

STROBOSCOPE Model for ARGESIM Benchmark C6 'Emergency Department - Follow-Up Treatment'

Photios G. Ioannou¹, Constantine A. Ioannou²

¹Dept. of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI, USA, *photios@umich.edu

²School of Information and Computer Sciences, University of California, Irvine, CA, USA

SNE 36(1), 2026, 61-64, DOI: 10.11128/sne.36.bn06.10769
Submitted: 2025-12-20
Received Improved: 2026-01-30; Accepted: 2026-02-02
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The STROBOSCOPE simulation system is used to model the follow-up treatment received by four types of patients in a hospital emergency department with two examining rooms, staffed by two doctors of different experience. Depending on their case, patients may be sent for an X-ray and to a plaster room one or two times, and may need to be seen again by the doctors. The effectiveness of three operational alternatives is compared.

Introduction

ARGESIM benchmark C6 [1] investigates how to model the follow-up treatment received by patients in a hospital emergency department, which included the following:

- One registration area (one person) that assigned 60% and 40% of patients to Casualty Wards 1 or 2.
- Casualty Ward 1 (CW1) with two experienced doctors.
- Casualty Ward 2 (CW2) with two inexperienced doctors (for simpler cases).
- One waiting area for CW1 and one for CW2.
- Two X-ray units with a common waiting queue.
- One room that applied or removed plaster.

The four patient types, and their ratios, were as follows:

1. X-ray patients (35%). Patients are seen in *CWk* before being sent for X-rays. Patients return to *CWk*, X-rays are examined, and patients leave.
2. Plaster removal (20%). Patients enter *CWk*, go for plaster removal, and then leave.

3. Plaster renewal (5%). Patients enter *CWk*, are sent for an X-ray, receive a new plaster, return for another X-ray, return to *CWk*, and leave.
4. Changing wound dressings. Patients are seen in *CWk* and then leave.

The time between patient arrivals is distributed exponentially with a mean of 0.3 minutes.

The durations of the various activities are distributed Triangular(*minimum, mode, maximum*) as follows:

Registration:	Triangular(0.2, 0.5, 1.0) minutes
CW1:	Triangular(1.5, 3.2, 5.0) minutes
CW2:	Triangular(2.8, 4.1, 6.3) minutes
X-ray:	Triangular(2.0, 2.8, 4.1) minutes
Plaster:	Triangular(3.0, 3.8, 4.7) minutes

1 STROBOSCOPE

The STROBOSCOPE simulation system [2] is based on three-phase activity scanning, which is well-suited to modelling cyclic operations and controlled routing of resources, such as those in ARGESIM Benchmark C6. Simulation models are networks of nodes (such as activities and queues) where resources spend time, connected by links that control both when activities can start and the flow of resources (from preceding to succeeding nodes). A graphical user interface (GUI) facilitates the development of simulation model networks (Figure 1).

2 Simulation Model

Figure 1 illustrates the STROBOSCOPE simulation model network for the ARGESIM benchmark C6, which was created using the custom STROBOSCOPE Graphical User Interface (GUI) in Microsoft Visio.

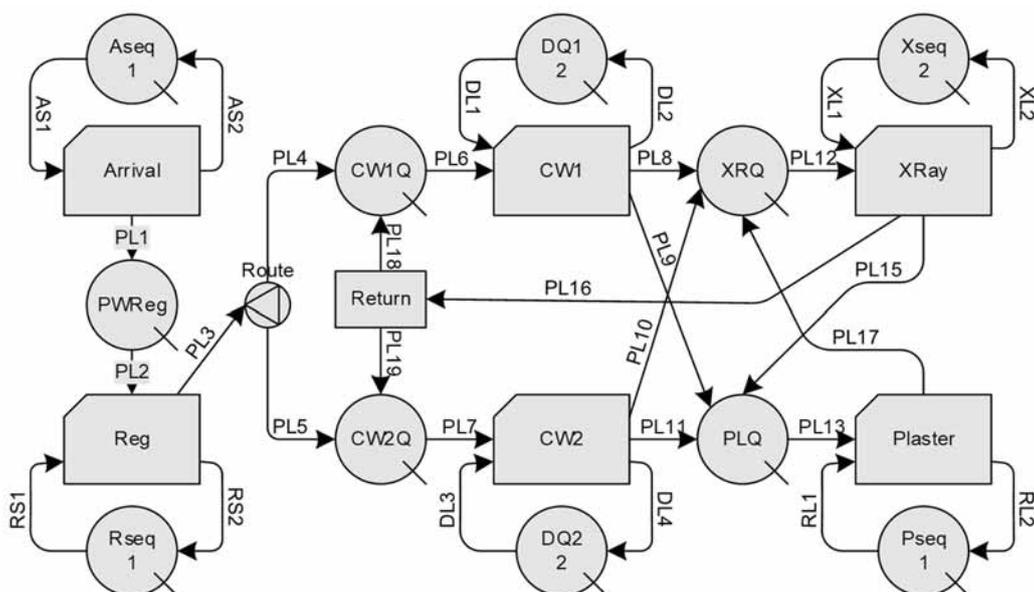


Figure 1: STROBOSCOPE Simulation Model Network.

Patients were modelled as characterized resources of type *Patient* (a class) with four subtypes: *P1*, *P2*, *P3*, and *P4*. Doctors were modelled as characterized resources of type *Doctor* with two subtypes: *D1* (experienced) and *D2* (inexperienced). Both *Patients* and *Doctors* have static and dynamic properties that depend on their subtype.

The queues in Figure 1 were initialized with the number of resources indicated inside each node. For example, queues *DQ1* and *DQ2* were initialized with 2 *Doctors* each, 2 *D1* in *DQ1* and 2 *D2* in *DQ2*.

The four queues in Figure 1 whose names end in “seq” were initialized with 1 or 2 units of the generic resource *Sequence* to start the serial instantiations of their successor conditional activities (combi).

For example, the *Sequence* resource in queue *Aseq* allowed the combi activity *Arrival* to create serial instances that model the arrival of patients. When each *Arrival* instance ended, it generated a *Patient* whose subtype was determined by random sampling, with probabilities *P1*: 35%, *P2*: 20%, *P3*: 5%, and *P4*: 40%.

Each arriving *Patient* then waited in queue *PWRReg* until it could register, *Reg*. It was then sent to the fork *Route*, which directed 60% of *Patients* to queue *CW1Q* and 40% to queue *CW2Q*, where they waited for a doctor in *DQk*, which enabled treatment to start in activity *CWk*.

When a *Patient* flowed through link *PL4* or *PL5*, its *Casualty Ward* (1 or 2) was stored in its *SaveValue WN*.

The number of X-rays (0, 1, 2) that a *Patient* had was stored in its *SaveValue nXRays*. This was increased by 1 each time the *Patient* was drawn by link *PL12*.

After the first examination by a *Doctor* in combi activities *CW1* or *CW2*, each *Patient* was released to the appropriate sequence of links depending on its subtype.

The appropriate routing of *Patients* was done by the *ReleaseWhere* attribute of each *releasing* link (from a combi to a queue) as follows:

- Links *PL8* and *PL10* allowed a *Patient* to flow through *only* if:
 - (a) Its subtype was *P1* and its *nXRays* was 0, or
 - (b) Its subtype was *P3* and *nXRays* was less than 2.
- Links *PL9* and *PL11* allowed a *Patient* to be released *only* if their subtype was *P2*.
- Link *PL15* allowed a *Patient* to be released *only* if its subtype was *P3* and its *nXRays* equaled 1.
- Link *PL16* allowed a *Patient* to be released *only* if:
 - (a) Its subtype was *P1* and its *nXRays* equaled 1, or
 - (b) Its subtype was *P3* and its *nXRays* equaled 2.
- Link *PL17* allowed a *Patient* to be released *only* if its subtype was *P3*.
- Links *PL18* and *PL19* allowed a *Patient* to be released *only* if its *savevalue WN* equaled 1 or 2, respectively.

A *Patient* that was not allowed to be released through any of the outgoing links was destroyed at the termination of the combi instance in which it resided.

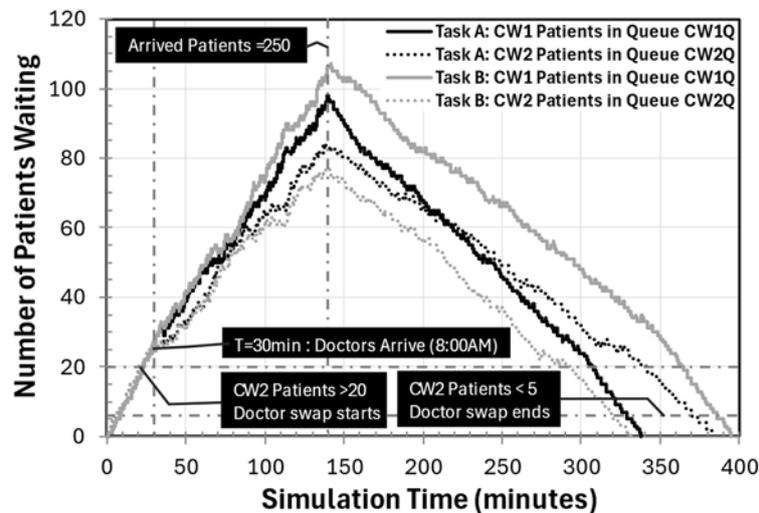


Figure 2: Number of *Patients* waiting in queues *CW1Q* and *CW2Q* on a typical day for both *Task-A* and *Task-B*.

For example, a *Patient* of subtype *P4* in an instance of *CW1* or *CW2* was destroyed along with that instance.

The fact that *Patients* started arriving at 7:30AM while *Doctors* started work at 8:00AM was modelled by the *Semaphores* of combis *CW1* and *CW2* that prevented them from starting when *SimTime* < 30 minutes.

3 Task A

Task A collected statistics on the average time needed to treat one *Patient* of each type, by running the simulation until 250 *Patients* had arrived (which represented 1 day).

Patient	Task A	Task B	Task C
All types	174	184	161
Type 1	237	252	164
Type 2	136	144	162
Type 3	252	266	175
Type 4	127	136	156
Avg. Std. Dev.	82	95	75
Closing time	13:48	14:30	13:52

Table 1: Averages for daily patient treatment time (min), daily standard deviation (min), and closing time.

The statistics in Table 1 are based on 100 replications (i.e., 100 days), each day allowing 250 *Patient* arrivals. These results show the average treatment time for all *Patients* and for each *Patient* type. Also shown is the average (over 100 days) of the daily standard deviation of all treatment times and the department average closing time.

4 Task B

Task B is an investigation into the effects of swapping one experienced *Doctor* (*D1*) from *DQ1* with an inexperienced *Doctor* (*D2*) in *DQ2*. The swap would start when the queue for Ward 2, *CW2Q*, exceeded 20 *Patients* and end when the queue *CW2Q* had fewer than 5 *Patients*. The times of *Doctor D2* when working in *CW1* were increased by 20% due to the more complex cases.

Insights into the effect of this swap are provided by Figure 2, which shows the number of *Patients* waiting in the Casualty Ward queues *CW1Q* and *CW2Q* on a typical day for both *Task A* and *Task B*.

Figure 2 shows that by 8:00AM (i.e., 30 minutes after the simulation started), the number of *Patients* arriving in *CW2Q* would already exceed 20 (for both *Task A* and *Task B*).

Figure 2 also shows that in all cases, the number of *Patients* in *CW2Q* would drop to below 5 only at the end of the day.

This means that the doctor swap would start at 8:00AM, as soon as the *Doctors* arrived for work in the morning, and would last practically the entire day.

Table 1 shows that all statistics resulting from the swap in **Task B**, i.e., the average daily patient treatment time, the average daily standard deviation, and the average closing time, were worse than those for **Task A**.

This was not unexpected because, as shown in Figure 2, queue *CW1Q* in Task A already had more *Patients* waiting than *CW2Q*. Thus, moving an experienced *Doctor D1* from *CW1* (the Ward with the larger queue) to *CW2* (the Ward with the smaller queue) in **Task B** was not a good idea. It could only make performance worse.

5 Task C

Task C investigated whether giving higher priority for service to patients who would return (a second time) to the same queues they had previously visited would reduce the daily standard deviation of treatment times.

In STROBOSCOPE, this change in *Patient* priority was implemented in a straightforward manner by changing the *Discipline* attribute for the queues for the Casualty Wards, for X-rays, and for plaster work (i.e., *CW1Q*, *CW2Q*, *XRQ*, *PLQ*), from the default *TimeIn* (i.e., *FIFO* based on the most recent times that *Patients* entered each queue) to *ResNum* (i.e., where *Patients* with a smaller *resource number*, i.e., those that arrived to the *system* earlier, would be sent to the front of the queue).

Table 1 shows that, as expected, the change in *Patient* priority reduced the average treatment time for *Patients P1* and *P3* (who made repeated visits to queues) and increased the average treatment time for *Patients P2* and *P3* (the easier cases) who visited the queues once.

The overall objective of **Task C** was achieved. The daily standard deviation of treatment times was reduced from 82 to 75 minutes.

6 Conclusion

The STROBOSCOPE simulation results in Table 1 are very close to previous solutions, e.g., [3] and [4]. Figure 2 is new and provides additional insights into the system.

The simulation model presented here requires only the basic modelling capabilities of STROBOSCOPE and is thus suitable for introductory educational purposes.

The network in Figure 1 used the standard drag-and-drop GUI elements implemented in Visio. Right-clicking the nodes and links in the GUI would activate dialog boxes (provided by a C++ add-on) that allowed the definition of node and link attributes as needed.

The GUI was used to create all the required STROBOSCOPE simulation model statements (text), which were then sent to STROBOSCOPE to perform the simulations.

References

- [1] Breitenecker F. Comparison 6: Emergency Department-Follow-up Treatment - Definition. Simulation Notes Europe SNE. 1992; 2(3): 30.
- [2] STROBOSCOPE Simulation System Software. Retrieved from www.STROBOSCOPE.org. Sept. 20, 2025.
- [3] Ölsböck K, Noyal M, Einzinger P. An Agent-based Approach to ARGESIM Benchmark C6 'Emergency Department – Follow-up Treatment' using Anylogic. Simulation Notes Europe SNE. 2013; 23(3-4): 193-194.
- [4] Pllana S, Fahringer T. An OO Programmed Approach to ARGESIM Comparison C6 'Emergency Department' with CSIM. Simulation Notes Europe SNE. 2004; SNE 14(40), 2004, 33.