*Proceedings, Construction Congress V, ASCE*
*Stuart D. Anderson,ed*

# State-Based Probabilistic Scheduling Using STROBOSCOPE's CPM Add-On

**by Julio C. Martinez[1] and Photios G. Ioannou[2]**

## Abstract

Critical path scheduling is a well-developed tool for construction planning at the project level. A substantial amount of research has focused on adding probabilistic capabilities to the technique. Simulation-based methods are almost universally accepted as the most effective because of their modeling versatility and power. Some scheduling models based on simulation recognize correlation between activity durations and allow the probabilistic selection of paths within a network. Very few techniques, however, allow activity duration and sequencing to be defined in terms of the dynamic information that becomes available as a project evolves.

This paper presents a program for probabilistic CPM scheduling designed as an add-on to the STROBOSCOPE simulation system. The add-on allows the duration of an activity to be specified as a formula that depends on any scheduling information that may be available when the activity starts. This includes the actual start date and duration of activities already started as well as the dates and floats for activities yet to start. This capability and the ability to combine probabilistic scheduling with construction process simulation make the CPM add-on and the STROBOSCOPE simulation system one of the most powerful project planning tools available today.

## Introduction

The Critical Path Method (CPM) is used to plan and control most modern construction projects. A CPM network represents the various activities that comprise a project and the precedence relationships between them. It can be

[1] Assistant Professor of Civil Engineering, Virginia Tech, Blacksburg, VA 24061-0105

[2] Associate Professor of Civil & Env. Engineering, University of Michigan, Ann Arbor, MI 48105

analyzed to determine the criticality and float of activities; the level of resources needed during each day of construction; and the dates at which important milestones will be achieved.

A CPM network is not necessarily an accurate representation of an actual construction plan. It assumes that activities have a fixed duration and that this duration is known at the beginning of the project. It also assumes that precedence relationships are fixed and absolutely necessary (i.e., all predecessors must finish before the successor can start).

As a project is built, the CPM network is updated to reflect the actual dates of the completed activities and better estimates of the duration of future activities. Often, the logic of the network changes to represent a revised construction plan.

The effectiveness of planning at the project level can be substantially enhanced if the plan can be formulated and evaluated more realistically ahead of time. This includes recognizing the uncertainty in the duration of activities as a function of the dynamic state of the project. Program Evaluation and Review Technique (PERT) was the first step in this direction. PERT can model probabilistic durations but its results are subject to merge-event bias and cannot model correlation between activity durations. Other tools typically overcome the merge-event bias problem by using Monte-Carlo simulation to model a CPM schedule (Van Slyke 1963). GERT (Phillips & Hog 1976) can additionally model uncertainty in the precedence of activities. VERT (Moeller & Digman 1981), MUD (Carr 1971) and DYNASTRAT (Morua Padilla 1986) can model correlation between the durations of activities and to a limited extent recognize the state of the project in progress. None of the existing tools, however, have the necessary flexibility and power to model uncertainty in the duration of activities as a true function of the state of the project, nor can they model the underlying process-level operations through concurrent simulation.

The CPM add-on for STROBOSCOPE presented here is capable of modeling activities with probabilistic durations that depend on the state of the project and other concurrent process-level simulations. This add-on serves as an example of how STROBOSCOPE can be used as a development vehicle for special-purpose simulators and more sophisticated project-level simulation tools.

## STROBOSCOPE

STROBOSCOPE is an acronym for STate and ResOurce Based Simulation of COnstruction ProcEsses. It is a simulation programming language based on activity cycle diagrams and the activity-scanning paradigm and was designed specifically for modeling construction operations. The STROBOSCOPE language is fully described in (Martinez 1996). Example applications that illustrate the language can be found in (Ioannou & Martinez 1996) and (Martinez & Ioannou 1995). Several other examples can be found in publications that appear in the references.

STROBOSCOPE can be extended via add-ons written in a high-level compiled language such as FORTRAN or C++. STROBOSCOPE can also be controlled via OLE Automation by other applications. A STROBOSCOPE add-on is a 32-bit MS Windows Dynamic Link Library that extends the STROBOSCOPE language with new statements, functions, and variables. The code within the add-on can call back into the simulation engine using STROBOSCOPE's Application Programming Interface (API) described in (Martinez 1996). Add-ons are loaded into STROBOSCOPE with the LOADADDON statement:

```
LOADADDON "C:\Program Files\Strobos\Add-Ons\CpmAddOn.dll";
```

Once an add-on is loaded, its statements, functions and variables can be used as if they were a standard part of the language.

## The CPM Add-On

When the CPM add-on is loaded, it provides the statements shown in Table 1.

**Table 1 - Statements registered by the CPM Add-On**

| Statements | Arguments | Function |
|---|---|---|
| CPMACTIVITY | 1) Activity Name<br>2) Duration Expression | Defines a CPM Activity with the provided name and duration sampled using the provided expression. |
| PRECEDENCE | 1) Predecessor Name<br>2) Successor Name | Indicates that the successor cannot start until the predecessor is finished. |
| CPMREPLICATE | 1) Number of Replications | Simulates the CPM network the number of times indicated and produces a report as described in the CPMREPORT statement. |
| DOCPM | | Performs a single forward and backward CPM pass. |
| CPMREPORT | | Prints the 90% Confidence Interval on project duration and a report showing the average duration, average early and late dates, average floats and criticality for each CPM Activity. |

Of particular interest is the second argument to the CPMACTIVITY statement. The argument is an expression that can include functions that sample from probability distributions and that can reference variables that access the state of project or of any concurrent process simulation.

Assume, for example, that a concurrent process simulation produces a dynamic variable named *RainfallLst48hrs* (in centimeters). This variable could then be used to define the probabilistic duration of an activity as follows:

```
CPMACTIVITY ExcSt14To17
Pertpg[0.9,1.1,1.4]+RainfallLst48hrs*0.3;
```

The duration of the `ExcSt14To17` activity defined above would be determined by sampling from a Beta distribution with 5th percentile 0.9 days, Mode 1.1 days, and 95th percentile 1.4 days; 0.3 days for each centimeter of rainfall accumulated during the 48 hours prior to the start of the activity would be added to the sampled value. The actual expression used to define the duration of the activity could be as complex as necessary, calling any number of functions and variables. The parameters of functions that sample from probability distributions can also include expressions or variables.

The CPM add-on automatically defines variables that access the state of the project. These variables are listed in Table 2.

Given these variables, CPM activities could be defined with statements such as (see Figure 1):

```
CPMACTIVITY CounterEq Pert[1,2,4];
VARIABLE CntrEqRemDur Max[0,CounterEq.EarlyFinish-
SimTime];
CPMACTIVITY FloorCvrngs Pert[2,4+CntrEqRemDur/2,8];
```

The first of the above statements defines the activity *CounterEq* with an optimistic duration of 1 day, most likely duration of 2 days, and pessimistic duration of 4 days. The second statement defines a function that returns the remaining duration of *CounterEq*. The last statement defines *FloorCvrngs* with an optimistic duration of 2 days, most likely duration of 4 days plus half a day for each day of remaining duration in *CounterEq*, and pessimistic duration of 8 days. An example such as this could be used to show that *CounterEq* interferes with *FloorCvrngs* due to a shared resource (e.g., they use the same crew or work space).

## Example

The following example, shown in Figure 1, is adapted from a classic CPM textbook (Harris 1978). It is the network for the construction of a fast-food outlet and includes the activities for the installation of counter equipment and floor coverings that were used as examples above.

**Table 2 - Variables defined by the CPM Add-On**

| Variable Prototype | Value Returned |
|---|---|
| *CPMActivity*.Critical | TRUE if *CPMActivity* is critical in the current run. |
| *CPMActivity*.Duration | The duration used for *CPMActivity* in the current run. |
| *CPMActivity*.EarlyFinish | The early finish of *CPMActivity* in the current run. |
| *CPMActivity*.EarlyStart | The early start of *CPMActivity* in the current run. |
| *CPMActivity*.Finished | TRUE if *CPMActivity* has finished in the current run. |
| *CPMActivity*.FreeFloat | The free float of *CPMActivity* in the current run. |
| *CPMActivity*.GoingOn | TRUE if *CPMActivity* is going on in the current run. |
| *CPMActivity*.LateFinish | The late finish of *CPMActivity* in the current run. |
| *CPMActivity*.LateStart | The late start of *CPMActivity* in the current run. |
| *CPMActivity*.Started | TRUE if *CPMActivity* has started in the current run. |
| *CPMActivity*.TotalFloat | The total float of *CPMActivity* in the current run. |

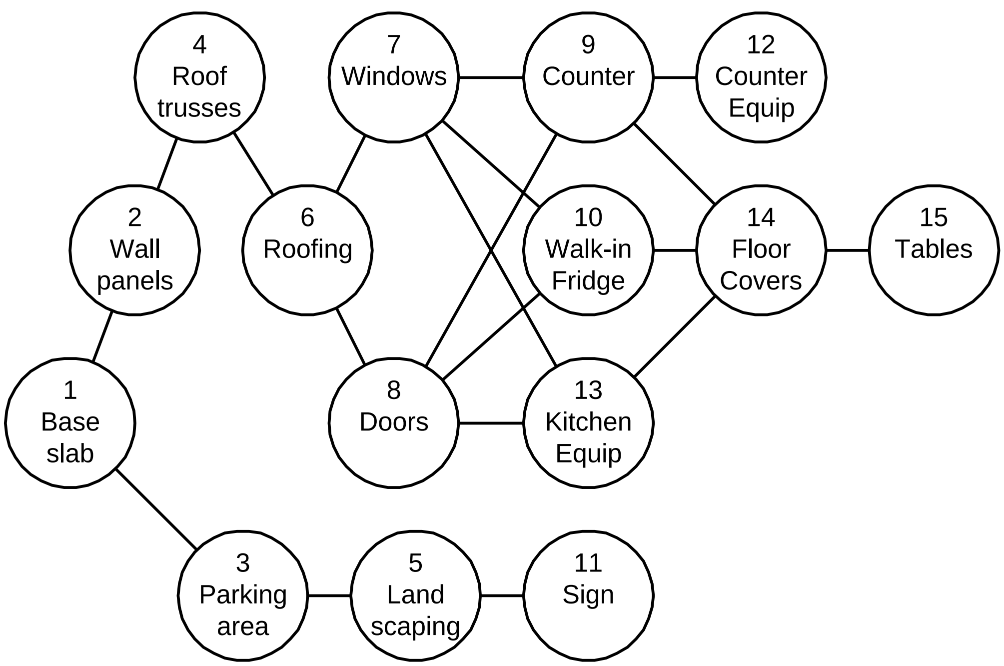


**Figure 1 - CPM Network for fast-food outlet construction**

The entire source-code for this model is shown below. Refer to the source-code for the probability distributions of the activity durations.

```
LOADADDON “CpmAddOn.dll”;

CPMACTIVITY BaseSlab Pert[3,6,12];
CPMACTIVITY WallPanels Pert[4,6,9];
CPMACTIVITY ParkingArea Pert[6,15,20];
CPMACTIVITY RoofTrusses Pert[1,2,5];
CPMACTIVITY Landscaping Pert[3,5,10];
CPMACTIVITY Roofing Pert[1,3,5];
CPMACTIVITY Windows Pert[1,2,4];
CPMACTIVITY Doors Pert[1,2,4];
CPMACTIVITY Counter Pert[3,8,10];
CPMACTIVITY WIFridge Pert[2,5,8];
CPMACTIVITY Sign Pert[3,4,6];
CPMACTIVITY CounterEq Pert[1,2,4];
VARIABLE    CntrEqRemDur Max[0,CounterEq.EarlyFinish-SimTime];
CPMACTIVITY KitchenEq Pert[4,10,15];
CPMACTIVITY FloorCvrngs Pert[2,4+CntrEqRemDur/2,8];
CPMACTIVITY Furnishings Pert[5,10,15];

PRECEDENCE BaseSlab WallPanels;
PRECEDENCE BaseSlab ParkingArea;
PRECEDENCE WallPanels RoofTrusses;
PRECEDENCE ParkingArea Landscaping;
PRECEDENCE RoofTrusses Roofing;
PRECEDENCE Roofing Windows;
PRECEDENCE Roofing Doors;
PRECEDENCE Windows Counter;
PRECEDENCE Doors Counter;
PRECEDENCE Windows WIFridge;
PRECEDENCE Doors WIFridge;
PRECEDENCE Landscaping Sign;
PRECEDENCE Counter CounterEq;
PRECEDENCE Windows KitchenEq;
PRECEDENCE Doors KitchenEq;
PRECEDENCE WIFridge FloorCvrngs;
PRECEDENCE Counter FloorCvrngs;
PRECEDENCE KitchenEq FloorCvrngs;
PRECEDENCE FloorCvrngs Furnishings;

CPMREPLICATE 1000;
```

The source-code used above consists almost entirely of statements defined by the CPM add-on. The only standard STROBOSCOPE statements used are VARIABLE and LOADADDON. STROBOSCOPE, running this code on a 200 MHz Pentium Pro under Windows NT 4.0, produces the following output:

```
STROBOSCOPE Model ProbCPM4.STR (1551032608)

Number of replications performed : 1000
Average Project Duration         : 44.95
90% CI on project duration       : [44.78,45.11]
```

| CPM Activity | Time | ESD | LSD | EFD | LFD | FF | TF | %Critic |
|---|---|---|---|---|---|---|---|---|
| Doors | 2.17 | 17.90 | 18.19 | 20.07 | 20.36 | 0.29 | 0.29 | 49.00% |
| ParkingArea | 14.35 | 6.42 | 20.94 | 20.77 | 35.29 | -0.00 | 14.52 | 0.00% |
| Roofing | 2.99 | 14.91 | 14.91 | 17.90 | 17.90 | 0.00 | -0.00 | 100.00% |
| FloorCvrngs | 4.56 | 30.39 | 30.39 | 34.96 | 34.96 | 0.00 | -0.00 | 100.00% |
| Counter | 7.48 | 20.36 | 22.91 | 27.84 | 30.39 | -0.00 | 2.55 | 13.80% |
| Windows | 2.17 | 17.90 | 18.19 | 20.07 | 20.36 | 0.29 | 0.29 | 51.00% |
| RoofTrusses | 2.35 | 12.55 | 12.55 | 14.91 | 14.91 | -0.00 | -0.00 | 100.00% |
| BaseSlab | 6.42 | 0.00 | -0.00 | 6.42 | 6.42 | 0.00 | -0.00 | 100.00% |
| WIFridge | 4.98 | 20.36 | 25.42 | 25.33 | 30.39 | 5.06 | 5.06 | 0.10% |
| Furnishings | 9.99 | 34.96 | 34.96 | 44.95 | 44.95 | 0.00 | -0.00 | 100.00% |
| KitchenEq | 9.89 | 20.36 | 20.50 | 30.25 | 30.39 | 0.15 | 0.15 | 86.10% |
| CounterEq | 2.16 | 27.84 | 42.78 | 30.00 | 44.95 | 14.94 | 14.94 | 0.00% |
| Sign | 4.15 | 26.28 | 40.79 | 30.43 | 44.95 | 14.52 | 14.52 | 0.00% |
| WallPanels | 6.13 | 6.42 | 6.42 | 12.55 | 12.55 | -0.00 | -0.00 | 100.00% |
| Landscaping | 5.50 | 20.77 | 35.29 | 26.28 | 40.79 | 0.00 | 14.52 | 0.00% |

```
-------------------------------------
Execution    Time = 3.578 seconds
```

Detailed statistics for each of the values shown in the report can be obtained by appending the REPORT statement to the code (this produces the standard STROBOSCOPE report). The detailed statistics would include the minimum, maximum, average, and standard deviation for each value. Notice that for all activities except *FloorCvrngs* the average duration is very close to the long term mean as given by the Pert approximation (a+4*M+b)/6. Without the interference from *CounterEq*, the average duration of *FloorCvrngs* would have been (2+4*4+8)/6=4.33. The average duration after 1000 iterations was 4.56, which reflects the extra time required when it overlaps with *CounterEq*.

The CPM add-on and its source code in C++ are included with STROBOSCOPE. Beyond its utility, the CPM add-on serves to illustrate STROBOSCOPE's powerful API and the ease with which it can be used to extend the language. Thus, it is a solid starting point for a more sophisticated project-level analysis tool. Despite this, the CPM add-on is very useful and perhaps the most attractive probabilistic scheduling tool available, especially for teaching and research. In fact, a recently completed Ph.D. dissertation at the University of California at Berkeley used the CPM add-on as the primary tool for validating a heuristic approach for the evaluation of the impact of correlation between activities (Wang 1996).

## Acknowledgments

The authors wish to thank the Horace H. Rackham Pre-Doctoral Fellowship Program at the University of Michigan and the National Science Foundation (Grant CMS-9415105) for supporting portions of the work presented here. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation or of the Rackham School of Graduate Studies at the University of Michigan.

## References

Van Slyke, R. M. 1963. "Monte-Carlo Methods and the PERT Problem," Operations Research.

Phillips, D. T. and Hogg, G. L. 1976. "Stochastic Network Analysis with Resource Constraints, Cost Parameters and Queing Capabilities Using GERT Methodologies," Computers and Industrial Engineering.

Moeller, G. L. and Digman, L. A. 1981. "Operations Planning With VERT," Operations Research.

Carr, R. I. 1979. "Simulation of Construction Project Duration," Journal of the Construction Division, ASCE.

Morua Padilla, E. 1986. "Resource Strategies for Dynamic Construction Project Management," Ph.D. Dissertation, Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI.

Martinez, J. C. 1996. "STROBOSCOPE: State and Resource Based Simulation of Construction Processes," Ph.D. Dissertation, Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI.

Ioannou, P. G. and Martinez, J. C. 1996. "Comparison of Construction Alternatives Using Matched Simulation Experiments," Journal of Construction Engineering and Management, ASCE, (112) 3, New York, NY.

Martinez, J. C. and Ioannou, P. G. 1995. "Advantages of the Activity Scanning Approach in the Modeling of Complex Construction Processes," Proceedings of the 1995 Winter Simulation Conference, Society for Computer Simulation, San Diego, CA.

Harris, R. B. 1978. "Precedence and Arrow Networking Techniques for Construction," John Wiley & Sons, New York, NY.