

Chapter 2

Introduction to Arena

Arena is an easy-to-use, powerful modeling and simulation software tool that allows the user to construct a simulation model and run experiments on the model. The software generates several reports as a result of a simulation run.

2.1 The Arena Window

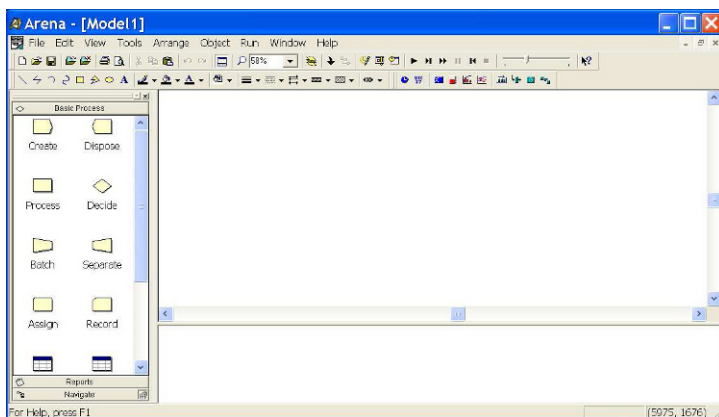


Fig. 2.1 The Arena window

After Arena starts, the computer screen shows the Arena window. Figure 2.1 shows what the Arena window typically looks like. This is the academic version 12.00.00. There are three main regions that can be identified in the main Arena window:

- The *Project Bar*, located on the left side of the Arena window and below the tool bars. The project bar contains three panels: the *Basic Process* panel, the *Report* panel, and the *Navigate* panel. Every panel contains several modules that are used in constructing simulation models.
- The *Model window flowchart view*, located on the right side of the Arena window and below the tool bars. This view is actually the workspace for the simulation model. It will contain all the model graphics: the flowchart, animation, and other drawings of the model.
- The *Model window spreadsheet view*, located on the right-hand side and below the flowchart view. The spreadsheet view shows the model data.

2.2 Arena Modules

The construction of simulation models with Arena, involves using modeling shapes, called *modules*, from the Basic Process panel. These modules are used as the building blocks in constructing a simulation model. There are two types of modules on a panel:

- *Flowchart modules*. The user places flowchart modules in the model window and connects them to form a flowchart, which describes the logic of model. The most common flowchart modules are: Create, Process, Decide, Dispose, Batch, Separate, Assign, and Record.
- *Data modules*. The user can edit these modules in the spreadsheet interface. These modules are not placed in the model window. The most common data modules are: Resource, Queue, Variable, Schedule, and Set.

A model is constructed in the model window flowchart view, which is the workspace of the model. From the Project bar, the user drags the flowchart modules needed into the model flowchart and connects the modules together. To edit a flowchart module, the user either double-clicks on the module shape and fills in the form, or edits its data in the spreadsheet.

The definition of a data module is carried out by clicking on the shape of then module in the Project bar to activate its spreadsheet. Then the user can edit the data of the module.

Flowchart modules exist both in the model workspace and as a row in the spreadsheet. Data modules exist only in spreadsheet view. Data modules can be added or deleted via the spreadsheet, while flowchart modules can only be added or deleted by placing the object or removing the object from the model workspace.

2.3 Using Arena

To build a simulation model and to carry out simulation runs with Arena, a user performs the following steps:

1. Construction of a basic model. Arena provides the model window flowchart view, which is a flowchart-style environment for building a model. The user selects and drags the flowchart module shapes into the model window and connects them to define process flow of the model.
2. Adding data to the model parameters. The user adds actual data (e.g., processing times, resource demands, others) to the model. This is done by double-clicking on module icons and adding data.
3. Performing a simulation run of the model. The user runs the simulation and examines the results.
4. Analysis of the simulation results provided by the automatic reports of Arena. The user can expand the statistics.
5. Modifying and enhancing the model according to the user needs.

Arena provides a family of application solution templates (ASTs) built on the Arena simulation system.

2.4 A Simple Model of The Carwash System

Vehicles arrive into a carwash shop to get a simple wash and clean up. The Carwash system consists of a single wash machine, which provides the actual service to the vehicles. Arriving vehicles join a line to wait for service. The vehicle at the head of the line is the one that is next to be serviced by the carwash machine. After the vehicle wash is completed, the vehicle leaves the system. The vehicles are considered the customers of the system, as they are the entities requesting service by the server (the wash machine). Figure 2.2 shows a graphical view of the conceptual model of the Carwash system.

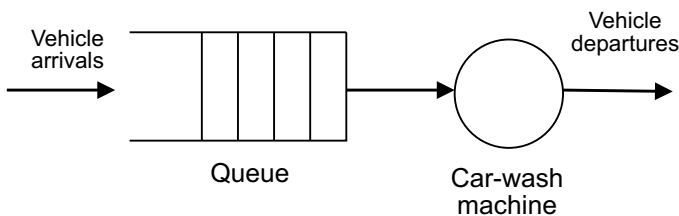


Fig. 2.2 Graphical view of the conceptual model of the Carwash system

The objects that flow through the system are the vehicles and in Arena, these objects are known as *entities*.

The first task in building the model is to define the flowchart of the model. To start the construction of a new model in Arena, the user activates the File menu and select New. A new model is given the name *Model1* by default.

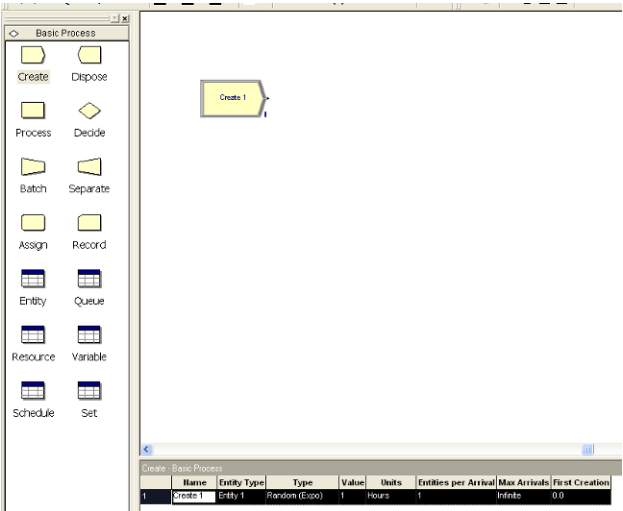


Fig. 2.3 Create module in model of the Carwash system

From the Project bar on the left-hand side of the Arena window, the user activates the Basic Process panel and drags the Create module into the flowchart view of the model window. Then the user drops the module shape in a convenient place, on the upper left-hand side of the flowchart view. Figure 2.3 shows the Create module in the flowchart view of the model window. This is the first module in constructing the model for the Carwash system. The figure shows only part of the Arena screen.

The Create module is the source of arriving vehicles into the system. To assign value to attributes or properties of the module, the user double-clicks on the module shape and enters the data requested in the small dialog window, as shown in Figure 2.4.

The name assigned to the module is *Arrival of vehicles*. The entity type is *Vehicles*. The arrival of vehicles occur randomly and the time between arrivals (the inter-arrival intervals) follow the behavior represented by an exponential probability distribution with mean value of 7.5 in minutes. The data for the module could also be filled in the spreadsheet view, below the model workspace.

The other modules needed to construct this simple simulation model are: a Process module and a Dispose module. As mentioned previously, these modules must be connected in such a manner as to represent the flow of entities through the system.

Create

Name:

Entity Type:

Arrival of vehicles

Vehicles

Time Between Arrivals

Type:

Value:

Units:

Random (Expo)

7.5

Minutes

Entities per Arrival:

Max Arrivals:

First Creation:

1

Infinite

0.0

OK

Cancel

Help

Fig. 2.4 Data in the properties of the Create module

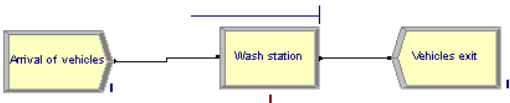


Fig. 2.5 Flowchart of the Carwash model

Process

Name:

Type:

Wash station

Standard

Logic

Action:

Priority:

Seize Delay Release

Medium(2)

Resources:

Resource: Wash machine, 1

<End of list>

Add...

Edit...

Delete

Delay Type:

Units:

Allocation:

Normal

Minutes

Value Added

Value (Mean):

Std Dev:

11.25

1.25

☒ Report Statistics

OK

Cancel

Help

Fig. 2.6 Properties of the Process module of the Carwash model

To place the two remaining modules in the model, the user selects the Create module, which is on the model view. The user then drags the Process module in the Basic process panel on the project bar to the flowchart model view and places it at any location to the right of the Create module. The two modules will automatically be connected. Now the user selects the Process module on the model view and drags the Dispose module in the Basic process panel on the project bar to the flowchart model view and places it at any location to the right of the Process module.

Figure 2.5 shows the complete flowchart of the Carwash model. The second module in the model is a Process module with assigned name *Wash station*. The third module is a Dispose module with assigned name *Vehicles exit*.

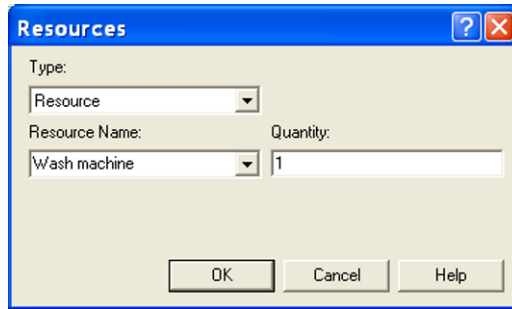


Fig. 2.7 The resources dialog window in the Carwash model

The properties of the Process module were assigned in a similar manner as for the Create module. The user double-clicks on the Process module and opens the dialog window of the module. The actual values to be used are then assigned to all the properties of the module, as shown in Figure 2.6.

The name assigned name to the Process module is *Wash station*. The Action selected for the module is *Seize, Delay, Release*. This means that a vehicle that arrives will wait until the resource becomes available, it will seize the resource, it will wait for the service interval, and then it will release the resource. The *Delay* is actually the processing time or the service interval.

To specify the resource that the vehicle needs for service, the user clicks the Add button on the Resources area of the Process module. A small Resources dialog window appears. The resource is specified and the name assigned to it is *Wash machine*. Figure 2.7 shows the Resources dialog window.

To setup the simulation runs, the user selects the Run menu and selects Setup. The dialog window that appears is shown in Figure 2.8. This figure shows the project title in the Project Parameters tab.

The Replications Parameters tab of the Run Setup dialog window is shown in Figure 2.9. The main parameter set is the replication Length, which is the simulation period.

Run Setup

Run Speed

Run Control

Reports

Project Parameters

Replication Parameters

Array Sizes

Project Title:

Carwash Model

Analyst Name:

Jose M. Garrido

Project Description:

Cars and other vehicles arrive into a carwash shop to get clean up. There is a single machine that carries out the actual wash of the vehicle. Arriving vehicles have to wait in a queue until service can be provided.

Statistics Collection

☐ Costing

☒ Queues

☐ Transporters

☒ Entities

☐ Processes

☐ Conveyors

☒ Resources

☐ Stations

☐ Activity Areas

☐ Tanks

OK

Cancel

Apply

Help

Fig. 2.8 The Project Parameters in the Run Setup dialog window of the Carwash model

Run Setup

Run Speed

Run Control

Reports

Project Parameters

Replication Parameters

Array Sizes

Number of Replications:

1

Initialize Between Replications

☒ Statistics

☒ System

Start Date and Time:

Tuesday , February 10, 2009 3:38:11 PM

Warm-up Period:

0.0

Time Units:

Hours

Replication Length:

14

Time Units:

Hours

Hours Per Day:

24

Base Time Units:

Hours

Terminating Condition:

OK

Cancel

Apply

Help

Fig. 2.9 The Replication Parameters on the Run Setup dialog window of the Carwash model

To run the simulation, the user selects Go from the Run menu, or presses F5. After the simulation runs to completion, Arena will ask the user whether he/she needs to open the reports.

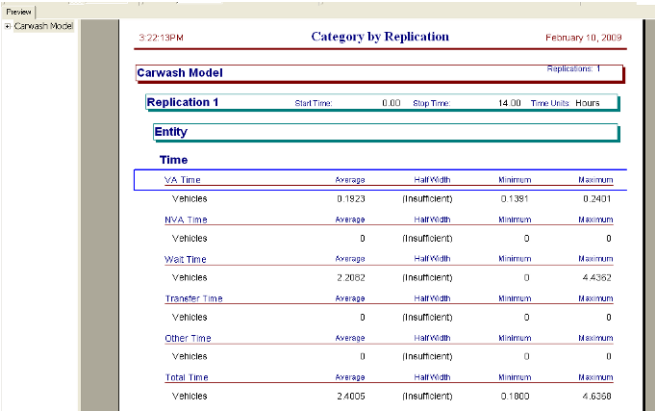


Fig. 2.10 The Category by Replication report of the Carwash model

The various reports produced by Arena after a simulation run can be accessed and opened from the Reports panel in the Project bar. Arena opens the Category Overview Report by default. Figure 2.10 shows part of the Category by Replications report. There is also a text file with a summary of results in the file Carwash.out.

From this report, the average wait time of a vehicle is 2.2082 minutes. The maximum wait time is 4.4362 minutes. The average time that a vehicle spent in the car wash shop is 2.4005 minutes. The total number of vehicles that arrived is 106 and the total number of vehicles that were serviced is 106, the rest remained in the system.

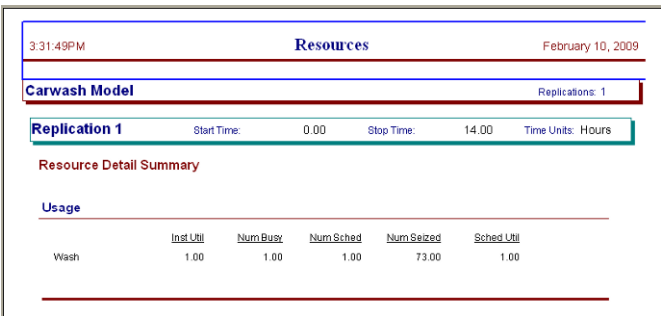


Fig. 2.11 The Resources report of the Carwash model

Figure 2.11 shows part of the Resources report produced after the simulation run. The resource (wash machine) *utilization* was 100%. The total number of times that the resource was seized was 73.

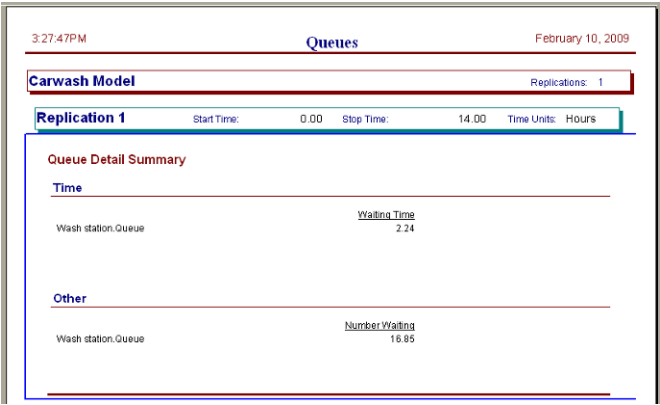


Fig. 2.12 The Queues report of the Carwash model

Figure 2.12 shows part of the Queues report. The average time that a vehicle spent on the queue was 2.24 minutes. The average number of vehicles waiting in the queue was 16.85.

Changing the mean inter-arrival interval to 14.5 minutes, changes the output results as shown in Figure 2.13. The total number of vehicles serviced is 49. The average waiting time is 1.969 minutes. The resource utilization is 0.6738 or 67.38%.

The basic animation possible with Arena consists of showing with a small visual object the arrival of an entity and its accumulation in the queue. The default picture for an entity can be changed by editing the Entity data module.

The other animation consists of showing visually the states of the resource, Idle and Busy. To place a picture to indicate each state, the user edits a picture in the Resource Picture Placement dialog by clicking the Resource button in the Animate bar, which is the second tool bar at the top of the Arena window.

Arena provides a facility to generate or draw dynamic plots of a simulation run. To define a plot, the user clicks on the Plot button on the Animate bar and enters the data on the Plot dialog window that appears.

2.5 Summary

Arena is a very versatile integrated simulation development tool. Constructing a simulation model involves identifying one or more flow objects known as entities that flow through the system and then building a flowchart of the model using Arena’s flowchart modules. The model is then enhanced by editing some of the data

Queue				
Time				
Waiting Time	Average	Half Width	Minimum Value	Maximum Value
Wash station.Queue	0.1969	(Insufficient)	0.00	0.9194
Other				
Number Waiting	Average	Half Width	Minimum Value	Maximum Value
Wash station.Queue	0.6892	(Insufficient)	0.00	5.0000
Resource				
Usage				
Instantaneous Utilization	Average	Half Width	Minimum Value	Maximum Value
Wash machine	0.6738	(Insufficient)	0.00	1.0000
Number Busy	Average	Half Width	Minimum Value	Maximum Value
Wash machine	0.6738	(Insufficient)	0.00	1.0000
Number Scheduled	Average	Half Width	Minimum Value	Maximum Value
Wash machine	1.0000	(Insufficient)	1.0000	1.0000
Scheduled Utilization	Value			
Wash machine	0.6738			
Total Number Seized	Value			
Wash machine	49.0000			

Fig. 2.13 Partial report after modifying the input parameters

modules. Arena provides a simple method to setup the simulation parameters and the model input parameters. A variety of reports with the corresponding simulation results is produced.

Exercises

- 2.1. Write a list of systems that would have a similar model as the one for the Carwash system.
- 2.2. Increase the inter-arrival interval of the arriving vehicles in the Carwash model and perform one or more simulation models. Comment on the differences in results compared to the results from the original simulation run.
- 2.3. Decrease the inter-arrival interval of the arriving vehicles in the Carwash model and perform one or more simulation models. Comment on the differences in results compared to the results from the original simulation run.
- 2.4. Increase the average processing interval of the vehicles in the Carwash model and perform one or more simulation models. Comment on the differences in results compared to the results from the original simulation run.
- 2.5. Decrease the average processing interval of the vehicles in the Carwash model and perform one or more simulation models. Comment on the differences in results compared to the results from the original simulation run.

2.6. After reading the documentation on how to construct a plot for a simulation run, add a plot to the carwash model. This plot should show the number of vehicles in the system as the simulation time progresses.

2.7. After reading the documentation on how to animation in a simulation run, add animation that shows the state of the wash machine (Idle and Busy). Also include another picture for the vehicles that arrive and wait in the queue.



<http://www.springer.com/978-1-4419-0515-4>

Object Oriented Simulation

A Modeling and Programming Perspective

Garrido, J.M.

2009, XIX, 431 p., Hardcover

ISBN: 978-1-4419-0515-4