Reddy Y.V., M.S. Fox, N. Husain, (1985), "Automating the Analysis of Simulation in KBS", Proceedings of Society for Computer Simulation MultiConference, Simulation Councils Inc: La Jolla, CA, pp. 34-40.

Automating the analysis of simulations in KBS

Y.V. Reddy
Mark S. Fox
Nizwer Husain

Intelligent Systems Laboratory
Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

1. Introduction

The purpose of simulation is to provide the system designer a tool to study the dynamics of a proposed or an existing system. A system may be viewed as a collection of interacting components. The interaction of system components produces change in the "state" of the system. The main objective of studying systems is to understand how this change in state occurs and how it may be controlled. Ziegler [1] identifies the following elements of modelling and simulation:

- The **real system** that refers to a source of observable data.
- The **experimental frame** that refers to a limited set of of circumstances under which the real system is to be experimented with.
- The **base model** which is capable of accounting for all the input-output behavior of the real system that is valid for all the allowable experimental frames.
- The **lumped model** that approximates the base model by "lumping" together system components.
- The **computer** used to generate the input-output pairs of the lumped model.

If the lumped model is a reasonable approximation of the real system we can understand the behavior of the real system by experimenting with its computer implementation. By understanding the behavior, we mean determining the effect of control parameters(input) on performance parameters(output).

In models built using traditional simulation systems this is accomplished by analyzing the voluminous performance data generated by their execution. In many systems there is little or no control over what performance data may be gathered. In KBS [2] [3], an object oriented system incorporating an Artificial Intelligence approach to simulation we introduced the notion of *Selective Instrumentation* where only the data relevant to a particular *goal* may be gathered and analyzed.

An instrument in KBS may be viewed as a probe attached to a schema representing an entity in the model and collects specified data whenever that schema is acted upon in some way. For example if we are interested in studying the performance of a **queue** we may attach a *queue-measurement* instrument which is activated whenever some object "arrives-at" or "leaves" the queue. whenever the instrument is activated it may simply record the value of the simulation clock and the length of the queue. This information is stored within the instrument itself and can be formatted in a variety of ways using a *report-generator* associated with the instrument. For example if we are interested in the time dependent behavior of the queue we may subject the data collected to an analysis by a *time-series-analyzer*. On the other hand if we are only interested in the maximum length of the queue we can subject the same data to analysis by a *descriptive-statistics-analyzer*.

The selective instrumentation facility originally provided in KBS goes a long way towards simplifying the process of data collection and analysis. However, this still requires the simulation user to manually instrument each schema and subject the collected data to an appropriate analysis. It will be ideal if the user can simply deposit the *goal* of the simulation and have the system select the appropriate instruments, attach them to the relevant schemata and analyze the data and report its conclusions. The analysis of data consists of evaluating a set of constraints representing the *subgoals* of the organization being modelled.

Once such a facility is created, we can develop a **rule-based** planning system which can suggest a system configuration to achieve the desired goal by consulting a domain specific knowledge base containing cause and effect relations as well as heuristics about desirable system configurations.

In this we present a detailed discussion of the issue of data analysis in simulations and outline our approach to its automation in KBS.

Section 2 presents a discussion of data analysis facilities found in commercially available simulation systems and points out the need for facilities to perform automatic analysis.

In section 3 we present a series of strategic and operational decisions that decision makers in a corporate distribution group may face. These are used to illustrate the concepts developed in subsequent sections.

Section 4 describes an approach to specification of goals of simulation in terms of a set of constraints on the performance of various entities of the system being modelled.

Section 5 presents a taxonomy of various types of performance data and the type of analyses that may be performed on them. This is followed by a discussion of how simulation models may be "instrumented" to collect these data.

In sections 7 and 8 we present a method of instrumenting a simulation model based on the specifications of its goal which is illustrated by an example from the corporate distribution model discussed in section 3.

In section 8 we present the need for automatic data analysis to include the facilities for detection of "causes" of degradation or improvement of a performance measure so that simulation studies can go beyond the stage of presenting **what** happened in the current scenario to **why** it happened.

Section 9 also presents a discussion of some of the approaches we are investigating in KBS in this area. Section 10 presents a brief outline of the current implementation of the system.

2. Data Analysis in Simulation

In the words of Hamming [4] the purpose of computing is "insight" not numbers. As it applies to simulation the purpose of computing is to increase understanding of the system being studied and be able to predict its behavior under hypothesized conditions. Most commonly used simulation systems such as GPSS [5], SIMSCRIPTII [6] and GASP [7] are designed to provide statistical information on various components of the model and as such cannot provide in-depth understanding of the model behavior which suggests the need for a system which can analyze the data with little help from the model user. The objectives of such an automatic analysis system is to develop facilities for:

- "Rating" a given scenario from a user specified perspective.
- Detecting the "forces" acting upon specified performance parameters.
- "Suggest" a scenario that satisfies a user specified goal.

In the remainder of this section we briefly describe the nature of data collection in the current simulation systems.

2.1. Data Collection in Current Systems

Most self-contained systems such as GPSS provide a set of statistics on the permanent entities of the system as well as transactions at the end of the simulation. Simulation systems such as GASP and SIMSCRIPTII that use a general purpose programming language for their implementation admit total flexibility in selecting how and when various data are collected. However, this requires "hard wiring" the computer program for data collection which can prove to be very cumbersome. The work by Klahr et al [8] mentions automatic analysis but details have not been available to the present authors. The various types of data that may be collected in these systems may be classified as:

- **Summary Data** -- This generally describes the average behavior of the system during the period of simulation which will be useful in comparing different scenarios.
- **Instance Data** -- This describes the state of the system as specified by the values of selected performance parameters at a certain point in time during the simulation period. This may be used to determine if a given scenario has yielded a desirable system state at a specified time.
- **Conditional Data** -- This describes the values of selected performance parameters when the system state causes certain conditions to be true.
- **Timeseries Data** -- This describes the values of selected performance parameters as a function of time which provides a more in-depth understanding of the system behavior than other types of data.

All these types of data may be collected by modifying the simulation program and may be dynamically displayed in case of interactive implementations. The idea of selective instrumentation introduced in KBS provides a more convenient means for collection and display of this type of data. As mentioned earlier, the purpose of automatic analysis is not just collection of data on performance parameters but detection of causes that are contributing to their change.

As a first step towards the realization of such a system we discuss a paradigm where the user constructs a *goal-schema*, an object representing the abstraction of a **corporate goal** specifying the expected values(or a utility function relating the value of a parameter and its contribution towards the realization of the organizational goals) of organizational parameters which are collected and analyzed by the system from the perspective of rating a given scenario.

In the next section we discuss the proposed uses of a KBS model of a corporate distribution system (i.e. a subsystem of a corporation charged with managing the storage and distribution of products among its suppliers, manufacturing units, regional distribution units and its customers) in strategic and tactical decision making which will be used to illustrate the ideas developed in this paper.

3. Decision Support in a Corporate Distribution System

In this section we briefly review various types of decisions that managers at various levels in a corporate distribution group are expected to make. In each case applicable performance measures are also identified. Before these are considered in detail let us describe the corporate organization.

> XYZ corporation is concerned with manufacture and distribution of a variety of personal computers and accessories. Individual components and complete systems are either manufactured in the corporation's own facilities or supplied by outside vendors and are stored in regional distribution centers. Actual sale of components and systems is performed by retailers or company owned stores whose inventory is replenished by the distribution centers under the control of a business unit. The corporate distribution group within the corporation is responsible for the design of the distribution network as well as operational planning which is responsive to the overall corporate goals.

3.1. Best Distribution Process

The alternative distribution processes that are applicable include:

- Regional distribution centers with retailer network
- Central warehousing with regional distribution center network
- Regional distribution centers without retailer network
- Distribution of components directly from the point of manufacturing

The effectiveness of these alternatives is determined by measuring:

- Total distribution cost which includes the cost of transportation, order administration as well as inventory stocking charge.
- Customer service
  - Number of stockouts
  - Order cycle time
  - Customer inventory investment
- Distribution cost per unit sold by each channel
  - Direct sales
  - Sales by retailers
  - Sales by company owned stores
- Inventory Investment

3.2. Best Distribution Structure

The best distribution structure that meets the specified organizational goals is determined by:

- Number of facilities
- Nodal linkages
- Location of facilities

The performance measures used in this are same as those specified for the case of determining the "best" distribution process.

3.3. Best Inventory Deployment Policy

This decision is tactical in nature and is concerned with location of inventory (manufacturer vs distribution center vs retailer) and the quantity of inventory by location. Again the performance measures used in this are same as before.

3.4. Resource Requirements

For a given scenario which includes a specified distribution process, distribution structure and inventory deployment policy we are concerned with determining the resource requirements at distribution centers and thruput centers. These include space and manpower. This decision is in the operational domain and requires the capture of man-hours and space used in various activities.

3.5. Other Tactical and Operational Decisions

In addition to those described above there are a number of other tactical and operational decisions that are encountered in the distribution domain. These include:

- Best modal policy(i.e. rules governing the selection of various modes of transportation)
- Transportation requirements
- Best mix of geocodes serviced by each facility
- Effects of interruption in transportation
- Effects of demand changes
- Best mix of geographic codes for each source when parts are available from multiple sources.

4. Simulation Goal Specification

As noted earlier, the purpose of a simulation model is to identify a scenario which realizes the goals of an organization. An organizational goal may be expressed as a composite of a set of constraints on performance measures of the system. For example, XYZ corporation may be concerned with developing a distribution structure which permits "in-process" inventory not to exceed six weeks worth of sales while assuring retailers that their stockout rate will not exceed 10% of the orders filled. This may be viewed as simply imposing two constraints on the in-process inventory and retailer stockouts.

Constraints are an inherent part of all organizational models. An analysis of factory organizations by Fox [9]resulted in the following categorization of constraints:

- **Organizational Goal** -- this is a constraint on measures of how an organization has performed or is expected to perform under a given scenario.
- **Physical Constraint** -- this is constraint on what an object can or cannot be used for.
- **Enabling Constraint** -- this specifies the conditions that must be satisfied before an object can be used or process begun.
- **Preference Constraint** -- this specifies preference for one object over another when both are available.
- **Availability Constraint** -- this specifies the unavailability of an object for other used during the time period under consideration.

All these constraints are equally applicable to the corporate distribution domain. However, from the automatic analysis perspective we will be only concerned with the organizational goal constraints. Consider the following organizational goals of the XYZ corporation's distribution system:

- Total distribution cost should not exceed 10% of the cost of products shipped.
- Customer stockouts should not exceed 5% of orders.
- Order cycle time should not exceed 10 working days.
- Retailer inventory at any time should not exceed total sale for 10 days.
- Distribution cost per unit sold should not exceed 10% of the cost of manufacturing.
- Total "in-process" investment should not exceed 6 weeks worth of sales.

Given these constraints on the organizational performance it should be possible to "rate" a given scenario to determine how close it comes to satisfying the desired goals. The steps involved in this process are:

- Represent each organizational goal as a constraint.
- Select and attach "instruments" to gather data
- Specify the procedure foı computing the performance measures.
- Execute the simulation for the given scenario.
- Evaluate each constraint and compute a coefficient of constraint satisfaction(CCS) which may be positive to indicate exceeding the desired goal or negative indicating falling short of the desired goal.
- Evaluate the scenario by computing a coefficient of goal satisfaction(CGS) as weighted average of constraint satisfaction coefficients.

4.1. Organizational Goal Constraints

An organizational goal constraint may be viewed as an expected value of some organization variable. The constraint when evaluated yields a coefficient of constraint satisfaction(ccs) which reflects whether the goal is surpassed($0 < ccs <= 1$), exactly satisfied($cs = 0$) or fell short ($-1 <= ccs < 0$). An organizational goal is a weighted sum of the individual goal constraint satisfaction coefficients which is a measure of the desirability of the given scenario. Constraint evaluation has been explored in the domain of job-shop scheduling by Fox [9]. In this we adapt those ideas to deal with analysis of simulation data.

In the remainder of this section we outline the knowledge representation needed to support this process. Throughout this paper we use the word schema to represent concepts, objects and other data structures. For a complete description of the schema representation see [10]. Figure 4-1 shows the representation for an organizational goal which when evaluated yields a "rating" which is a measure of the desirability of the given scenario. This rating is derived by combining the ratings of individual constraints specified as the value of the slot: "contributing-constraints". Other slots in the goal schema specify the procedure for evaluation and display of the results. Figure 4-2 shows the representation of a constraint. The slot: "constrained-by" of the constraint schema contains the details of the instrumentation needed to evaluate a constraint.

```
{{KBS-goal
   RATING:
      comment: "a goodness or badness indicator"
   CONTRIBUTING-CONSTRAINTS:
      comment: "the individual goal constraints"
   STATUS:
      comment: "whether active or inactive"
   GRAPH:
      comment: "a kiviat graph displays rating"
   EVALUATION-SCHEDULE:
      comment: "governs when to evaluate goal"
   GOAL-SCHEDULER:
      comment: "interprets evaluation schedule"
   EVAL-FN:
      comment: "goal evaluation function"
   REPORT-FN:
      comment: "function to display goal state"

}}
```

Figure 4-1: Composite Goal Schema

```
{{ constraint
   CONSTRAINED-BY:
   comment: "specification for constraint"
   CONTEXT:
   comment: "decides if constraint applies"
   IMPORTANCE:
   comment: "relative importance of constraint"
   RATING:
   comment: "a goodness or badness indicator"
   VALUE:
   comment: "unrated raw value of constraint"
}}
```

Figure 4-2: Constraint Schema

5. Characterization of Performance Data

In this section we propose a knowledge representation scheme to characterize, collect and analyse performance data in a KBS model. Examples from the XYZ model will be used to illustrate the concepts.

A KBS model may be characterized as a network of schemata representing the permanent entities in the system through which schemata representing temporary entities pass through as the model is executed. For example, in the XYZ model Vendors, Manufacturers and Distribution Centers may be considered the permanent entities whereas orders and loads may be considered as the temporary entities. The performance parameters we want to measure may be related to either the permanent or temporary entities. Before we specify the methods of measuring performance data we should characterize the various types of data that may be collected. These are described in the remainder of this section.

5.1. Single Point Data

This data type is used to describe the value of a single parameter in the model represented as the value of a slot in a schema. For example the value of total inventory held by a distribution center may be represented as a **Single Point Data**. This type of data may be obtained by accessing a slot value or by computing a function. The only kind of analysis that can be performed on this is to apply a "range constraint" which determines if it is in an acceptable range.

```
{{on-hand-quantity
   INSTANCE: "single-point-data"
   ANALYSIS: "range-checking"
   DATA: 300}}
```

5.2. Multipoint Data

This type of data is used to represent the value of a parameter observed a number of times as in sampling and stored as a list. As in the case of Single Point Data this may be obtained by accessing a slot value or by computing a function. Multipoint Data can be analysed by computing descriptive statistics such as Max, Min, Range, Mean, Mode, Median and Variance as well as by curve fitting to establish the theoretical distribution underlying the data. For example, the changing nature of total value of inventory held by a distribution center may be represented as Multipoint Data.

```
{{inventory-levels
    INSTANCE: "multi-point-data"
    ANALYSIS: mean median
    DATA: 37 30 56}}
```

5.3. Timeseries Data

This type of data is used to represent the value of a parameter as it changes over time. This is similar to the Multipoint data except that the time of observation as well as the value of the parameter are recorded. In addition to the descriptive statistics, Timeseries data can be subjected to spectral analysis [11] to elicit the time dependent nature of the performance parameter. Using this, for example, we may be able to deduce that the total value of inventory held by a distribution center is at its lowest at the beginning of each quarter thus suggesting a further analysis to determine the causes of such a behavior.

```
{{inventory-levels
    INSTANCE: "timeseries-data"
    ANALYSIS: time-average
    DATA: (1Jan84 37) (2Jan84 30) (3Jan84 56)}}
```

5.4. Value Over a Set Data

This type of data is used to represent value of a parameter which represents the performance of a class of entities. The data themselves may be of any type: Single Point, Multipoint or Timeseries. For example, the total value of inventory held by each of a specified set of distribution centers may be represented by using this data primitive. In addition to the previously mentioned analyses that can be performed on "cross sections" (e.g. For each distribution center) many analyses reflecting aggregations (e.g. Average value of inventory for all distribution centers located in a geographic area) may be performed.

In the next section we describe the instrumentation facility that is needed to collect various types of performance data.

6. Instrumentation of Simulation Models

The task of data collection is concerned with recording the changes in the value of a parameter. This can be accomplished by constantly monitoring the parameter and recording every change or by sampling. The former yields greater accuracy albeit with greater computational overhead whereas the latter approach may be satisfactory in many cases and thus the selection of the data collection method is subjective. In the KBS environment the monitoring is accomplished by attaching **demons** to slots whereas sampling is done by scheduling data collection events or as a separate action during the execution of regular events. Since the data collection in KBS is analogous to physical measurements (using measuring instruments) the notion of an instrument has been introduced and is represented as:

```
{{KBS-instrument
    IS-A: "instrument"
    PURPOSE:
    INSTRUMENT-TYPE:
        Range: (OR "data-collection" "data-display")
    INSTRUMENT-MODE:
        Restriction: (OR demon event scheduled)
    SLOT-TO-DEPOSIT:
        Comment: slot value to be monitored or
                 the event slot when this will be
                 executed as one of the event actions
    SCHEMA-TYPE:
        Comment: the names of generic schemata to which
                 this instrument applies
    INSTRUMENT-SCHEDULE:
        Comment: event schedule for scheduled instrument
        Restriction: (TYPE "is-a" "event-schedule")
    DATA:
        Restriction: (OR
                (TYPE "instance" "point-data")
                (TYPE "instance" "multipoint-data")
                (TYPE "instance" "timeseries-data")
                (TYPE "instance" "set-data"  ))
        Comment: this schema defines the type of the data
                 and how it is collected
    ACTION:
        Comment: the data collection or display function
    ATTACHMENT-FUNCTION:
        Comment: the attachment procedure}}
```

Once an instrument schema is defined the information contained in it is used to attach it to an appropriate part of the model which when executed results in the collection of the desired data which can be subjected to desired analyses.

7. Goal Directed Instrumentation

One of the tasks involved in automatic analysis is the creation and attachment of appropriate instrument schemata and then subjecting the data thus collected to the desired analysis. For example, in the XYZ model one of our goals is to understand the behavior of inventory levels at XYZ-stores(i.e. stores owned by the XYZ corporation) located in a given geographic area so that we may develop suitable logistics to keep the inventory in those XYZ-stores at a desired level. In the remainder of this section we will describe the various steps involved and the knowledge representation needed at each step.

7.1. Goal Specification

Consider the goal:

```
Provide  an  analysis  of  inventory
levels of the component:  "xa50"  for
XYZ-stores  located  in  "geographic-
code": 25
```

This goal calls for generating a set of instruments which can measure the inventory levels and perform an analysis. By analysis we assume the intended analysis to be generation of descriptive statistics. However, it should be possible to specify a variety of analyses on the same data. From this statement of the goal we can deduce the following:

- Schema class: "xa50-inventory"
- Schema set: "related to XYZ-stores located in geocode:25 through the inventory-for relation"
- Parameter to be observed: "on-hand"
- Analysis: "descriptive statistics"

Using this information we can create the instruments described in the next subsection.

7.2. Generation of Instruments

```
{{xa50-inventory-instrument
    INSTANCE: "KBS-instrument"
    INSTRUMENT-TYPE: "data-collection"
    INSTRUMENT-MODE: demon
    SLOT-TO-DEPOSIT: "on-hand"
    SCHEMA-TYPE:"xa50-inventory"
    DATA: "xa50-inventory-on-hand-data"}}
```

Figure 7-1: Inventory Instrument Schema

```
{{xa50-inventory-on-hand-data
    INSTANCE: "set-data"
    SCHEMA-SET: set of xa50 inventory schemata
                derived from the instrument
    SLOT: "on-hand"
    ANALYSIS: "descriptive-stats"
    DATA: to be filled by the instrument}}
```

Figure 7-2: Inventory Data schema

The specification of the instrument and data schemata is to be derived manually. However, it will be possible to automate this process in future using a natural language deductive reasoning system

8. An Example of Scenario Rating

In this section we will illustrate the procedure for rating a scenario via an example from the XYZ model. Consider a composite organizational goal to increase customer satisfaction while keeping the distribution overheads low. This goal may be broken-down into two subgoals:

- Customer stockouts should not exceed 5% of orders.
- Distribution cost per unit sold should not exceed 10% of the

cost of manufacturing.

{{XYZ-goal1
INSTANCE: "KBS-goal"
CONTRIBUTING-CONSTRAINTS: "satisfy-customer"
"economize-distribution"
EVALUATION-SCHEDULE: "daily-at-midnight" }}

Figure 8-1: An example of XYZ corporation's goal

The corporate goal specified by the schema: "XYZ-goal1" is a composite of two organizational goal constraints: "satisfy-customer" and "economize-distribution". These are shown in figures 8-2 and 8-3 respectively. Each goal constraint is assigned an importance rating based on the role it plays in the overall corporate plan. Each goal constraint schema points to a constraint specification schema which specifies an utility function and the needed data collection instruments. Figures 8-4 and 8-5 show the constraint specifications for the "satisfy-customer" and "economize-distribution" constraints.

{{satisfy-customer
INSTANCE: "goal-constraint"
CONSTRAINED-BY: "satisfy-customer-spec"
IMPORTANCE: 0.70 }}

Figure 8-2: Retailer Satisfaction Goal Constraint

{{ economize-distribution
INSTANCE: "goal-constraint"
IMPORTANCE: 0.30
CONTEXT: economize-distribution-precon
CONSTRAINED-BY: "economize-distribution-spec"}}

Figure 8-3: Distribution Cost Reduction Goal Constraint

In order to evaluate the goal constraint: "satisfy-customer" we need to measure the total orders filled as well as the number of stockouts. This is accomplished by depositing the instruments: "measure-stockouts" and "measure-total-orders" shown in

{{ satisfy-customer-spec
INSTANCE: "goal-constraint-spec"
APPLY: eval-customer-satisfaction
UTILITY: "stockouts-utility-graph"
INSTRUMENTS: "measure-stockouts"
"measure-total-orders" }}

Figure 8-4: Specs for Customer Satisfaction Constraint

{{ economize-distribution-spec
INSTANCE: "goal-constraint-spec"
APPLY: eval-distribution-costs
UTILITY: "distribution-utility-graph"
INSTRUMENTS: "measure-manf-cost"
"measure-distribution-costs" }}

Figure 8-5: Specs for Distribution Goal Constraint

figures 8-6 and 8-7 respectively.

{{measure-stockouts
INSTANCE: "KBS-instrument"
INSTRUMENT-TYPE: "data-collection"
INSTRUMENT-MODE: event
ACTION: extract-stockout-info
SLOT-TO-DEPOSIT: "back-orders"
SCHEMA-TYPE:}}

Figure 8-6: Instrument to measure Stockouts

{{measure-total-orders
INSTANCE: "KBS-instrument"
INSTRUMENT-TYPE: "data-collection"
INSTRUMENT-MODE: event
ACTION: sum-up-total-orders}}

Figure 8-7: Instrument to measure Total Orders

The data needed to evaluate the "economize-distribution" goal constraint is collected by depositing the instruments: "measure-distribution-cost" and "measure-manf-cost" shown in figures 8-9 and 8-8 respectively.

{{measure-manf-costs
INSTANCE: "KBS-instrument"
INSTRUMENT-TYPE: "data-collection"
INSTRUMENT-MODE: event
ACTION: compute-manf-costs}}

Figure 8-8: Instrument to measure Manufacturing Costs

{{measure-distribution-costs
INSTANCE: "KBS-instrument"
INSTRUMENT-TYPE: "data-collection"
INSTRUMENT-MODE: event
ACTION: compute-distribution-costs}}

Figure 8-9: Instrument to measure Distribution Costs

The instrument action functions will be tailored to extract the desired information and update the "data" in the instrument. Note that instrument data may also be directly available in the schema it is attached to.

Having defined the goal, its constraints and the instruments, it is then connected to the model. When a goal is connected the relevant instruments are automatically attached to the model. The actual evaluation of the goal is an event scheduled to occur at some future date according to the "evaluation-schedule", at which time the "eval-fn" is executed. However, it may also be evaluated manually at any time as desired.

A goal evaluation function (e.g. eval-KBS-goal) will normally retrieve the "contributing-constraints" of the organizational goal. For each constraint if the "context" of the "constraint" applies then the constraint is evaluated by the function in the "apply" slot to compute a "rating". This rating weighted by the "importance" of the constraint contributes to the overall "rating" of the organizational goal. If the goal is evaluated more than once it should be possible to observe the direction and decide how good or how bad the organization is doing.

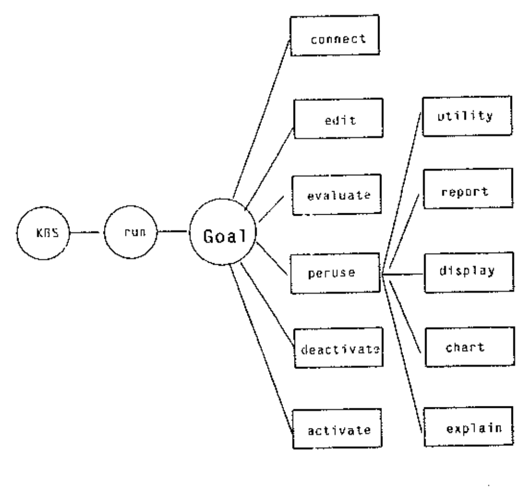

Figure 8-10: The Goal Command tree

The goals command hierarchy 8-10 illustrates the various commands available to a KBS user to create and manipulate goals.

9. Automatic Analysis

So far we have discussed methods used to create an organizational goal and derive a rating of the current scenario which is useful in comparing alternate scenarios. However, what we really want the simulation to do is to "suggest" why a particular aspect of the goal has not been met. This requires us to monitor the performance parameter and record what other parts of the model are manipulating it. Because of the object orientation of KBS it is not difficult to record the dynamic behavior of the model since the parameters of an entity (i.e. an object in the model) can only be affected by executing an **event message** (i.e. a command to execute an event with the current object as the focus). However, recording the dynamic changes of a parameter and what events caused that change is not sufficient to establish "cause and effect" since we do not know why the affecting event is caused (in general an event may be caused by multiple causes).

Our approach to establishing cause and effect behavior focuses on first *recognizing* the class of behavior to be understood and selecting a *(change in) representation* which is sufficient to identify cause.

9.1. Behavioral Classification

In order to analyze the performance of a simulation model it is necessary to understand the nature of different performance parameters and associate them with models that can explain their behavior. For example we may categorize the performance parameters as follows:

- **level** -- this represents the value of a parameter which may be incremented or decremented. Examples of such parameters are:
  - o inventory level
  - o queue length

  Parameters of this type may be studied by associating with them an underlying **producer/consumer** model.

- **cycle time** -- this represents the value of a parameter that is a sum of several individual times spent by a transaction (i.e. a temporary object) at various facilities (i.e. permanent objects/entities) in its entire life-cycle (between creation and destruction of the object) or time spent between two designated objects.

  Examples of this are:
  - o time spent by a shipment between the shipping location and destination.
  - o time spent by an order(request for goods) between the originator and the final order processing facility.

  Parameters of this type may be studied by associating a simplified queueing model derived from the captured dynamic behavior.

- **utilization** -- parameters of this type are used to represent performance of **server** type facilities. These also may be studied by associating a simplified queueing model derived from the captured dynamic behavior.

9.2. Flow Model Analysis

In the situation where *levels* are being analyzed and the pattern of events does not contain feedback loops. Consider the case of analyzing inventory levels which stock out often. It is important to determine what flows of orders and deliveries lead to the stock out situation, and where additional flow may relieve the problem.

A flow model may be constructed by recording the stimulus-response pattern of events; for each event the values of its parameters and the events cuased by it are recorded. A model can be created showing the magnitude of event flow between nodes in the simulation model. A graph theoretic analysis can then be performed to ascertain bottlenecks.

9.3. System Dynamics Approach to Simulation

In the case where the model contains feedback, a system dynamics [12] approach may be utilized. In the system dynamics approach the system is structured as a network of **causal feedback loops** necessary to explain the behavior of key performance parameters of the system and thus emphasizes **causal thinking**. The various steps involved in this approach are:

- **Develop a causal loop diagram** -- this shows the chain of causes that affect a parameter.
- **Develop a flow diagram** -- this represents the system as a collection of **levels** (i.e. the values of performance parameters) and **rates** (i.e. rate of change of other parameters that affect the level variables positively or negatively)
- **Develop a set of difference equations** -- this represents the mathematical relation ship between levels and rates (ofcourse rates are also related some levels)

Having developed the difference equations representing the system it is a simple matter to study the dynamic behavior of level variables under alternate scenarios. The simulation system DYNAMO [13] may be used to implement the model defined as a set of difference equations.

Consider again the XYZ model. We are interested in deriving a scenario where the "on-hand" inventory for any part stocked at a retailer is not allowed to become negative. In order to do this we will first execute our current model and record the level of inventory for the specified part over time. An analysis of this time series indicates that "stockouts" (i.e. on-hand inventory <= 0) have occurred at various points in time. Now we need to understand why this has occurred which may be done by constructing a simplified model of its behavior. The simplified model of this level parameter may look as follows:

- On-hand value is **decremented** by the execution of the "fill-order" event which happens regularly. We can derive the "rate of consumption" by analyzing the size of orders filled.
- On-hand value is **incremented** by the execution of the "shipment-arrival" event. The "rate of supply" can be computed by analyzing the shipments that are associated with each "shipment-arrival" event.

Using this model we can derive a difference equation:

```
on-hand[t] =   on-hand[t-1]
             - rate-of-consumption[t-1,t]
             + rate-of-supply-rate[t-1,t]
```

Using this equation we can compute the "steady-state" value for the on-hand inventory parameter. However, in most practical cases the cause and effect will be separated by a number of intervening events as well as different events may independently contribute to the increase or decrease of a level parameter.

In case a parameter is altered by more than one event we can easily record the changes contributed by each event and account for them separately in our difference equation. For example if retailers allow customers to return products purchased by them then we can alter our previous equation as follows:

```
on-hand[t] =
        on-hand[t-1]
      - rate-of-consumption[t-1,t]
      + rate-of-supply[t-1,t]
      + rate-of-customer-returns[t-1,t]
```

In the above example we simply computed the rate-of-supply by using the shipment sizes actually observed. However what we are really interested in is what determines the shipment size and the frequency of occurrence of the shipment-arrival event. This cannot be automatically determined by analyzing the **event network** alone. What we really need to capture is why or under what circumstances a particular event occurs. As part of the event behaviour definition the model builder should provide various conditions under which an event takes place (i.e. parameter value and the object) which can be used in propogating the causes of change in the parameter being studied.

Once again in our example assume that a distribution center in the XYZ model ships 10% of its excess inventory (i.e. inventory above a prespecified level) to the retailer under consideration. Now we can rewrite our difference equations of our model as:

```
on-hand[t] =
      on-hand[t-1]
    - rate-of-consumption[t-1,t]
    + rate-of-customer-return[t-1,t]
    + rate-of-supply[t-1, t]


rate-of-supply[t-1, t] =
    0.1 * (distribution-center.on-hand[t-1]
        - distribution-center.fixed)
```

Bhaskaran and Reddy [14] describe a strategy for capturing and verifying dynamic behavior of KBS models (i.e. event networks and parameters affected and accessed during simulation) which can be used to derive simplified models of individual performance measures.

10. Results

The "XYZ-goal-1" discussed in the example on rating scenarios was actually implemented on a model of a distribution network. After a few days of simulated time the goal was evaluated and reported.

Goal: XYZ-goal-1 **CLOCK:** Fri Jul 13 00:00 1984

Purpose: To control distribution overheads

Constraint	importance	rating
Distribution Costs	0.3	0.5
Customer satisfaction	0.7	-0.1

Overall goal rating: 0.08

This result indicates that customer satisfaction was *bad* because it was rated negatively but the distribution costs were economical and were rated at 0.5 which is *good*, thus bringing the overall rating into the good region.

```
Overall goal rating
   = wt. avg. of individual constraint ratings
   = (0.3 * 0.5  + 0.7 * -0.1)/(0.3 + 0.7) = 0.08
```

11. Conclusion

It has been shown in detail how goals, constraints and instruments work in harmony to rate model scenarios. This means we can collect data from a model run, define and connect goals to the model and be able to tell how good the model is behaving with respect to the organizational goals. The instruments, goals and constraints are constructed manually but data-collection and reporting are done automatically. Future work will focus on further automating the process of constructing instruments from a natural language description of the goal. In addition, work is also under way to detect causal relationships between model variables so that it will also be possible to suggest changes to be made to the model in order to obtain desirable model scenarios.

References

1. B.P. Ziegler, *Theory of Modelling and Simulation*, John Wley and Sons, New York, 1976.
2. 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.
3. Y.V.Reddy and Mark S. Fox, "Knowledge Representation in Organization Modeling and Simulation: A detailed Example," *Proceedings of the Thirteenth Annual Pittsburgh Conference on Modeling and Simulation*, April 1982.
4. R.W. Hamming, *Numerical Methods for Scientists and Engineers*, Mcgraw-Hill, New York, 1962.
5. IBM, *GPSS/360 Introductory Users Manual*, 1973.
6. P.J. Kiviat, R. Villaneuva, H.M. Marcovitz, *The SIMSCRIPT II Programming Language*, Prentice-Hall, Englewood Cliffs, N.J., 1969.
7. Pritsker A.A.B., *The GASP IV simulation language*, New York: John Wiley and Sons, 1977.
8. McArthur D and Sowizral H, "An Object Oriented Language for Constructing Simulations," *Proceedings IJCAI*, 1981,
9. Mark S. Fox, *Job-Shop Scheduling: A Study of Constraint-Directed Reasoning*, PhD dissertation, Computer Science Department & Robotics Institute, Carnegie-Mellon University, Pittsburg, Pa, 1982.
10. Wright,J.M. and Fox,M.S., "SRL/1.5 User Manual," Tech. report, Carnegie-Mellon University, 1983, Intelligent Systems Laboratory,The Robotics Institute, Pittsburgh,PA.
11. G.S. Fishman, *Concepts and Methods in Discrete Event Digital Simulation*, John Wley and Sons, New York, 1973.
12. N.Roberts R. Deal, D. Anderson, M. Garet, W. Shaffer, *Introduction to Computer Simulation- A system Dynamics Modelling Approach*, Addison-Wesley, 1983.
13. A.L.Pugh IIIMIT press, *DYNAMO II User's Manual*, 1976.
14. V. Bhaskaran, Y.V.Reddy, "Self-Understanding and Automatic Verification of Simulation Models," *Unpublished*, 1984, .