

SIMULATION OF COMPLEX CONSTRUCTION PROCESSES

Photios G. Ioannou
 Julio C. Martinez

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

ABSTRACT

Eight successively refined simulation models for the earthmoving operations involved in the construction of a dam provide the foundation for illustrating the ease and effectiveness of modeling complex construction processes by using STROBOSCOPE, a modern simulation system based on a programming language that follows the activity-scanning paradigm. This approach and the use of characterized resources in a stochastic environment make the typical engineering calculations for heavy equipment performance relatively easy to implement and significantly more realistic and accurate.

1 INTRODUCTION

Traditionally there have been three major impediments to the wide-spread use of simulation to model complex construction processes. The first was that available simulation languages employed the process interaction paradigm and were developed to suit the needs of manufacturing and not construction. This difficulty was partially addressed by the advent of construction-oriented simulators, such as CYCLONE, RESQUE, COOPS, etc. All these systems have been based on activity-scanning, a paradigm suitable for modeling construction processes because of its use of the notion of activities and thus the similarity to CPM. However, these systems are not languages and thus lack the power to model complex construction processes (Martinez and Ioannou, 1995).

Today, the first two problems have been largely resolved with the advent of STROBOSCOPE (STate and Resource Based Simulation of CONstruction ProcEsses), a system developed by the authors that combines activity-scanning with a powerful programming language. However, the third difficulty still remains. Construction is a conservative industry with very high stakes that has not yet been sold on the ease and effectiveness of simulation modeling or the accuracy of its results. Thus, model development, validation and

verification are extremely important, especially when the real decision makers do not have any prior experience or confidence in the capabilities of simulation modeling.

This paper describes how a construction simulation model can be designed and implemented at eight levels of increasing complexity and accuracy and then animated to verify that it is indeed a faithful representation of reality. The example used is that of earthmoving for the construction of a dam. This example goes through a process of successive model refinement that serves both as a modeling case study and as an illustration of the fact that it is relatively easy to model the performance of heavy construction equipment by performing complex engineering calculations under stochastic conditions that exceed the capabilities of traditional methods. The final model is then verified through animation.

2 EARTHMOVING FOR DAM CONSTRUCTION

An earthmoving contractor needs to move 200,000 m³ of soil for the construction of a dam. Soil will be loaded at the borrow area using two types of heavy loaders. The soil is loaded into two types of heavy trucks that travel to the dam embankment, dump, and return empty. The density of the loose soil moved is 1690 kg/m³. A total of 3 loaders and 11 trucks will be used on this job. Tables 1 and 2 show the number and characteristics of the loaders and trucks by type and model.

Table 1: Loader Data

Model	CAT 936E	CAT 950E
Machines available	2	1
Bucket Size (m ³)	2	2.4
Soil loaded in loader's bucket, S (m ³)	Bucket Size * $N[1,0.05]$	Bucket Size * $N[1,0.04]$
Time to load a scoop of soil in truck (min)	0.40+0.05 S	0.43+0.035 S
Cost (\$/hr)	45	62

Table 2: Truck Data

Model	CAT 769C	CAT 773B
Machines available	8	3
Capacity (m ³)	23.6	34.1
Mass when empty (kg)	31,178	39,396
Flywheel Power (kW)	336	485
Transmission efficiency	0.82	0.84
Retarding power (kW)	434	526
Max. speed (km/hr)	75	62
Actual speed (km/hr)	Calc. speed *N[1,0.03]	Calc. speed *N[1,0.02]
Dumping time (min)	1	1.12
Maneuvering time at loading area (min)	0.6	0.72
Cost (\$/hr)	117	175

Trucks at the borrow area queue up and are loaded in FIFO order. Each truck is loaded by one loader. Before loading can begin, a truck must leave the waiting queue and maneuver into position. The maneuvering time depends on the truck as shown in Table 2. After this time passes, the loader starts putting scoops of material into the truck.

The amount of soil in each scoop, S , is random and depends on the size of the loader's bucket as shown in Table 1. The time for a loader to place one scoop of material into a truck has a fixed and a variable component. The latter is proportional to the amount of soil actually loaded, S , as shown in Table 1.

Soil continues to be loaded for as long as the amount of empty space in the truck exceeds one half of the loader's bucket size. Thus, the last scoop brings the loaded material to at least within half a loader's bucket from the truck's capacity. This means that the total material loaded may occasionally exceed the nominal capacity of the truck.

When a truck is loaded it starts hauling the soil to the dam embankment. The haul/return route from the borrow area to the dam crosses a small river. The road from the borrow to the river is 2.3 km long. It descends at a 4% grade and has a rolling resistance of 7%. The road from the river to the dam is 1.7 km long. It ascends at a 6% grade and has a rolling resistance of 4%.

The speed of a truck on any given route is determined on the basis of engineering calculations using the data in Table 2 (instead of using rimpull or retarding curves) and the slope and rolling resistance characteristics of the road. Trucks are assumed to travel at constant speed (acceleration and deceleration times are disregarded). The calculated speed is adjusted for random variability as shown in Table 2.

Trucks cannot dump material at the dam without the help of a spotter. There is only one spotter and she can only pay attention to one truck at a time.

3 BRIDGE DECISION PROBLEM

The existing bridge cannot support heavy trucks and the contractor must provide one or more new temporary low-cost bridge(s) for his trucks to cross the river.

There are two types of temporary bridges to choose from. A Type I bridge costs \$95,000 and can hold only one truck at a time. Moreover, trucks must pass very slowly (it takes 1.75 minutes to cross over). A Type II bridge costs \$165,000 and can hold up to six trucks at the same time. Because the bridge is narrow, however, these trucks must be traveling in the same direction, separated at least 0.30 minutes from each other. The time to cross a Type II bridge is also 1.75 minutes.

The earthmoving contractor must determine the type and number of bridges that gives the lowest project cost and the corresponding total cost and time to complete the project.

4 BRIDGE CROSSING POLICIES

Based on the contractor's past experience with one-way bridges, it appears that the best crossing policy is to let all trucks going in one direction cross the bridge without reversing the flow of traffic for as long as there is a queue. When no more trucks need to cross the bridge in the current direction and the bridge is clear, the trucks in the other direction are allowed to cross in the same manner. If the bridge is empty and no trucks are waiting to cross the bridge (on either side), then the next direction of travel is established by the first truck to arrive at the bridge.

It must be pointed out that under very heavy traffic it is customary to impose a limit on the time available for crossing the bridge in any one direction to allow traffic to cross in the other direction. When this time limit is reached (e.g., three minutes), no more trucks going in that direction are allowed to enter the bridge. When all the trucks finish crossing, then the trucks at the other end are allowed to pass in the same manner. It is easy to show that this situation does not apply to the problem at hand because the truck cycle-time is long (relative to the time to cross the bridge) and the number of trucks is limited.

5 MODELING METHODOLOGY

Successively refined simulation models for this problem are presented below. These models were developed using the STROBOSCOPE system.

STROBOSCOPE is a general-purpose discrete-event simulation system based on the activity-scanning paradigm that is particularly suited for modeling complex construction processes. STROBOSCOPE models are networks of nodes and links (similar to activity cycle diagrams). The heart of the system is a simulation programming language that provides full access to dynamic variables, the dynamic properties of resources, and the state of the simulation. It allows modeling stochastic resource production, utilization, and consumption; smart resource allocations; characterization of resources created at runtime by combining other resources; and dynamic decisions regarding the sequence of activities. STROBOSCOPE is a 32-bit program for Windows NT and Windows 95.

A complete description of STROBOSCOPE appears in (Martinez 1996). Example applications can be found in (Martinez, Ioannou, and Carr 1994; Martinez and Ioannou 1994, 1995; Ioannou and Martinez 1995, 1996a, 1996b). The STROBOSCOPE program, its documentation, and several examples are available via anonymous ftp from "grader.engin.umich.edu".

The networks for the simulation models shown in this paper were developed using STROBOSCOPE's graphical user interface (GUI). This GUI is built on top of a commercial drawing program (Visio 3 or 4) that has been enhanced by the authors with specially programmed "drag-and-drop" graphics elements (nodes and links) that serve as primitives for constructing STROBOSCOPE networks interactively. A custom dynamic-link-library (DLL) provides a plethora of tabbed dialog-boxes for data input, extensive error-checking, and the ability to compile and communicate to STROBOSCOPE all the information entered. This is accomplished either "live" via OLE automation or by saving ready-to-use simulation input files to disk.

6 EARTHMOVING SIMULATION MODELS

A series of eight simulation models for the earthmoving problem are presented below. At first, these models will ignore the bridge decision problem and concentrate on how to model the earthmoving operation as accurately as possible using generic resources that have no properties or attributes (as in CYCLONE). We later introduce characterized loader and truck resources to allow for engineering calculations and finally we extend that model to solve the bridge decision problem.

6.1 Model 1—Simple Earthmoving

The network for the first model is shown in Figure 1. It shows that the *combi* (conditional) activity *Load* cannot start unless there are resources in the queues *LoadersWait* and *TrucksWait* that precede it. When *Load*

starts, it removes one resource from each of these two queues (a loader and a truck) via the links LD1 and TR1. At the end of *Load*, the loader is returned to the queue *LoadersWait* through link LD2 and the truck passes through link TR2 to the *Haul* activity which is then allowed to start. At the end of the *Haul* activity, the truck is passed to the *Dump* activity and then to the *Return* activity. Finally, it returns to the queue *TrucksWait*.

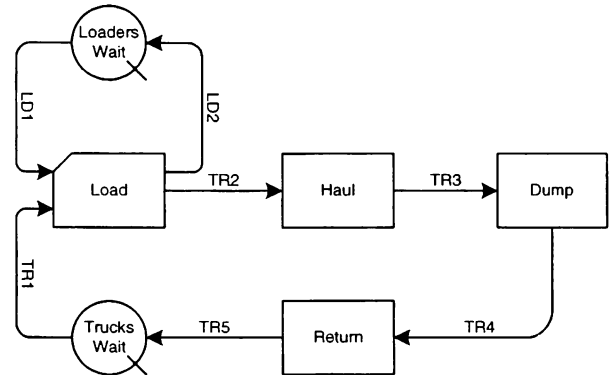


Figure 1: STROBOSCOPE Network for Dam Construction Simulation Models 1 and 2

To have a complete model, we need (in addition to the network) to define the problem parameters, resource types, and provide simulation control statements.

The following define the problem parameters:

```
VARIABLE NumLoaders 3;
VARIABLE NumTrucks 11;
VARIABLE LdrHrCst 50; /($/hr)
VARIABLE HlrHrCst 135; /($/hr)
```

We also need to define *Loader* and *Truck* as generic resource types and initialize the queues *LoadersWait* and *TrucksWait*:

```
GENTYPE Loader;
GENTYPE Truck;
INIT LoadersWait NumLoaders;
INIT TrucksWait NumTrucks;
```

In our first model, the duration of the activity *Load* is the total time it takes the average loader to load an average truck completely. (In reality, this time depends on the specific properties of the loader and truck actually used.) We shall assume that the duration of *Load* follows a Normal distribution with a mean of 7 and a standard deviation of 1.2 minutes.

```
DURATION Load Normal[7,1.2];
```

These parameters for the Normal distribution were chosen so that the results from the first model will be close to those of the more detailed models that follow.

The Haul activity represents the total time it takes to haul soil to the dump location and should thus have a duration that depends on the specific properties of the truck, the road, and the material hauled. Here we will again assume a Normal distribution with appropriately chosen parameters for the average truck. The duration of the *Return* activity is specified in a similar manner:

```
DURATION Haul Normal[11.66,0.3];
DURATION Return Normal[7.68,0.33];
```

Based on the problem data, the average truck has a capacity of 26.67 m³ and moving 200,000 m³ requires a total of 7,500 dumps. Thus, simulation should run until the total number of instances of *Dump* reach this limit.

```
SIMULATEUNTIL Dump.TotInst>=7500;
```

The total cost of the project depends on the time to move the soil and the hourly equipment costs:

```
VARIABLE TotalFleetCost
  (NumLoaders*LdrHrCst+
   NumTrucks*HlrHrCst)*SimTime/60;
DISPLAY "Total Working Time : "
  SimTime/60 " Hours";
DISPLAY "Total Equipment Cost: "
  TotalFleetCost " ($)";
```

6.2 Model 2—Hard-wired Resource Drawing Loop

As a first refinement to the above model, the duration of activity *Load* will be redefined as the sum of twelve IID random variables, each representing the time to load an average scoop of material (i.e., 2.22 m³) into the truck. Thus, each of these distributions depends only on the properties of the loader (i.e., its “size”) and not on the truck being loaded. The capacity of the average truck (i.e., 26.67 m³) is reflected in the number of random variables that must be added up (i.e., twelve).

```
DURATION Load
  'Normal[1,0.44]+Normal[1,0.44]+
   Normal[1,0.44]+Normal[1,0.44]+
   Normal[1,0.44]+Normal[1,0.44]+
   Normal[1,0.44]+Normal[1,0.44]+
   Normal[1,0.44]+Normal[1,0.44]+
   Normal[1,0.44]+Normal[1,0.44]+
   Normal[1,0.44]+Normal[1,0.44]';
```

As expected, the results of this model are by design equivalent to those of the first model.

6.3 Model 3—Resource Drawing Amount

We will now model soil as another resource and allow it to flow through the network via links SL1-SL4, as shown

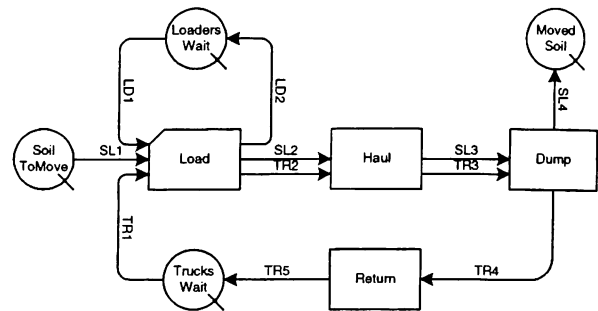


Figure 2: STROBOSCOPE Network for Dam Construction Simulation Models 3, 4, and 5

in Figure 2. To do so we define *Soil* as a generic resource and initialize the *SoilToMove* queue.

```
VARIABLE AmountOfSoil 200000; / m3
GENTYPE Soil;
INIT SoilToMove AmountOfSoil;
```

Each instance of the *Load* activity should not simply remove (draw) one unit of *Soil* from the *SoilToMove* queue (one is the default amount). Instead, the amount of soil drawn must equal the size of the average truck and must be specified explicitly by defining the *draw amount* attribute of link SL1:

```
DRAWAMT SL1 26.67;/ Truck capacity in m3
```

Simulation should now continue and stop only when the *SoilToMove* queue becomes empty.

```
SIMULATE;
```

The *actual* amount of soil drawn by the combi *Load* travels through the network via links SL1-SL4. Thus, every time *Dump* finishes, it *automatically* releases a truckload of soil (26.67 m³) into the queue *MovedSoil*.

6.4 Model 4—Resource Drawing Loop

The network in Figure 2 can be refined by taking advantage of the resource drawing attributes of link SL1. These attributes allow us to model the duration of activity *Load*, as well as the drawing of resources out of the *SoilToMove* queue, the way it really happens, i.e., one scoop at a time. This is best shown by example:

Drawing through link SL1 should continue until the number of “draws” (i.e., scoops) reaches 12:

```
DRAWUNTIL SL1 SL1.nDraws>=12;
```

On average, each of these scoops removes 2.22 m³ of soil from *SoilToMove* and transfers it to *Load*:

```
DRAWAMT SL1 2.22;/ m3 per scoop
```

The distribution of the time it takes to load a scoop is defined as follows:

```
DRAWDUR SL1 Normal[0.58,0.33];
```

We can now define the duration of activity *Load* to be equal to the sum of all (i.e., twelve) sampled draw durations through link SL1:

```
DURATION Load SL1.SumDrawDur;
```

The elegance of this model should now be obvious. The number of random variables that must be added to give the duration of a single instance of activity *Load* is no longer hard-wired in a long formula that adds up twelve random variables. Instead, the drawing process is treated like a loop that in this case is repeated twelve times (i.e., it is controlled by the number 12 in the DRAWUNTIL statement).

The draw duration that is sampled in each loop does not make simulated time go by. Only activities can do that. Instead, it provides a mechanism by which we can sample the duration associated with each draw and collect statistics over the entire set of samples (sum, average, standard deviation, minimum, and maximum). The number of samples that are collected in this manner (in this case, twelve) depends on the number of draws needed to complete an operation (in this case, to load an entire truck scoop by scoop). These statistics (e.g., the sum of draw durations) can then be used to establish the duration of a real activity (such as *Load*) that makes simulated time go by.

6.5 Model 5—Soil Loaded and Dumped is Random

Now that soil is drawn from the *SoilToMove* queue scoop by scoop, it is possible to refine the network in Figure 2 further so that the amount of soil drawn in each scoop is random. This will also make the total amount of soil hauled and dumped by each truck different, and presents us with the opportunity to make the duration of the *Haul* activity sensitive to the truck's total weight.

This statement defines the amount of soil drawn through SL1 link as a random variable:

```
DRAWAMT SL1 2.22*Normal[1,0.05]; / m3
```

The duration of activity *Haul* can now be adjusted based on the amount of soil actually loaded in the truck relative to the average truck capacity (i.e., 26.67 m³):

```
DURATION Haul 'Normal[11.66,0.3]*
    Haul.Soil.Count/(26.67)';
```

This statement makes a dynamic calculation by using a *predefined resource variable* to access the state of the simulation at run time. The predefined variable is *Haul.Soil.Count*. It returns the amount of *Soil* that is currently held by the instance of activity *Haul* whose

duration we are about to compute.

6.6 Model 6—Characterized Resources

A more realistic model requires that loaders and trucks be modeled as characterized resources. Characterized resources must belong to a resource type (e.g., *Loader*) that defines the *properties* (i.e., static attributes) that all resources of that type must have (e.g., *Power*, *BucketSize*, etc.). Within a type, such as *Loader*, we must define *subtypes*, such as *S936E* and *S950E*, for which we can specify the specific values of these properties. Thus, the characterized types and subtypes for the loaders and trucks in this problem can be defined as follows:

```
CHARTYPE Loader BucketSize
    LoadVariability FixedLdTm
    VarLdTmFact CostPerHr;
SUBTYPE Loader S936E 2 0.05 0.40 0.05 45;
SUBTYPE Loader S950E 2.4 0.04 0.43 0.035
    62;
CHARTYPE Truck Capacity EmptyWgt Power
    Eff RetPower MaxSpeed ...;
SUBTYPE Truck S769C 23.6 31178 336000
    0.82 434000 75/3.6...;
SUBTYPE Truck S773B 34.1 39396 485000
    0.84 526000 62/3.6...;
```

In addition to the above CharType (static) properties, characterized resources can have *SaveProps*, i.e., properties whose values can be assigned and changed during simulation run-time. Thus, individual resources that belong to the same subtype have exactly the same static properties but have *SaveProps* that have different values from one resource to the next. For the problem at hand it is convenient to define truck *SaveProps* to hold the amount of soil loaded into a truck (*AmtLoaded*), and the rolling resistance (*RR*) and grade of the road (*GR*):

```
SAVEPROPS Truck AmtLoaded RR GR;
```

The definition of characterized resource types and subtypes allows modeling the loading process with complete accuracy. As shown by the superscripts on the links LD1 and TR1 in Figure 3, activity *Load* first draws the next truck (via TR1), then the next loader (via LD1) and then the soil (via SL1). Thus, the *Load* instance being created already holds a specific loader and truck when soil is about to be drawn. This means that drawing through SL1 should continue until the amount of soil deposited into *Load* is within 1/2 of the loader's bucket from the truck's capacity ("the" means the loader and truck held by the *Load* instance being created):

```
DRAWUNTIL SL1 'Load.Soil.Count >=
    Load.Truck.Capacity-
    Load.Loader.BucketSize/2';
```

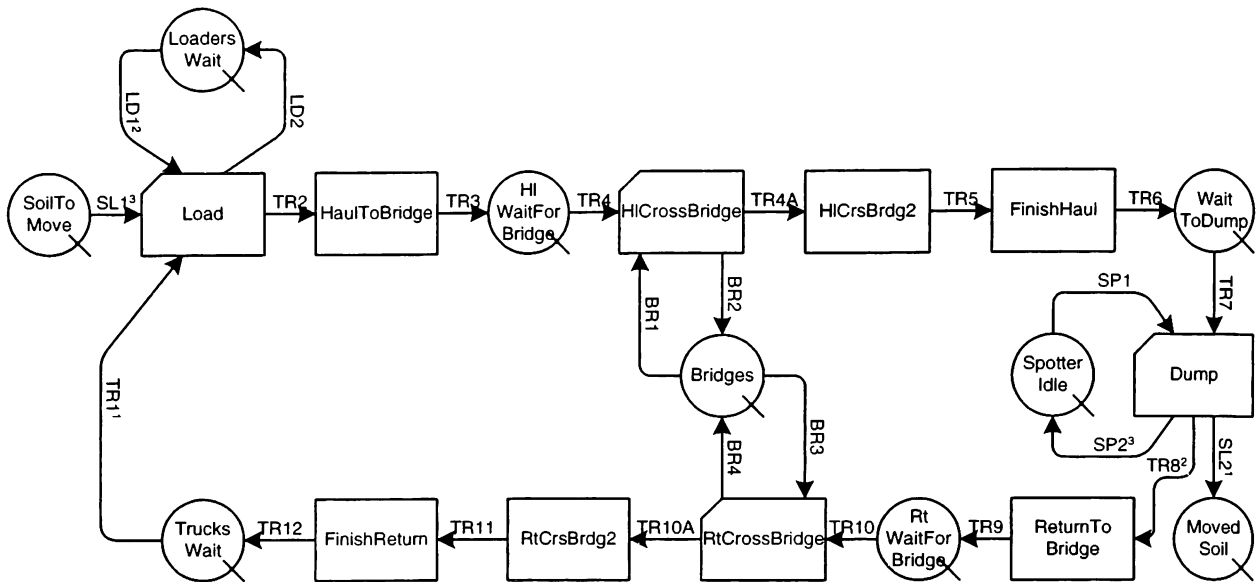


Figure 3: STROBOSCOPE Network for Dam Construction Simulation Model

The amount of soil drawn by each scoop should equal the size of *the* loader's bucket modified by its random load variability:

```
DRAWAMT SL1 'Load.Loader.BucketSize*
Normal[1,Load.Loader.LoadVariability]';
```

The draw duration consists of a fixed and a variable component that depends on *the* amount of soil drawn:

```
DRAWDUR SL1 'Load.Loader.FixedLdTm+
SL1.LastAmtDrawn*
Load.Loader.VarLdTmFact';
```

The duration of *Load* equals the time for *the* truck to maneuver into loading position plus *the* sum of the sampled draw durations:

```
DURATION Load 'Load.Truck.ManeuverTm
+SL1.SumDrawDur';
```

The amount of soil loaded in the truck is stored (for later use) in the truck's *SaveProp AmtLoaded*:

```
ONRELEASE TR2 ASSIGN AmtLoaded
Load.Soil.Count;
```

The duration of *Haul* depends on *AmtLoaded* relative to the truck's capacity:

```
DURATION Haul 'Normal[11.66,0.3]*
Haul.Truck.AmtLoaded/
Haul.Truck.Capacity'
```

Given that each truck has *AmtLoaded* as a *SaveProp*, there is no reason for soil to flow through the network.

Instead, the amount of soil dumped into the *MovedSoil* queue is set equal to *the* truck's *AmtLoaded*:

```
RELEASEAMT SL2 'Dump.Truck.AmtLoaded'
```

6.7 Model 7—Engineering Calculations

The effective grade *EG* is the sum of the road properties stored in *RR* and *GR*:

```
VARPROP Truck EG RR+GR;
```

A truck's effective power depends on whether the effective grade is positive or negative:

```
VARPROP Truck EffectivePow 'EG>=0 ?
Power*Eff : RetPower'; /Watts
```

The force that the engine has to overcome to move up/down the grade (in Newtons) is given by:

```
VARPROP Truck Force '(EmptyWgt
+AmtLoaded*SoilDensity)*Abs[EG]*9.8';
```

The truck's speed (m/sec) equals its effective power over force (but must not exceed its maximum speed):

```
VARPROP Truck Speed
Min[EffectivePow/Force,MaxSpeed];
```

Road properties are assigned to truck properties while it flows through the links (similarly for *TR5*, *TR8*, *TR11*):

```
ONRELEASE TR2 ASSIGN RR 0.07;
ONRELEASE TR2 ASSIGN GR -0.04;
ONRELEASE TR2 ASSIGN AmtLoaded
Load.Soil.Count;
```

The duration of *HaulToBridge* equals distance over speed (converted from seconds to minutes)

```
DURATION HaulToBridge 'DistSourceToBridge
/HaulToBridge.Truck.Speed/60';
```

The duration of *Dump* depends on the type of truck dumping (only one truck dumps at a time):

```
DURATION Dump Dump.Truck.DumpTm;
```

6.8 Model 8—Final Model with Type I or II Bridges

The final network is shown in Figure 3. At first we need to define a few problem parameters:

```
VARIABLE NumBridges 1; / No. of bridges
VARIABLE TimeToCrossBridge 1.75;/minutes
/LeadTime: TypeI:1.75min, TypeII:0.30 min
VARIABLE LeadTime 0.30;
VARIABLE CostOfABridge 165000;/or $95,000
```

At the end of *HCrossBridge* the bridge is made available to trucks going in the same direction. The duration of *HCrossBridge* is (I) 1.75, or (II) 0.30 minutes for proper spacing between Trucks.

```
DURATION HlCrossBridge LeadTime;
DURATION RtCrossBridge LeadTime;
```

The duration of *HlCrsBrdg2* is the remaining time:

```
DURATION HlCrsBrdg2 'TimeToCrossBridge-
LeadTime';
DURATION RtCrsBrdg2 'TimeToCrossBridge-
LeadTime';
```

The direction of traffic on a Type II bridge cannot reverse (i.e., *HCrossBridge* cannot start), even if the bridge is available, if there are trucks finishing a crossing in the other direction (i.e., if *RtCrsBrdg2* has any current instances). This assumes that only 1 Type II Bridge is available:

```
SEMAPHORE HlCrossBridge
!RtCrsBrdg2.CurInst;
SEMAPHORE RtCrossBridge
!HlCrsBrdg2.CurInst;
```

7 CONCLUSION

The results for models 1-7 are shown in Table 3 and the final results in Table 4. As expected, the results for models 1-7 are very close and much less than the final results (they do not include the cost of any bridges). Table 4 shows that using two Type I bridges is the best alternative with a an expected total cost of \$710,000 and a duration of 322 hours. This time is very close to that for models 1-7 so no further improvement is possible.

Table 3: Results of Preliminary Simulation Models

Model	Cost	Time (hr)
1	\$521,954	319
2	\$522,236	319
3	\$522,339	319
4	\$520,400	318
5	\$521,156	319
6	\$517,858	321
7	\$511,180	317

Table 4: Simulation Results for Bridge Selection

No of Bridges	Type I Bridges		Type II Bridges	
	Total Cost	Time	Total Cost	Time
1	\$802,467	439 hr	\$715,433	341 hr
2	\$709,787	322 hr	\$869,101	334 hr
3	\$798,318	318 hr		

To verify that the final model was correct and to illustrate its results to others it was animated using PROOF Animation. A snapshot of the animation is shown in Figure 4.

ACKNOWLEDGMENTS

The development of STROBOSCOPE has been supported by the University of Michigan Horace H. Rackham School of Graduate Studies and the National Science Foundation (Grant No CMS-9415105).

REFERENCES

- Ioannou, P. G., and J.C. Martinez. 1995. Evaluation of alternative construction processes using simulation. In *Proceedings of the 1995 Construction Congress*, San Diego, CA, October 22-26, ASCE.
- Ioannou, P. G., and J.C. Martinez. 1996a. Animation of complex construction simulation models. In *Proceedings, Third Congress on Computing in Civil Engineering*, Anaheim, CA, June 17-19, ASCE.
- Ioannou, P. G., and J.C. Martinez. 1996b. Comparison of construction alternatives using matched simulation experiments. *Journal of Construction Engineering and Management*, ASCE, (122) 3.
- Martinez, J. C., Ioannou, P. G., and R. I. Carr. 1994. State and resource based construction process simulation. In *Proceedings of the First Congress on Computing in Civil Engin.*, ASCE, Washington, DC.
- Martinez, J.C., and P.G. Ioannou. 1994. General purpose simulation with STROBOSCOPE, In *Proceedings*,

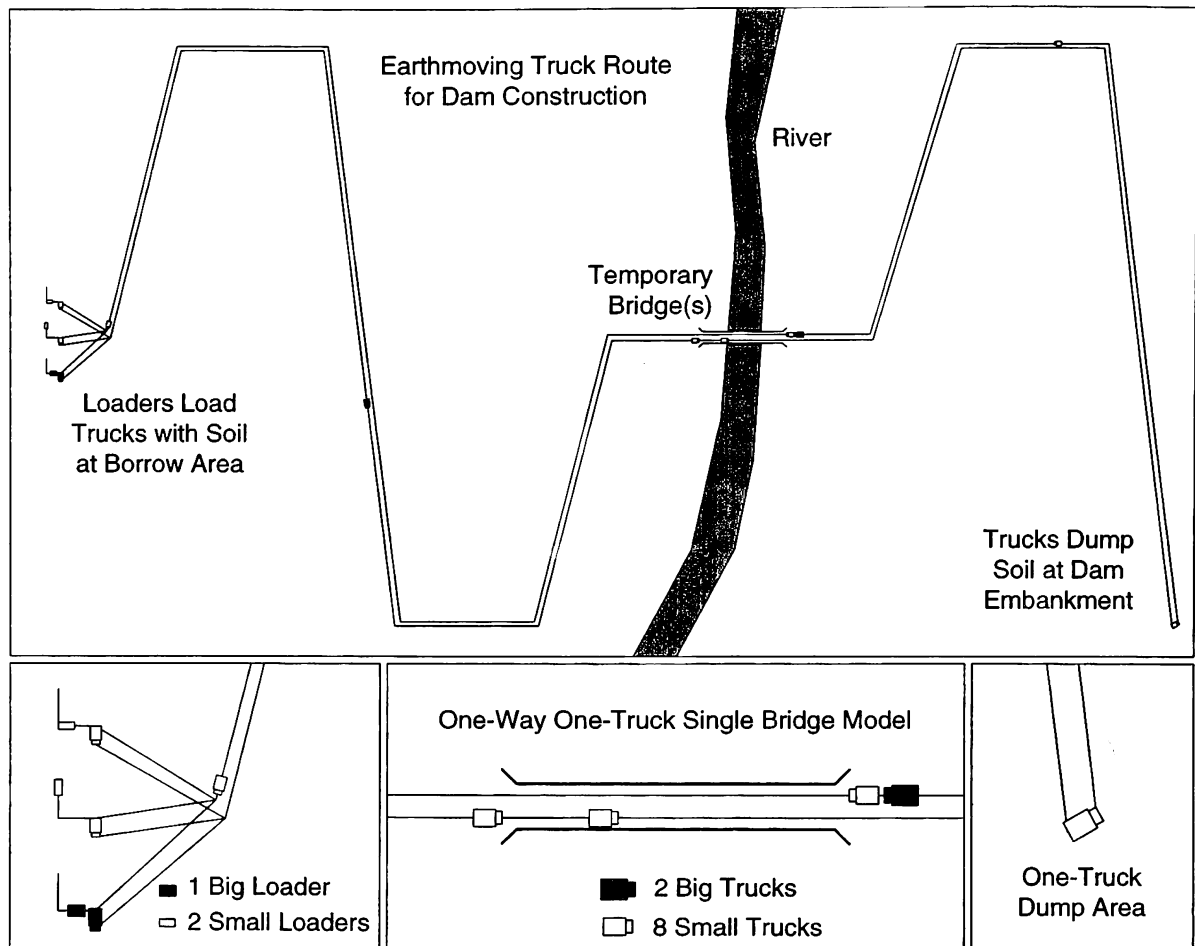


Figure 4: Earthmoving Layout for Dam Construction (PROOF Animation)

1994 Winter Simulation Conference, Orlando FL, December 11-14, IEEE, Piscataway, New Jersey.

Martinez, J.C., and P.G. Ioannou. 1995. Advantages of the activity scanning approach in modeling complex construction processes. In *Proceedings, 1995 Winter Simulation Conference*, Washington, DC, December 3-6, IEEE, Piscataway, New Jersey.

Martinez J.C. 1996. STROBOSCOPE—State- and resource-based simulation of construction processes, Ph.D. thesis, Civil and Environmental Engineering Dept., Univ. of Michigan, Ann Arbor, Michigan.

AUTHOR BIOGRAPHIES

PHOTIOS G. IOANNOU is Associate Professor in the Dept. of Civil and Environ. Engin. at the Univ. of Michigan. He received a Diploma in Civil Engin. from the National Technical Univ., Athens, Greece, in 1979; and a SMCE and Ph.D. from MIT in 1981 and 1984. He has just completed a six-year term as Chairman of the Computing in Construction Technical Committee of the

ASCE. He co-developed the UM-CYCLONE construction process simulation system with R.I. Carr, supervised the design and development of COOPS by L.Y. Liu, and served as chairman of J.C. Martinez's Ph.D. dissertation committee. His research interests are primarily focused on the areas of construction decision support systems and construction process modeling.

JULIO C. MARTINEZ is Post Doctoral Research Fellow in Civil Engineering at the University of Michigan. He designed and implemented the STROBOSCOPE simulation language as part of his Ph.D. dissertation research. He received a Civil Engineer's degree from Universidad Catolica Madre y Maestra (Dominican Republic) in 1986, an MS in Civil Engin. from the University of Nebraska in 1987, and an MSE and a Ph.D. in Construction Engineering and Management from the University of Michigan in 1993 and 1996. His research interests are in computer applications to construction engineering and management.