# KNOWLEDGE BASED SIMULATION OF CONSTRUCTION

Robert I. Carr
University of Michigan
Ann Arbor, Michigan, USA

David Y. Chang
Texas A & M University
College Station, Texas, USA

Eloy Morua-Padilla
Costa Rica Institute of Technology
San Jose, Costa Rica

## INTRODUCTION

This paper describes two knowledge based simulation systems for construction process and project planning, management, and control. It extends the preliminary models described at the Second Symposium on Robotics in Construction at Carnegie Mellon University two years ago.

RESQUE is a resource based discrete event Monte Carlo simulator for construction processes. It simulates mixed resource types and mixed and conditional strategies for resource allocation and activity durations which are typical of complex construction processes. RESQUE uses simple process charts to represent construction processes. Its process definition language (PDL) characterizes and tracks resource properties and defines resource calling, holding, handling, and routing functions. It simulates process performance to estimate actual costs for each configuration and strategy. It is implemented on an Apollo workstation and on an IBM AT using a DBase III Plus front end for user friendly accessibility.

DYNASTRAT is a Monte Carlo simulation based program that analyzes alternative resource allocation strategies for construction projects which have uncertain activity durations. It encapsulates decision rules to describe dynamic strategies that fit samples of labor and material costs, weather, productivities, and other random variables. DYNASTRAT models activity cost and time as random variables, models interdependencies among project activities and between their durations and costs, assesses weather effects on duration and cost of activities, incorporates simulated project status in decision making, provides a framework to model assignment of resources under uncertain project conditions, and evaluates the impact that different strategies may have on project duration and cost.

The domains of the two models differ in their level of scale. A construction project can be considered a group of activities or processes. RESQUE is designed for the microlevel of process planning; whereas, DYNASTRAT is designed for the macrolevel of project planning. RESQUE can be used to calculate probability distributions which are used as input to DYNASTRAT.

## RESQUE

RESQUE is an acronym for a resource based queuing network simulation. It is a general process simulator which was developed specifically to model repetitive, cyclical construction processes. A construction process is a fundamental work module in accomplishing a project. Examples are excavation, backfill, concrete placement, pipe hanging, and conduit laying processes.

### CONSTRUCTION PROCESS CHARACTERIZATION

A process can be divided into its basic operations. For example, (a) shovel excavates and loads dirt to a truck, (b) truck hauls the dirt, (c) truck dumps dirt, and (d) unloaded truck returns to excavation site. Construction process resource characteristics which are important in simulating and analyzing production are these:

1. <u>Resources activate operations.</u> Operations start when the resources they require become available to them.
2. <u>Resources are expensive.</u> Resource cost is the major cost variable in construction processes, and resources cost money even when they are idle.
3. <u>Resources are limited.</u> Quantity of project equipment, workers, and space are limited.
4. <u>Resource flows are sequenced.</u> Resource flows among operations are sequenced (1) by physical or technical requirements and (2) by managers' allocation decisions.
5. <u>Resource flows can be cyclic.</u> Resources can be utilized repetitively by the same operations.
6. <u>Resource flows can be delayed.</u> Resources can be idle, awaiting other resources needed to start an operation.
7. <u>Resources are not homogeneous.</u> Non-identical resources are expected. Trucks in an excavation process can differ in their hauling capacity and speed.
8. <u>Resource properties govern the operation executions.</u> Resource requirements and cycle time of an operation can depend on properties of involved resources.

RESQUE MODELING OBJECTIVES

RESQUE was developed to meet the following objectives, in recognition of the resource characteristics described above and experience with other simulation tools (GPSS, MUD, CYCLONE, INSIGHT).

1. Describe a process in an easily understood process chart which depicts operations, resources, and resource flows among operations.
2. Allow resource representation at the level of detail which is required to describe processes. For example, (1) resources can be grouped by resource types, but (2) individual resources can be differentiated within a resource type.
3. Model interactions of non-identical resources. Duration and resource requirements of an operation are sensitive

to properties of individual resources.

4. Test different process alternatives without complicating the process chart. Different resource selections and resource allocation plans for a process can be formulated and tested without modifying the process chart.
5. Generate performance data that helps pinpoint process bottlenecks. Resource delays at different locations and the production of operations in a process should be tracked and reported to evaluate resource productivity for different process alternatives.

RESQUE MODELING ELEMENTS

Resources in a construction process must be accounted for at all times, either in an operation or idle. In a RESQUE model, resource states are represented by three elements, which are connected by an arc element to depict possible resource flows. These elements are modeled on those of CYCLONE, but each has properties which extend it past CYCLONE.

1. <u>Combi Elements:</u> A Combination, or Combi, element is an operation that can operate on a combination of resources from more than one immediate source. All required resources must be available to activate a Combi. A Combi can be considered a logical AND element. Predecessors of a Combi must be Que elements which will be defined later.
2. <u>Normal Elements:</u> A Normal element is an operation that does not require resources from more than one immediate source. Instead, it operates on resources coming from any one of its preceding elements, which can be Normals and Combis. A Normal can be considered a logical OR element.
3. <u>Que Elements:</u> A Queue, or Que, element is a state in which resources wait in a process. Delays occur when resources at a Que wait for resources from other Ques to arrive to start a Combi. Therefore, Ques must precede Combis, and Combis can only follow Ques.

4. <u>Arc Elements:</u> An Arc element is a possible resource flow path. An Arc connecting a Que and a Combi establishes a resource requirement, i.e., required resources must flow from the Que to the Combi to start the operation. An Arc emanating from a Combi or Normal shows a possible path that resources released from operations may take according to user specifications.

A RESQUE process chart is a simple network consisting of these four basic elements, which delineates operations and resource flows in a process.

RESOURCE CHARACTERIZATION AND TRACKING

RESQUE allows non-identical resources to flow through the same chain of operations, which requires the ability to model the resource-operation dependency mentioned earlier. Therefore, it is necessary to characterize and track resource properties which can affect operation functions.

Each resource unit in a RESQUE model is characterized by two user defined parameters: the resource type name and the optional integer value attribute that represents a resource property. As a result, resources of the same type can have different properties. For example, a resource of "truck" type with an attribute value of 1 is different from that with an attribute value of 2. The one with attribute of 1 might represent a 10-ton truck and the other might be a 20-ton truck.

In RESQUE, resources flow in sets during simulation. A set is an aggregate of one or more resource units in a strict hierarchical structure. That is:

1. A set can have only one parent, called the set header.
2. A set may contain subsets, subsets of subsets, etc.
3. Set headers do not have parents, and each subset header can have only one immediate parent.

Resources involved in a construction process can all be described by the set representation. An empty truck is a set by itself, and a truck loaded with structural steel is a set of which the truck is the set header and each structural steel member is a subset within the set. RESQUE tracks the characterization (type name and attribute value) of each resource unit in a set.

A resource dependent variable (RDV) is a variable whose value depends on attribute values of specific resources. RDV's are used to model operation function parameters whose values are resource property dependent, and they control these functions during process simulation. Users define the dependency rules which can be referred to for determining the values of RDV's. The RDVLIST statement in RESQUE contains all such user defined rules.

Therefore, the attribute of each resource unit in a set can be referred to for determining the value of a function parameter that depends on the attribute. For example, the duration of the operation to erect each structural steel member can be dependent upon the size or other attribute of each member. The structure of sets can be dynamically changed during simulation according to handling definitions in operations. RESQUE always tracks the latest structure of each set, as well as the characterization of each resource unit in the set.

RESQUE OPERATION FUNCTIONS

Functions can be defined in operations (both Combis and Normals) to specify resource interactions and to control dynamic resource flows in simulation. These can vary according to different model variables such as resource properties and simulated time. The operation functions are as follows:

1. Resource Calling Function (CALL): A Combi calls resources from each of its preceding Ques. In a call, users specify (a) Que element from which resources are called, (b) preference by which resource sets are drawn from the Que, and (c) number of resource sets called from the Que. Each Combi can have its own resource calling plan which can vary with the properties of involved resources. Different calling plans can be

defined without complicating the process chart. The calling function can also be controlled by a condition which must be true before the calling is executed, such as a required level of some model variable.

2. Resource Holding Function (DUR): Resources involved in an operation are held for a specific time period over which they are being operated on. This duration defines the time that one operation cycle takes. Resources in a Combi can be held for more than one operation cycle in order to complete handling tasks (described below). The duration parameter can be a constant, a random variable, or an RDV, whose value depends on attributes of specific resources in the same operation.

3. Resource Handling Function: Every construction operation involves resource handling, of which there are three basic tasks. Transporting is to move a resource set without changing its content. Assembling is to merge resource sets from different sources into a new set. Disassembling means to remove specified resources from a set. RESQUE will model assembling and disassembling tasks, and will model transporting as a special case of disassembling. Combis can do assembling and disassembling, and Normals can do disassembling. A typical disassembly at a Combi will be described in the CALL by type and attribute of resources to be disassembled, allowable number of units in the group, and number of operation cycles needed to disassemble one resource unit. Each of these can be an RDV.

4. Resource Routing Function (DEST): The routing function controls model resource flows. Resource sets are routed to Ques and Normals, sent to an assembling base (if in a Combi), or destroyed. Routing rules can be deterministic, probabilistic, conditional, resource attribute based, resource quantity based, Que length based, operation activation based, or a combination of these.

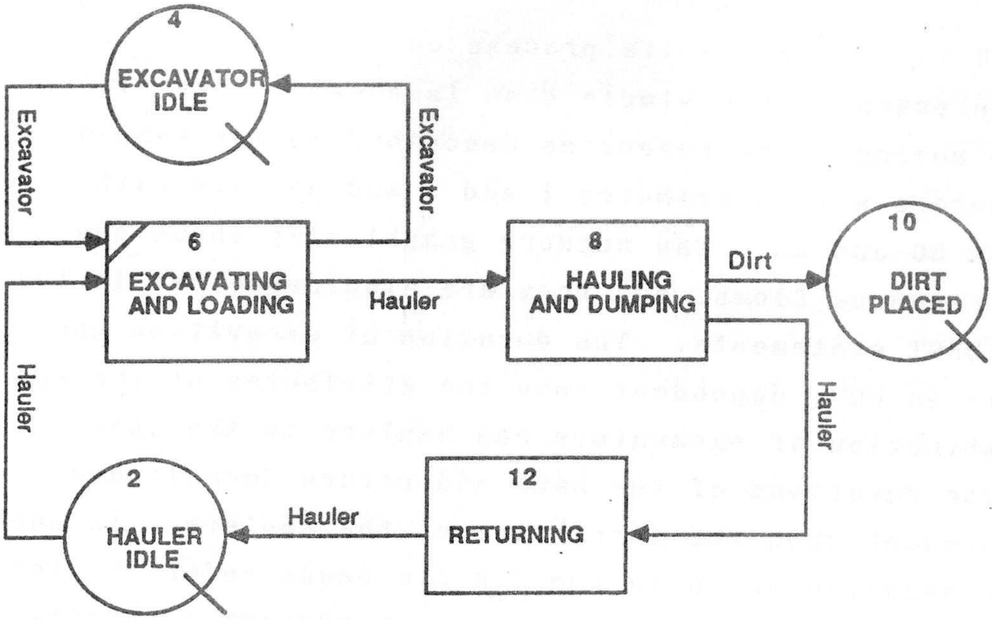
RESQUE IMPLEMENTATION

RESQUE is defined by its process definition language (PDL). An example of a simple case is shown in the figure for earth moving. The resources described in the RESLIST are excavators with attributes 1 and 2 and haulers with attributes 30 and 50. The network graphically shows the possible resource flows, but they are also described in the CALL and DEST statements. The duration of excavation and loading is an RDV, dependent upon the attributes of the specific combination of excavators and haulers in the operation. The durations of the haul and return Normals are RDV's dependent upon the attributes of the haulers. In each case, the negative value in the DUR statement refers to the rule in the RDVLIST, and the rule in the RDVLIST identifies the appropriate statement in the DURLIST.

RESQUE is programmed in Fortran, developed on an Apollo workstation and since implemented on an IBM AT. We have developed an interactive program in DBase III Plus which guides a user through input in a manner which is much easier than direct input into the PDL. The DBase III Plus program translates from the interactive screens into the PDL. Examples of a screen print from two templates for the earth-moving example are shown, on which the input information is marked with a heavy pen. The first template shows the completed screen for input for calling the hauler resource for excavating and loading. The second shows the completed screen for input of duration of excavating and loading.

Each of templates allows input for the RDVLIST, in response to the question "RDV?<Y,N>". The first template has "N" for "no", and the second has "Y" for "yes" for the durations. This second shows that the duration of Combi #^, Excavation and Loading, is dependent upon the combination of hauler and excavator. Excavator 1 requires 7 to 9 minutes for a 30 ton hauler or 10 to 12 minutes for a 50 ton hauler, for example.

RESQUE is now limited in its resource representation,




```
PROCESS, EARTH MOVING, 1

CONDLIST, 1
  1, CC, 10, GE, 600

RESTLIST, 4
  2, EXCAVATOR, 1, N1
  2, EXCAVATOR, 2, N2
  4, HAULER, 30, N3
  4, HAULER, 50, N4

DURLIST, 8
  1, 2, 7., 9.
  2, 2, 10., 12.
  3, 2, 3., 5.
  4, 2, 5., 7.
  5, 2., 8., 11.
  6, 2, 10., 12.
  7, 2, 6., 9.
  8, 2, 9., 11.

RDVLIST, 6
  1, 2, HAULER, -2, 30, -3, 50
  2, 4, EXCAVATOR, 1, 1, 3, 2
  3, 4, EXCAVATOR, 2, 1, 4, 2
  4, HAULER, 3, 30, 5, 50
  5, HAULER, 5, 30, 6, 50
  6, HAULER, 7, 30, 8, 50

COMBI, 6, EXCAVATING AND LOADING
  CALL, 2, P1
  DEST, 8
  CALL, 4, P2
  DEST, 4
  DUR, -1

NORMAL, 8, HAULING AND DUMPING
  DEST, 12
  GEN, DIRT, , -4, 10
  DUR, -5

NORMAL, 12, RETURNING
  DEST, 2
  DUR, -6

QUE, 2, HAULER IDLE, 30, 50

QUE, 10, DIRT PLACED

QUE, 4, EXCAVATOR IDLE, 1, 2
```

- Que No : (4)
- Priority:<L,F,G,S,or H> ?(G)
  {L:LIFO,F:FIFO,G:LARGEST,
  S:SMALLEST,H:ATTRIBUTE }
- Calling Quant.:RDV?<Y,N>(N) Q = (1)
- Holding Cycle :RDV?<Y,N>(N) T =
- Handling Patt.<A,B,S>?(A)
  {A:ASSEM,B:BATCH DISASSEM,S:STEP DISASSEM}

RESOURCE DEPENDANT VARIABLES (R.D.V.) LIST

| Rule | Que# | Res. Type | RDV<Y,N>? | Attr_1 | RDV<Y,N>? | Attr_2 |
|---|---|---|---|---|---|---|

GENERATE <Y,N>?(N)

DURATION OF COMBI # 6

Duration : RDV<Y,N>?(Y)

R.D.V. LIST

| Rule | Que# | Res. Type | RDV<Y,N>? | Attr_1 | RDV<Y,N>? | Attr_2 |
|---|---|---|---|---|---|---|
| 1 | 2 | HAULER | Y Rule : -2 | 30 | Y Rule : -3 | 50 |
| 2 | 4 | EXCAVATOR | N Ref.#: 1 | 1 | N Ref.#: 2 | 2 |
| 3 | 4 | EXCAVATOR | N Ref.#: 3 | 1 | N Ref.#: 4 | 2 |

DURATION LIST

Ref.# Dist. Type<1,2,3,4>? {1:Const.,2:Uniform,3:Normal,4:Beta}

| Ref.# | Dist. Type | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | Uniform : | Max. = | 7.000 | Min. = | 9.000 |
| 2 | 2 | Uniform : | Max. = | 3.000 | Min. = | 5.000 |
| 3 | 2 | Uniform : | Max. = | 10.000 | Min. = | 12.000 |
| 4 | 2 | Uniform : | Max. = | 5.000 | Min. = | 7.000 |

and its model definition is input-intensive. It is being fitted with a more mature knowledge based representation through an object oriented paradigm in which resources and operations are encapsulated units. Each unit will have its own knowledge, which includes resource properties and interaction rules. To define a specific process, one needs only to select a set of operations and resources from the system's construction processing knowledge base. With the encapsulated knowledge, the system can analyze process design and performance. The simulator is suitable as a simulation module in construction robot program generation and analysis, operating upon the knowledge base that describes robot operating characteristics and project attributes.

## DYNASTRAT

### CONSTRUCTION PROJECT CHARACTERIZATION

A construction project is an interrelated set of chains of activities in which the project duration is determined by the path of activities that produces the longest duration. One calculates the duration of a project by estimating the duration of each activity in the project, or at least the chain which is expected to be critical. Activity durations are functions of the level of resources assigned and their productivity.

Estimates of costs of each activity can produce the estimated cost of building a project. The costs vary with the prices and productivities of the resources. The productivity of resources is a function of weather, resource proficiency, supervision, site conditions, and level of resources assigned. On many projects, weather effects, resource prices, resource productivity, supervision, and site conditions are random variables, which makes activity and project durations and costs random variables.

The sum of expected values for uncertain activity durations is not an accurate estimator of project duration because of merge event bias. Construction activities' share dependencies on weather, productivities, supervision, etc., and these interdependencies must be modeled.

Limitations on resources force resource allocation decisions, and resource assignments are critical in determining activity starts, productivities, durations, and direct costs and project duration, overhead costs, and total costs. These decisions are regularly made under conditions of uncertainty.

DYNASTRAT MODELING OBJECTIVES

DYNASTRAT has been designed to meet the following objectives, in light of the project characterization above.

1. Model uncertainty in activity durations and costs.
2. Model activity and project costs as important variables for making resource allocation decisions.
3. Model correlations among activities, and between activity durations and their costs.
4. Consider assignment of limited resources under uncertain project conditions.
5. Account for dynamic changes introduced by the progress of the project.
6. Allow construction managers to assess the impact that different resource allocation decisions can have on the duration and cost of uncertain projects.

DYNASTRAT VARIABLES

The model can be characterized in an influence diagram by its decision variables and by its state variables. The decision variables are under the control of the manager and are shown by rectangles. The state variables are random, under the control of nature, and are shown by circles. Intermediate outcomes calculated by the model are shown in hexagons.

DYNASTRAT MODULES

DYNASTRAT consists of several modules, or engines, each of which performs a different and specialized function. They are fully integrated in their structures, and their

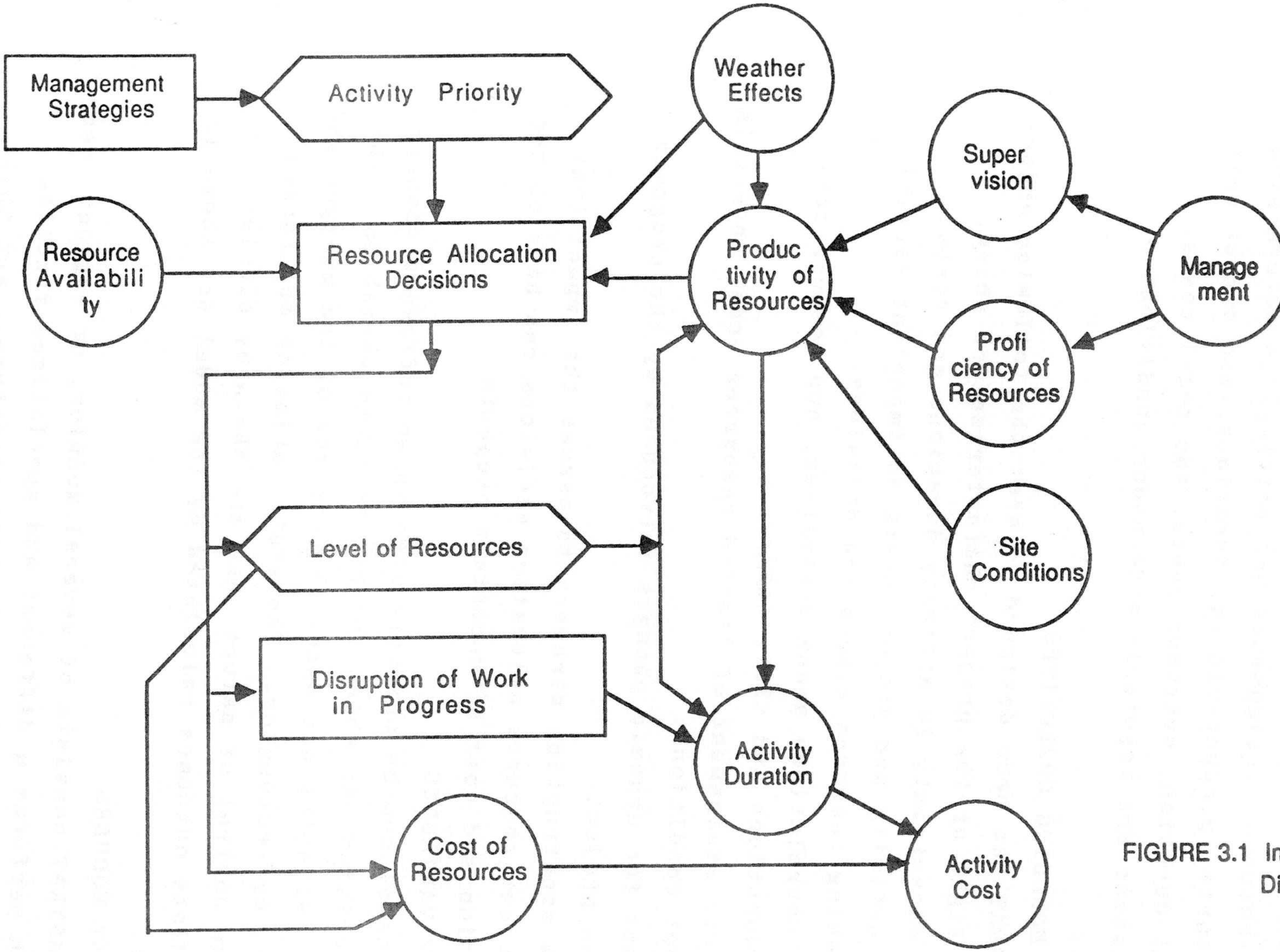



FIGURE 3.1 Influence Diagram

relationships are shown in the flow chart. The modules are described below:

1. Network Logic Engine determines which activities are eligible to start and calculates CPM values: total and free float, early and late start and finish dates, project duration, criticality indexes, and critical path.
2. Interdependent Random Variables Sampling Engine generates random samples of random variables, including weather, that affect activity duration and cost. It calculates weather correction factors and cost and duration modifying factors and combines these samples to model correlations among activities.
3. Dynamic Management Strategies Engine estimates priority of activities, chooses crew size to start the allocation at scheduling dates, and determines when activities in progress can be disrupted to reassign their resources to activities of higher priority.
4. Resource Allocation Engine selects activities for allocation of resources. Based on the current set of management strategies, it assigns priorities to these activities, selects crew sizes, and allocates resources. It also tracks resource availability and predicts and updates equipment delivery dates.
5. Progress Simulation Engine simulates daily progress on each activity according to productivity of resources, allocated crew size, and weather conditions generated by random sampling.
6. Activity Completion Forecast Engine forecasts future activity completions and status, as input to resource allocation, at the start and at control dates. This incorporates current status of project and knowledge obtained from activities already finished, and it forecasts start and finish dates, floats, cost, duration, and priority of activities in progress/not yet started.
7. Cost Simulation Engine calculates total direct cost of each activity, according to its duration, crew sizes,

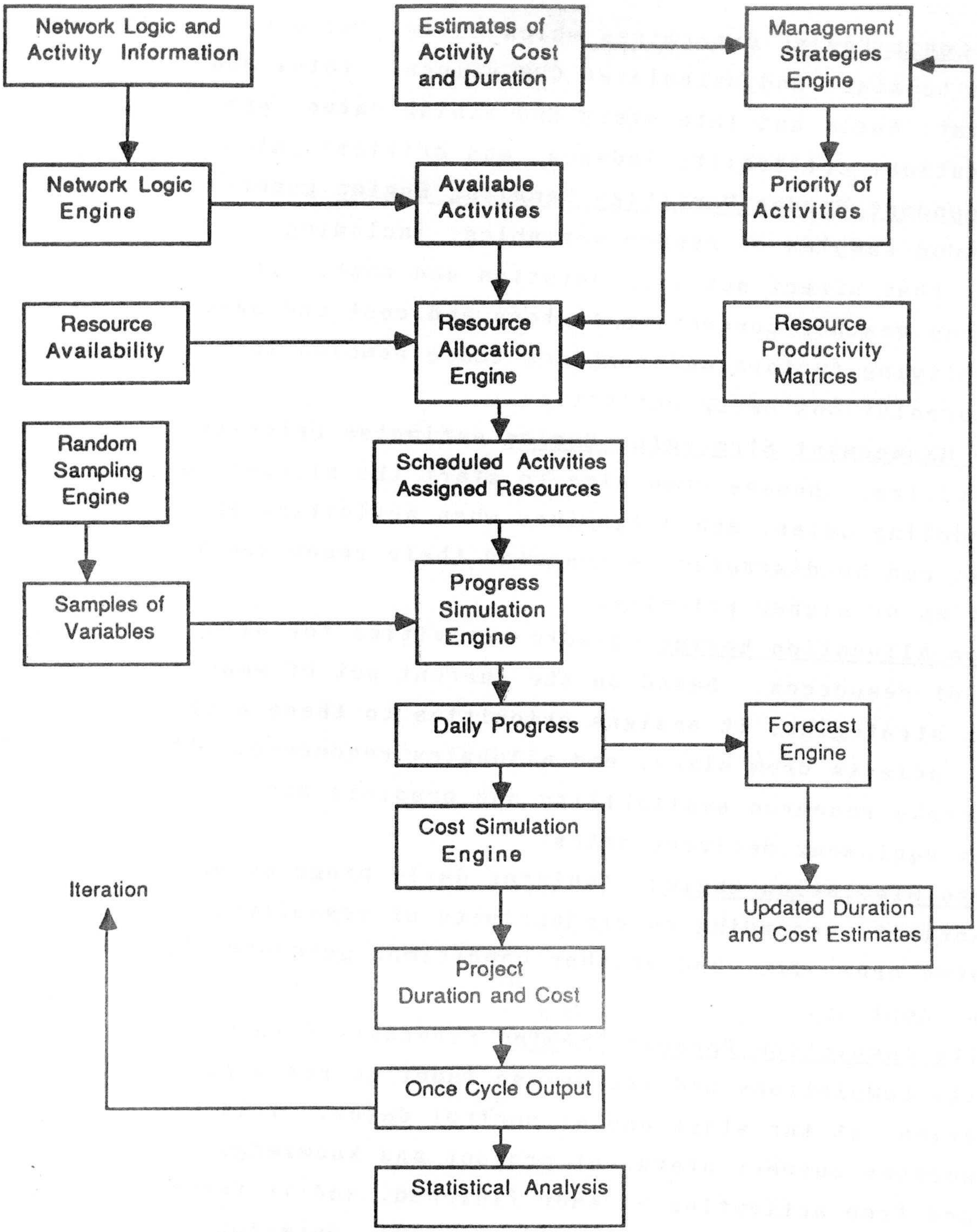



FIGURE 3.2 DYNASTRAT General Flow Chart

and resource and material costs. It also calculates fixed and variable overhead and total project costs.

DYNASTRAT INPUT

1. Activity and network data provide the precedence logic of the activities and the total output necessary to finish each activity.
2. Probability distributions and interdependencies of random variables which affect duration and cost of activities. These include daily resource costs, total output required, crew and equipment relative productivities, and material cost. Some are unique to a particular activity, and others are shared among activities.
3. Resource and productivity matrix for each activity describe the composition (type and number of each resource) of different possible crew sizes and their expected productivities under standard conditions.
4. Resource availability for labor and equipment.
5. Project labor and equipment hourly costs and material unit costs.
6. Project overhead cost which cannot be directly related to individual activities are input as fixed cost plus cost per day of project duration.
7. Management strategies for making dynamic activity and resource scheduling decisions. Strategies include managers' priorities among the set of rules which determine (a) how priorities are calculated, (b) order of activities, (c) interruption of activities in progress, and (d) level of resources assigned to each activity.
8. Estimates of durations, costs, and criticality to be used as base values for application of strategies.
9. Definition of control date occurrences.
10. Probability density functions for daily weather conditions from which program will simulate weather.
11. Weather correction factors for activities which define sensitivity of each activity to different possible weather conditions.

OVERVIEW OF PROJECT SIMULATION

At every scheduling or control date, DYNASTRAT checks the network logic to determine which activities can start, identifies their resource requirements, and checks to see whether resources are available for these activities. If they are, the model schedules as many of these activities as possible by following the set of management strategies. Using the generated daily weather data and the weather correction factors of the activities scheduled, DYNASTRAT simulates the daily progress achieved on each activity on every simulated day, calculates cumulative progress to date, actual daily cost incurred, and cumulative cost to date.

At dates defined in input as control dates, the model obtains the status of the project and forecasted durations and costs of the remaining activities based upon progress to date and information to date on the outcomes of the variables of uncertainty that affect duration and cost. The network is recalculated using this information, and the priorities of all activities are updated to provide a new basis for scheduling activities.

When required cumulative simulated daily progress is achieved for each activity, the model stops the simulation and records the activity actual duration and cost. When all activities have reached their required cumulative progress, the model performs CPM and cost calculations to obtain actual and late start and finish dates, floats, critical path(s), and project duration and cost.

In its current implementation, DYNASTRAT is a specific expert system that can be converted into a generic project planning expert system. DYNASTRAT's network logic, interdependent random variables sampling, dynamic management strategies, resource allocation, progress simulation, activity completion forecast, and cost simulation engines provide a knowledge base and inference engine that compares alternative management strategies for a project. This can be augmented with a general knowledge base of resource costs and availabilities. Managers need only input the network logic of a particular project and generic classification of each activity to determine the preferred set of decision rules to follow during construction.

This material is based upon work supported by the National Science Foundation under Grant No. CEE-8409939.