Fujimoto
Systems Laboratory
Systems Planning & Data Processing Department
Kawasaki Steel Corporation
Hibiya Kokusai Building 2-3, Uchisaiwai-cho
2-chome, Chiyodaku, Tokyo 100 JAPAN

ABSTRACT

Super Simulation Shell (SSS) is a simulation system based on KEE™ and SimKit™. The purpose of SSS is to provide steel plant engineers with an environment in which they can easily build computer simulation models. These engineer have difficulties using existing systems to solve problems such as the determination of a transporter's potential, the number of machines, operation control, facility layout and estimation of in-process inventory and lead time.

SSS is based on technologies of object oriented programming, some of them are provided by KEE and SimKit, and others come from the concepts and components in steel plants. SSS also includes functions to easily input attributes on screen, to make animation for model validation, to suggest control strategies and to collect output data for analysis.

This paper describes the basic concept and examples of SSS.

KEE™ and SimKit™ are trademarks of IntelliCorp, Inc.

1 INTRODUCTION

In steel works, the transportation cost is very expensive. The weight of products is about 20 metric tons. Time interval of production is about 2 minutes. A fully automated transportation system is expected. This means that the system is more complex, and more sophisticated.

To design such transportation systems, we used to do discrete event simulation models in GPSS. But GPSS is difficult to use for those who are not specialists in simulation.

The purpose of *Super Simulation Shell* (SSS) is to provide steel plant engineers with an environment in which they can easily build computer simulation models. To achieve the purpose of SSS, SSS must satisfy the following problems.

1) The system must be interactive and have animation facilities.
2) The domain and tasks can be limited to transportation problems.
3) The system must implement our experiments in GPSS.
4) The system must process "If-Then" rules which represented empirical knowledge.

To meet these requirements, we have implemented SSS on SimKit. The SimKit system is an integrated set of general purpose simulation and modeling tools built in and on the KEE software development system. It takes full advantage of the expressive representation, powefull reasoning, and user-friendly interface tools that KEE provides.

This paper describes the basic concept and examples of SSS.

2 OUTLINE OF SSS

2.1 Task and Domain Description

SSS was designed and implemented by six members of Kawasaki Steel Corporation.

The problems can be summarized as follows:

1) Capacity planning of transportation systems.

We must develop transportation systems in a factory or among factories. To do so, we use simulation models. The main purpose of simulation is to determine the capacity of these transportation systems. So, the facilities which we must prepare are all transportation machines in steel works. For example, in a factory, there are overhead cranes, AGVs, conveyors, transporters, stock yards and lines. Among factories, there are diesel locomotives, lift cars, torpedo cars (which carry molten iron), trucks and trailers. For these facilities, which have different functions, we prepared models.

2) Rule generation to operate transportation systems.

Today, the operation control of transportation systems is mainly rule-based one. The quality of the operation rules affects the performance of transportaion systems. Therefore, it is important for us to generate and estimate the operation rules before deploying transportation systems.

We prepared an environment in which a user can easily treat the rules with KEE's knowledge base system.

3) Optimization of inventory and line balancing.

The major facilities in the integrated steelworks have different process cycles and different performance for each product. If we improve the process cycles, the production capacity of the facility will increase. And, if the process cycles increases, the other products will wait and the inventory will increase.

We implemented an interactive interface and graphical output facilities to solve these process scheduling problems.

4) Reduction of lead time.

As described above, it is important to consider the problems of the lead time.

To analyze this, we implemented functions for collecting various output data.

2.2 The Features of SSS

The key features of SSS emerged from KEE and SimKit. We expanded the original features of KEE and SimKit. In the following, we show the features of SSS.

1) Object Oriented Programming

Object oriented programming is suitable for developing simulation models with the structure of target domain and objects behaviors. And, hierarchical representation of objects with inheritance makes the model implementation easy.

SSS provides us with all the object hierarchy in a steel plant. Table 1 presents the list of objects.

2) Interactive Modeling

The SimKit environment is wholly interactive. An interactive modeling environment simplifies "What-If" analysis. In addition, we developed specific input windows with four types of input for various attributes of objects.

3) Rule based Control

Rule based control systems allow heuristic knowledge to be represented as "If-Then" rules. They are useful for implementing operational control of transportation systems.

4) Easy Animation

It takes a long time to make animations. However, using the mechanisms of active values or daemons of KEE, we developed an special-purpose and easy-to-use animation functions.

5) LISP Programming

Symbolic LISP programming enriches the descriptive power of simulation models more than Fortran programming does. This is more friendly for programmers.

Knowledge based simulations are more understandable, interactive, accessible, and extensible than simulations built using "conventional" technologies.

Table 1: List of Objects

classification	name of objects	function	classification	name of objects	function
Basic Parts	SOURCE	Item creation in the model	Transporters	TRANSFER	General carrier
	SERVER	Item processor		CRANE	Crane
	QUEUE	Temporary item holder		AGV	Vehicle
	SINK	Item terminator		DRIVER	Tractor
	ITEM	Parts of a model to be processed		CONTAINER	Container
Facilities	LINE	Multiple processor		FRONTAGE	Frontage
	ASSEMBLE	Assemble facilities		CONVEYOR	Conveyer
	BATCH	Batch processing facilities	Others	MOTOR	Conveyer controller
	FURNACE	Furnace		OPERATOR	Operator
	SKID	Skid		Control Tower	Controller
	YARD	Store house		ST	Station

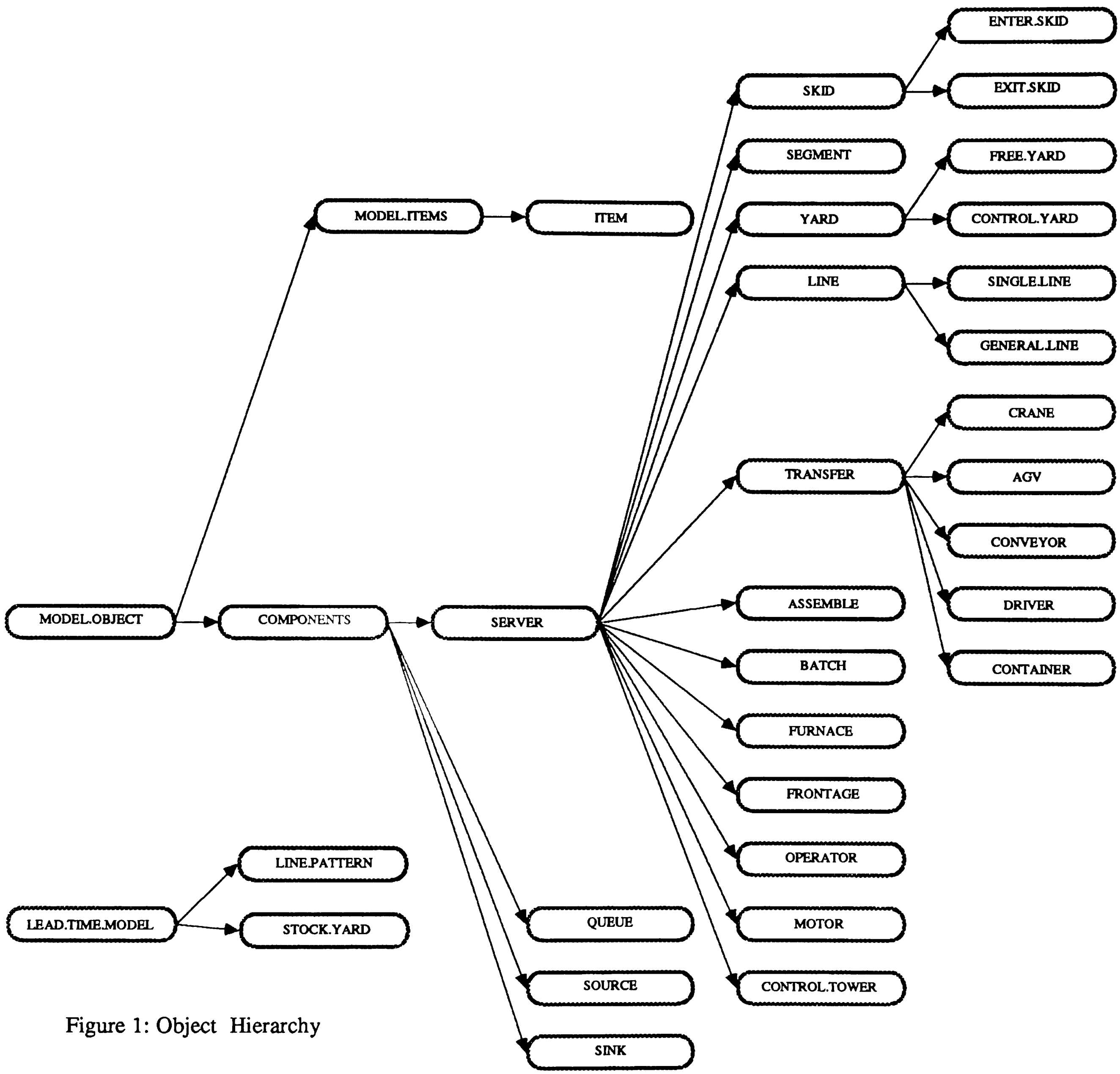

Figure 1: Object Hierarchy

3 INSIDE THE SYSTEM

3.1 Object Hierarchy

Figure 1 presents the class/subclass hierarchy defined in this system, which is classified by function. The objects for calculating lead time are independent of the other objects.

3.2 Rule Description

When the event that required decision making or optimization will occur, it is useful to use "If-Then" rules that represent experts empirical knowledge. An example is when an AGV is carring an item and it must decide which path is better to use. If-Then rules are useful, because of their understandability. SSS provides users with interfacing to bridge KEE's powerfull rule processing capability.

Figure 2 presents a sample of "If-Then" rules. This function depends on KEE, which can do more powerful reasoning.

```
1) (Idle.Line.Rule
     (IF (a Downstream of ?Component is ?Down)
           (Cant.Find (the Contents of ?Down is ?X)
        Then
        (Change.to
           (the Preferred.Downstream of ?Component is ?Down))))
2) (Shorter.Line.Rule
     (IF (a Downstream of ?Component is ?D1)
           (a Downstream of ?Component is ?D2)
           (LISP (Not (EQ ?D1 ?D2)))
           (LISP (<= (Length (Get.Values ?D1 'Contents))
                          (Length (Get.Values ?D2 'Contents))))
        Then
        (Change.to
           (the Preferred.Downstream of ?Component is ?D1))))
```

1) A rule taking priority of putting items into empty lines
2) A rule to put items into shortest waiting line

Figure 2: The example rules

3.3 Interface

SSS has four kinds of interactive windows with pop up menus. The first is a model type window that is used to input general attributes (e.g., capacity, position, etc). The second is a cycle-time window that is used to input an attribute about time (e.g., processing time, item generation time, carrying time and speed, etc). The third is a data assignment window that supports functions to assign data to an item, when an event occurs. For example, it allows assigning a processing time to an item according to the kind of product. The fourth is a data collecting window that supports functions to collect various analysis data.

These windows are controlled by daemons. Figure 3 presents an example of the cycle-time window.

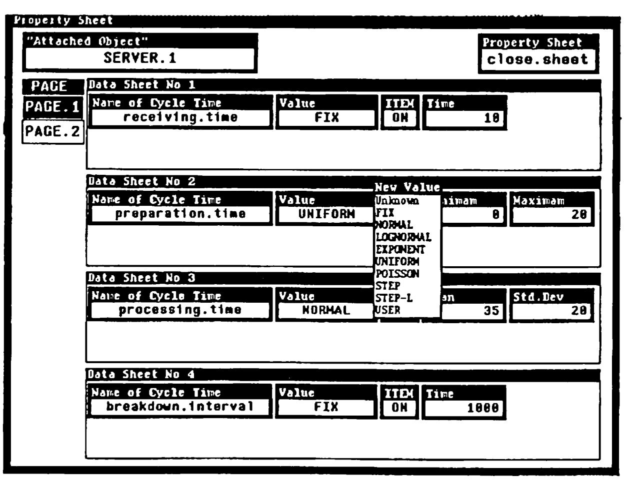

Figure 3: The cycle-time window

4 EXAMPLE

Figure 4 is an example of a coil transportation system which is placed at the end of hot strip mill.

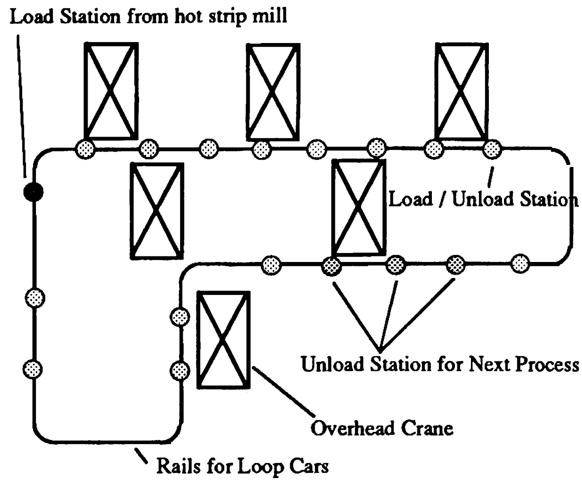

Figure 4: Layout of Coil Stock Yard

This transportation system includes six overhead cranes, some loop cars, and coils transported. Loop cars carry the hot coils which came out from the hot strip mill to the coil stock yards. They also carry the cooled coils in the coil stock yards to the next process station. The job of loading and unloading coils from the loop cars is done by overhead cranes.

The purposes of simulation are to determine the number of overhead cranes and loop cars. The major feature of this simulation is that the overhead cranes have autonomous knowledge that determines the timing of loading and unloading coils from the loop cars. This knowledge was elicited through the simulation.

5 CONCLUSIONS

SSS offers a number of advantages for improving the simulation modeling environment. The first advantage is rapid prototyping. Simulation development in the domain of transportation systems is performed especially rapidly. The interactive, accessible, extensible and visible environment helps this model building process.

The second advantage is that this system makes the development of simulation models with rules much easier. Rules can be generated and estimated by the resulting simulation system.

The third advantage is that SSS contributes immensely to lowering costs in a transportation system by labor savings and optimal design on investment. Today, in Kawasaki Steel Corporation, SSS is installed in three major works (Chiba works, Mizushima works and Chita works).

And SSS is being refined by using it in many aspects of simulation problems.

The only disadvantage is that it takes a long time to run models. Improving execution performance is a future goal.

ACKNOWLEDGMENT

KEE and SimKit's programming environment helped us to develop this simulation system. The authors and Kawasaki Steel Corp. would like to express their gratitude to IntelliCorp, Inc. , the developer of KEE and SimKit. And they must say thanks to thier colleagues, Mr. Ishida, Mr. Nakagawa and Mr. Fujiwara. Their abilities and efforts have contributed greatly to the success of this system.

REFERENCES

Stelzner, J., Dynis, J., Cummins, F. 1989. The SimKit™ System: Knowledge-Based Simulation and Modeling Tools in KEE. In Proceedings of the 1989 Winter Simulation Conference, eds. E.A. MacNair, K.J. Musselman, and P. Heidelberger, 232-234.

Kaminski, J., Cosic, C., Strohm, G., Kepner, J., Bycura, J. 1989. Knowledge-Based Modeling and Simulation Components. In Proceedings of the 1989 Winter Simulation Conference, eds. E.A. MacNair, K.J. Musselman, and P. Heidelberger, 222-231.

Rothenberg, J. 1989. Tutorial: Artificial Intelligence and Simulation. In Proceedings of the 1989 Winter Simulation Conference, eds. E.A. MacNair, K.J. Musselman, and P. Heidelberger, 33-39.

AUTHORS' BIOGRAPHIES

ZEN-ICHIROU MURO is a staff assistant manager at Kawasaki Steel Corporation in Chiba works, JAPAN. He received a B.Eng. in Information science from Tsukuba University in 1981. Since joining Kawasaki Steel Corporation in 1981, he has been involved with research and development of production control systems in steel works using operations research methods. His current research interest is simulation coupled with artificial intelligence. He is a member of ORSJ and AISJ and IPSJ.

NORIHIKO YANAGIHARA is a staff assistent manager at Kawasaki Steel Corporation in Mizushima works, JAPAN. He received a B.Eng. and M.Eng. in Mechanical Engineering from Waseda University in 1980 and 1982, respectively. Since joining Kawasaki Steel Corporation in 1982, he has been involved with research and development of production control systems in steel works using operations research methods. His current research interest is applications of operations research methods. He is a member of ORSJ.

HIDEO FUJIMOTO is a staff researcher at Kawasaki Steel Corporation in head office, JAPAN. He received a B.Eng. in Industrial Engineering from Waseda University in 1980. Since joining Kawasaki Steel Corporation in 1980, he has been involved with research and development of production control systems in steel works using operations research methods. His current research interest is applications of artificial intelligence. He is a member of ORSJ.