EZStrobe Model for ARGESIM Benchmark C10 'Dining Philosophers Problem II'

Photios G. Ioannou[1*], Marios C. Papaefthymiou[2], Constantine A. Ioannou[2]

[1]Dept. of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI, USA, *photios@umich.edu
[2]School of Information and Computer Sciences, University of California, Irvine, CA, USA

SNE 36(2), 2026, 73-76, DOI: 10.11128/sne.36.bn10.10772
Submitted: 2025-12-21
Received: 2026-01-30; Accepted: 2026-02-10
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The EZStrobe add-on for the STROBOSCOPE simulation system is used to model the classic "Dining Philosophers" problem, where five philosophers sit around a table in front of five plates of food and alternate between thinking and eating. There are only 5 chopsticks available, and because philosophers need two chopsticks to eat, at most two can be eating at the same time. Thus, activities compete for limited resources. The EZStrobe simulation model is presented, and the reasons for the differences in its results from previous solutions are examined.

Introduction

The *ARGESIM Benchmark C10* [1] depicts five philosophers seated around a table, each with a plate of food in front of them. The philosophers alternate between thinking and eating, which requires the use of two chopsticks. Only five chopsticks are available, positioned between the five plates. Thus, only two philosophers could be eating at the same time. The durations of the thinking and eating activities are such that, although random, they could result in more than one philosopher wanting to start eating at exactly the same simulation time and thus competing for the available chopsticks.

1 EZStrobe - STROBOSCOPE

EZStrobe is an add-on for the STROBOSCOPE discrete-event simulation system [2] that provides a graphical drag-and-drop modelling user interface (GUI) that runs within Microsoft Visio using VBA (Figure 1).

Both EZStrobe and STROBOSCOPE are based on three-phase activity scanning, which is especially suited for cyclic operations, such as those in *ARGESIM Benchmark C10*.

EZStrobe models are simpler and utilize only one predefined type of generic resource named EZs. Simulation models are networks of nodes where resources spend time (such as activities and queues), connected by links that control both when activities can start and the flow of resources (from preceding to succeeding nodes).

An important objective of EZStrobe is that it does not require the user to be proficient in the STROBOSCOPE modelling language. Additionally, the EZStrobe GUI is designed to communicate graphically nearly all the key details of a simulation model network, such as the duration and priority of activities, the initial contents of queues at the start of simulation, the *enough* attributes and *draw amounts* of *drawing* links, and the *release amount* of *releasing* links. Only the global simulation controls, such as the initial seed, the simulation stopping conditions, and the number of replications, are hidden from view. They are accessed in a dialog box that appears when the user right-clicks the Visio model page.

2 EZStrobe Simulation Model

The EZStrobe simulation model network is shown in Figure 1. The rectangles with clipped corners are the conditional activities (*combis*) "Think*k*" (blue) and "Eat*k*" (green) for each philosopher *k*.

A key modelling concept in EZStrobe (and STROBOSCOPE) is that a combi activity can only be preceded by queues, and queues can only precede combi activities. Each combi activity cannot start and create an instance unless it has the required resources, which are provided by the directly preceding queues.

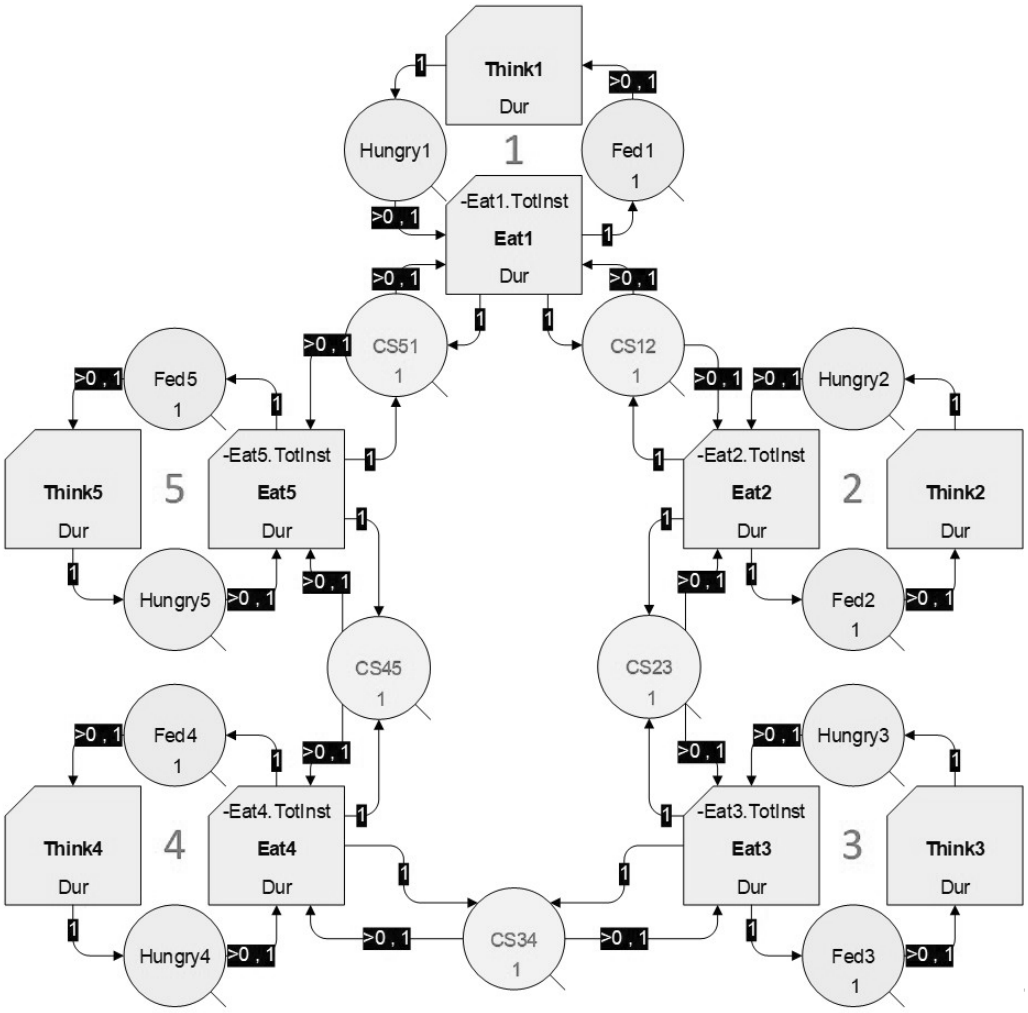

Figure 1: EZStrobe Simulation Model Network for ARGESIM Benchmark C10 'Dining Philosophers'.

The startup of each combi activity is managed by its *drawing links*, which are the links from each preceding queue to the combi, and through which a new combi instance draws resources when it starts. To ensure that *all* preceding queues will have *enough* resources to support a new combi activity instance, each *drawing link* has two attributes (defined below): the '*enough*' and the '*draw amount*', which are displayed in white letters inside a black box on top of each drawing link.

The *enough* attribute is a logical condition about the contents of the preceding queue. The default *enough* expression is ">0", which checks that the contents of the queue are greater than zero.

This means that the preceding queue has enough resources to support the start of a new instance of the succeeding combi activity when the queue is not empty.

A combi activity can start a new instance and draw resources *only* when the *enough* attributes for *all* its incoming drawing links return the logical value *true*.

The *draw amount* is the number of resources moved from the queue to the combi each time the combi starts and creates an instance. The default draw amount is 1.

Similarly, each *releasing link* from a combi to a queue displays its *release amount* with white letters in a black box.

The *release amount* is the number of resources released to the succeeding queue each time a combi instance finishes. The default *release amount* is 1.

Each of the five "Fed*k*" queues represents a philosopher who has just eaten and is ready to start thinking. At the start of the simulation, they are initialized with one resource, as shown at the bottom of each queue.

Each of the five "CS*ij*" queues represents the chopstick between philosophers *i* and *j*. They are also initialized with one resource each.

In this model, the *enough* attributes for all drawing links are the default ">0". Thus, at time zero, all five "Think*k*" can start because their preceding queues "Fed*k*" are not empty (they are initialized with 1 resource). Later, a combi "Eat*k*" will only be able to start when *all three* of its preceding queues "Hungry*k*", "CS*jk*", and "CS*kl*" are not empty.

Each time the simulation clock advances, *all* combi activities are first *sorted* according to their *priority* attribute and then examined one by one to determine if they can start, in that order. If all the *enough* attributes of all drawing links for the examined combi return the value *true*, then the activity creates an instance and removes *draw amount*, i.e., "1" resource from each preceding queue as shown on the drawing link.

Thus, in an EZStrobe simulation model with the default *enough* link attributes, each combi activity "Eat*k*" first checks that both chopsticks it needs are available, and then creates an instance and draws both chopsticks, one after the other. Thus, it is impossible for only one chopstick to be drawn and for deadlock to occur, unless the model is specifically contrived (which serves no practical purpose).

This model sets the priorities of the "Eat*k*" activities by the dynamic expression "-Eat*k*.TotInst" (shown at the top of the "Eat*k*" combi activity nodes in Figure 1). Larger values of the activity priority attribute result in higher priority. Thus, top priority is given to the combi activity "Eat*k*" that has had the *least total instances* up to now (to make eating equitable). (Priorities could also reflect the philosophers' current waiting-to-eat time.)

The durations of activities "Think*k*" and "Eat*k*" follow discrete uniform distributions in the interval [1, 10] that were implemented by the variable *Dur*, defined by the expression: 'Int[Uniform[0,10]] + 1'.

This ensured that more than one "Eat*k*" activity would want to start at exactly the same simulation time and thus compete for the adjacent chopsticks.

In this model, whenever adjacent "Eat*k*" activities could start at the same simulation time, the one with the fewest number of instances up to now will be the one examined first and be able to start a new instance and take both adjacent chopsticks.

3 Results

The model shown in Figure 1 was run for 100,000 time units (this run took 0.3 seconds). A subset of the default statistics reported by STROBOSCOPE is shown in Tables 1 and 2.

As expected, the statistics for "Thinking" and "Eating" are close to the moments $m = 5.5$ and $\sigma = 2.87$ for the discrete uniform distribution in the interval [1, 10].

However, the Avg=4.19 and StDev=4.39 for the times when the philosophers are "Waiting" to eat are less than half of those in [3] and [4], which report values closer to Avg=11.5 and StDev=8.

State	P1	P2	P3	P4	P5	All
Thinking	5.53	5.51	5.41	5.51	5.54	5.50
	2.87	2.88	2.84	2.86	2.84	2.86
Waiting	4.13	4.19	4.27	4.18	4.17	4.19
	4.32	4.45	4.54	4.28	4.37	4.39
Eating	5.50	5.46	5.48	5.46	5.46	5.47
	2.90	2.87	2.87	2.88	2.86	2.88

Table 1: Philosophers' times in respective states (average and standard deviation).

State	CS1	CS2	CS3	CS4	CS5	All
Waiting	2.10	2.11	2.11	2.12	2.10	2.11
	2.57	2.58	2.57	2.60	2.56	2.58
Utilization	72%	72%	72%	72%	72%	72%

Table 2: Chopstick times in respective states (average and standard deviation) and utilization.

The chopstick "Utilization" of 72% in Table 2 is also less than the 92% reported in [3] and [4]. (Previous solutions do not report the "Waiting" times for chopsticks.)

A possible explanation for the differences in the reported statistics is that the default *enough* attribute of links in EZStrobe does *not allow* a hungry philosopher to draw just the chopstick on his left without also drawing the one on his right and start eating. Drawing just the chopstick on the left would also block the philosopher on their left from eating and increase their waiting time, too, as they would have to wait for the philosopher on the right to start and finish eating. It might also cause a deadlock when all five philosophers are hungry and have all drawn the chopsticks on their left.

By design, the *enough* link attributes in EZStrobe and STROBOSCOPE do not allow deadlock to occur by *ensuring first* that an activity can draw *all the resources it needs, before* allowing the activity to start a new instance and draw *any* resources. Clearly, this is the preferred strategy since it reduces the time philosophers wait before they can eat to less than half (from 11.5 to 4.19).

To investigate the system's behaviour, this model was also run with a different priority rule for the "Eat*k*" activities. Specifically, higher priority was given to the "Eat*k*" activity for the philosopher who had currently been waiting the longest time to eat. The resulting statistics were indistinguishable from those in Tables 1 and 2.

4 Conclusion

The simulation model presented here is particularly suitable for educational purposes, as it can be easily developed in EZStrobe using drag-and-drop graphics.

All the modelling elements (i.e., activities, queues, and links) shown in Figure 1 are instances of predefined intelligent EZStrobe shapes in Visio that were dragged, positioned, and connected on the model page.
The only additional requirement was to double-click these shapes and define the initial contents of queues, as well as the expressions for the duration and priority of activities in the custom dialogs that would appear.

To define the time at which the simulation should stop, the user right-clicks the model page, which opens the appropriate dialog. To run the simulation, the user right-clicks the model page and selects the option "Run simulation" from the menu that appears.

This instructs EZStrobe to construct the appropriate simulation model using the STROBOSCOPE language and send it to STROBOSCOPE for processing (all of which are hidden from the user). STROBOSCOPE then performs the simulation and presents the results in its own window.

References

[1] Breitenecker F, Schmidt B. Comparison 10: Dining Philosophers II. Definition of ARGESIM Benchmark C10. Simulation Notes Europe SNE. 1996; 06(18): 32-33.

[2] STROBOSCOPE Simulation System Software. Retrieved from www.stroboscope.org. Sept. 20, 2025.

[3] Legourski V, Huang Y, Cevan O, Breitenecker F. Statechart Modelling for ARGESIM Benchmark C10 'Dining Philosophers Problem II' using Simulink/Stateflow. Simulation Notes Europe SNE. 2008; 18(1): 39-40.

[4] Löscher T, Breitenecker F. A Petri Net-based solution to ARGESIM Comparison C10 'Dining Philosophers II' using MATLAB and PetriSim. Simulation Notes Europe SNE. 2005; 15(43): 29.