Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

KNOWLEDGE-BASED ASSEMBLY OF SIMULATION NETWORKS USING CONSTRUCTION DESIGNS, PLANS, AND METHODS

Iris D. Tommelein
Robert I. Carr

Department of Civil and Environmental Engineering
The University of Michigan
Ann Arbor, MI 48109-2125, U.S.A.

Abdalla M. Odeh

Department of Civil Engineering
Jordan University of Science and Technology
Irbid, JORDAN

ABSTRACT

An object-oriented and interactive computer system is presented that integrates project- and process-level planning. This system, named CIPROS, realistically models construction processes by matching resource properties with design component properties and operation durations. It uses a modular representation to create discrete-event simulation networks and to relate simulation output to the design and construction plan of the facility to be built. CIPROS users must identify and describe attributes of components to be constructed, based on the facility's design drawings and specifications, and they must develop a CPM plan. They must also select a construction method to perform each activity by retrieving the appropriate elemental simulation network from a library of networks that represent such methods. CIPROS then pieces the networks together based on sequential relationships from the plan and property values input from the drawing and specification data. The latter initialize the simulation network resources that make up the constructed facility. To complete the simulation network, users must specify the construction resources that are available to perform the work and which may be shared by activities. CIPROS comprises a fully- operational discrete-event simulation engine that is called once a network is completed. Besides producing statistical reports that are instrumental in assessing the quality of the construction plan, CIPROS can also be used to check the degree of facility completion as the simulation progresses.

1 INTRODUCTION

During construction planning, anticipated work is typically represented by activities in a CPM schedule. Individual activities describe at a high level of abstraction what work needs to be done. Precedence relationships between activities describe how activities must be sequenced to accomplish all work.

Activities can also be represented at a more detailed level of abstraction to characterize individual resources (e.g., trades, materials, and equipment) and their interactions within and across activities. At this detailed process level, it is meaningful to also model the extent and duration of such interactions and the degree of uncertainty associated with them, so that discrete-event simulation can aid in imitating the behavior of the real-world system over time and provide data to support advance planning.

One can also use design data to characterize the wide variety of components that must be assembled or manufactured in-place to construct the facility. While one-of-a-kind designs comprise many standard types of components (such as footings, beams, and columns, which represent classes of components), each individual component (termed a class instance) has property values (including its actual dimensions, material, and location in the facility, the number of reinforcing bars needed to build it, the amount of concrete to be placed in it, etc.) which represent the values of attributes of the instance). Such attributes are not represented in most construction management models, e.g., when CPM is used to create a construction plan, though they largely define what construction resources a contractor must bring to the site and how long individual activities will take.

A realistic construction process model must therefore use this detailed but otherwise disparate construction management data, as is provided in design drawings, specifications, precedences among activities, available contractor resources, and typical construction methods. This can be done by linking disparate data sources to one another, across their levels of abstraction, to form a comprehensive unit. It is then necessary to create a discrete-event simulation engine that can use individual component characteristics without unduly complicating the discrete-event model formulation task. This is achieved in the CIPROS model that is presented here.

2 CIPROS MODEL

CIPROS is a pseudo-acronym for Construction Integrated PROject and process planning Simulation system. It uses discrete-event simulation to represent stochastic activity durations, model resource availability and interactions, and model component attributes that dictate resource requirements and activity durations. Through this integration, realistic characteristics of components and resources are available to process simulation.

2.1 Modeling Procedure

Creating a discrete-event simulation model using CIPROS includes seven consecutive steps that use CIPROS' knowledge-based representation and simulation primitives. These items, shown in parentheses, are explained later in the paper. The user must perform the following steps:

Step 1. Define Project Design and Specifications: specify a plan to build a facility by entering design drawing and specification data in object-oriented format. (This data will fit within CIPROS' project taxonomy and component class hierarchy.)

Step 2. Create Activity-Level Plan and Relate Activities to the Design Components: specify the activities needed to build this design by entering the activities by number, name, and scope and by showing their precedence relationships.

Step 3. Choose Construction Method for each Activity in the Plan: For each activity, select a construction method. (Methods are defined by process charts and operations functions.)

Step 4. Initialize Product Components: CIPROS uses the design and specification data to initialize the product components that relate to activities in the plan (i.e., they will initialize CInQues in the method that was chosen to perform the activity).

Step 5. Identify Construction Resources: provide the type and quantity of construction resources for each method (i.e., link the RInQues from each chart to elements in CIPROS' Resource Class Hierarchy, so that the appropriate attributes can be inherited and used in simulation).

Step 6. Custom-Tailor Simulation Network: CIPROS pieces the elemental simulation networks that describe construction methods together in the sequential order defined by the plan. Generally, each process has its own operations and queues. However, if operations of different processes share resources, the user must define such queues and assign priorities.

Step 7. Simulate and Interpret Results: The construction plan is simulated for a user-defined number of replications.

2.2 Example Application

Consider the construction of a simple concrete structure that has: two types of footings (Footing-1 and Footing-2), two types of columns (Column-1 and Column-2), two types of walls (Interior-Wall-1 and Exterior-Wall-1), one type of slab (Slab-1), and a single type of roofing (Roof-1).

The individual components have additional characteristics. For example, this project has eight components of type Footing-1 and four of Footing-2. Each footing of type Footing-1 is made of concrete of type Concrete-4 and includes five elements of Rebar-1 and seven of Rebar-2, it is 0.6 m long by 0.3 m wide by 0.2 m high. Each footing of type Footing-2 is also made of concrete of type Concrete-4, but it includes eight elements of Rebar-1 and ten of Rebar-2, and it is 0.4 m by 0.4 m by 0.2 m. Columns, walls, slab, and roof would have their own characteristics.

Estimating rules determine that 0.054 m^3 of soil must be excavated for each footing of type Footing-1 and 0.048 m^3 for those of type Footing-2 (a rule-of-thumb might be to take 1.5 times the volume of the footing). Similarly, 0.0378 m^3 of concrete will fill each footing of type Footing-1 and 0.0336 m^3 each of type Footing-2 (this is 5% more than the calculated volume of each element). Also, two forms of type Form-1 and two of type Form-2 will shape each Footing-1, while four of Form-3 will shape Footing-2.

To model the placement of rebar and concrete using a traditional simulation model, a user must construct a process diagram with several resource queues, as elements in a queue are considered to be identical for modeling purposes. If Rebar-1 and Rebar-2 are not identical, one queue should hold Rebar-1 and another queue Rebar-2, despite the fact that all pieces of rebar may have to go through similar processing steps of handling and placing. A single queue might be used to hold the resource Concrete-4 with a total amount of 0.4368 m^3, but some counting mechanism must be crafted to guarantee that the exact fraction thereof is placed in each Footing-1 and the other fraction in each Footing-2. CIPROS does not need such duplication of queues that hold similar components and concoction of counting mechanisms that use combinations of available primitives.

3 CIPROS KNOWLEDGE BASES

CIPROS uses an object-oriented representation to distinguish among individual components at simulation run time. The CIPROS discrete-event simulator handles distinctions on an as-needed basis, according to which class resources belong to and what differentiates among individuals in each class. This greatly simplifies model formulation, realism, and reusability. To achieve these results, simulation input must be specified in knowledge bases that contain project design data, individual component data, and construction resources.

3.1 Project Taxonomy

The example can be cast in a hierarchy that describes the design elements in the project, which is termed the project taxonomy (Figure 1, Top). Some attribute values, such as unit dimensions of a component, must be input explicitly. Other attribute values, such as volume of concrete per component, can be specified as a formula (sometimes called a method or a demon) that computes this volume based on the components' dimensions.

3.2 Class Hierarchy of Product Components

For each component, the user must specify the CIPROS generic component class hierarchy to which the component belongs. For example, Footing-1 and Footing-2 belong to the class Footing (Figure 1, Center). As will be clarified later, this specialization is necessary for CIPROS to apply its knowledge of generic component construction methods for building to specific project components (these components will initialize CInQues).

3.3 Class Hierarchy of Construction Resources

CIPROS also contains a class hierarchy of contractor resources to be used in construction activities (Figure 1, Bottom). They are classified as labor, equipment, or materials in support of construction, such as carpenters, cranes, and form work. Labor refers to individual workers or to crews composed of workers with the same or different skills. Equipment includes both the machines themselves and attachments for them as attachments often determine a machine's functionality. The user can specify the contractor resources to be used for construction, to which CIPROS will then apply its knowledge of construction methods using those specific construction resources (they will initialize RInQues).

3.4 Activity-level Construction Plan

The construction of a CIPROS simulation network is based on fleshing out the activity-level construction plan. A traditional way to plan construction work is to break a project down at a manageable level of detail into activities. A common convention is that activities pertain to a certain type of work and involve the continuous application of selected resources. Activities are sequenced with the traditional finish-to-start relationships to form the project plan. Figure 2 represents a possible plan to construct part of the example project. The space required for each footing will be excavated. Then, formwork and reinforcing will be placed. Following this, each footing will be cast. Of course, the time it takes to perform any of these activities for any individual footing will depend on that footing's attribute values. For example, it will take longer to place rebar in footings of type Footing-2 than in those of type Footing-1 because more pieces of rebar are involved. The scope of an activity refers to the project components to which the activity applies and is defined by an ordered list of component class(es) each with its number of units. These components must exist in the component hierarchy. In Figure 2, the user has placed all of Footing-1 and Footing-2 in the same activities. Each activity must thus complete all of Footing-1 and Footing-2 before a following activity can start (this is the finish-to-start precedence relationship).

After determining the topology of the activity network, the user must select a construction method for each activity, select the construction resources needed by each method, and assign activity durations. CIPROS will use its previously-defined hierarchy of construction resources to retrieve the attributes that are relevant to each process.

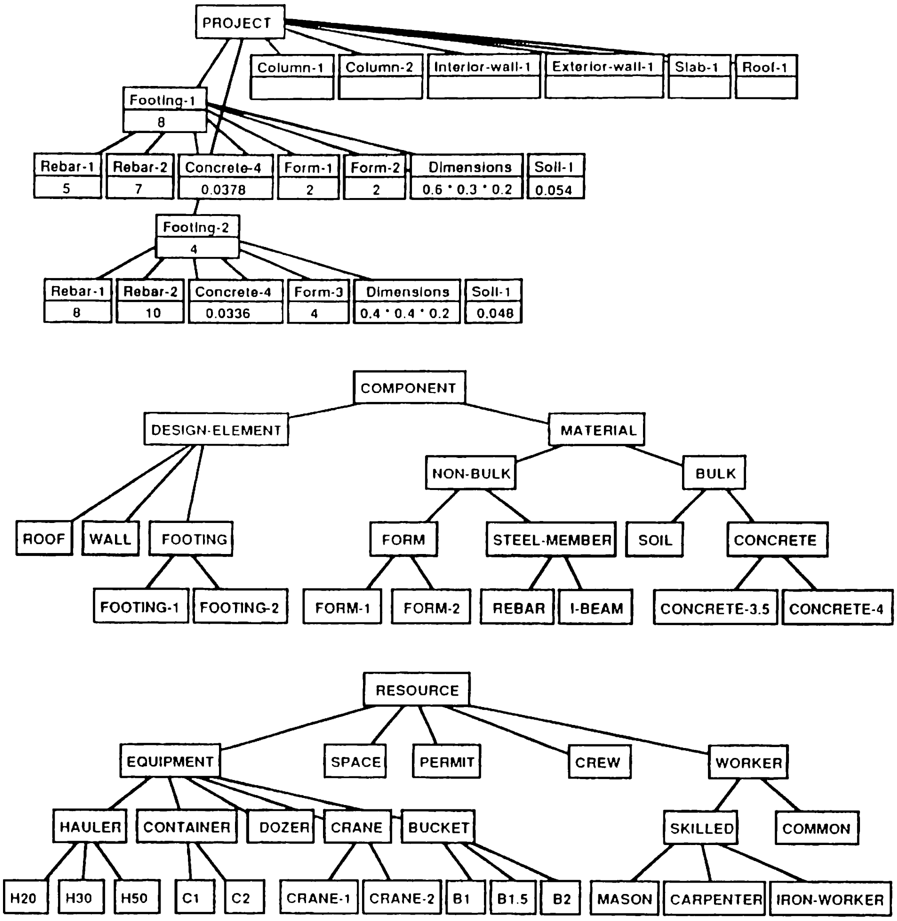

Figure 1: Hierarchical Knowledge Bases: Project Taxonomy (Top), Component Class Hierarchy (Center), and Resource Class Hierarchy (Bottom)

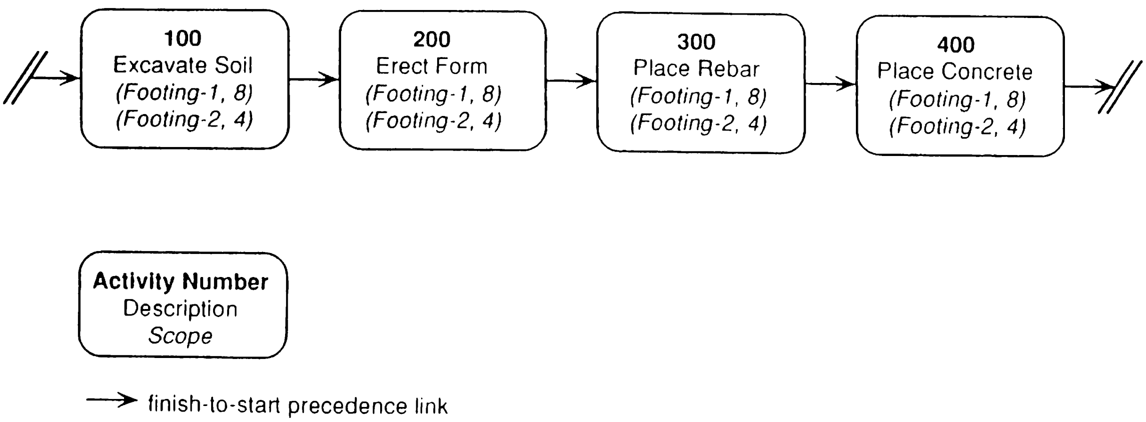

Figure 2: Activity-level construction plan.

3.5 Library of Construction Techniques and Methods

Construction methods form the link between project and process planning. Methods are fairly standard, because most construction work is routine. For example, concrete delivery, transport, and placement operations are innate to the process of placing concrete. The sequencing of these operations in the process is basically the same for footings, first-level columns, or second-level columns regardless of the type and quantity of concrete in each footing or column, though their durations may vary. The actual quantities can be determined at run time, and duration distributions can be selected when concrete placement is being simulated.

Accordingly, methods are predefined in a CIPROS knowledge base by means of elemental discrete-event simulation networks, comprising a process chart and operation functions. Processes have input and output queues with inputs and outputs that are initially specified only at an abstract class level. This is the key to realistic process modeling in CIPROS: these classes will be instantiated at run time, based on information drawn from the actual design drawings and specifications and the construction resources provided by the contractor, in the order given by the construction plan. The operation functions will be applied to the inputs of the network and their evaluation can be based on those inputs.

Standard processes are classified in a knowledge base of construction techniques such as concrete placing, rebar placing, and form erecting. For each technique, the knowledge base includes processes representing different construction methods that specify which and how construction resources will be used to do the work. For example, concrete can be placed by one of several methods, such as crane and bucket, concrete pump, or concrete buggy.

4 PROCESS CHARTS

A process chart is a network of modeling elements (operations, queues, and arcs) that describes entity states and flows in a process. The generic term entity is used here to denote a unit such as a product component or a construction resource that flows in simple or compound form through the simulation network. A set is defined as an assembly of one or more entities in a strict hierarchical structure. That is, an entity may have more than one child but can have only one parent. For example, a hauler and a steel member each represent a set. A hauler loaded with three steel members is a set, in which the steel members are children of the hauler entity. The set is referenced by the set header entity, here, the hauler. The CIPROS simulator tracks the set's structure and the characteristics of each entity in the set throughout simulation. The modeling elements that make up the process chart cannot themselves depict all relevant information. Operation functions that are separate from the process chart are used to define entity tasks and interactions and to control entity flow in the process.

4.1 CIPROS Modeling Elements

Cyclone (Halpin and Riggs 1992) is the best known discrete-event simulation system used in construction. Its modeling capabilities are based on a small, comprehensive set of primitives. The CIPROS modeling elements carry the same names as Cyclone's, but their functionality is extended (Figure 3).

- Normal Operation (Normal): an unconstrained operation that does not require entities from more than one immediate source. A Normal is equivalent to a logical "or" in a network path, because it is activated upon arrival of any entity routed to the Normal. A Normal is represented graphically by a rectangle.
- Combination Operation (Combi): an operation constrained by a combination of entities from more than one immediate predecessor. A Combi is equivalent to a logical "and" in a network path, because it operates on a user-defined combination of entities of different types called the Combi ingredients. A Combi can only be activated when all its ingredients are available. It is represented graphically by a rectangle with a diagonal line at the top left corner.

In CIPROS, Combis have operation functions. These pertain to determining which number of entities must be withdrawn from a preceding Queue, how long an entity should be captured by the Combi, etc. These operation functions are defined with the links that connect a Queue to the Combi. They are described with the handling tasks, later in this paper.

- Arc: Represents the flow path of entities released from an operation (i.e. Combi or Normal). An Arc is represented graphically by a line with an arrow head to show the direction of flow.

Link: Represents entity flows from Queues and depicts entity requirements and entity tasks that will be applied to the entities joining the Combis that follow. Links are classified by these entity tasks:

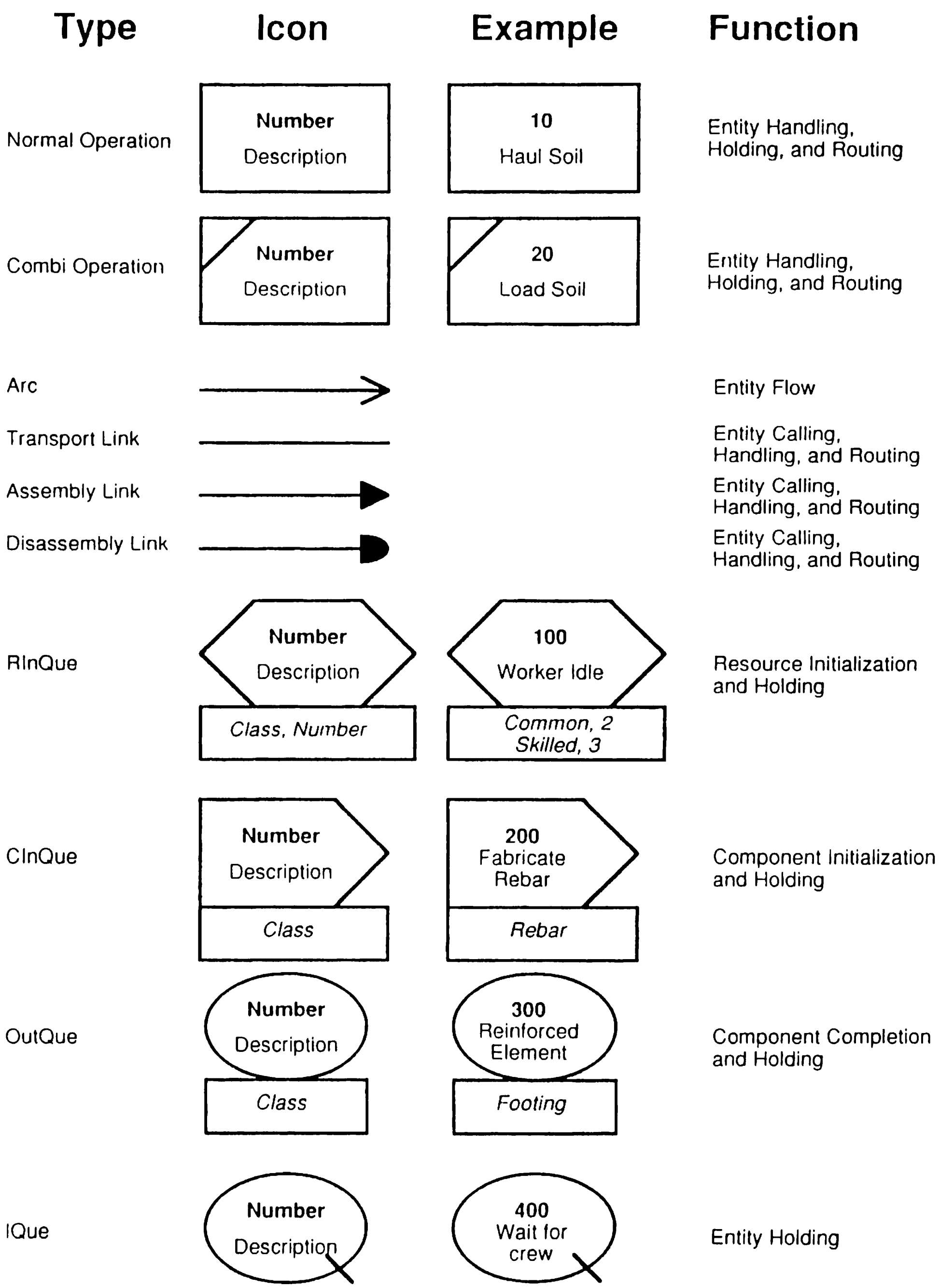

Figure 3: CIPROS Modeling Elements and Their Functions

- ♦ Transport Link: Represented by a line.
- ♦ Assembly Link: Represented by a line with an arrow head.
- ♦ Disassembly Link: Represented by a line with a rounded head.

Queue: Represents an idle state of entities or sets. Process delays occur when entities at one Queue wait for entities to arrive at Queues to start their common succeeding Combi operation. Besides representing a wait state, Queues also serve two other functions. First, Queues are used to match construction resources to product components at the start of the simulation of the process. Second, Queues are used to track the process status. This enables CIPROS to schedule succeeding processes. To serve these functions, Queues are classified as: o Resource Input Queue (RInQue): Represents a Queue of entities, named input resources, such as the construction resources a contractor brings to the site (e.g., a crane or form work). These entities set the production rate of Combis and Activities but do not get consumed in the process. An RInQue is graphically represented by a hexagon with a rectangle under it. Construction resources are defined by ranked pairs of their class(es) and their number, as shown in the rectangle. Resources of different classes are initialized in the queue in the order of their listing.

- ♦ Component Input Queue (CInQue): Represents a Queue of entities, named input components, such as design elements specified by architects or engineers (e.g., concrete and rebar to be placed in a beam). These entities generally get handled, assembled, or converted in the process. A CInQue is graphically represented by a pointed box with a rectangle under it. The input components of a CInQue are defined by their type, as shown in the rectangle. Only one type of component can be defined in each CInQue.
- ♦ Component Output Queue (OutQue): Represents a Queue of entities, named output components (e.g., a beam that has been cast). These entities result from performing the process. An OutQue is graphically represented by an oval with a rectangle under it. The output components of an OutQue are defined by their type, as shown in the rectangle. Only one type of component can be defined in an OutQue. An OutQue provides a counter that updates the process completion status whenever a component of the defined type joins this Queue.
- ♦ Intermediate Queue (IQue): Represents a Queue that is neither an input nor an output. In contrast with other Queues, which are strictly typed, an IQue can hold either resource or component entities. An IQue is graphically represented by an oval with a diagonal line across its lower right side that forms the letter "Q".

CIPROS process charts are small, comprehensive units that show how construction resources can be applied to perform a certain activity. In short, they describe construction methods as elemental simulation networks. Examples will be shown later. These charts depict the static structure of the processes. They are complemented by operation functions that define the dynamic nature of the processes.

4.2 Operation Functions

Operation functions specify entity interactions and control entity flows during simulation. They allow users to model different process strategies without altering the process network. Process strategies are generally related to entity use or entity routing. An example of a strategy is "use largest available hauler" in a loading operation. In a Normal operation, these functions are attributes of the operation and thus apply to all entities joining the Normal. However, in the case of a Combi operation, where entities from different Queues are involved, the functions (except the operation duration) are attributes of the Link between the Combi and a preceding Queue. The operation functions in CIPROS are:

- Operation Duration
- Set Destination
- Set Calling Priority
- Handling task
  - ♦ Transportation
  - ♦ Assembly
  - ♦ Disassembly
- Resource allocation and capturing

5 DISCRETE EVENT SIMULATION

CIPROS simulates the project construction plan as follows:

- It identifies activities in the project plan that can start and their corresponding methods. If concurrent activities share resources, precedence goes to those with lower number.
- It identifies product components and construction resources to initialize CInQues and RInQues in each elemental process network representing the chosen methods.
- It simulates the processes. That is, CIPROS retrieves the appropriate attribute values needed to withdraw elements from the queues, activates nor-

mal and combi operations according to these values, and accordingly samples durations from the probabilistic distribution of each operation duration. When a process is terminated, the status of components that appear in the OutQue is updated. The activity modeled by this process is herewith terminated and its successors will start, that is, their queues are initialized. In this way, CIPROS steps through the entire plan, until no more activities can be performed.

Note that components do not flow between processes but are initialized when an activity starts. This is the essential link between the process simulation and the CIPROS knowledge bases. Resources are initialized at the earliest start of the activities whose processes compete for them. The activity products are project components that are routed to output queues and result in activity completion. The initialization of these resources and input components triggers the start of the process by making available the entities required for process operations.

The simulator tracks the state of product components, construction resources, and activity precedence throughout simulation. Process statistics are similar to those produced by CYCLONE and provide information related to the process operations, tasks within processes, and queues, and thus the involved resource utilization, delays, and bottlenecks.

The statistical reports pertain to the last replication and the overall simulation. At the activity level, they report start and finish times of each activity and its process and the activity's duration. These output reports are instrumental in measuring the quality of a construction plan and help identify areas for improvement.

5.1 Implementation Software and Hardware

CIPROS is implemented under PC Windows* using the ACTOR (Franz 89, Whitewater Group 90) object-oriented programming language. The CIPROS architecture is generic; its modeling concepts apply to a variety of construction facilities, techniques, and methods. In its current implementation, however, the scope of CIPROS' knowledge bases was limited to construction of structural systems for high-rise buildings.

6 CONCLUSIONS

CIPROS realistically models construction processes by integrating data from design drawings and specifications, construction plans, in combination with a library of construction methods. The availability of these knowledge bases provide leverage to the CIPROS simulation engine that can exploit diversity in resources. The model also alleviates knowledge-intensive simulation network construction and initialization tasks. This results in a model that extends beyond current planning and simulation models in its ability to integrate activity- with process-level planning. This new methodology provides a successful means for integrating designer, planner, and contractor knowledge into a unified system. It demonstrates how data generated by each participant in the facility engineering process contributes to the overall system definition.

The CIPROS object-oriented, interactive, knowledge-based system implements the presented methodology and delivers proof of concept of its usefulness and practicality. Its sophisticated discrete-event simulation engine takes advantages of data stored in the various knowledge bases. The tight integration of design and specification data, project planning, and process simulation achieved in CIPROS, illustrates that discrete-event simulation can and should be an integral part of construction management tools.

7 REFERENCES

Franz, M. 1989. Object-oriented Programming Featuring Actor. Scott Foresman and Co., Glenview, IL.

Halpin, D.W., Riggs, L.S. 1992. Planning and Analysis of Construction Operations. Wiley-Interscience, New York, NY.

Whitewater Group 1990. ACTOR User's Manual Vols. 1 and 2. Evanston, IL.