Fox M.S., N. Sathi, V. Baskaran, J. Bouer, (1986), "Simulation Craft: An Expert System for Discrete Event Simulation", Proceedings of 1986 Eastern Simulation Conferences, March 10-12, 1986, Norfolk, VA.

Simulation Craft™ – an expert system for discrete event simulation

Mark Fox, Neena Sathi, V. Baskaran, Jack Bouer

Carnegie Group Inc

650 Commerce Court

Station Square

Pittsburgh, Pennsylvania 15219

ABSTRACT

Simulation Craft™ is an artificial-intelligence based simulation system that increases the productivity of simulation experts and enables non-experts to develop simulation models. Simulation Craft incorporates the knowledge of simulation experts to create three expert systems; one each for model creation, model execution, and model analysis. These expert systems perform tasks, such as helping the user specify the model, designing the simulation experiment, and analyzing the simulation results, which would otherwise have to be performed by simulation experts. Thus Simulation Craft reduces the time as well as the programming and simulation experience required to create, execute, and analyze simulation models.

INTRODUCTION

Simulation is a valuable tool for predicting the behavior of systems under various conditions. Successful discrete event simualations prepared with conventional simulation languages (e.g., GPSS, SIMSCRIPT, SLAM, SIMULA, and SIMAN) have been used in the management of job-shops, inventory systems, message switching systems, and transporation, distribution and communication networks. [1]

Unfortunately preparing simulations with conventional simulation languages is a long and difficult process which requires speciailized knowledge. To develop a simulation model with a conventional language, a user must have experience programming in that language. The user must also be very familiar with statistical concepts to determine the simulation tests to run and the results from the simulation. [2] Because of these requirements, most simulations are prepared by experts not by the person who has the problem that requires simulation.

This creates communication difficulties. The person with the problem must specify it to the expert in sufficient detail so that the expert can prepare the simulation. After the simulation is completed, the simulation expert must describe the results to the lay person in terms that he/she can understand.

Much research has been performed in applying artificial-intelligence techniques to the task of simulation development. [3, 4, 5, 6, 7] Simulation Craft, which is being developed by Carnegie Group, is a simulation package that employs artificial-intelligence techniques to overcome many of the deficiencies of conventional simulation languages.

Simulation Craft increases the productivity of simulation experts and enables non-experts to develop simulation models. These goals are achieved by embedding simulation expertise in the system thus reducing the time as well as the programming and simulation experience required to construct, execute and analyze simulation models.

With Simulation Craft, the user constructs simulation models in terms of well known domain objects. For example, factory floor simulation objects include machines, work-centers, and orders etc.. Simulation Craft visually monitors the simulation as it runs by showing the movement of simulation objects on the screen. After the run is completed, Simulation Craft conducts an automated analysis of simulation results to provide concise answers to user questions. All of this is accomplished in a fraction of the time required to perform simulations using conventional simulation systems.

COMPARISON OF SIMULATION CRAFT WITH CONVENTIONAL SIMULATION SYSTEMS

Conventional discrete simulation systems are divided into (a) simulation languages (e.g., GPSS) and (b) specialized systems which are targeted for specific application areas (e.g., MAP/1).

While they do provide a flexible simulation facility, the simulation languages are difficult to use. Users must learn to program in the language before they can construct simulations. They also have to translate their problems into the language's concepts because the languages use abstract objects, such as servers and gates, not those suited for a particuliar domain. For these reasons model creation requires a great deal of time.

The specialized simulation systems do not require as much programming skill as the simulation languages, but they are limited in the types of scenarios they can simulate and the output they provide. The specialized languages only include a limited set of objects and activities. Users are not able to construct accurate simulations if their situation does not conform to these objects and activities. For example, many of the specialized languages for manufacturing could not model machines such as autoclaves which have complicated loading algorithms. Furthermore, the user is often limited to only a standard set of reports which might not contain the information needed.

Another weakness of the simulation languages and the specialized systems is that they burden the user with voluminous data on transactions and servers which the user must sift through to find answers to questions. In addition, the systems do not provide guidance on additional scenarios the user should simulate. For example if an initial simulation run indicates that a communication network is processing certain messages slowly, the user has to go through the simulation output to identify possible network bottlenecks. After identifying the bottlenecks the user has to determine a possible solution and modify the simulation code to incorporate a test for this solution.

Simulation Craft overcomes the deficiencies of conventional systems by employing artificial intelligence techniques. These techniques enable Simulation Craft to have an easy-to-use model creation interface which translates graphic input into complicated models. In Simulation Craft, models are created just by placing graphic objects (e.g., warehouses for a distribution problem) on a display. When a graphic object is placed, the model is elaborated to include the object, its attributes and its relations with other objects. This elaboration is performed using knowledge stored in a library for the particuliar application area. The user does not have to translate his situation into Simulation Craft terms, because the objects and the rules for their placement were created specifically for the application area. However, the user is not restricted to the objects that the library provides. The system's flexible knowledge representation capability allows the user to define his own objects and simulation events.

Simulation Craft's execution facility allows the user to develop an intimate understanding of the model and explore what-if scenarios. Simulation Craft monitors the running simulation visually which makes the model much more transparent to the user. The user can see situations, such as a bottleneck, developing on the screen which creates greater understanding and trust in the model. In addition, with Simulation Craft's interactive scenario management, the user can stop and alter the simulation at any point and run both the original and altered scenario or the altered scenario alone.

Simulation Craft performs the analysis of the simulation results using goals and rules that incorporate the knowledge of simulation experts. This automated analysis frees the user from the time-consuming task of analyzing the voluminous simulation data and determining how to alter the simulation model.

SYSTEM ARCHITECTURE

Simulation Craft is divided into three modules:

- Expert Model Builder - constructs the simulation model.
- Expert Model Executor - provides guidance on designing the simulation experiment (e.g., structuring the runs) and then runs the simulation model
- Expert Model Analyzer - analyzes the data produced by the runs.

Simulation Craft is designed to be application-independent. It is tailored for specific application areas by using libraries of simulation objects, events, rules and standard reports. The libraries for the application area also contain routines for application-specific problems. For example the factory floor library contains scheduling routines for routing orders through the factory floor.

Carnegie Group will provide the libraries for many application areas. In addition, sophisticated users can create their own libraries or modify an existing library.

Simulation Craft is implemented in Common Lisp using *Knowledge Craft*™ Carnegie Group Inc's knowledge engineering system. All of the data on simulation objects, activities and events are stored in schemata using CRL™, the frame-based language in Knowledge Craft. Each schema has slots and values to store attributive and relational information about an entity such as a machine's capacity. Relations link schemata to one another, and information in one schema can be tranferred to another. [8]

The remainder of this article discusses each of the Simulation Craft modules.

EXPERT MODEL BUILDER

In Simulation Craft, simulation models are constructed graphically. The user is provided with a simulation display and a palette and sub-palettes of standard simulation objects for the application area.

Layout of Resources

The user constructs the layout of resources (e.g., machines, nodes in a communication network) in the simulation by selecting objects from the palette and placing them on the simulation display. When the object is placed in the display, the schema for an instance of that object is automatically added to the simulation model. The factory floor palette, which is based on information in the factory floor library, is shown in Figure 1. This palette contains icons for "machines", "order generators", and "parts".

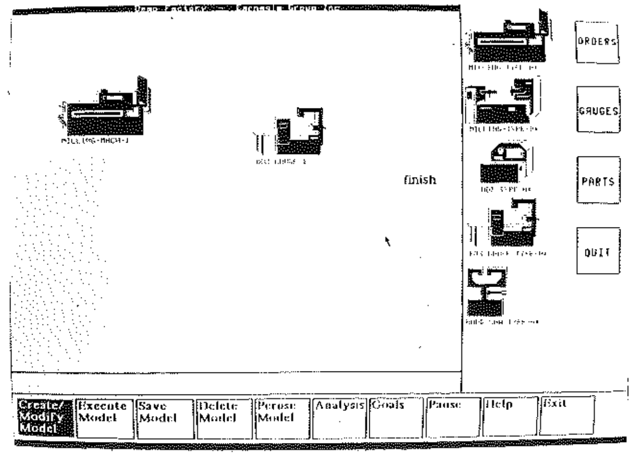

Figure 1: Palette

The standard palatte objects are taken from the application-area library, and users can define their own objects to supplement the standard choices. The user can also add an object that is not in the library by using the palette's creation feature.

In cases where there are a large number of palette objects, users can locate objects by using Find/Filter™, a keyword and logic programming search facility. In the Find portion, the user specifies the entity type he wants to search (e.g., machines, tools), and the Filter portion then allows the user to specify keyword values to locate the specific entity he wants. Figure 2 shows the keyword specification and output for a query to find machines that cost more than $1,000,000.

Intelligent Action Amplification

Intelligent amplification of users' actions are performed by rules defined in the library. In addition to placing objects in the simulation, users can also group, delete and relate objects.

For example, when entities are grouped on the screen, the simulation

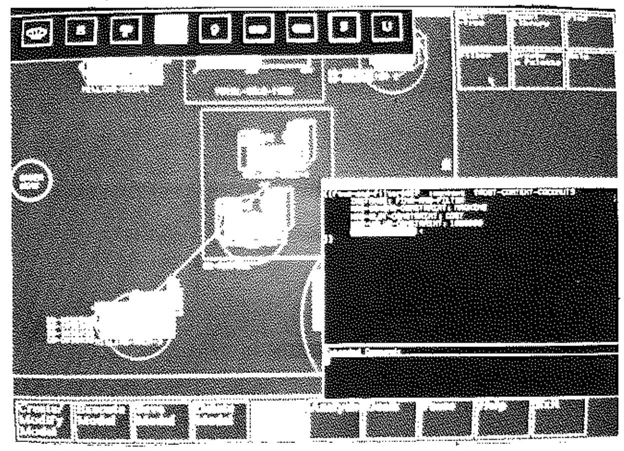

Figure 2a: Find/Filter

model is updated so that it treats the entities as a group. In the factory floor domain, when machines are grouped into a work center, then orders are sent to the work center and then routed to individual machines rather than being sent to machines directly. (This type of grouping is shown in Figure 3.) The exact meaning of grouping is defined in the library since it is application specific.

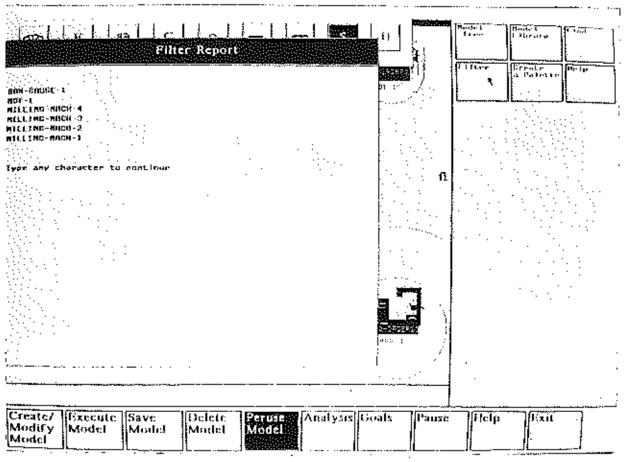

Figure 2b: Find/Filter

The grouping feature is generalized. Any entity or group can be ganized into higher level groups so that ,for example, work centers

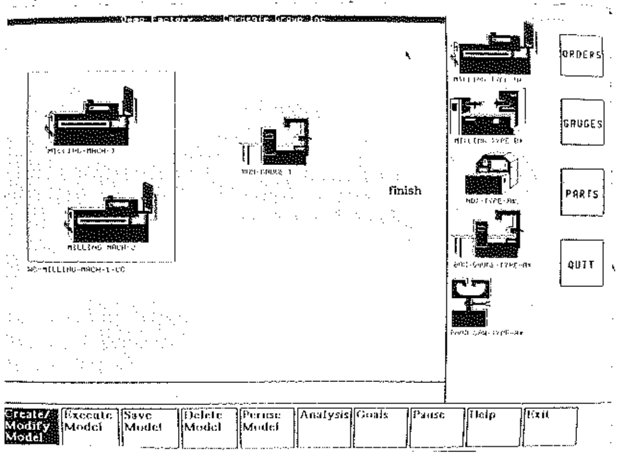

Figure 3: Grouping Machines

can be grouped into cost centers or divisions.

Transaction and Activity Specification

In addition to specifying the resources, the user specifies the transactions (e.g., orders in the factory domain, messages in a communication network) in the simulation and the activities that will process these transactions.

The specification of activities is done graphically. (Figure 4 shows a specification of the factory operations used to produce a part for an order.) The user will specify the resources required for each activity by either pointing to the resources or picking from a list.

Simulation Craft uses an Activity/State model to allow a flexible specification of activities. [9] The sequence of operations can be parallel or sequential and objects can be joined (e.g., assembly operations) or split (e.g., splitting messages in a network). The routing between operations can be deterministic, probablistic or conditional. Similiarly the routing at the resource level is also extremely flexible. The user can specify the queue discipline for each resource, such as machines, by picking from a list (e.g., first-in first-out, lowest or highest value of a user specified parameter).

Knowledge Representation

Simulation Craft's knowledge representation enables it to translate simple graphic input into additions to the model. As was mentioned above, once the object is placed in the display, the schema for an instance of that object is automatically added to the simulation model. The schema will contain the attributes and behavior of the object. An example of a schema for a prototype Deburring Machine could be as

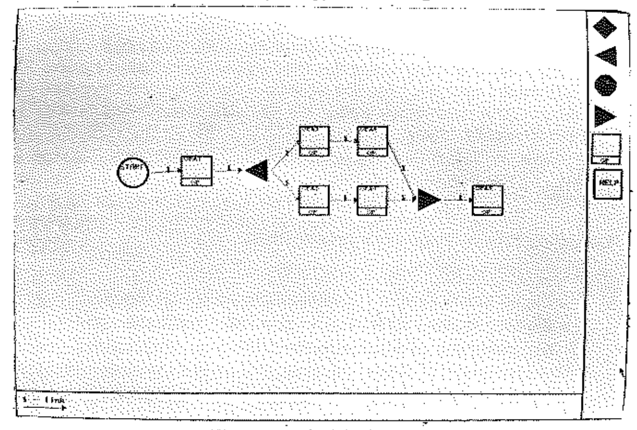

Figure 4: Activity Layout

follows:

```
{{Deburring-Machine
  IS-A: Discrete-Machine
  INSTANCES+INV:
    Deburring-Machine-1
    Deburring-Machine-2
  CAPACITY: 1
  OPERATORS-REQUIRED: 1
  MEAN-TIME-BETWEEN-
    BREAKDOWNS:
  CURRENT-ORDER:
  QUEUE:
  MAXIMUM-QUEUE-SIZE
  ARRIVE-BEHAVIOR:
    Standard-Debur-Arrive
  LOAD-BEHAVIOR:
    Standard-Debur-Load
  UNLOAD-BEHAVIOR:
    Standard-Debur-Unload}}
```

In the schema the names of the slots (e.g., capacity) are capatilized while the values of slots are not.

Some slots specify attributes and behavior. Slots, such as CAPACITY and MEAN-DOWN-TIME are attributes of the deburring machine. The behavior slots contain the names of routines that govern the behavior of the machine when an order arrives at the machine and is loaded and unloaded.

Other slots specify relations. The IS-A slot relates the Deburring-Machine prototype to Discrete-Machine which is a higher level concept. The INSTANCE+INV slot relates the Deburring-Machine prototype to specific instances of deburring machines so that information that is true of all deburring machines, such as behavior, is automatically transferred from the prototype to the instances. Through the relation slots, hierarchies and networks of schemata can be constructed. Figure 5 shows a hierarchy of machines, work centers, and cost centers for a sample factory.

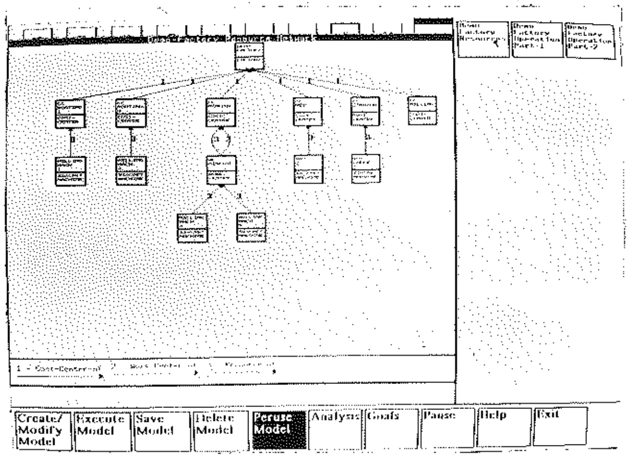

Figure 5: Schema Hierarchy

The user will be queried for the values of some slots, but others will be defaults which the user can override. [10, 11]

Event Generation

By specifying the activities, the user will be automatically specifying the events for the simulation (e.g. arrival of order at the machine, loading and unloading of the order).

The library also has a domain specific language for specifying additional events. For the factory floor application area, this language allows users to specify behavior for machines (e.g., machine breakdowns and maintenance) operators (e.g., shift changes) and operations.

Instruments

Simulation Craft has a flexible instrumentation capabilty that allows users to specify statistics that they want to measure. Instruments can measure values of a parameter ,such as queue length, as it changes over time as well as the number of times a particuliar value appears (e.g. record number of resource breakdowns)

Consistency and Completeness

When the user specifies the model, Simulation Craft checks that no information is missing or inconsistent. Examples of problems it might discover are:

- Missing information (Has the user entered resources for all activities)
- Incorrect assignment of resources (Has the user specified that the lathing operation uses milling machines?)
- Overcommittment of resources (Does the model use more than 100% of a single or group of operator's time?)

Modification

The user is able to modify the model, for example to include an additional machine or operation. Consistency and completeness checking will be performed after the modification just as it was after creation.

Version Management

Users can save and delete models. In addition, the user may create and save different versions of the same model by modifying the original verion. The system will track and graphically display the relations between different versions of the model.

EXPERT MODEL EXECUTOR

After creating the model, the user will run it to obtain results. The system uses its expert knowledge to specify the execution parameters of the model. The running simulation is displayed graphically on the screen. At any point the user can stop the model and alter it or attach visual gauges.

Design of Experiment

Before the user can answer questions using simulation, the simulation experiment must be designed which involves identifying the options to be tested, the data to be collected, and the length of the simulation runs. These tasks usually must be performed by a simulation expert, but Simulation Craft performs these tasks for the user by using rules and goals which incorporate simulation experts' knowledge and heuristics.

Goal-Specification

In most cases simulation is used to evaluate alternative solutions to a problem. To spare the user the task of examining voluminous simulation data to determine which solution is best, Simulation Craft will automatically evaluate the solutions on the basis of user specified goals, which are specific to the application area. The user picks goals from a menu and indicates their importance. Figure 6 shows a sample menu for specifying goals.

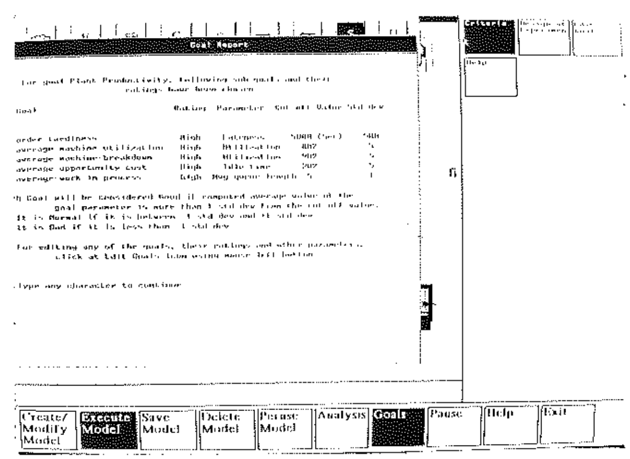

Figure 6: Specification of Goals

After the user picks goals, Simulation Craft will automatically create:

- instruments to obtain the data required to evaluate the performance versus goals
- evaluation functions for the goal

For example to measure performance relative to a late order goal, Simulation Craft will create an instrument to measure the number of late orders and a function that rates performance for a given percentage of late orders.

Graphic Display of The Running Model

The running simulation is displayed graphically on the screen in the form of transactions moving between resources and queues at resources. (see Figure 7) This graphic display helps the user follow the simulation so that he can see problems such as bottlenecks as they develop. The user can also zoom in on particuliar aspects of the display, such as a work center.

Model Reduction

Simulation Craft's model reduction capability allows the user to run the model at various levels of abstraction, to provide both detailed and summary-level views of the model. For example, the user could run the model at a detailed level that showed the operation of individual machines, or run the model at a summary level that just shows the operations of work centers. [12]

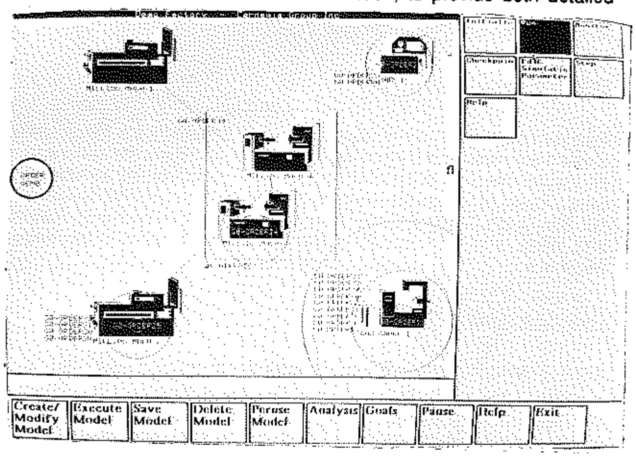

Figure 7: Running Model

Interactive Scenario Management

The user can stop the simulation at any point and either alter the model or employ a step option. Altered models can be saved. The stepping facility allows the user to step through a user-specified portion of the simulation a single event or multiple events at a time. The user might use this option to step through the model when he sees a problem, such a large amount of work in process, developing.

Gauges

The user can attach gauges to the simulation to display statistics, such as machine idle time, which are otherwise not displayed. The user can also input data through the gauges. Figure 8 shows a sample gauge.

EXPERT MODEL ANALYZER [13]

The Expert Model Analyzer applies artificial intelligence to the analysis task, effectively providing an "expert simulation analyst". Rules and goals are used to compare and identify problems in the scenarios that are being simulated. This automated analysis frees the user from the time-consuming task of analyzing the voluminous simulation data and determining how to alter the simulation model.

Goal-Analysis After the simulations are run, Simulation Craft will produce reports that compare the performance of the scenarios on the basis of the goals. The reports will rate the performance of individual scenarios as well as compare scenarios. Figure 9 shows a goal report for an individual scenario.

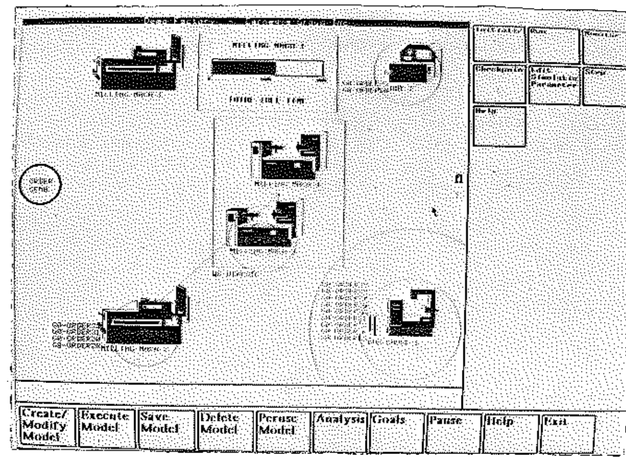

Figure 8: Gauge

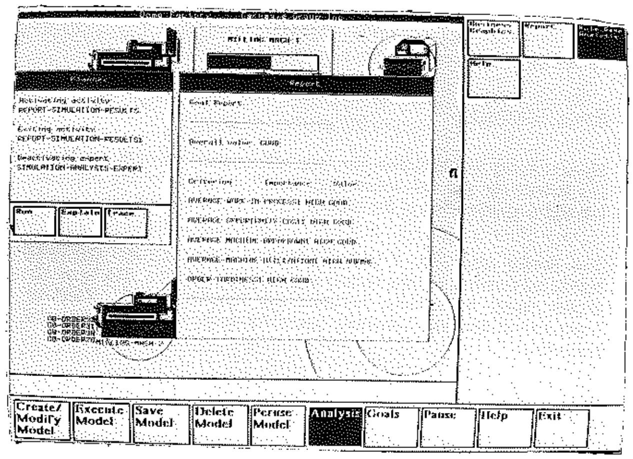

Figure 9: Goal Report

Rule-based Analysis of Scenarios

Besides comparing options, Simulation Craft uses rules to detect deviant behavior, such as a resource shortage, and specify corrective action. Problems are identified using domain knowledge about cause-effect relationships. [14]

A simple example of a rule to detect a bottleneck machine could be as follows:

```
IF a machine has a
   high queue length
   and high utilization rate

THEN the machine may be
     a bottleneck
```

Figure 10 shows the output of rules that identify machines that are under-utilized or are likely to break down.

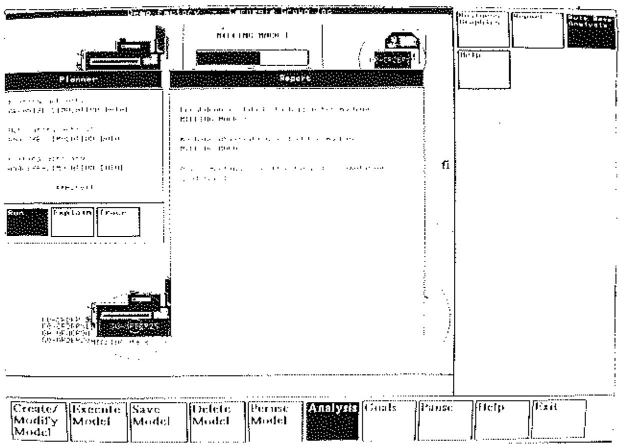

Figure 10: Rule-based Analysis

Additional Analysis Tools

Simulation Craft also contains a full statistical analysis and graphical display capability, so that users can conduct further analysis. Figure 11 shows an example of a histogram, which is one of Simulation Craft's graphical display capabilities.

SUMMARY Performing successful simulations is a long and difficult process which requires substantial expertise. Simulation Craft reduces the time and expertise required for simulation by incorporating the knowledge of simulation experts in expert systems that help the user create, execute and analyze the model.

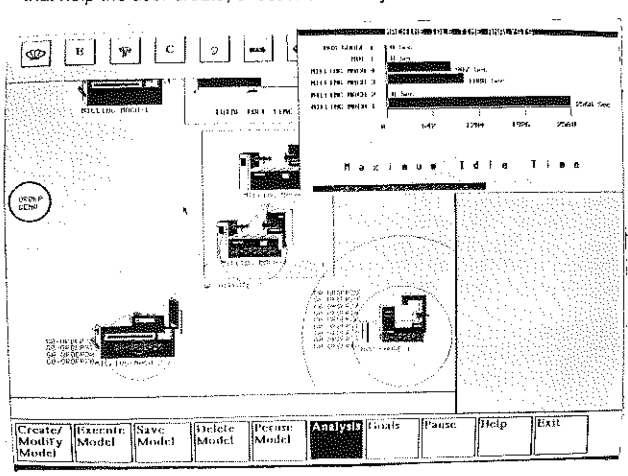

Figure 11: Histogram

[1] REFRENCES

[1] Jerry Banks and John S. Carson II, "Process-interaction Simulation Languages", *SIMULATION*, May 1985 44:5 225-235.

[2] Robert A. Sargent, "Statistical Analysis of Simulation Output Data", *Symposium on the Simulation of Computer Systems IV*, August 1976 pp. 39-50.

[3] G.K. Hutchinson, "The Automation of Simulation", *Winter Simulation Conference Proceedings*, 1981, pp 489-495.

[4] Y.V. Reddy, Mark S. Fox, Kevin Doyle, John Arnold, "INET: A Knowledge-Based Simulation Model of a Corporate Distribution System" The Robotics Institute, Carnegie Mellon University, Pittsburgh PA, 1983.

[5] James D. Hollan, Edwin L. Hutchins, Louis Weitzmam, "Steamer: An Interactive Inspectable Simulation-Based Training Systems" *AI Magazine*, Summer 1984.

[6] McArthur D. and Sowizrat H., "An Object Oriented Language for Constructing Simulations", *Proceedings IJCAI, 1981.*

[7] Sarah E. Goldin and Phillip Klahr, "Learning and Abstraction in Simulation", *Proceedings IJCAI, 1981* pp. 212

[8] "Knowledge Craft, Version 2.0 Users Manual", June 5, 1985, Carnegie Group Inc, Pittsburgh, PA.

[9] Arvind Sathi, Mark S. Fox, Michael Greenberg, "Representation of Activity Knowledge for Project Management", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 5, September 1985.

[10] Mark S. Fox and Y.V. Reddy, "Knowledge Representation in Organization Modeling and Simulation: Definition and Interpretation," *Proceedings of the Thirteenth Annual Pittsburgh Conference on Modeling and Simulation*, April 1982.

[11] Mark S. Fox and Y.V. Reddy, "Knowledge Representation in Organization Modeling and Simulation: Definition and Interpretation: A Detailed Example", *Proceedings of the Thirteenth Annual Pittsburgh Conference on Modeling and Simulation*, April 1982.

[1]TM Knowledge Craft, Simulation Craft, CRL and Find/Filter are trademarks of Carnegie Group Inc.