

USING DESIGN DRAWINGS AND PROJECT PLANS TO CONSTRUCT DISCRETE-EVENT SIMULATION NETWORKS

Odeh, Abdalla M., Tommelein, Iris D., and Carr, Robert I.

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

ABSTRACT

CIPROS* is knowledge-based simulation system that enables its users to relate project design drawings and specifications to process networks and associated resources. Each process is defined in terms of actions, corresponding project components, availability and attributes of resources, a network of operations and queues, and operation functions. CIPROS infers process precedence from user-defined action dependencies. The CIPROS knowledge base contains process networks and operation functions corresponding to different construction techniques and methods. In addition to the knowledge base and a user interface, CIPROS includes a generic discrete-event simulator. Project simulation produces statistical reports for management review, instrumental to assessing the quality of a construction plan and identifying areas for improvement. The system thus augments construction plans with knowledge pertaining to the interaction and characteristics of construction resources and models the uncertainty inherent in construction operations. Integrating construction plans with resource information and probabilistic operation durations is key to building executable construction plans.

KEY WORDS: discrete-event simulation, knowledge-based systems, project planning, process planning, integration, object-oriented programming, construction

1. INTRODUCTION

This paper introduces an object-oriented simulation model, called CIPROS*, that integrates project- and process-level planning. Construction project planning involves the mapping of facility design drawings and specifications into construction techniques and methods required for its construction. The use of general design features and common construction techniques and methods in automated plan construction has been limited to artificial intelligence (AI) plan-generating tools such as PLANEX¹⁵, BUILDER⁶, OARPLAN⁷, and SIPEC¹⁰. With the exception of PLANEX, these tools plan at the activity level, and do not consider the timing nor resource consumption of activities. Moreover, the uncertain nature of activity durations, resource characterization, interactions, and the resulting imbalances and loss of productivity at the construction process or operational levels are not addressed by such planning tools. These characteristics dominate process-level planning models, where simulation applies.

Discrete-event simulation has been used extensively to model repetitive construction processes. These models are decision-support systems that imitate the behavior of real-world systems over time and generate statistical data that provides users with insights into resource constraints and interactions. In construction, CYCLONE⁸ is the best-known discrete-event simulation system. CYCLONE's powerful modeling capabilities are based on a very small set of primitives, yet the system lacks capabilities for resource characterization and tracking. Some of CYCLONE's restricted expressiveness has been overcome by RESQUE⁵'s characterization of primitive resource and ability of resource tracking. This new model, like CYCLONE, fails to exploit design features and control mechanisms required for project plan simulation.

Discrete-event simulation has also been used to model uncertain durations of construction project activities. MUD^{2,3} modeled projects of related activities sharing resources and weather. DYNASTRAT^{4,11} extended MUD to include costs and allocation of resources.

* knowledge-based Construction Integrated PROcess and PROject Simulation.

CIPROS' contribution lies in providing a knowledge-based architecture for integrating project and process planning with discrete-event simulation; the model does not handle uncertain durations of project-level activities. CIPROS integrates process and project planning by considering: (1) the facility defined by general design drawings and specifications; (2) processes representing common construction techniques and methods; (3) the uncertain nature of construction operations; and (4) the availability, interaction, and characteristics of construction resources. Its users define construction plans by relating project design drawings and specifications to construction processes and resources. CIPROS then simulates the project construction for a user-defined number of replications and produces statistical reports. These reports provide data on process-level resource utilization and form a basis for comparing alternative construction plans.

2. CIPROS MODEL

CIPROS is an object-oriented simulation model that enables its users to formalize and test construction plan by exploiting knowledge about common facility design features and construction techniques and methods. The CIPROS modeling blocks, shown in Figure 1, consist of a hierarchy of specialized classes, a knowledge base of common construction techniques and methods, input, a simulation engine, and simulation output. The input blocks are entered by the user and define the facility design drawings and specifications and the construction plan.

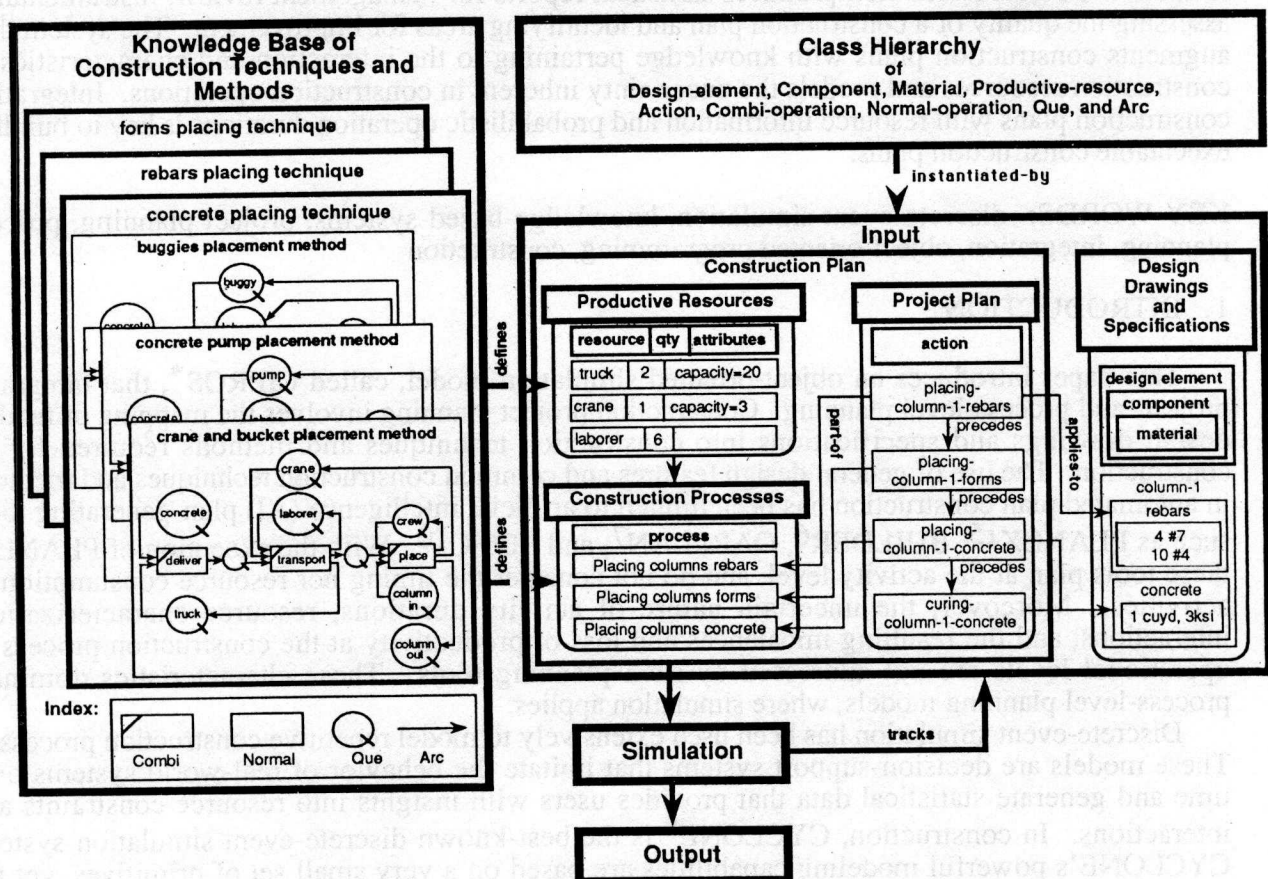


Figure 1: CIPROS Architecture

The construction plan consists of the project plan, construction processes, and productive resources. The user encodes the project plan by defining processing requirements of components and their precedence. These requirements are instances of actions included in the class hierarchy such as *placing-concrete* and *placing-rebars* component actions.

A process is an aggregate of actions that apply to components of the same type and have a common construction technique and method. Construction techniques and methods are presented by modular process charts and operation functions. The process chart is a network of operations,

queues, and arcs that depict entity states and flows in the process. Operation functions complement the process chart and define entity tasks and interactions, and control entity flows in the process. Process action precedence must be defined to fit the requirements of the project plan. This precedence controls material flow between processes. Processes may share queues of productive resources, as may be the case for user-defined instances of classes in the model's resource hierarchy.

The model contains a knowledge base of common methods associated with generic construction techniques such as the *crane and bucket placement method* of the *concrete placing technique* shown in Figure 1. The user may select process techniques and methods from those defined in the knowledge base or define new ones by relating instances of operation, queue, and arc classes provided in the class hierarchy.

CIPROS simulates the project construction plan for a user-defined number of replications. The simulator tracks the state of project components and action precedence throughout simulation and produces statistical reports suitable for assessing the plan's quality and identifying areas of improvement. The following subsections give more detail on the input, knowledge base of construction techniques and methods, simulation, and the simulation output reports.

2.1 Design Drawings and Specifications

The characterization of project components is a key ingredient of the CIPROS architecture. A specific facility's design drawings and specifications are entered by a system user as instances of specialized classes provided in the class hierarchy, in a taxonomy of entities representing the project components. A *project component* can be a design element, a design element component, or a material entity. *Design elements* represent design features of the structural, architectural, mechanical, and electrical systems depicted by the design drawings. A design element is an aggregate of components and a component is an aggregate of material entities of the same type. More precisely, a user must classify material entities that make up the design element based on their type and required processing, and relate them to the design element components with a part-of relationship. Figure 2 shows that the *concrete-component-1* and *rebar-component-1* are part-of *column-1*, and that *rebar-1* and *rebar-2* are part-of the *rebar-component-1*.

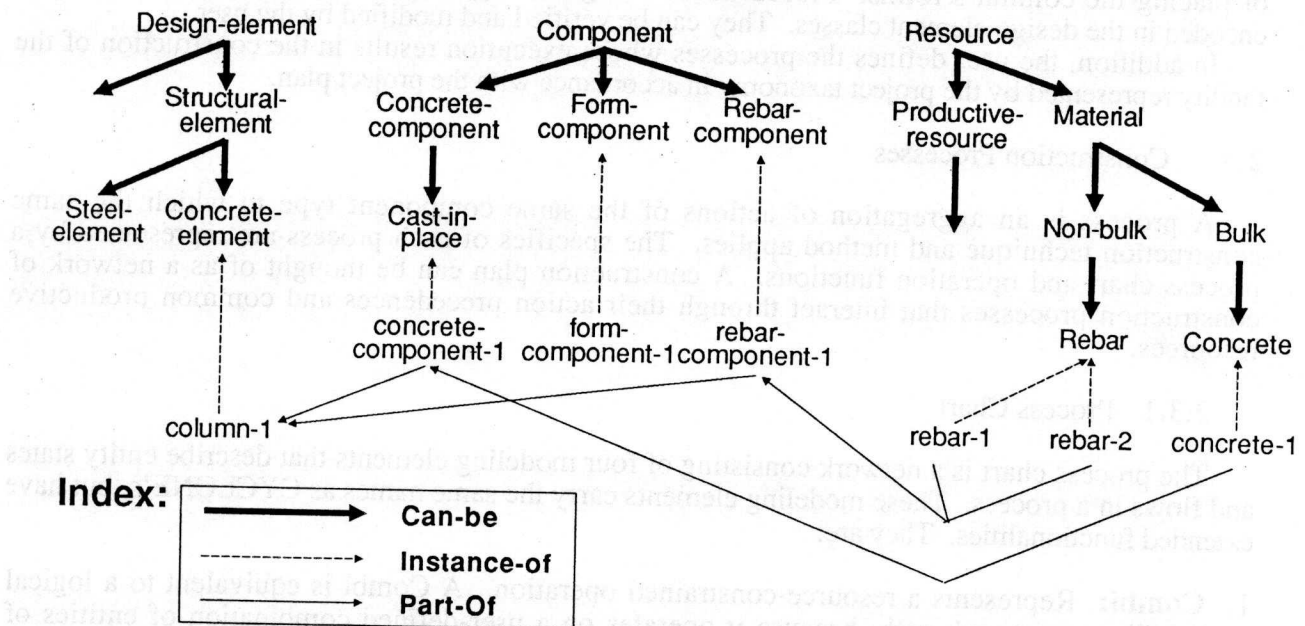


Figure 2: Project Taxonomy

Materials are classified as *bulk* and *non-bulk*. Instances of bulk materials are divisible, non-assignable entities. In the process of *placing columns concrete*, for example, the concrete loaded in a bucket constitutes one entity that can be split and placed in any of the involved columns. Instances of non-bulk materials are indivisible, assignable entities, such as windows and doors.

Typically, each design element has unique non-bulk materials, so each non-bulk material entity is assigned to one design element.

Components are classified by the type of materials of which they are composed and their processing requirements. *Concrete-component*, *form-component*, and *rebar-component* are examples of components classified by material (Figure 2). *Cast-in-place* and *pre-cast* (not shown) specialize the class *concrete-component* based on processing requirements. Embedded knowledge in component classes includes material type, set of possible actions, and action precedence among components of the same design element. Each of these are default values that must be confirmed by the user.

Design elements are classified into a general class and specialized subclasses based on their components. For example, the *structural-element* class is divided into *concrete-element* and *steel-element* (Figure 2). User-defined attributes, such as the design element's shape and dimensions, are used by the model to infer the number or quantity of component materials. Embedded knowledge in design element classes includes a set of components that are prompted to the user for possible selection upon instantiation.

2.2 Project Plan

The user encodes the project plan by defining component requirements, component actions, and action precedence. Component requirements define *temporary components*. These components are required for construction but are not defined in the design drawings and specifications of a design element. The component actions define the processing requirements of a component, such as procuring, fabricating, and placing a rebar component. Figure 3 shows that the concrete component of *column-1* requires *forms*. The actions *placing* and *curing* apply to the *concrete* component of *column-1*. These are encoded in the *cast-in-place-concrete* class and are prompted to the user for selection upon instantiation of this class.

Precedence relationships among component actions define technical dependencies or desired sequencing. An action precedence defines conditions on components that must be met before the action can be applied. That is, it defines the set of actions whose completion triggers the start of the dependent action. For example, the action of placing a column's concrete is preceded by the action of placing the column's forms. Precedences among a design element's component actions are encoded in the design element classes. They can be verified and modified by the user.

In addition, the user defines the processes whose execution results in the construction of the facility represented by the project taxonomy in accordance with the project plan.

2.3 Construction Processes

A process is an aggregation of actions of the same component type to which the same construction technique and method applies. The specifics of each process are represented by a process chart and operation functions. A construction plan can be thought of as a network of construction processes that interact through their action precedences and common productive resources.

2.3.1 Process Chart

The process chart is a network consisting of four modeling elements that describe entity states and flows in a process. These modeling elements carry the same names as CYCLONE's, but have extended functionalities. They are:

1. **Combi:** Represents a resource-constrained operation. 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.
2. **Normal:** Represents an unconstrained operation. 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.

3. **Queue or Que:** Represents a state in which entities wait in a process. Delays occur when entities at a Que wait for entities from other Ques to arrive to start a Combi operation. Therefore, Ques must precede Combis, and Combis can only follow Ques.
4. **Arc:** Represents an entity flow path.

Each operation (a Combi or a Normal) and Que in the process network is given a unique identification number. In the case where Combi successors of a Que compete for its entities, the Combi of higher number gets priority.

The chart depicts the static structure of the process. It is complemented by operation functions that define the dynamic nature of the process.

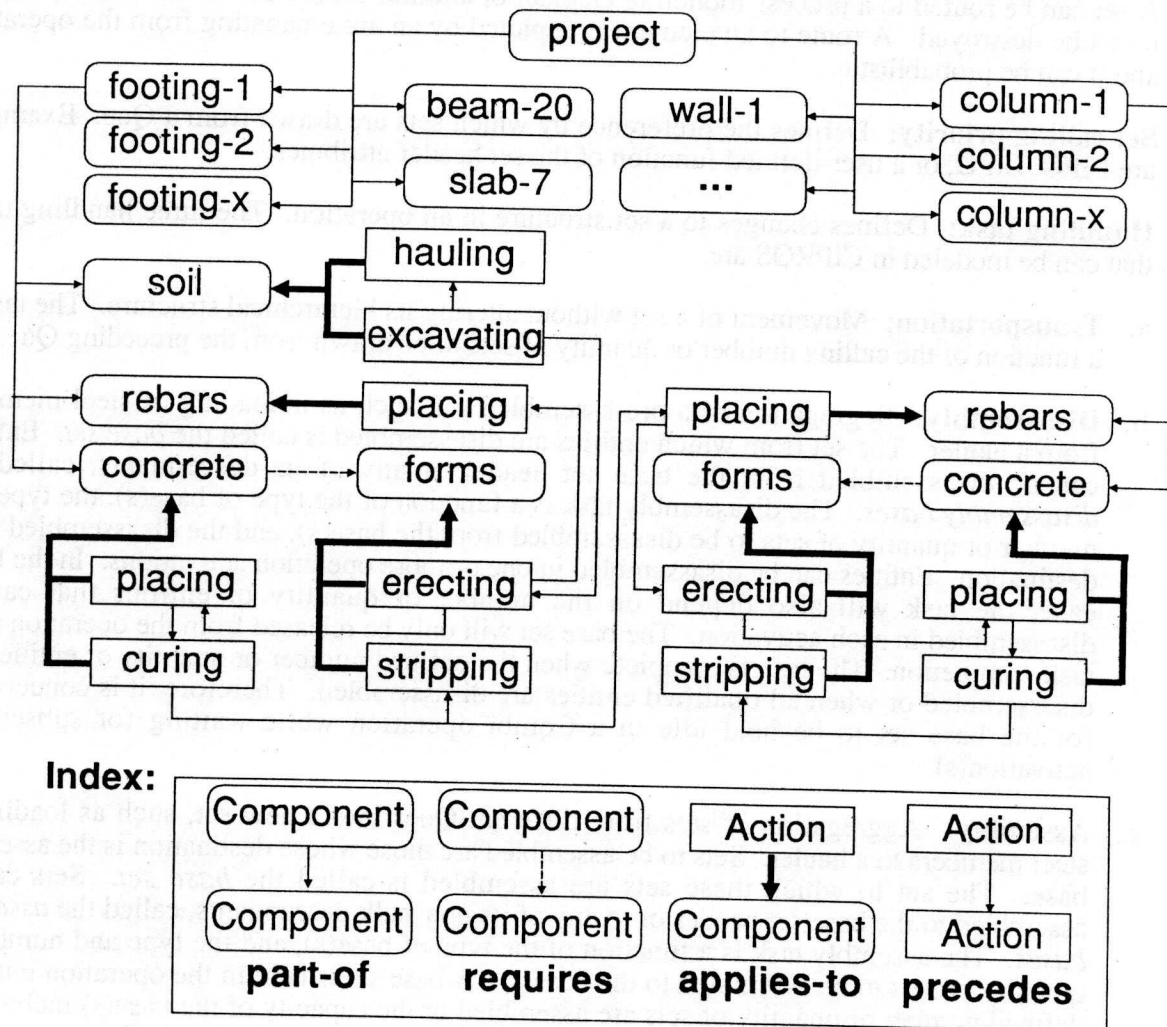


Figure 3: Project Plan

2.3.2 Operation Functions

Entities flow in simple or compound forms through the simulation network. We define a *set* 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 represents 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.

User-defined operation functions specify entity interactions and control set flows in simulation. They allow the user to model different process strategies without altering the process network. Resource strategies are generally related to resource use or resource routing. An example of a

strategy is "use largest available hauler" in a loading operation. In the case of a Normal operation, these functions are attributes of the operation and thus apply to all sets joining the Normal. However, in the case of a Combi operation, where sets from different Ques are involved, the functions (except the operation duration) are attributes of the Arc linking the Combi with a preceding Que. The operation functions in CIPROS are:

1. **Operation duration:** Defines the random variable representing the duration for each operation activation. This variable may depend on attributes of entities involved in the operation.
2. **Set destination:** Defines the route to be followed by a set after its handling in operations. A set can be routed to a process modeling element or another set (i.e., it can be assembled), or it can be destroyed. A route to an element is depicted by an arc emanating from the operation and it can be probabilistic.
3. **Set calling priority:** Defines the preference by which sets are drawn from a Que. Examples are FIFO, LIFO, or a user-defined function of the set header attributes.
4. **Handling task:** Defines changes to a set structure in an operation. The three handling tasks that can be modeled in CIPROS are:
 - a. **Transportation:** Movement of a set without altering its hierarchical structure. The task is a function of the calling number or quantity of sets to be drawn from the preceding Que.
 - b. **Disassembly:** Segregation of a pre-assembled set, such as unloading of steel members from a hauler. The set from which entities are disassembled is called the *base set*. Entities can be disassembled from the base set header or any of its descendants, called the *disassembly bases*. The disassembly task is a function of the type of base(s), the type and number or quantity of sets to be disassembled from the base(s), and the disassembled set's destination. Entities can be disassembled in one or more operation activations. In the latter case, the task will also depend on the number or quantity of entities that can be disassembled in each activation. The base set will only be released from the operation upon task completion. The task is complete when the defined number or quantity of entities are disassembled or when all qualified entities are disassembled. Therefore, it is conceivable for the base set to be held idle in a Combi operation while waiting for subsequent activation(s).
 - c. **Assembly:** Aggregation of sets from different Ques into a new set, such as loading of steel members to a hauler. Sets to be assembled are those whose destination is the assembly base. The set to which these sets are assembled is called the *base set*. Sets can be assembled to the base set header or to any of its non-bulk descendants, called the *assembly bases*. The assembly task is a function of the type of base(s), and the type and number or quantity of sets to be assembled to the base(s). A base set is held in the operation until the defined number or quantity of sets are assembled or the capacity of the base(s) included in the set is exhausted. Capacities of a base and its ancestors are updated whenever a set is assembled-to or disassembled-from it. The quantity that can be assembled to a base is limited to its capacity as well as the capacity of its ancestors in the set hierarchy. Because the number or quantity of sets that are assembled in each operation activation is defined by another task (i.e.; transportation or disassembly), it is conceivable for the base set to be held idle in a Combi operation waiting for subsequent activation(s).
5. **Resource allocation and capturing:** Defines the type and maximum number of transported resources to be allocated or captured by an assembly or disassembly task. Limiting the number of resources that can be allocated or captured by a set's task allows parallel processing of multiple sets within the same operation. On one hand, captured resources are locked into the task and can only be freed upon task completion. On the other hand, allocated resources are freed and routed to their destination at the end of each operation activation.

The construction plan consist of a series of interrelated processes whose execution results in the construction of the facility represented by the project taxonomy. The relationships between processes and process interaction are described next.

2.3.3 Process Interaction

In CIPROS, processes are related through process precedence, common productive resources, and material flow between processes. The process precedence is that of its actions as defined by the project plan.

CIPROS tracks the state and dependencies of each action in the system and schedules its successors accordingly. The actual start of an action in a process is controlled by its module associated with the process. A module is a user-defined aggregate of design elements. The precedences of all actions pertaining to components in a module must be met before these actions can start in the process. For example, the project plan may show that the action of *placing-column-1-concrete* is scheduled for possible start when the action of *placing-column-1-forms* is completed. Assuming that column-1 through column-10 constitute a module in the concrete placing process, all the actions of *placing columns forms* (instantiated by *placing column-1 forms* to *placing column-10 forms*) must be completed before the *placing columns concrete* actions (instantiated by *placing column-1 concrete* to *placing column-10 concrete*) can start in the process.

Each process has its own operations and Ques. Processes may share productive resource Ques based on numbering priority criteria. Material output of one process may be input to another process. By our convention, processes cannot share material Ques. The flow of materials between processes is controlled by action precedence and is carried out by CIPROS' simulator.

Restricted material flows between processes coupled with operation functions provide the modularity in processes that we needed to define the CIPROS knowledge base of construction techniques and methods.

2.4 Construction Techniques and Methods

We have formalized commonly recurring sequences of construction operations as *processes*. For example, concrete delivery, transport, and placement operations can be identified for the process of *placing <component> concrete*, regardless of the resources involved. The operations and their sequencing in the process are basically the same, for example for first-level columns and second-level columns.

We have catalogued processes in a knowledge base of generic construction techniques and methods. *Techniques* define processes' static structures, while *methods* encompass the selection, allocation, and interaction of processes' productive resources. Techniques are represented by process networks of operations, queues, and arcs such as the *concrete placing technique*, *rebars placing technique*, and *forms placing technique*, shown in Figure 1. For each of these techniques, the knowledge base includes processes representing different construction methods such as the *crane and buckets* -, *concrete pump* -, and *buggies placement method* of the *concrete placing technique*, also shown in Figure 1. The process requirement of *laborer* and the assigned tasks depend on the selected concrete placement method.

These process networks have input queues with uninstantiated resources. By choosing a construction method from those enlisted, CIPROS prompts the user with an input dialog for specifying the number and attributes of productive resources to be allocated to the process.

2.5 Simulation and Output Reports

CIPROS simulates the project construction plan for a user-defined number of replicates. The simulator tracks the state of project components and action precedence throughout simulation and produces the following types of statistical reports for each process:

1. **Process statistics:** the start and finish times of the process and each of its actions.
2. **Operation statistics:** (a) the first activation time, (b) the last activation time, (c) the number of activations, and (d) the duration average and standard deviation.

3. **Que statistics:** According to user's preference, Que statistics can be based on the number or quantity of entities. These statistics include: (a) the average and standard deviation of waiting time, (b) the average and standard deviation of Que size, (c) the maximum and minimum size, (d) the total number or quantity of entities that have entered and left the Que, and (e) the time-weighted distribution of the Que size.

These output reports are not only instrumental in measuring the quality of a construction plan; they help identify areas for improvement of an existing plan.

3. CIPROS IMPLEMENTATION

The CIPROS architecture is generic; its modeling concepts apply to a variety of construction facilities, techniques, and methods. We are not attempting to represent the enormous variety in facility design features, materials, resources, and construction techniques and methods. Instead, we want to deliver proof-of-concept that our architecture is appropriate for the task we have set out to model. Accordingly, we have narrowed the scope of CIPROS and focus on the construction of structural systems for high-rise buildings. The class hierarchy of project components and resources, and the knowledge base of construction techniques and methods included in the model reflect this limitation.

The model is currently in the development stage. It is being implemented in a PC WindowsTM environment using the ACTOR¹ object-oriented programming language. ACTOR was chosen for its portability, execution speed, and ability to produce executable files. The CIPROS simulation engine, class hierarchy, and the corresponding input environment have been implemented. The input environment is user friendly with menu-driven windows, input dialogs that guide users throughout the model development stage, and scanning facilities that allow easy viewing and updating of input parameters so that users can easily develop alternative plans. We are now encoding the knowledge base of construction methods and techniques.

4. CIPROS CAPABILITIES AND LIMITATIONS

CIPROS takes advantage of the expressiveness and expandability of its implementation environment. Its capabilities of resource characterization, resource tracking, and control allow for realistic modeling of complex systems. CIPROS' strength lies in its ability to take into account specific information about a project, enabling its users to map such information to construction processes with common productive resources, and simulating these processes in accordance with the user-defined project plan.

The CIPROS object hierarchy and knowledge base are expandable. Depending on the requirements of the project at hand, the user may use the model's existing classes or create new specialized classes. A newly-defined class can be a subclasses of an existing class, and can inherit variables and methods from it. Thus, users need not develop new classes from scratch. Furthermore, the user may define and add new processes representing construction techniques and methods to the knowledge base. Once a new class or process is defined, it becomes part of the model, and can be instantiated repeatedly.

CIPROS processes are modular. Except for input queues of common productive resources, each process has its own network. Consequently, users can model different entity interactions, and resource utilization strategies by altering input parameters without effecting the process network. Process modularity coupled with the knowledge base of construction techniques and methods enables the user to model alternative construction plans with relative ease.

As we have not completed the implementation of CIPROS, we have not engaged in experimentation with the system, so we cannot assess system performance in terms of computation speed. Neither can we comment at this time on the storage requirements for and the completeness and appropriateness of the class hierarchies and the process knowledge base.

Some conceptual limitations we are aware of are those inherent to simulation models stemming from uncertain and uncontrollable future events scheduling. The present model cannot take decisions based on events external to the operation. An entity with an incomplete task along with its captured resources could be held idle in an operation for an long period of time due to improper resource strategies of other operations in the process. In such a case, the operation has neither knowledge nor control over when the entities needed for task completion will be available. This is

in contrast with real-life systems, in which managers can coordinate and control entity interactions and flows of different processes.

A resource utilization strategy that defines completion of an assembly task, especially those involving non-bulk entities, might lead to unanticipated and undesired situations. The general issues are: What must be done to complete an almost-complete task when no more resources are available? Should a task be terminated when its available indivisible, non-bulk entities are less in quantity to what is required for task completion? For how long should an idle entity with an incomplete assembly task wait before the task should be terminated? When can resources captured in an incomplete task be freed? The solution to such issues is based on knowledge regarding the future flow of materials to the operation, and such knowledge is generally not available in the simulation model. One way of dealing with such questions is to allow freeing of captured resources or terminating assembly tasks if a user-defined time has elapsed since the task entity became idle. Alternatively, the assembly task may be terminated if a user-defined percentage of task completion has been achieved at the end of operation activation.

5. FUTURE RESEARCH

Future research can enhance the model's capabilities to include cost and preemption functions, automated input-data entry, and additional decision support. Automating some tasks now delegated to users is possible by linking CIPROS to other programs. For example, an interface could translate CAD design drawings to object-oriented frames, and an AI planner could generate a project plan by inferring component actions, action precedence, and aggregating actions into processes. Other knowledge-based systems can then reason about appropriate techniques and methods for construction processes and resource allocation. Additional decision support can be provided by expert systems^{12,13,14} modeling complex management decisions. They may remedy some of the problems encountered during simulation. Users will then interact with CIPROS to deal with the interesting issues, such as those regarding resource assignment and use strategies.

6. CONCLUSIONS

CIPROS extends beyond current planning and simulation models in its ability to integrate project and process planning. The model encodes design features and common construction techniques and methods in the form of a specialized class hierarchy and a knowledge base, that can be instantiated and selected by users. In addition to defining project design drawings and specifications, the user must provide the anticipated construction plans. At the process level, CIPROS enables its users to model the uncertain nature of construction operations and resource availability and interaction, and the sharing of construction resources by processes. CIPROS' discrete-event simulation system tracks productive resources and the completion of construction activities associated with design elements. Comprehensive output reports summarize simulation data at the process level. This data can then be compiled to be meaningful at the project level, where alternative project plans and their effects on construction performance can be analyzed.

7. REFERENCES

1. *ACTOR User's Manual*, Volumes 1 and 2, The Whitewater Group, Evanston, IL, 1990.
2. Carr, R. I., "Simulation of Construction Project Duration," *Journal of the Construction Division*, ASCE, 105 (2) 117-128, June 1979.
3. Carr, R. I. and Meyer, W.L., "Planning Construction for Repetitive Building Units," *Journal of the Construction Division*, ASCE, 106 (2) 185-194, June 1980.
4. Carr, R. I., Chang, D. Y., and Morua, E., "Knowledge-based Simulation of Construction," *Proceedings 4th International Symposium on Robotics and Artificial Intelligence in Building Construction*, Building Research Station, Technion, Haifa, Israel, June 1987, pp. 168-189.
5. Chang, D. Y., Hoque, Md. S., "A knowledge-based simulation system for construction process planning," *Proceedings of the 6th Symposium on Automation and Robotics in Construction*, 348-355, San Francisco, CA, June 1989.
6. Cherneff, J., Logcher, R., and Sriram, D., "Integrating CAD with Construction-Schedule Generation," *Journal of the Construction Division*, ASCE, 5 (1) 64-84, Jan 1991.

7. Darwiche, A., Levitt, R. E., and Hayes-Roth, B., "OARPLAN: Generating Project Plans by Reasoning about Objects, Actions, and Resources," *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, 2 (3) 1989.
8. Halpin, D.W., Woodhead, R.W., *Design of Construction and Process Operations*, John Wiley and Sons, New York, 1976.
9. Ito, K., Ueno, Y., Levitt, R. E., and Darwiche, A., "Linking Knowledge-Based Systems to CAD Design Data with an Object-Oriented Building Product Model," Center for Integrated Facility Engineering, Technical report No.012, Stanford University, Stanford, CA, August 1989.
10. Kartam, N. A. and Levitt, R. E., "Intelligent Planning of Construction Projects," *Journal of Computing in Civil Engineering*, ASCE, 4 (2) 155-176, April 1990.
11. Morua Padilla, E., Carr, R.I., "Resources strategies for dynamic project management," *Journal of the Construction Division*, ASCE, (forthcoming, probably June 1991).
12. O'Keefe, R., "Simulation and expert systems -A taxonomy and some examples," *Simulation* 46 (1) 10-16, January 1986.
13. Reddy, R., "Epistemology of knowledge based simulation," *Simulation* 49 (4) 162-166, April 1987.
14. Shannon, R. E., Mayer, R., and Adelsberger, H. H., "Expert systems and simulation," *Simulation* 44 (6) 275-284, June 1985.
15. Zozaya-Gorostiza, C., Hendrickson, C., and Rehak, D. R., *Knowledge-Based Planning for Construction and Manufacturing*, London: Academic Press Inc., 1989.