CHAPTER 3

A Guided Tour Through Arena

As we were honest enough to admit in Chapter 2, we really used Arena to carry out the "hand" simulation in Section 2.4, as well as the multiple replications and the modified model with the double-time arrivals in Section 2.6. In this chapter, we'll lead you on a tour through Arena by having you start up Arena, browse through the existing model we built for the hand simulation, run it, and then build it from scratch. Then we'll use just these basic building blocks in a case study exploring a question of real interest, whether it's better to have specialized serial processing or generalized parallel processing, and the effect of variability on such a decision. We'll also explore the Arena user interface, get you into the help and documentation systems, discuss different ways to run your simulation, and illustrate some of the drawing and graphics tools.

Section 3.1 gets you to start Arena on your computer, and in Section 3.2, you'll open an existing model and look around. In Section 3.3, you'll go through the model in some detail, browsing the dialog boxes and animation, running the model, and taking a look at the results; in Section 3.4, you'll construct this model from scratch. Section 3.5 contains the case study mentioned above. In Section 3.6, we'll briefly go over many of Arena's pieces and capabilities, including what's available in the menus and toolbars and the drawing and printing capabilities. There's a broad and deep help system in Arena, with all of the detailed technical documentation, which is the subject of Section 3.7. There are a lot of options for running and controlling simulations, which are discussed in Section 3.8.

By the end of this chapter, you'll have a good feel for how Arena works and have an idea of the things you can do with it. You'll be able to work effectively with Arena to build simple models and maybe take a stab at doing some not-so-simple things as well by cruising the menus and dialogs on your own, with the aid of the help and documentation systems. While you can probably make some sense out of things by just reading this chapter, you'll be a lot better off if you follow along in Arena on your computer. More information on building your own models with Arena is discussed in Chapter 4 and beyond.

3.1 Starting Up

Arena is a true Microsoft® Windows® operating system application, so its look and feel will already be familiar to you, and all the usual features and operations are there. In addition, Arena is fully compatible with other Windows software, like word processors, spreadsheets, and CAD packages, so you can easily move things back and forth (Chapter 10 goes into detail about Arena's interaction and communication with other software).

By the way, we're assuming you're already comfortable with the basics of working with Windows, such as:

- Disks, files, folders, and paths.
- Using the mouse and keyboard, including clicking, double-clicking, and right-clicking.
- Operating on windows, like moving, resizing, maximizing, minimizing, and closing.
- Accessing things from menus. We'll use notation like "*M* > *C* > *S* > *T*" to mean open the *M* menu, choose *C* from it, then choose *S* from the submenu menu (if any), then choose the tabbed page labeled *T* (if any), etc.
- Using the *Control*, *Alt*, and *Shift* keys. By "*Ctrl+whatever*," we'll mean to hold down the *Ctrl* key and press "*whatever*" (this will also apply for *Alt+whatever* and *Shift+whatever*). If "whatever" is a keyboard key, it's not case-sensitive. "Whatever" could also be a mouse click, like *Ctrl+Click* to extend a selection to include additional items.
- *Cut* (or the menu command *Edit* > *Cut* or the shortcut key combination *Ctrl+X*), *Copy* (or *Edit* > *Copy* or *Ctrl+C*), and *Paste* (or *Edit* > *Paste* or *Ctrl+V*) of text and other items.
- Filling out dialog boxes by entering and editing text entries, pressing buttons, selecting and clearing (that is, unchecking) check boxes, clicking exactly one from a list of option buttons (also called radio buttons), and selecting items from drop-down list boxes.

If any of these things are unfamiliar to you, it would probably be a good idea for you to go through a tutorial on Windows before moving on.

Go to your computer, on which Arena is already installed per the instructions that came with it (see Appendix E for instructions on installing the academic version of Arena, which is what's on the CD packaged with this book). Approach the computer cautiously but with confidence—if it senses you're afraid, it could attack. Locate the Arena icon, or a shortcut to it, and double-click on it (or launch Arena by starting Windows and clicking the *Start* button, then *Programs* > *Rockwell Software* > *Arena 10.0*, and finally the *Arena 10.0* icon). In a moment, the Arena copyright window will come up; if you're running an academic version (which is what's on the CD with this book) or an evaluation version, you'll get a message box to this effect, which you should read and then click *OK* (or just click the *Enter* key on your keyboard since the *OK* button is already selected by default).

At the top left of the Arena window are the *File*, *View*, *Tools*, and *Help* menus (in addition to several other menus if a blank model file was automatically opened when Arena started up). You'll also see toolbars with various buttons, only a few of which are available unless you have a model file open:

🗋 Create a *New* blank model file. This is equivalent to the menu command *File* > *New* and to the keyboard operation *Ctrl+N*.

📂 Display a dialog box to open a previously saved model; equivalently *File* > *Open* or *Ctrl+O*. You may need to navigate around to other folders or disks to find what you want.

📂 *Template Attach* (*Templates*, of which there are several, contain the modeling elements); equivalently *File* > *Template Panel* > *Attach*. The template files (with file name extension*.tpo*) are in the Template folder, which in turn is in the Arena 10.0 folder. You can also right-click in the Project Bar on the left (see Figure 3-1), then *Template Panel* > *Attach* from the pop-up that appears.

📂 *Template Detach* (when you don't need the modeling elements in the active panel any more); equivalently *File* > *Template Panel* > *Detach* or right-click in the active panel (which you want to detach) the Project Bar on the left, then *Template Panel* > *Detach* in the pop-up.

↖? *Context Help* to provide help on a menu or toolbar command. Click on it to add the question mark to your mouse arrow and then click on a toolbar button or menu command to get help on it; closing that help window returns the mouse pointer to arrow-only.

Tooltips provide another source of quick and brief help (even quicker and briefer) on toolbar buttons. If your mouse remains motionless over a button for a second or two, a little box shows up with the name of the button. If you want to know more about that button, you could use ↖? as just described, or maybe look it up (now that you at least know its name) in Arena's help system; more on that in Section 3.7. If you get tired of being pestered by tooltips at every turn, you can turn them off via *View* > *Toolbars* > *Toolbars* and clear (uncheck) the *Show Tooltips* option.

When you're done with your Arena session and want to get out, click ✕ at the upper-right corner of the Arena window, or *File* > *Exit*, or *Alt+F4,* or right-click in the Arena window bar at the very top and select Close from the pop-up.

3.2 Exploring the Arena Window

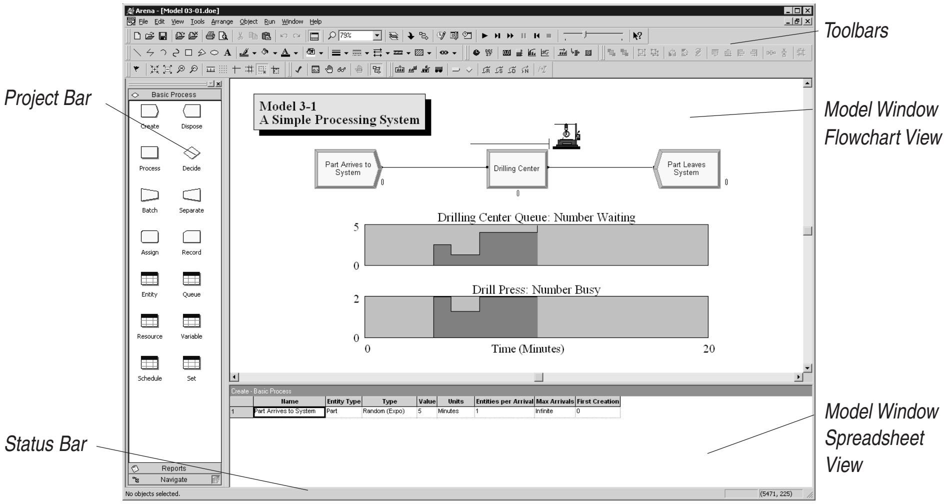
In this section, we'll open an existing model, use it to look around the Arena window so you can get familiar with where things are, and introduce some basic Arena terminology.

3.2.1 Opening a Model

The ready-made model for the hand simulation can be found via *File* > *Open* (or just click 📂 to bring up the Open dialog box). File names appear in a scrolling box, and you can also navigate to other folders or drives. Find the file named `Model 03-01.doe`; the file name extension *.doe*[1] is the default for Arena files. In a typical installation using the CD that came with this book, it will be in the Book Examples folder, which is in turn in the Arena 10.0 folder. Click on this file name (highlighting it), and then click the *Open* button (or just double-click the file name).

You should get an Arena window that looks something like Figure 3-1 (you might see different toolbars and buttons on your computer or see some things in different places). We'll call this Model 3-1.

[1] In its early development, Arena was code-named "Bambi." We're not making this up.



Create - Basic Process

	Name	Entity Type	Type	Value	Units	Entities per Arrival	Max Arrivals	First Creation
1	Part Arrives to System	Part	Random (Expo)	5	Minutes	1	Infinite	0

Figure 3-1. Arena Window for the Simple Processing System, Model 3-1

3.2.2 Basic Interaction and Pieces of the Arena Window

As shown in Figure 3-1, the Arena window with this model open is divided into several pieces.

On the right, taking up most of the screen, is the *model window*, which is actually inside the Arena window. If you had several Arena models open at once you'd have a separate model window for each of them, all inside the Arena window, just as in word-processing or spreadsheet software. Switch between model windows by just clicking in them (if the one you want is visible), or use the Arena Window menu to select from the entire list. If you have a lot of models open, you can cycle among them via *Ctrl+Tab*, or you might want to minimize some of them to icons with the ▁ button in each one. The Window menu also has commands (*Cascade*, *Tile*, etc.) for how you'd like to arrange the open models or their minimized icons. Create a new (blank) model window via 🗋 (or *File > New* or *Ctrl+N*), save the active model window via 💾 (or *File > Save* or *Ctrl+S*) or *File > Save As,* and open a previously saved model window via 📂 (or *File > Open* or *Ctrl+O*). Resizing and repositioning a model window works just like any Microsoft® Windows® operating system application.

The familiar cut, copy, and paste operations work within Arena as well as between Arena and other applications. For instance, you might have several Arena model windows open, and you might want to copy some objects from one to another. Just select the objects with the mouse (*Ctrl+Click* to extend the selection, or drag a box across them if they're positioned that way), copy them to the Clipboard (*Ctrl+C* or 📄 or *Edit > Copy*), switch to the other window, and paste them in (*Ctrl+V* or 📋 or *Edit > Paste*). After choosing the paste operation, the mouse pointer changes to cross hairs that you click where you want the northwest corner of the selection to land. Or, you might have Arena open simultaneously with a spreadsheet in which there's a long number you want to put into an Arena dialog text box. Copy the number from the spreadsheet cell, switch to Arena (either via the Windows® Taskbar or by using *Alt+Tab* to cycle through the open applications), position the insertion pointer in the Arena dialog box where you want the number, and paste it in. If you're writing a report in a word processor and want to paste in a "snapshot" of an Arena screen, go to Arena and get it into the state you want to photograph, press the *Prnt Scrn* (Print Screen) key, switch over to your word-processing document, and paste the shot where you want it; if you want just the active window (like a dialog box you want to document), press *Alt+Prnt Scrn* instead, then paste it into the word-processing document.

The model window can be split into two regions, or *views*: the *flowchart view* and the *spreadsheet view*. Often it's helpful to see both the flowchart and spreadsheet views of the model window at the same time. But you can choose to see only one of the views and thus devote all of the real estate in the model window to it by clearing the menu command *View > Split Screen* or clicking ▤ so that it does not appear to be pushed in; in this case, to see one or the other view, just single-click on either a flowchart (□) or spreadsheet (▦) module in the Project Bar on the left of your screen. The flowchart view contains the model's graphics, including the process flowchart, animation, and other drawing elements. The spreadsheet view can display model data such as times and other parameters, and allows you to enter or edit them (right now it happens to be showing details about something called "Create – Basic Process"). Many model parameters can be viewed and edited in either the flowchart view or the spreadsheet view, but the spreadsheet view gives you access to lots of parameters at once, arranged in compact groups of similar parameters convenient for editing, especially in large models. The horizontal line splitting the flowchart and spreadsheet views (if both views are visible) can be dragged up or down to change the proportion of the model window allocated to the two views.

Down the left edge of the Arena window in Figure 3-1 is the *Project Bar*, which hosts *panels* containing the objects with which you'll be working, displaying one panel at a time. Right now the Project Bar is displaying the Basic Process panel, which contains fundamental building blocks, called *modules*, that are useful in a wide variety of simulation models.

Below the Basic Process panel on the Project Bar is a horizontal button labeled "Reports," which will display another panel containing a road map to the results of a simulation after it's run; click on this button to make this panel visible, and then click on the Basic Process button to make that panel visible again.

The Navigate panel allows you to display different views of a model, including different submodels in a hierarchical model (Model 3-1 doesn't have submodels so the only view in the Navigate panel is Top-Level, though if you click on the + to its left, you open a tree that has three entries for our model, which we'll discuss in Section 3.2.3 below). If the small button (▣) on the right of the horizontal Navigate button is pressed, you'll also get a "thumbnail" of the model window in the top of the Navigate panel, with a translucent blue box showing the location and zoom size of the active window's current view. Clicking anywhere in the thumbnail changes the current view to that spot. Drag the blue box around to pan to other regions of the window, or resize the blue box (hover the pointer over an edge and then click-drag) to change the “altitude” of the zoom. The +/– toggle in the circle in the thumbnail’s upper right determines whether the blue box is shown relative to the entire window’s capacity (the “+” choice), or takes into account what region in the window actually has content (the “–” choice).

The Project Bar is usually docked to the left edge of the Arena window, but it can be torn off and “floated” anywhere on your screen, or it can be docked to the right edge of the model window if you prefer. You’ll usually need the Project Bar to be visible while working on a model, but if you’d like more room just to look through things, you can push the small ✖ button at the upper right of the Project Bar, or clear *View > Project Bar* to hide it (re-check *View > Project Bar* to display it again).

There are several other panels that come with Arena, perhaps depending on what you licensed. These include Advanced Process (with different and “smaller” building blocks for more detailed modeling), Advanced Transfer (containing many options for moving entities around), and Blocks and Elements (which together give you full access to the SIMAN simulation language that underlies Arena; see Pegden, Shannon, and Sadowski, 1995). Yet more panels contain constructs for specialized applications, like modeling contact centers and high-speed packaging lines. As mentioned earlier, to make the elements in a panel available for use in your model, you need to *Attach* the panel to your model via *File > Template Panel > Attach* or the *Template Attach* button (📂), or by right-clicking in a panel and selecting *Template Panel > Attach* in the pop-up. Panel files have the file name extension *.tpo* and are typically in the Template folder inside the Arena 10.0 folder. If you want Arena to attach certain panels to each new model you start, do *Tools > Options > Settings* and type the file names of those *.tpo* panel files into the *Auto Attach Panels* box there.

At the very bottom of the Arena window is the *Status Bar*, which displays various kinds of information on the status of the simulation, depending on what’s going on at the moment. Right now the only thing it shows are the (x, y) coordinates in the world space (see Section 3.2.3) of the location of the mouse pointer. While the simulation runs, the Status Bar will display, for instance, the simulation clock value, the replication number being executed, and the number of replications to be run. You can hide the Status Bar by clearing (unchecking) *View > Status Bar*.

3.2.3 Panning, Zooming, Viewing, and Aligning in the Flowchart View

The particular flowchart view of the model window you see in Figure 3-1 is just one of many possible *views* of the model and the big *world space* in which the flowchart depiction of a model lives. The world space’s center has (x, y) coordinates (0, 0), and it extends for thousands of units in all four directions from there; these units are just positional and don’t have any particular physical meaning (call them furlongs or youdels[2] if you like). To maximize the size of the model window within the Arena window, click ▣ if it’s visible in the upper right corner of the model window. Likewise, to maximize the Arena window itself to consume your entire screen, click its ▣ button.

[2] Apologies to Gene Woolsey.

To see different parts of the flowchart view, you can pan around using the scroll bars on the lower and right edges, or the arrow keys (try it; to navigate via the keyboard, you must first make the model window active by clicking in it). You can also zoom in (with the button or the + key or *View > Zoom In*), or zoom out (with the button or the – key or *View > Zoom Out*) to see parts of the model from different "altitudes." To pan/zoom automatically to see all the model at the closest possible zoom, click (or *View > Views > All*, or the * key). If you want to go back to the preceding view (maybe you messed up), click (or *View > Previous*). If you're at a relatively high altitude but spy a region that you'd like to view up close, select *View > Views > Region* (or hit the "[" key) to change the mouse pointer to cross hairs, click on one corner of the rectangular region you want to see, then again on the opposite corner—Arena will pan and zoom to see all of that region at the closest possible zoom (that is, lowest possible altitude). Another way to pan and zoom is via the thumbnail option in the Navigate toolbar, described in Section 3.2.2.

If you get to a view you like (and to which you'd like to be able to return instantly), you can save it as a *Named View* and assign a hot key to it. Pan and zoom to the view you want to save, then select *View > Named Views* (or hit the the **?** key or the button), and then click *Add*. You must give the view a descriptive Name, and you can optionally assign a hot key to it as well. To jump back to this view at any time, select *View > Named Views* (or hit the **?** key or the button), click on the view you want, and press the *Show* button. You can also access your Named Views in the Navigate panel of the Project Bar by clicking the + to the left (in this case of Top-Level) to open up a tree of the Named Views; just click on an entry to go to that view. Yet another way (probably the fastest way) to get to a Named View is to hit the hot key assigned to it; you'll have to remember what the hot keys are, or maybe document them in the model with some text, as described in Section 3.6.3. Hot keys for Named Views are one of the few places in Arena where characters are case-sensitive (for example, "a" and "A" are different). Named Views can be accessed at any time, even while the simulation is running. We've set up three Named Views for Model 3-1: all (hot key **a**), logic (hot key **l**), and plots (hot key **p**). Try them out.

New Arena models start out in a specific "Home" pan/zoom configuration, just to the southeast of the (0, 0) position in the world space, to which you can return by pressing the Home key on your keyboard (or *View > Views > Home*). To see the largest possible area of the world space (from the maximum altitude), select *View > Views > Max*.

To get your visual bearings, you can display a background grid of little dots by checking *View > Grid* (or by clicking). If you further want to cause newly placed items to snap to this grid, check *View > Snap* (or click). Both of these actions are toggle keys; that is, you just repeat the action to undo it. To snap existing items to the grid, first select them (maybe using *Ctrl+Click* to keep extending your selection, or dragging a rectangle across them if they're arranged that way) and then *Arrange > Snap Object to Grid* to adjust their positions to align with the grid points. To customize the spacing of the grid points, select *View > Grid & Snap Settings*; the units are (x, y) values in the measurement units of the world space. You can also display Rulers on the top and left edges, with units' being Arena world units, by pushing the button or checking *View > Rulers*.

You might want to align objects in the flowchart view precisely, either horizontally or vertically, with respect to their edges or centers, and there are several options for doing this. One option is to establish horizontal and vertical *Guides* in the flowchart view; make sure Rulers are displayed, then click the *Guides* button (┼) or select *View* > *Guides*, and then drag down from the horizontal ruler on the top or drag to the right from the vertical ruler on the left to place a dashed blue Guide line (you can have several of each). Then, click the *Glue to Guides* button (⊞) or select *View* > *Glue to Guides*, and then drag objects toward the guides until a red positional square lights up for edge or center positioning. Once objects are glued to a guide, you can drag the guide, and all objects glued to it will move together and stick with their alignment to the guide. Another option is to select the objects you want to align and then use the Arrange menu; select Align to line the selected objects up on their Top, Bottom, Left, or Right edges; select Distribute to space them evenly horizontally or vertically with spacing specified in Settings. *Arrange* > *Flowchart Alignment* lines up flowchart modules (see Section 3.2.4 next) in the current set of selected objects.

3.2.4 Modules

The basic building blocks for Arena models are called *modules*. These are the flowchart and data objects that define the process to be simulated and are chosen from panels in the Project Bar. Modules come in two basic flavors: *flowchart* and *data*.

Flowchart modules describe the dynamic processes in the model. You can think of flowchart modules as being nodes or places through which entities flow, or where entities originate or leave the model. To put an instance of a flowchart module of a particular type into your model, drag it from the Project Bar into the flowchart view of the model window (you can drag it around later to reposition it). Flowchart modules are typically connected to each other in some way. In the Basic Process panel, the kinds of flowchart modules available are Create, Dispose, Process, Decide, Batch, Separate, Assign, and Record; other panels have many additional kinds of flowchart modules. Each type of flowchart module in the Basic Process panel has a distinctive shape, similar to classical flowcharting (see Schriber, 1969) and suggestive of what it does. But in other panels (such as the Advanced Process panel), there are many more flowchart-module types than there are reasonable shapes, so they're all represented by simple rectangles. Some panels (like Advanced Transfer) use colors in the rectangles to distinguish different types of flowchart modules, and some panels (like the specialized ones for contact centers and packaging) use more elaborate graphics for them. One way to edit a flowchart module is to double-click on it once it's been placed in the flowchart view of the model window, to bring up a dialog box pertaining to it. Another way to edit flowchart modules is to select a module type (for example, click on a Create or a Process module), either in the Project Bar or in the flowchart view of the model window, and a line for each flowchart module of that type in the model shows up in the spreadsheet view of the model window (if it's visible), where you can edit the entries. This gives you a compact view of all the instances of flowchart modules of that type in your model, which is useful in large models where you might have many such instances.

Data modules define the characteristics of various process elements, like entities, resources, and queues. They can also set up variables and other types of numerical values and expressions that pertain to the whole model. Icons for data modules in the Project Bar look like little spreadsheets. The Basic Process panel's data modules are Entity, Queue, Resource, Variable, Schedule, and Set (other panels contain additional kinds of data modules). Entities don't flow through data modules, and data modules aren't dragged into the model window; rather, data modules exist "behind the scenes" in a model to define different kinds of values, expressions, and conditions. You don't double-click on a data module to edit it, but just single-click it in the Project Bar and a spreadsheet for that type of module will appear in the spreadsheet view of the model window (which must be visible), which you can then edit or extend by double-clicking where indicated to add additional rows. While the default is to edit modules from the spreadsheet view, if you double-click on the number in the left-hand column (or right-click and select Edit via Dialog), you can also edit in the dialog mode. Unlike flowchart modules, you don't have more than one instance of a data module in a model; however, there could be many rows in the spreadsheet for a data module, each typically representing a separate object of that type (for instance, if your model has three different queues, the Queue data module will display three rows, one for each queue, in its spreadsheet).

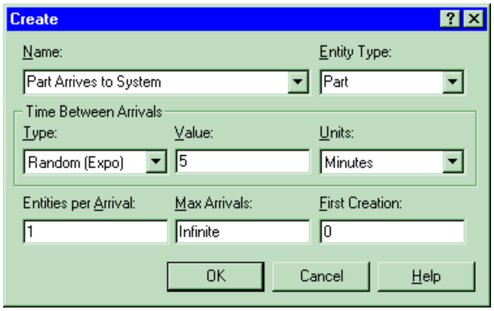
Flowchart and data modules in a model are related to each other by the names for objects (like queues, resources, entity types, and variables) that they have in common. Arena keeps internal lists of the names you give to these kinds of objects as you define them, and then presents these names to you in drop-down lists in the appropriate places in both flowchart and data modules, which helps you remember what you've named things (and protects you from your own inevitable typos...or, at least, keeps you consistent in your lousy typing so your model will still run).

3.2.5 Internal Model Documentation

If you rest your mouse pointer on a module symbol or other object, you'll see a Data Tip. Data Tips have two parts, a *default description* and a *user-defined description*. The default tip will describe some generic information about the object, such as its name and type. The user-defined part will display exactly what is entered in the Object Properties Description field. You can enter text in this field by right-clicking on an object and selecting Properties. Display of these Data Tips can be toggled via *View > Data Tips*. By default, both are enabled, but either the default or the user-defined Data Tips may be disabled.

In addition to the *module* descriptions, you can also enter a *Project Description* that provides some context to the entire model. This is a good place to document what the model does, why it was created, assumptions you have made, and similar information. This is entered in *Run > Setup > Project Parameters* (see Figure 3-15 later in this chapter).

While Data Tips are a useful feature, particularly when your models get large, you might wonder if there is another way to make use of this information. There is. *Tools > Model Documentation Report* creates a custom report summarizing all your model data. It provides some options about what information to include, then generates an HTML report.



Create

Name: Part Arrives to System

Entity Type: Part

Time Between Arrivals

Type: Random (Expo)

Value: 5

Units: Minutes

Entities per Arrival: 1

Max Arrivals: Infinite

First Creation: 0

Figure 3-2. The Create Property Dialog Box for Model 3-1

3.3 Browsing Through an Existing Model: Model 3-1

To see how Model 3-1 is set up, we'll now walk you through the flowchart and data modules in the model window and indicate how they're related. Then we'll run this model and look at the results. After that, in Section 3.4, we'll show you how to build this model from scratch.

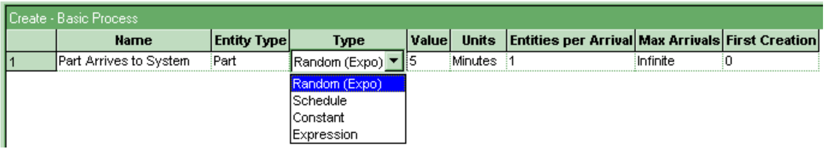
3.3.1 The Create Flowchart Module

We'll start with the Create module, which we named Part Arrives to System, at the left of the flowchart view of the model window. Note that this module is an instance of the general Create module, which we've specialized for our needs in this particular model.

The Create module is the "birth" node for arrival of entities to our model's boundary, representing parts in this case, into the model from outside. Double-click it to open a dialog box like the one in Figure 3-2.

In the Name box, we've typed Part Arrives to System as the name of this particular Create module (rather than accepting the bland default Name), which is what appears inside its shape in the flowchart view and in its data tip. We entered Part as the Entity Type; there's only one entity type in this model but in general there could be many, and naming them separately keeps them straight and allows you to customize them in useful ways (like separating times or numbers in system by entity type).

Across the center of the dialog box is a bordered area called Time Between Arrivals, where we specify the nature of the time separating consecutive arrivals of Part entities originating in this module. In the Type box, we selected Random (Expo) (using the list box arrow ▾) so that the interarrival times will be generated as draws on a random variable; in particular, from the exponential distribution (see Appendix C if you need to brush up on your probability and Appendix D for definition of the exponential and other probability distributions). In the Value box, we typed 5, and in the Units box, selected Minutes to tell Arena that we mean 5 minutes rather than 5 seconds or 5 hours or 5 days. While the number we typed in was "5," we could have typed "5." or "5.0" since Arena is generally quite robust and forgiving about mixing up integers and real numbers.



Create - Basic Process

	Name	Entity Type	Type	Value	Units	Entities per Arrival	Max Arrivals	First Creation
1	Part Arrives to System	Part	Random (Expo)	5	Minutes	1	Infinite	0

Figure 3-3. The Create Spreadsheet for Model 3-1

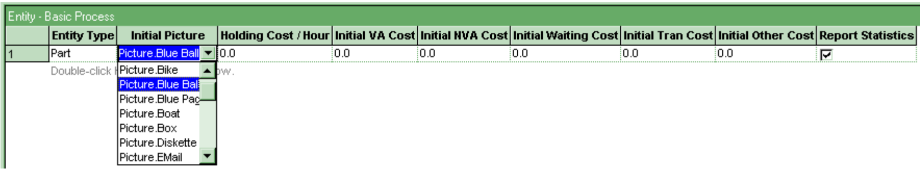
In the bottom row of boxes, we said that the number of Entities per Arrival is 1 (the default, so that Parts arrive one at a time rather than in a batch of several), that we don't want to put a cap on the maximum number of arrivals (if we did, this Create module would be "turned off" after that), and that the first Part should arrive right away, at time 0 (rather than after an initial time period that might or might not have the same definition as times between successive arrivals).

To close this Create dialog box, click *Cancel* or ✕ at the upper right; if you'd made any changes that you wanted to retain, you'd click *OK* instead.

An alternative way to edit the Create flowchart module is via the spreadsheet view in the model window. If you click the Create module in the flowchart view of the model window (or on any instance of the Create module there if there were several of them in your model), or on the general Create module shape in the Project Bar, a spreadsheet for your Create module(s) shows up in the spreadsheet view of the model window, as in Figure 3-3. By clicking or double-clicking in each of the fields in this spreadsheet, you can edit the entry or select from options; Figure 3-3 shows the list of options for Type accessed via the drop-down list there (drop-down lists for fields will be offered wherever they make sense). If you had multiple Create modules in your model, each representing a separate source of incoming entities, there would be a separate row in the Create spreadsheet for each. This is convenient in large models to edit many things quickly, or just to get an overview of all the Create modules at once. Selecting a particular Create module in either the flowchart or spreadsheet view selects that module in the other view. By right-clicking in a row of the spreadsheet, you're given the option of editing that module via the dialog box as in Figure 3-2. If you right-click in a numerical field (here or pretty much anywhere in Arena), you can also choose Build Expression to get very useful assistance in putting together a perhaps-complicated algebraic expression that could use many different Arena variables, probability distributions, mathematical functions, and arithmetic operations (we'll discuss the Expression Builder in Sections 3.4.10 and 4.2.4). You can change the field widths in a spreadsheet by dragging left and right the solid vertical bars separating them.

3.3.2 The Entity Data Module

One of the things we did in our Create module was to define an Entity Type that we called Part. By selecting the Entity data module in the Project Bar, the Entity spreadsheet for your model shows up in the spreadsheet view of the model window, as in Figure 3-4. Here you can see and edit aspects of the types of entities in your model. In Figure 3-4, the drop-down list for the Initial Picture field is shown, indicating that we decided that our Part entities will be animated as blue balls when the simulation runs. There are several fields



Entity - Basic Process

	Entity Type	Initial Picture	Holding Cost / Hour	Initial VA Cost	Initial NVA Cost	Initial Waiting Cost	Initial Tran Cost	Initial Other Cost	Report Statistics
1	Part	Picture.Blue Ball	0.0	0.0	0.0	0.0	0.0	0.0	☑

Figure 3-4. The Entity Spreadsheet for Model 3-1

for defining the costing data for entity types. A check box at the end lets you ask for Report Statistics on this entity type, including the average and maximum time in system observed for these types of entities during the run. We only have one entity type in our model, but if you had several, each would have its own row in the Entity spreadsheet.

3.3.3 The Process Flowchart Module

Our Process module, which we named `Drilling Center`, represents the machine, including the resource, its queue, and the entity delay time there (part processing, in our case). Open it by double-clicking on its name, and you should see the dialog box in Figure 3-5.

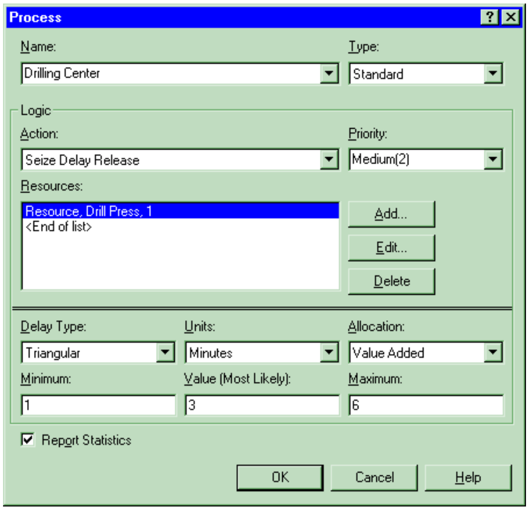
After entering the Name `Drilling Center`, we selected `Standard` as the type, meaning that the logic for this operation will be defined here in this Process module rather than in a hierarchical submodel. Skipping to the bottom of the dialog box, the Report Statistics check box allows you a choice of whether you want output statistics like utilizations, queue lengths, and waiting times in queue.

The Logic area boxes take up most of the dialog and determine what happens to entities in this module.

The Action we chose, `Seize Delay Release`, indicates that we want this module to take care of the entity's seizing some number of units of a Resource (after a possible wait in queue), then Delay for a time representing the service time, and then Release unit(s) of the Resource so that other entities can seize it. Other possible Actions are simply to Delay the entity here for some time (think of it like a red traffic light, after which the entity proceeds), Seize the Resource and then Delay (but not Release the Resource), or Delay and then Release the Resource that previously had been Seized; several Process modules could be strung together to represent a wide range of processing activities.

You can specify different Priorities for entities to Seize the Resource. Here and elsewhere in Arena, lower numbers mean higher priority.

Define the Resource(s) to be Seized or Released in the Resources box; click *Add* to add a Resource to this list. You can define or edit a particular Resource line by double-clicking its line, or selecting it and then clicking *Edit*, to bring up the Resources dialog box, as in Figure 3-6. Here you define the Resource Name and the Quantity of units (e.g., individual servers) that Seizing entities will Seize and that Releasing entities will Release (this is *not* where you specify the number of units of the Resource that exist—that's done in the Resource data module's Capacity field and will be discussed later). Listing more than one Resource means that Seizing entities must Seize the specified Quantity of each Resource before starting to be processed, like a machine and two operators, and Releasing entities will Release the specified Quantity of the corresponding Resources.



Process

Name: Drilling Center

Type: Standard

Logic

Action: Seize Delay Release

Priority: Medium(2)

Resources:
Resource, Drill Press, 1
<End of list>

Delay Type: Triangular

Units: Minutes

Allocation: Value Added

Minimum: 1

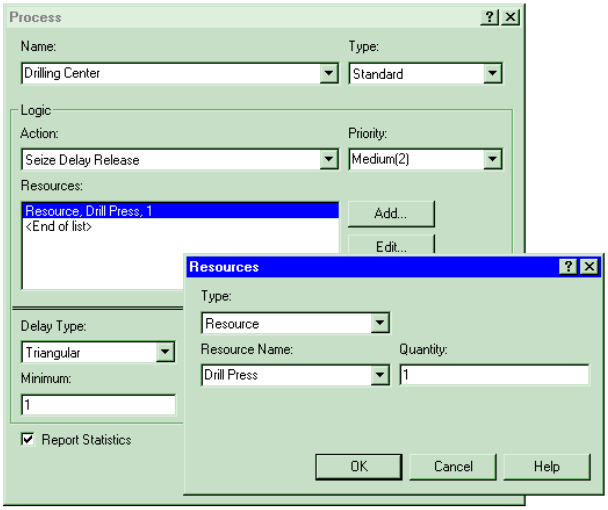
Value (Most Likely): 3

Maximum: 6

☑ Report Statistics

Figure 3-5. The Process Property Dialog Box for Model 3-1

Returning to the Process dialog box in Figure 3-5, the Delay Type drop-down list box offers three probability distributions (Normal, Triangular, and Uniform), a Constant, or a general Expression. The Units field determines the time units for the numerical Delay duration, and the Allocation field relates to how this delay is to be charged. The prompts on the next line change to match your choice of Delay Type. Note that the Expression option for the Delay Type allows you great flexibility in defining the Delay duration, including any other Arena probability distribution; right-clicking in the Expression field lets you bring up the Expression Builder (see Sections 3.4.10 and 4.2.4) to help you.



Process

Name: Drilling Center

Type: Standard

Logic

Action: Seize Delay Release

Priority: Medium(2)

Resources:
Resource, Drill Press, 1
<End of list>

Delay Type: Triangular

Minimum: 1

☑ Report Statistics

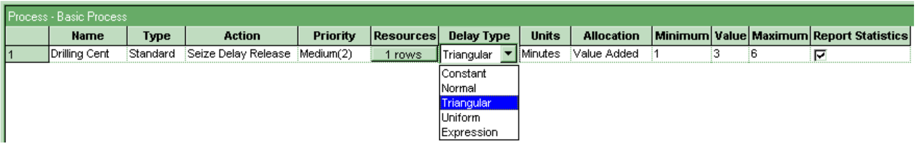
Resources

Type: Resource

Resource Name: Drill Press

Quantity: 1

Figure 3-6. The Resources Dialog Box for Model 3-1



Process - Basic Process

	Name	Type	Action	Priority	Resources	Delay Type	Units	Allocation	Minimum	Value	Maximum	Report Statistics
1	Drilling Cent	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Minutes	Value Added	1	3	6	☑

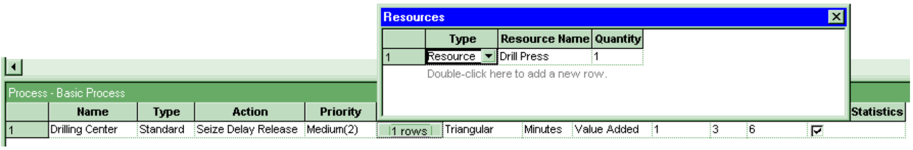
Figure 3-7. The Process Spreadsheet for Model 3-1

Close the Process dialog box with the *Cancel* button; again, if you had made changes that you wanted to retain, you'd click *OK*.

Figure 3-7 illustrates the Process spreadsheet, seen if you select any Process module instance in the flowchart view of the model window, or the general Process module in the Project Bar, with the drop-down box for Delay Type shown (where we've selected `Triangular`). If you had multiple Process modules in your model, there would be a row for each one in the Process spreadsheet. As with Create modules, this provides an alternative way to view simultaneously and edit the fields for your Process module(s). If you click on the "1 Rows" button in the Resources field, a secondary spreadsheet appears (see Figure 3-8) that allows you to edit, add, and delete resources equivalent to the Resources dialog box of Figure 3-6 (you need to click the X button at the upper right of the Resources secondary spreadsheet to close it before you can go on).

3.3.4 The Resource Data Module

Once you've defined a Resource as we've done in this Process module (in our case, we named the resource `Drill Press`), an entry for it is automatically made in the Resource data module; click on it in the Project Bar to view the Resource spreadsheet in Figure 3-9. This spreadsheet allows you to determine characteristics of each Resource in your model, such as whether its Capacity is fixed or varies according to a Schedule (that drop-down list is shown in Figure 3-9, where `Fixed Capacity` has been selected). You can also cause the Resource to fail according to some pattern; try clicking on the "0 Rows" button under the Failures column heading to bring up a secondary spreadsheet for



Resources

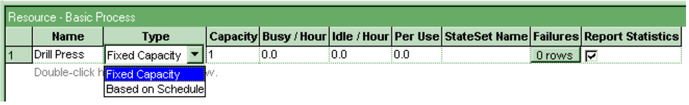
	Type	Resource Name	Quantity
1	Resource	Drill Press	1

Double-click here to add a new row.

Process - Basic Process

	Name	Type	Action	Priority								Statistics
1	Drilling Center	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Minutes	Value Added	1	3	6	☑

Figure 3-8. The Resources Secondary Spreadsheet in the Process Spreadsheet for Model 3-1



Resource - Basic Process

	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	Drill Press	Fixed Capacity	1	0.0	0.0	0.0		0 rows	☑

Double-click h w.

Figure 3-9. The Resource Data Module Spreadsheet for Model 3-1

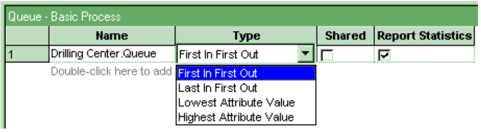
this (the failure pattern is defined in the Failure data module in the Advanced Process panel, which you might have to attach to the Project Bar for your model).

3.3.5 The Queue Data Module

If the `Drill Press` resource is busy when an entity gets to the Process module, the entity will have to queue up. The Queue spreadsheet, seen in Figure 3-10, appears in the spreadsheet view if you select the Queue data module in the Project Bar. Here you can control aspects of the queues in your model (we only have one, named `Drilling Center.Queue`), such as the discipline used to operate it, as shown in the Type list in Figure 3-10 (`First In First Out` is the default and is selected). You could, for instance, rank the queue according to some attribute of entities that reside in it; if you chose Lowest Attribute Value, the queue would be ranked in increasing order of some attribute, and an additional field would show up in the line for this Queue in which you would have to specify the Attribute to be used for ranking.

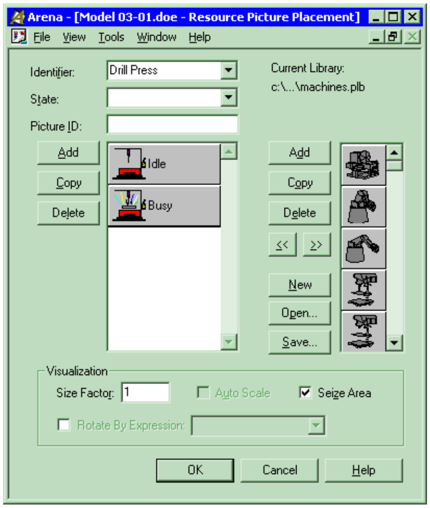
3.3.6 Animating Resources and Queues

Speaking of queues, you might have noticed the ————⊣ just above the Process module in the flowchart view. This is where the queue will be animated, and the Process module acquired this graphic when we specified that we wanted entities to Seize a Resource there.


Queue - Basic Process

	Name	Type	Shared	Report Statistics
1	Drilling Center.Queue	First In First Out	□	☑
	Double-click here to add	First In First Out Last In First Out Lowest Attribute Value Highest Attribute Value		

Figure 3-10. The Queue Data Module Spreadsheet for Model 3-1


Arena - [Model 03-01.doe - Resource Picture Placement]

Identifier: Drill Press

State:

Picture ID:

Current Library:
c:\...\machines.plb

Idle

Busy

Visualization

Size Factor: 1

□ Auto Scale

☑ Seize Area

□ Rotate By Expression:

Figure 3-11. The Resource Picture Placement Dialog Box for Model 3-1

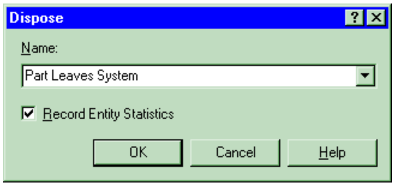
And while we're on the subject of animation, you've no doubt noticed the above and to the right of the Process module and positioned at what will be the head of the queue animation. This is a Resource animation and will change appearance during the simulation depending on whether the `Drill Press` Resource is Idle or Busy. This did not come "free" with the Resource specified in the Process module; rather, we added it to our model via the *Resource* button () in the *Animate* toolbar. Double-click on the icon to get the Resource Picture Placement dialog, as in Figure 3-11. This allows us to pick pictures from libraries (files with extension *.plb* to their name, usually found in the Arena 10.0 folder) to cause the Resource to be animated differently depending on the state it's in. While you can get an idea of how this works at this point, we'll discuss Resource animation in Sections 3.4.8 and 4.3.3.

3.3.7 The Dispose Flowchart Module

The Dispose module represents entities leaving the model boundaries; double-click its name to bring up the dialog box in Figure 3-12; the Dispose spreadsheet is in Figure 3-13. There's not much to do here—just give the module a descriptive Name and decide if you want output on the Entity Statistics, which include things like average and maximum time in system of entities that go out through this module and costing information on these entities.

3.3.8 Connecting Flowchart Modules

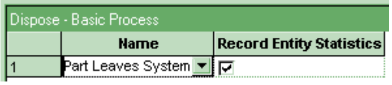
The Create, Process, and Dispose modules are connected (in that order, going left to right) by lines called *Connections*. These establish the sequence that all parts will follow as they progress from one flowchart module to another. To make the Connections, click *Connect* () or equivalently select *Object > Connect*, which changes the mouse pointer to cross hairs. Click on the *exit point* (▶) from the source module and finally on the *entry point* (■) on the destination module (you can make intermediate clicks if you want this connection to be a series of line segments). To help you hit these exit and entry points, which might appear quite small if you're zoomed out to a high altitude, Arena lights up a



Name: Part Leaves System

☑ Record Entity Statistics

Figure 3-12. The Dispose Property Dialog Box for Model 3-1


Dispose - Basic Process

	Name	Record Entity Statistics
1	Part Leaves System	☑

Figure 3-13. The Dispose Spreadsheet for Model 3-1

green box to indicate that you can click now and hit an exit point, and a red box for an entry point.

If you have many connections to make (maybe you placed a lot of flowchart modules to rough out your model), after each one you can right-click in a blank spot of the flowchart view and select Repeat Last Action from the pop-up menu to keep connecting. And if you have *really* a lot of connections to make, you can double-click on the Connect button (or do *Object > Connect* twice in a row) and not even have to bother with the right-click pop-up; when you're done and you want to get out of this, right-click or hit the Escape key (*Esc*).

If *Object > Auto-Connect* is checked, Arena will automatically connect the entry point on a newly placed module to whichever other connect-out module is selected when you place the new module.

If *Object > Smart Connect* is checked, then new Connections are automatically laid out to follow horizontal and vertical directions only, rather than following free-form diagonal directions according to where the connected modules are (unless you make intermediate clicks while drawing a connection, in which case you get the intermediate points and diagonals). This is pretty much a matter of taste and has no bearing on the model's operation or results.

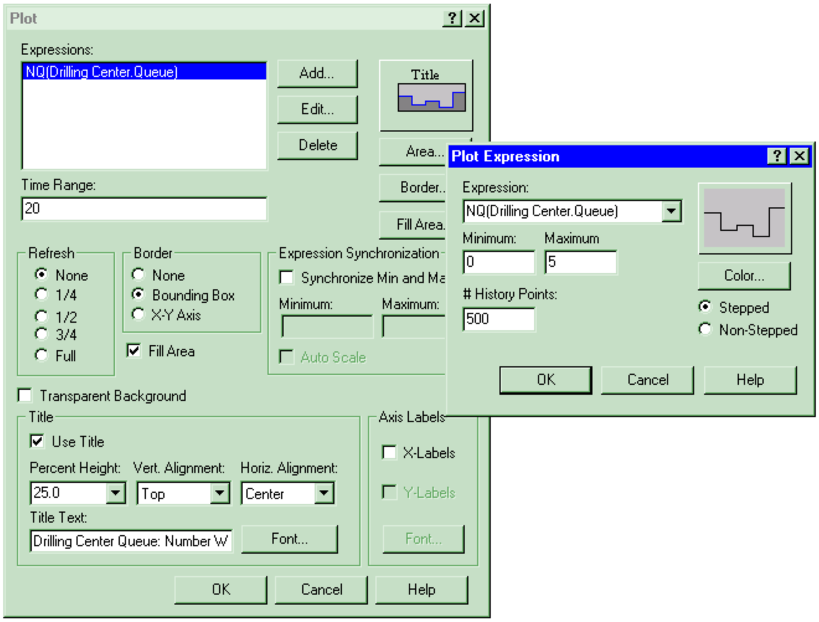
If *Object > Animate Connectors* is checked (or, equivalently, *Animate Connectors* () is pushed in), then Arena will show entity icons (in our case, the blue balls) running down the connections as the transfers happen when the simulation runs. This is just to let you know during the animation that these transfers are occurring—as far as the simulation and statistics collection are concerned, they are happening in zero simulated time (or, equivalently, at infinite speed). We'll show you how to model non-zero travel times between model locations in Section 4.4, including how they're animated.

If for some reason you want to move a connection point (entry or exit) to somewhere else relative to its module, you can do so but you must first right-click on it and select Allow Move from the pop-up.

3.3.9 Dynamic Plots

The two plots were created via the *Plot* button () from the *Animate* toolbar. They'll dynamically draw themselves as the simulation runs, but then disappear when it's over (we'll show you how to make more detailed plots, which also stick around after the run ends, in Section 7.2.1).

Double-click on the top plot (the one for the queue length) to get the Plot dialog box on the left side of Figure 3-14. In the Expressions window, we have just one entry, so we'll get just one curve on this plot. This entry got there by clicking *Add* on the Plot dialog box to bring up a Plot Expression dialog box (which looks like the right side of Figure 3-14 after it's filled out), where we entered in the Expression box there `NQ(Drilling Center.Queue)`, the number of entities in this queue, which Arena will automatically update as the simulation proceeds. Right-clicking in this Expression box allowed us to use the Arena Expression Builder to help us enter the correct text here (more on the Expression Builder in Sections 3.4.10 and 4.2.4).


Plot

Expressions:
NQ(Drilling Center.Queue)

Time Range: 20

Refresh: ● None ○ 1/4 ○ 1/2 ○ 3/4 ○ Full

Border: ○ None ● Bounding Box ○ X-Y Axis

☑ Fill Area

Expression Synchronization: □ Synchronize Min and Ma
Minimum: Maximum:
□ Auto Scale

□ Transparent Background

Title: ☑ Use Title
Percent Height: 25.0 Vert. Alignment: Top Horiz. Alignment: Center
Title Text: Drilling Center Queue: Number W

Axis Labels: □ X-Labels □ Y-Labels

Plot Expression

Expression: NQ(Drilling Center.Queue)
Minimum: 0 Maximum 5
History Points: 500
● Stepped ○ Non-Stepped

Figure 3-14. The Plot and Plot Expression Dialog Boxes for the Queue-Length Plot for Model 3-1

In the Plot dialog box, go ahead and double-click this Expression (or select it and then click *Edit*) to get the filled-out Plot Expression dialog box shown on the right side of Figure 3-14. The Minimum is the smallest *y*-axis value for this curve, and the Maximum is (you guessed it) the maximum *y*-axis value we want to allow for this curve. Here, we know that the model will start off empty and idle, so it's clear that the Minimum should be 0. However, the Maximum is *a priori* pretty much a guess, which might have to be adjusted after you make your run and see what the actual maximum queue length turns out to be; if your guess on the Maximum is too big, you'll squish your plot toward the bottom, and if you underguess the Maximum, you'll decapitate it (Arena can scale automatically, however, as discussed below). The # History Points is the maximum number of corners on the plot you want to allow for at any given time; if you see your plot dissolving from the left as the simulation runs, you should increase this value. Since this is a queue-length plot, it will be piecewise-constant, so the Stepped appearance is appropriate. The *Color* button in the Plot Expression dialog box allows you to change the color of the curve (we chose black), which might be useful if you're plotting several curves for several different Expressions on the same set of axes. Close the Plot Expression dialog with its *Cancel* button to get back to the Plot dialog box.

Back in the Plot dialog box, we entered `20` for the Time Range to allow room on the *x*-axis for a plot over the whole 20-minute simulation run (the units, which we want to be minutes, are in the Base Time Units for the model, as discussed in Section 3.3.11). Since this is wide enough for the whole run, we selected None for Refresh (the fractions under

Refresh are the portion of the plot that shifts off the left edge as needed to make room for that much in the near future on the right edge). We feel safer with a Bounding Box around the whole Border, and we checked Fill Area to flood the area under the curve with a color. If we were plotting several curves in the same graph, we could scale them on a common *y*-axis by checking Synchronize Min and Max and enter our guesses for the Minimum and Maximum here for all curves; in this case, if we next checked Y-Labels, we could then check Auto Scale to get Arena to change the *y*-axis scale as needed for all curves, and then our guesses on the extremes would just be the initial values (this works even if you're plotting just one curve). You can add a Title to your plot; the fields in that area should be self-explanatory (except maybe Percent Height, which is the percent of the total plot height consumed by the Title). The X-Labels option would label the extreme values of the *x*-axis, but we'll do our own custom labeling in Section 3.3.10; the Y-Labels option would display on the vertical axis the lowest and highest values reached by the curve(s) in the plot. The *Area*, *Border*, and *Fill Area* buttons under the plot thumbnail on the right allow you to select colors for those elements (we chose light gray for the background, dark gray for the fill area flooded under the curve, and black for the border—okay, call us boring). Click *Cancel* to close this Plot dialog box.

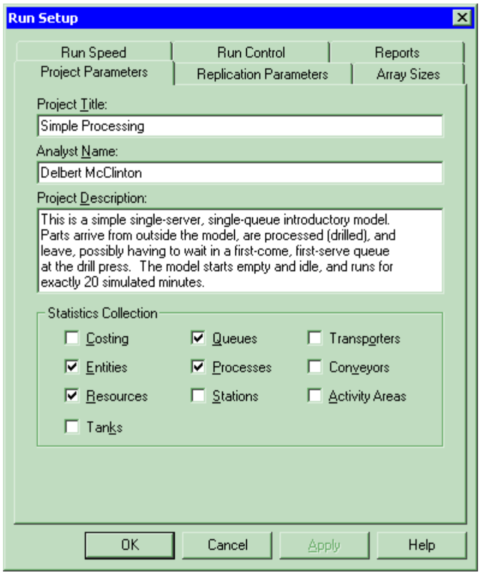
The size of the plot is determined by dragging the handles on its borders. Click (once) on the plot and try this (don't worry, there's an Undo). Actually, you have to specify an initial size of the plot after you fill out the dialogs, but you can change this later, as well as drag it around to relocate it.

The Plot and Plot Expression dialog boxes for the Drill Press: Number Busy plot are similar, so we won't go through them in detail (but go ahead and open them to look at them). The only really different thing is that the Expression whose value we want to plot on the *y*-axis is `NR(Drill Press)`,which we know will always be either 0 or 1, so we specified the Maximum in the Plot Expression dialog box to be `2` to make for an attractive and tasteful graph. As before, we used the Expression Builder in the Expression box of the Plot Expression dialog box to figure out that this is the right name and syntax.

3.3.10 Dressing Things Up

The various labels in the model window, like the title at the upper left and axis labels for the plot, were done via the *Text* button (A) on the *Draw* toolbar. You can control the usual things like font, size, and style from there. To go to a new line in the text use *Ctrl+Enter*. To change the text color, select the text (single-click on it), and use the *Text Color* button (A ▾) to select either the color on the underline there (click on the A in this case) or to choose a different color (click on the ▾ in this case) that will become the new underline color for future text. You can also resize or rotate text by selecting it and dragging the underline bar.

The *Draw* toolbar also has things like boxes, ellipses, polygons, and lines, as well as the means to control their colors and styles, which you can use to decorate your model window, depending on your artistic creativity and talent (as you can see, ours is severely limited). This is how we made the simple shadow box behind the model title in the upper left of the model window. The *Arrange* toolbar and menu have buttons and commands that allow you to manipulate objects, such as grouping, flipping, sending a draw object to



Project Title: Simple Processing

Analyst Name: Delbert McClinton

Project Description:
This is a simple single-server, single-queue introductory model.
Parts arrive from outside the model, are processed (drilled), and
leave, possibly having to wait in a first-come, first-serve queue
at the drill press. The model starts empty and idle, and runs for
exactly 20 simulated minutes.

Statistics Collection

□ Costing	☑ Queues	□ Transporters
☑ Entities	☑ Processes	□ Conveyors
☑ Resources	□ Stations	□ Activity Areas
□ Tanks		

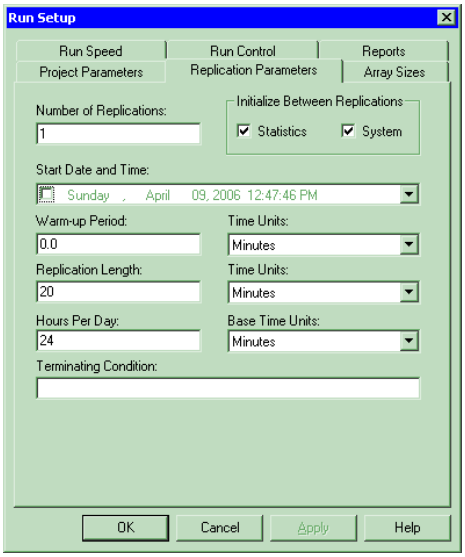
Figure 3-15. The Run* > *Setup* > *Project Parameters Dialog Box for Model 3-1

the back or front of a stack of objects, and so on. We'll talk more about artwork in Sections 3.6.2 and 3.6.3.

3.3.11 Setting the Run Conditions

Things like run length and number of replications are set via *Run* > *Setup*, which brings up a dialog box with five tabbed pages. Figure 3-15 shows the tab for Project Parameters, where we specify a Project Title, Analyst Name, and Project Description, as well as select what kind of output performance measures we want to be told about afterwards. We also chose to document our model internally via entering a brief Project Description.

Figure 3-16 shows the *Replication Parameters* tab of Run Setup, which controls a number of aspects about the run(s). We default the Number of Replications field to `1` (which we'll accept for now since we're only concerned with modeling at the moment, although you know better, from Section 2.6.2). We'll default (that is, not use) the Start Date and Time field, which is for associating a specific calendar date and time with a simulation time of zero. You can also specify a Warm-up Period at the beginning of each replication, after which the statistical accumulators are all cleared to allow the effect of possibly atypical initial conditions to wear off. We specify the Length of Replication to be `20` and select the time unit for that number to be `Minutes`. The Hours Per Day box defaults to `24` (to answer the question you obviously have about this, it could be convenient to define a day to have, say, 16 hours in the case of a two-shift manufacturing operation, if it's customary to think of time in days). The Base Time Units box specifies the "default" time units in which time-based outputs will be reported, as well as how Arena will interpret some time-based numerical inputs that don't have an accompanying Time Units box (such as the Time Range in the Plot dialog box in Figure 3-14). The



Number of Replications: 1

Initialize Between Replications: ☑ Statistics ☑ System

Start Date and Time: □ Sunday , April 09, 2006 12:47:46 PM

Warm-up Period: 0.0 Time Units: Minutes

Replication Length: 20 Time Units: Minutes

Hours Per Day: 24 Base Time Units: Minutes

Terminating Condition:

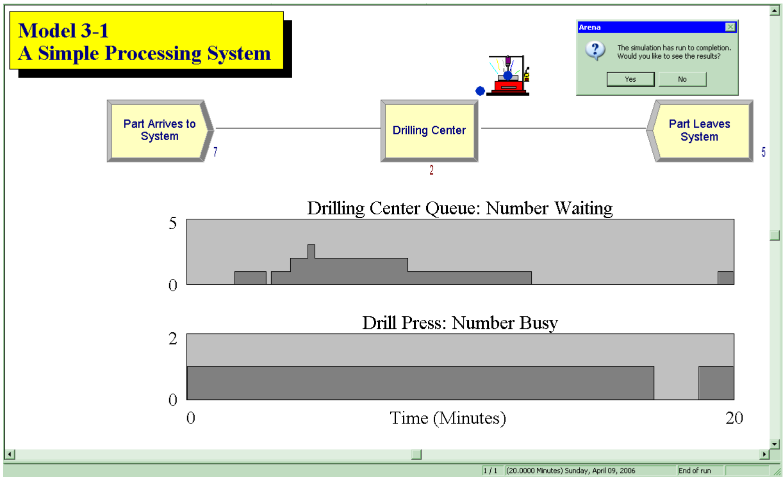
Figure 3-16. The Run > *Setup* > *Replication Parameters Dialog Box for Model 3-1*

Terminating Condition box allows you to establish complex or state-dependent termination rules; see Section 12.5.2 for an example where we want the simulation to keep running until the results achieve the statistical precision we'd like. Model 3-1, however, will simply terminate at time 20 minutes. Close the Run Setup dialog box by pressing *Cancel*.

Speaking of termination, you must specify in every Arena model how you want it to terminate. This is really part of modeling. Arena can't know what you want, so does not include any kind of "default" termination. In fact, in most cases, your simulation will just continue running forever or until you intervene to stop it, whichever comes first. In Section 3.8, we'll show you how to pause and then kill your run if you need to.

3.3.12 Running It

To run the model, click the *Go* button (►) in the *Standard* toolbar (or *Run* > *Go* or press the *F5* key); note that the buttons in this group are similar to those on a video player. The first time you run a model (and after you make changes to it) Arena checks your model for errors (you can do this step by itself with the ✓ button on the *Run Interaction* toolbar, or *Run* > *Check Model* or the *F4* key); if you have errors, you'll be gently scolded about them now, together with receiving some help on finding and correcting them. Then you can watch the model animation run, but you'll have to look fast for a run this short unless your computer is pretty laid back. During the animated run, you see the Part entities (the blue balls) arriving and departing, the Resource Picture changing its appearance as the Resource state changes between Idle and Busy, the Queue changing as Part entities enter and leave it, the digital simulation clock in the Status Bar advancing, and the plots being drawn. The counters next to the flowchart modules display different quantities depending



The simulation has run to completion.
Would you like to see the results?

1 / 1 (20.0000 Minutes) Sunday, April 09, 2006 End of run

Figure 3-17. Ending Animation State of Model 3-1

on the module type. For the Create module, it's the number of entities that have been created. For the Process module, it's the number of entities that are currently in process there (in service plus in queue), and for the Dispose module, it's the number of entities that have left the system. There are other ways to run your model, and we'll discuss some of them in Section 3.8.

The final state of things should look something like Figure 3-17, except that we've moved the Arena dialog box (asking about seeing the results) out of the way. The plots display the same information as in Figure 2-3 from the hand simulation. The clock in the Status Bar is frozen at its final value, and at that point we see that the Resource is Busy operating on one part, with one part waiting in queue (in agreement with the final state in the hand simulation in Section 2.4.3 and the bottom row of Table 2-2). The final values of the counters next to the flowchart modules are also as they were at the end of the hand simulation in Section 2.4.3.

The Arena box that appears at the end of the run asks if you'd like to see the summary results, which we'll do next in Section 3.3.13. After you look at those reports (or if you choose not to), your model window will appear to be "hung" and you can't edit anything. That's because you're still in *run mode* for the model, which gives you a chance to look at the plots and the final status of the animation. To get out of run mode and back to being able to edit, you have to click *End* (■), just like on a video player.

3.3.13 Viewing the Reports

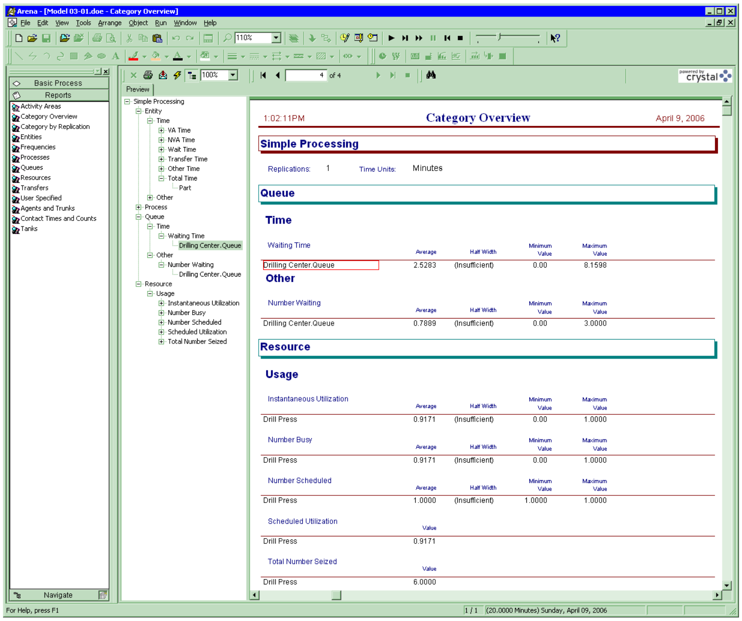
If you'd like to see the numerical results now, click *Yes* in the Arena box that appears at the end of your simulation, as shown near the top right of Figure 3-17. This opens a new reports window in the Arena window (separate from your model window). The Project Bar now displays the Reports panel, which lists several different Reports you can view, such as Category Overview, Category by Replications, and Resources. Clicking on each of these reports in the Project Bar opens a separate report window in the Arena window (use the Arena Windows menu to see what you have open). Don't forget to close these report windows when you're done viewing them since they don't go away on their own if you simply go back to your model window; if you change your model and then re-run it, you might wind up with several different report windows open and it could get confusing to figure out which one goes with which variant of your model. Actually, when making changes in your model to investigate the effects of different parameter settings or assumptions, you probably should change the name of your *.doe* file slightly, since Arena will simply overwrite previous results to the same report file name if you don't, and you'll lose your previous results. (The Arena Process Analyzer, discussed in Sections 3.6.1 and 6.5, provides a far better way to manage the activity of running multiple variants or scenarios of your model and keeping track of the results for you.)

The default Arena installation automatically brings up the Category Overview Report, which gives you access to most of the results; the other reports listed in the Project Bar repeat a lot of this, but provide more detail. Down the left edge of the report window itself is a tree, which you can expand by clicking the + signs in it (and re-contract with the – signs), giving you a kind of hyperlinked outline to this entire report. The report itself is organized into pages, through which you can browse using the ▶, ▶|, |◀, and ◀ buttons at the top left of the report window. If you want to print some or all of the pages in the report being displayed, click the 🖶 button in the report window (not the similar-looking button above it in the Arena window, which will be dimmed and thus inactive anyway if the report window is active). If you'd like to export the report to a different file, including several common spreadsheet and word-processor formats, click 📥 in the report window and follow the directions there.

But if you're looking for just a few specific results, it's better to click around on the +'s and –'s in the tree outline in the report window. For instance, to see what happened with the queue during our simulation, we clicked down a sequence of + signs into the Queue section of the report (specifically, Simple Processing → Queue → Time → Waiting Time → Drilling Center.Queue), eventually getting to the Waiting Time information, as shown in Figure 3-18. What's selected in the tree is displayed and outlined in the report to the right, and we see from that line that the average waiting time in queue was 2.5283 (the report reminds us that the Base Time Units are minutes), and the maximum waiting time was 8.1598 minutes (both of which agree with the hand-simulation results in Section 2.4.4). A little further down in this view of the report window, under "Other" (to which we could jump directly by clicking on its entry in the tree), we see that the average number waiting (i.e., length) of the queue was 0.7889 part, and the maximum was 3, both of which agree with our hand simulation in Section 2.4.4.

Browse through this report and note that the output performance measures in Table 2-3 are all here, as well as a lot of other stuff that Arena collected automatically (we'll talk more about these things later). By following the branches of the tree as indicated below, you'll find, for example:

- Simple Processing → Entity → Time → Total Time → Part: The average total time in system was 6.4397 minutes, and the maximum was 12.6185 minutes.



1:02:11PM Category Overview April 9, 2006

Simple Processing

Replications: 1 Time Units: Minutes

Queue

Time

Waiting Time	Average	Half Width	Minimum Value	Maximum Value
Drilling Center.Queue	2.5283	(Insufficient)	0.00	8.1598

Other

Number Waiting	Average	Half Width	Minimum Value	Maximum Value
Drilling Center.Queue	0.7889	(Insufficient)	0.00	3.0000

Resource

Usage

Instantaneous Utilization	Average	Half Width	Minimum Value	Maximum Value
Drill Press	0.9171	(Insufficient)	0.00	1.0000

Number Busy	Average	Half Width	Minimum Value	Maximum Value
Drill Press	0.9171	(Insufficient)	0.00	1.0000

Number Scheduled	Average	Half Width	Minimum Value	Maximum Value
Drill Press	1.0000	(Insufficient)	1.0000	1.0000

Scheduled Utilization	Value
Drill Press	0.9171

Total Number Seized	Value
Drill Press	6.0000

Figure 3-18. Part of the Category Overview Report for Model 3-1

- Simple Processing → Resource → Usage → Instantaneous Utilization → Drill Press: The utilization of the drill press was 0.9171 (i.e., it was busy 91.71% of the time during the simulation). The different measures of utilization are discussed in Section 4.2.5.
- Simple Processing → Process → Other → Number In → Drilling Center: During the simulation, seven entities entered the drilling center Process module.
- Simple Processing → Process → Other → Number Out → Drilling Center: During the simulation, five entities left the drilling center Process module (two fewer than entered, which is how many parts were in the drilling center at termination). This value of 5 also represents the total production in this model, since parts exit the system immediately after leaving the drilling center.
- Simple Processing → Entity → Time → Wait Time → Part: Of the five parts that exited the system, their average wait time in all queues (of which there's only one in this model) was 3.0340 minutes, and the maximum was 8.1598 minutes. The reason the average here differs from the average waiting time in queue = 2.5283 is that the 3.0340 here counts the waiting times of only those five parts that exited the system, while the 2.5283 counted the waiting times of all 6 parts that left the queue. The two maxima, however, are the same in this run since that maximum was achieved earlier in the simulation (the maxima would not necessarily always be equal).
- Simple Processing → Entity → Other → WIP → Part: The work in process (WIP) averaged 1.7060 parts and hit a maximum of four parts at some point(s) in time.

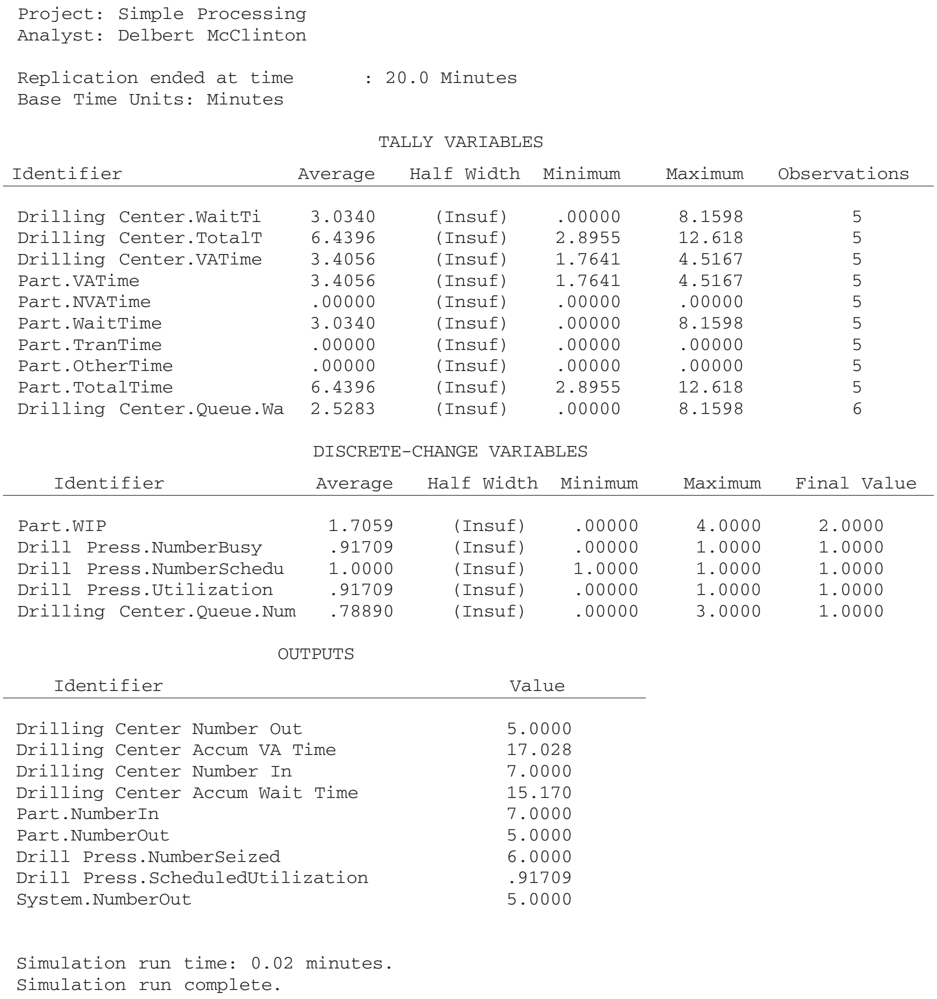
Many of the numbers in the reports (as in our hand simulation in Chapter 2) can be classified as *tally, time-persistent,* or *counter* statistics:

- *Tally statistics* are those that result from taking the average, minimum, or maximum of a list of numbers. For example, the average and maximum total time in system (6.4397 and 12.6185 minutes, respectively) are tally statistics since they're respectively the average and maximum of the total times in system of the five parts that left the system during the simulation. Tally statistics are sometimes called *discrete-time statistics* since their "time" index (1, 2, 3, ...) is a discrete indexing of the time order in which the observations were made.
- *Time-persistent statistics* are those that result from taking the (time) average, minimum, or maximum of a plot of something during the simulation, where the *x*-axis is *continuous* time. Time-persistent averages involve the accumulated area under the plotted curve (i.e., an integral). The average and maximum of the number of parts in queue (0.7889 and 3 parts, respectively) are time-persistent statistics, as is the instantaneous utilization of the drill press (0.9171). Time-persistent statistics are also known as *continuous-time statistics*.
- *Counter statistics*, as the name suggests, are accumulated sums of something. Often, they are simply nose counts of how many times something happened, like the number of parts that left the drilling center (five in our simulation) or that entered the drilling center (seven). But they could be accumulations of numbers that are not all equal to 1; for instance, the accumulated waiting time at the drilling center was 15.1700 minutes, representing the sum (not average) of the waiting times observed in queue there. In the Category Overview report, this number can be found via Simple Processing → Process → Accumulated Time → Accum Wait Time → Drilling Center. Another counter statistic is at Simple Processing → Resource → Usage → Total Number Seized → Drill Press, where we see that the drill press resource was used (either to completion for the part or just started) six times.

If you close a report window, you can view it later as long as you don't delete (or overwrite) the Microsoft® Access database file that Arena creates as it runs. This Access file is named `model_filename.mdb`, where `model_filename` is what you named your *.doe* model file (so in our case, the Access file is named `Model 03-01.mdb`). On the Project Bar, select Reports and then click on the report you want (such as Category Overview) to view it again. The way this works is that Arena uses third-party software called Crystal Reports® from Business Objects (http://www.businessobjects.com) to read the Access database file, extract the useful stuff from it, and then display it to you in the report-window format we've described above.

Right now you might be thinking that this report structure is pretty serious overkill just to get a handful of numbers out of this small model, and it probably is. However, in large, complicated models it's quite helpful to have this structure to organize the myriads of different output numbers and to help you find things and make some quick comparisons and conclusions.

In addition to the reports described above, Arena produces a very compact (to the point of being cryptic) report of many of the simulation results, as a plain ASCII text file named `model_filename.out` (so `Model 03-01.out` for us), as shown in Figure 3-19. Some of the labels are a little different; e.g., "DISCRETE-CHANGE VARIABLES" are the same as time-persistent statistics. You'll find in this file many of the numbers in the reports we talked about above (minor roundoff-error discrepancies are possible), and even a few that are not in the reports above (like the number of observations used for tally statistics). For some purposes, it might be easier and faster to take a quick look at this rather than the report structure we described above. However, the order and arrangement and labeling of things here is decidedly user-hostile; this format



```
 Project: Simple Processing
 Analyst: Delbert McClinton

 Replication ended at time      : 20.0 Minutes
 Base Time Units: Minutes

                                   TALLY VARIABLES

Identifier                 Average   Half Width  Minimum    Maximum   Observations
__________________________________________________________________________________

 Drilling Center.WaitTi     3.0340     (Insuf)     .00000     8.1598          5
 Drilling Center.TotalT     6.4396     (Insuf)    2.8955      12.618          5
 Drilling Center.VATime     3.4056     (Insuf)    1.7641      4.5167          5
 Part.VATime                3.4056     (Insuf)    1.7641      4.5167          5
 Part.NVATime               .00000     (Insuf)     .00000     .00000          5
 Part.WaitTime              3.0340     (Insuf)     .00000     8.1598          5
 Part.TranTime              .00000     (Insuf)     .00000     .00000          5
 Part.OtherTime             .00000     (Insuf)     .00000     .00000          5
 Part.TotalTime             6.4396     (Insuf)    2.8955      12.618          5
 Drilling Center.Queue.Wa   2.5283     (Insuf)     .00000     8.1598          6

                             DISCRETE-CHANGE VARIABLES

    Identifier                 Average   Half Width  Minimum    Maximum   Final Value
__________________________________________________________________________________

 Part.WIP                     1.7059     (Insuf)     .00000     4.0000     2.0000
 Drill Press.NumberBusy       .91709     (Insuf)     .00000     1.0000     1.0000
 Drill Press.NumberSchedu    1.0000      (Insuf)    1.0000      1.0000     1.0000
 Drill Press.Utilization      .91709     (Insuf)     .00000     1.0000     1.0000
 Drilling Center.Queue.Num    .78890     (Insuf)     .00000     3.0000     1.0000

                          OUTPUTS

    Identifier                              Value
___________________________________________________

 Drilling Center Number Out                5.0000
 Drilling Center Accum VA Time             17.028
 Drilling Center Number In                 7.0000
 Drilling Center Accum Wait Time           15.170
 Part.NumberIn                             7.0000
 Part.NumberOut                            5.0000
 Drill Press.NumberSeized                  6.0000
 Drill Press.ScheduledUtilization          .91709
 System.NumberOut                          5.0000


 Simulation run time: 0.02 minutes.
 Simulation run complete.
```

Figure 3-19. SIMAN Summary Report File (Model 03-01.out) *for Model 3-1*

is actually a leftover from earlier versions of Arena and in fact goes back to the early 1980s and the underlying SIMAN simulation language. However, if, out of nostalgia or a mania for compactness, you want this to be the default report that Arena gives you if you ask for one after the run ends, select *Run* > *Setup* > *Reports* and in the Default Report list scroll down and pick SIMAN Summary Report (*.out* file).

The exact meaning of report labels like Average, Minimum, Maximum, and Time-Average should be clear to you by now. But the reports also refer here and there to Half Widths (though we never get numbers for them in this model and are just scolded that we're somehow Insufficient). As you might guess, these will be half widths of confidence intervals (they'll be at level 95%) on the expected value of the corresponding performance measure, provided that our simulation produces adequate data to form them.

If we were doing more than one replication (which we're not in this case), Arena would take the summary results for an output performance measure from each replication, average them over the replications, compute the sample standard deviation from them, and finally compute the half width of a 95% confidence interval on the expected value of this performance measure. This is exactly what we did by hand in Section 2.6.2 for several of the output performance measures there, as given in Table 2-4. Exercise 3-1 asks you to do this with Arena to reproduce (except maybe for round-off) the results in Table 2-4.

If we're interested in the *long-run* (or *steady-state*) behavior of the system after any initialization effects have worn off, we might choose to make just one (really) long replication or run; this issue is taken up in Section 7.2.3. If we do so, we might see half-width numbers in the reports if our run is long enough, even though we're doing only one replication. Arena tries to compute these half widths by breaking the single long run into batches whose means serve as stand-ins for truly independent summary results over replications for making confidence intervals. We'll talk about this more in Section 7.2.3, including a discussion of exactly how these half widths are computed from the simulation output data. The reason you might get "Insufficient" (or, sometimes, "Correlated") instead of a numerical value for the half width is that the run must be long enough to provide data in sufficient quantity and of adequate quality to justify the validity of the half widths; if your run is not long enough for this, Arena simply declines to deliver a value on the theory that a wrong answer is worse than no answer at all.

3.4 Building Model 3-1 Yourself

In this section, we'll lead you through the construction of Model 3-1 from scratch. What you end up with might not look exactly like our Model 3-1 cosmetically, but it should be functionally equivalent and give you the same results.

Before embarking on this, we might mention a couple of little user-interface functions that often come in handy:

- Right-clicking in an empty spot in the flowchart view of the model window brings up a small pop-up box of options, one of which is to repeat the last action (like placing a module in your model, of which you may need multiple instances). This can obviously save you time when you have repetitive actions (though it won't make them any more interesting). Other options here include some views, as well as running or checking the model.

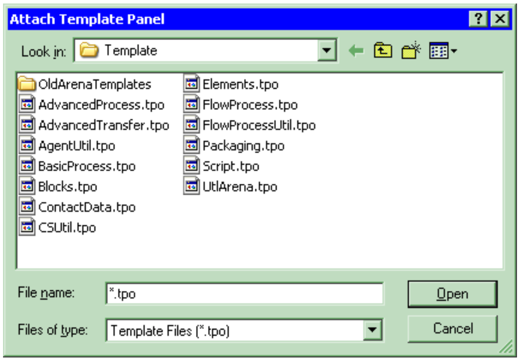
- *Ctrl+D* or pressing the *Ins* key duplicates whatever is selected in the flowchart view of a model window, offsetting the copy a little bit. You'll then probably want to drag it somewhere else and do something to it.

3.4.1 New Model Window and Basic Process Panel

Open a new model window with 🗋 (or *File > New* or *Ctrl+N),* which will automatically be given the default name `Model1`, with the default extension *.doe* when you save it. You can change this name when you decide to save the contents of the model window. Subsequent new model windows during this Arena session will get the default names `Model2`, `Model3`, and so on. You might want to maximize your new model window within the Arena window by clicking □ near the model window's upper-right corner (if it's not already maximized).

Next, attach the panels you'll need if they're not already there in the Project Bar. For this model, we need only the Basic Process panel, which is a file called `BasicProcess.tpo`, typically in the Template folder under the Arena 10.0 folder. Click 📂 (or *File > Template Panel > Attach* or right-click on the Project Bar and select *Attach*) to open the Attach Template Panel dialog in Figure 3-20, where you open the Basic Process panel `BasicProcess.tpo` (click on it, then click the *Open* button, or just double-click it). You can tell Arena to attach certain panels automatically to the Project Bar of new model windows via *Tools > Options > Settings* by typing the panels' file names (including the *.tpo* extension) into the Auto Attach Panels box.

The attached panel will appear in the Project Bar, with icons representing each of the modules in this panel. Right-click in the panel to change the icon size or to display text only for the modules. Right-clicking in a panel also allows you to detach it if you accidentally attached the wrong one (you can also detach the visible panel via 📂 or *File > Template Panel > Detach*). You can detach a panel even if you've placed modules from


Attach Template Panel

Look in: Template

OldArenaTemplates
AdvancedProcess.tpo
AdvancedTransfer.tpo
AgentUtil.tpo
BasicProcess.tpo
Blocks.tpo
ContactData.tpo
CSUtil.tpo
Elements.tpo
FlowProcess.tpo
FlowProcessUtil.tpo
Packaging.tpo
Script.tpo
UtlArena.tpo

File name: *.tpo

Files of type: Template Files (*.tpo)

Figure 3-20. The Attach Template Panel Dialog

that panel in your model. If your display isn't tall enough to show the panel's full height, use the scroll bar at its right to get to it all.

3.4.2 Place and Connect the Flowchart Modules

This model requires one instance of each of three flowchart modules: Create, Process, and Dispose. To add an instance of a flowchart module to your model, drag its icon from the Project Bar into the flowchart view of the model window and drop it about where you want it (you can always drag things around later). To help you line things up, remember Grid (∷), Snap (⊡), Snap to Grid (⊞), Rulers (▭), Guides (┼), Glue (⊡), *Arrange > Align*, and *Arrange > Distribute* from Section 3.2.3.

If you have *Object > Auto-Connect* checked and you dragged the modules into your model in the order mentioned above (without de-selecting a module after dropping it in), Arena will connect your modules in the correct order; if you have *Object > Smart Connect* checked, those connections will be oriented horizontally and vertically.

Figure 3-21 shows how these modules look in the flowchart view of the model window just after we placed them (both Connect toggles were checked), with the Dispose model selected since it was the last one we dragged in. If you did not have *Object > Auto-Connect* checked, you'll need to connect the modules yourself; to do so, use *Connect* (⊞) on the modules' exit (►) and entry (■) points, as described in Section 3.3.8. Recall as well from Section 3.3.8 that, during the animation, you'll see entities running along the connections if the *Animate Connectors* button (⊞) is pressed (or equivalently, *Object > Animate Connectors* is checked) just to let you know that they're moving from one flowchart module to the next. However, this movement is happening in zero simulated time; Section 4.4 describes how to represent positive travel times in your model.

3.4.3 The Create Flowchart Module

Open the "raw" Create module by double-clicking it to get the dialog box in Figure 3-22, where we need to edit several things. We first changed the Name of this instance of the Create module from the default, Create 1, to `Part Arrives to System`, and we changed the Entity Type from the default to `Part`. In the Time Between Arrivals area of the dialog box, we accepted the default Random (Expo) for the Type, changed the default

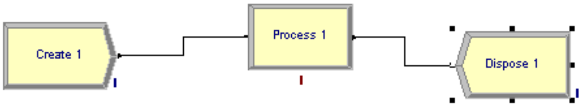
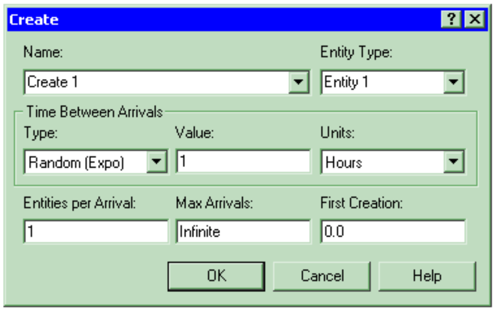


Figure 3-21. Initial Placement of the Flowchart Modules



Create

Name: Create 1

Entity Type: Entity 1

Time Between Arrivals

Type: Random (Expo)

Value: 1

Units: Hours

Entities per Arrival: 1

Max Arrivals: Infinite

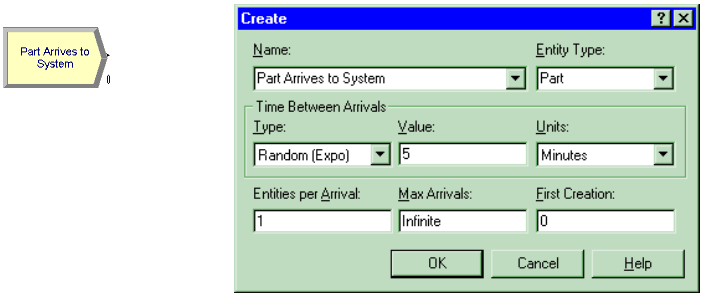
First Creation: 0.0

Figure 3-22. The Create Dialog Box

Value of 1 to 5, and selected Minutes as the Units from the drop-down list there (note that the default is Hours). We accepted the defaults for the three boxes in the bottom row of the dialog box and clicked *OK* to save our changes; at this point, the Create dialog box should look like Figure 3-2. Recall that we can also view and edit flowchart modules via their spreadsheet view, as detailed in Section 3.3; the completed spreadsheet for this Create module was shown earlier in Figure 3-3. Note that the new name of this Create module now appears in its shape in the flowchart view of the model window.

3.4.4 Displays

As we introduce new modules and new concepts, we'll try to lead you through each dialog box (or, equivalently, spreadsheet). Even though the Create module is fairly simple, the above description was fairly lengthy. To convey this more compactly, we'll use visuals, called *Displays*, as shown in Display 3-1. Note that there are three parts to this display. The upper-right portion has the filled-in dialog box (which in Display 3-1 is the same as Figure 3-2). In some cases, it may show several related dialogs. The upper left shows the module with which the dialog box is associated. Later, this portion may also show buttons we clicked to get the dialog box(es) shown on the upper right. The bottom portion of the display is a table showing the actions required to complete the dialog box(es). The left column of the table defines the dialog boxes or prompts, and the right column contains the entered data or action (italics) like checking boxes or pressing command buttons. In general, we'll try to provide the complete display when we introduce a new module or a new secondary dialog box of a module we've already covered. For modules that aren't new, we'll normally give you only the table at the bottom of the display, which should allow you to recreate all the models we develop easily. There may be more items or prompts in a dialog box than we show in a display; for those we accept the defaults.



Name: Part Arrives to System
Entity Type: Part
Time Between Arrivals
Type: Random (Expo)
Value: 5
Units: Minutes
Entities per Arrival: 1
Max Arrivals: Infinite
First Creation: 0

Name	`Part Arrives to System`
Entity Type	`Part`
Time Between Arrivals area	
Type	`Random (Expo)`
Value	`5`
Units	`Minutes`

Display 3-1. The Completed Create Dialog Box

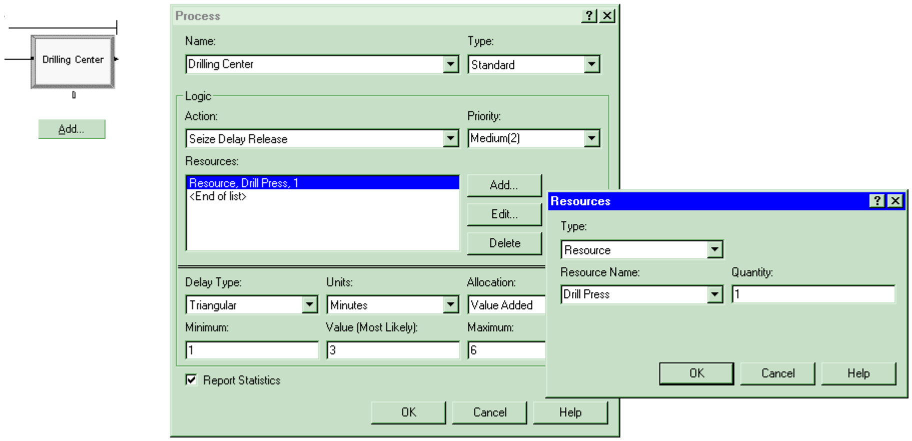
This might be a good time to save your model; choose a name different from ours (which was `Model 03-01.doe`), or put it in a different folder.

3.4.5 The Entity Data Module

Now that you've defined the `Part` Entity Type in your Create module, you might want to say some things about it. The only thing we did was change its initial animation picture from the default "Report" (▤) to "Blue Ball" (●). To do this, click the Entity data module in the Project Bar to display it in the spreadsheet view of the model window (see Figure 3-4 in Section 3.3.2). In the Initial Picture list, click `Picture.Blue Ball` (clicking the arrow opens the list for scrolling).

3.4.6 The Process Flowchart Module

Display 3-2 indicates what's needed to edit the Process flowchart module. Since the Action you specify includes Seizing a Resource, you *must* hit the *Add* button to define which resource is to be Seized; this brings up the Resources secondary dialog box, which is shown in the display as well. You also might want to make the area for the queue animation longer; click the queue animation (the ———⊣), then drag its left end back to the left (hold down the *Shift* key to constrain its angle to horizontal, vertical, or 45° diagonal).



Name: Drilling Center
Type: Standard
Logic
Action: Seize Delay Release
Priority: Medium(2)
Resources:
Resource, Drill Press, 1
<End of list>
Delay Type: Triangular
Units: Minutes
Allocation: Value Added
Minimum: 1
Value (Most Likely): 3
Maximum: 6
☑ Report Statistics

Type: Resource
Resource Name: Drill Press
Quantity: 1

Name	`Drilling Center`
Action	`Seize Delay Release`
Resources (secondary dialog via *Add* button)	
Type	`Resource`
Resource Name	`Drill Press`
Quantity	`1`
Delay Type	`Triangular`
Units	`Minutes`
Minimum	`1`
Value	`3`
Maximum	`6`

Display 3-2. The Completed Process Module

3.4.7 The Resource and Queue Data Modules

Once you've defined this Process module, your model has both a Resource and a Queue, with names you specified (the name for a Queue is `whatever.Queue`, where `whatever` is the Name you gave to the Process module for this Queue). If you need to specify non-default items for a Resource or a Queue (which we don't in this model), you'd use the Resource and Queue data modules, as described in Sections 3.3.4 and 3.3.5.

3.4.8 Resource Animation

While not necessary for the simulation to work, it's usually nice to animate your Resources. This lets you show their state (just Idle vs. Busy for this model), as well as show entities "residing" in them during processing.

Click on the *Resource* button (▮) in the *Animate* toolbar to bring up the Resource Picture Placement dialog box. To associate this picture to the Resource, click the Identifier arrow to choose the Resource Name, `Drill Press`. In the list of pictures on the left side of the dialog box, select `Inactive` and then press the *Delete* button to its left; do the same for `Failed`.

Now, if your Drill Press really looks like a white square when it's Idle and a green square when it's Busy, you're all set...but probably it doesn't. If you double-click on one square you get into the Picture Editor where you can try your hand at drawing your drill press in both its corresponding states. Or, if you have a graphics file somewhere depicting your drill press (maybe from a digital camera), you could copy and paste it into the Picture Editor.

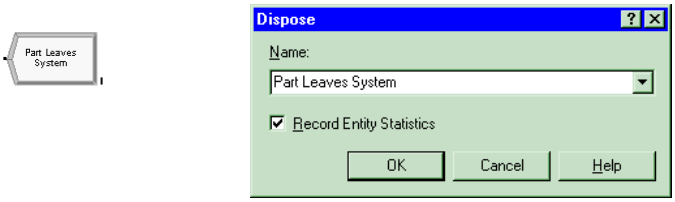
Instead, let's pick some attractive artwork out of one of Arena's picture libraries, on the theory that if you were any good at art or photography you probably wouldn't be reading *this* book. So go ahead and close the Picture Editor to get back to the Resource Picture Placement dialog box. To open an Arena picture library, click *Open* along the right column and navigate to the Arena 10.0 folder where you'll see a list of files with *.plb* file name extensions. Open `Machines.plb` to view a gallery on the right of stunning creations from the rustbelt collection. Scroll down a bit and click, then click the white-square Idle button on the left (depressing it), and then click << to copy this picture to the left where it becomes the Idle picture instead of the white square. Similarly, copy on the right to become the Busy picture on the left. Finally, check the Seize Area box at the bottom so that the Part being processed will show up in the animation. Your Resource Picture Placement dialog box should look like Figure 3-11 in Section 3.3.6 at this point. We'll have more to say about Resource pictures in Section 4.3.3.

3.4.9 The Dispose Flowchart Module

The final flowchart module is the Dispose; Display 3-3 shows how to edit it for this model (the only thing to do is improve on the default Name).

3.4.10 Dynamic Plots

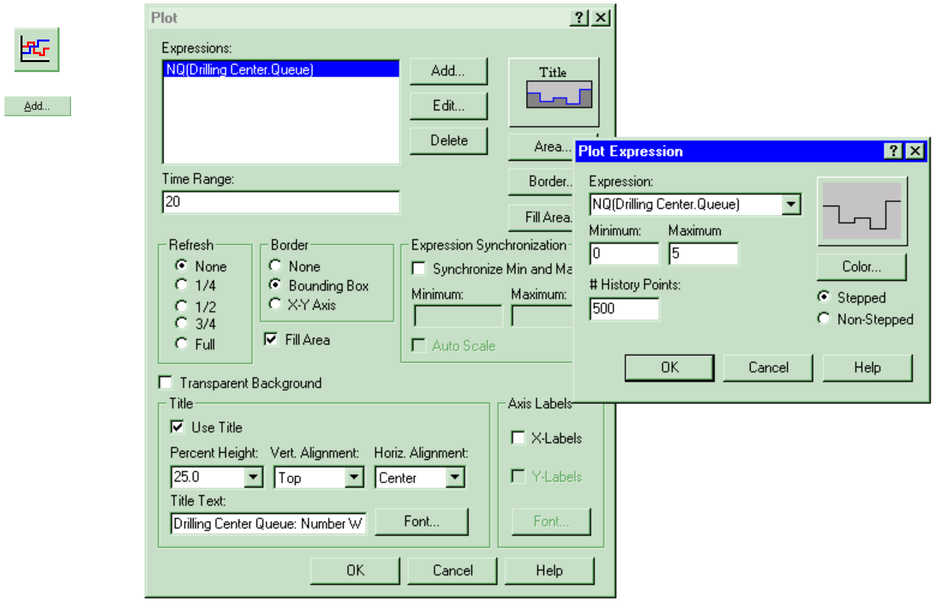
We described most of the entries and properties of the two animated plots earlier in Section 3.3.9. To make such a plot from scratch, press the *Plot* button ( ) on the *Animate* toolbar to get a blank Plot dialog box, and then proceed as indicated in Display 3-4. Remember that initially you might have to guess at the Maximum *y*-axis value in the Plot Expression dialog box, and perhaps adjust it after you have a feel for the results. Also, when you're done filling in the dialog and click *OK*, your mouse pointer becomes cross hairs; click to determine the location of one corner of the plot, then again to determine the opposite corner (of course, you can resize and reposition the plot later).



Name:
Part Leaves System
☑ Record Entity Statistics

Name	`Part Leaves System`

Display 3-3. The Completed Dispose Module



Add...

Plot

Expressions:
NQ(Drilling Center.Queue)

Add... Edit... Delete

Title Area... Border... Fill Area...

Time Range:
20

Refresh
● None
○ 1/4
○ 1/2
○ 3/4
○ Full

Border
○ None
● Bounding Box
○ X-Y Axis

☑ Fill Area

Expression Synchronization
□ Synchronize Min and Ma
Minimum: Maximum:
□ Auto Scale

□ Transparent Background

Title
☑ Use Title
Percent Height: 25.0
Vert. Alignment: Top
Horiz. Alignment: Center
Title Text: Drilling Center Queue: Number W
Font...

Axis Labels
□ X-Labels
□ Y-Labels
Font...

OK Cancel Help

Plot Expression

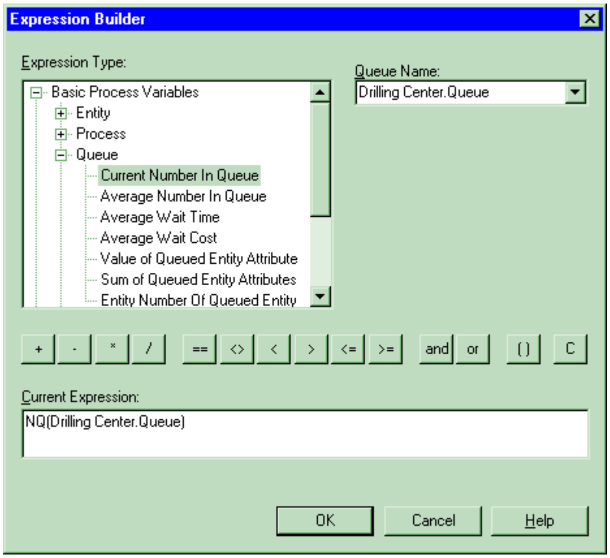
Expression: NQ(Drilling Center.Queue)
Minimum: 0
Maximum: 5
History Points: 500
Color...
● Stepped
○ Non-Stepped

OK Cancel Help

Plot Expressions (secondary dialog via *Add* button)	
Expression	`NQ(Drilling Center.Queue)`
Maximum	`5`
Color	*black*
Plot	
Time Range	`20`
X-Labels	*clear* (i.e., *uncheck*)
Title - Use Title	*select*
Horiz. Alignment	*Center*
Title Text	`Drilling Center Queue: Number Waiting`

Display 3-4. The Completed Plot Dialog Box for the Queue-Length Plot

While it's perfectly legal just to type in `NQ(Drilling Center.Queue)` manually for the Expression in the Plot Expressions secondary dialog box, you'd first have to know that this is the right thing to type in, which you very well might *not* know. This is one of many places in Arena where you can enter a general algebraic expression, and to do so correctly, you often need to know the exactly correct names of various objects in your model (like `Drilling Center.Queue`) and built-in Arena functions (like `NQ`, which returns the current number of entities in the queue named in its argument). To help you out with this memory-intensive task (that's your memory, not your computer's), Arena provides something called the Expression Builder, which you can access by right-clicking in any box that calls for some kind of Expression, and then



Queue Name: Drilling Center.Queue

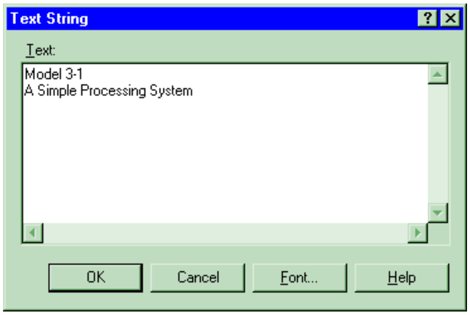
Current Expression: NQ(Drilling Center.Queue)

Figure 3-23. The Expression Builder for a Queue-Length Expression

selecting Build Expression. Figure 3-23 shows the Expression Builder window after we expanded the Expression Type tree on the left to get at what we want for the queue-length plot. The label on the box in the upper right (now Queue Name) will change depending on what we select in the Expression Type tree; here it provides a drop-down list where we can specify the queue for which we want to know the Current Number in Queue (we only have one Queue in this model so it's a short list). The Current Expression box at the bottom is the Expression Builder's answer, and clicking *OK* at this point pastes this text back into the Expression box from where we started with our right-click. You can still edit and modify the expression there, as you can in the Current Expression box of the Expression Builder, perhaps using its calculator-type buttons for arithmetic operations, and using the tree at the left to look up the names of other quantities.

The plot for the number busy at the drill press is quite similar, with only two differences from Display 3-4, both of which are in the Plot Expressions secondary dialog box. First, make the Expression `NR(Drill Press)`, an expression you can discover with the Expression Builder via the Expression Type path Basic Process Variables → Resource → Usage → Current Number Busy, and select `Drill Press` (the only entry) under Resource Name. Finally, make the Maximum `2` since we know this curve will always be at height zero or one.

To make the two plots visually harmonious, we made them the same size and aligned them vertically.



Text:

Model 3-1
A Simple Processing System

Figure 3-24. The Text String Dialog Box

3.4.11 Window Dressing

Model 3-1 has several text labels in the flowchart view of the model window to help document things as well as indicate what's what during the animation. These were produced via the *Text* button (**A**) on the *Draw* toolbar, which opens a Text String dialog box like the one in Figure 3-24. Type in your text (use *Ctrl+Enter* to go to a new line), perhaps change its font (Times Roman, Arial, etc.), font style (Italics, Bold, etc.), and size via the *Font* button, and then click *OK*. Your mouse pointer becomes cross hairs, which you click to place the northwest corner of the text entry in the flowchart view of your model window. You can drag it around later, as well as resize and reorient it using the underline below it that appears when you select it (hold down the shift key while reorienting it to constrain it to be horizontal or vertical or on a 45º line). To change the text color, select the text and use the *Text Color* button (A ▾) to select either the color of the underline there (click on the A in this case) or to choose a different color (click on the ▾ in this case and select a color from the palette), which will also become the new underline color.

The yellow backdrop box behind the model label and its shadow were made with the *Box* button (□) on the *Draw* toolbar; clicking this button turns your mouse pointer to cross hairs, after which you click once to determine one corner of the box and again to determine the opposite corner. You can change the fill color by first selecting the box and then clicking the Fill Color button (🪣 ▾); change the border color with the Line Color button (✎ ▾). To create faux three-dimensional effects like shadows, you can cleverly "stack" and offset objects (like a yellow box on top of a slightly shifted black box) using options like Send to Back from the *Arrange* toolbar and menu.

If you'd like to drop a graphics file (*.gif, .jpg,* etc.) into the flowchart view of your model window, use *Edit > Insert New Object*, select *Create from File*, next *Browse* to choose the file you want, press *Open*, then *OK*, and finally drop it in with the cross hairs.

Clearly, you could spend a ruinous amount of time on this kind of stuff. Without implying anything about anybody, we offer it as a simple empirical observation that the higher up in an organization you go to present your simulation, the more effort is probably justified on graphics.

3.4.12 The Run > Setup Dialog Boxes

To establish the run conditions, use the *Run > Setup* menu command, where you'll find tabbed pages that control various aspects of how your simulation will execute. You'll need to edit just two of these pages.

The first one is the Project Parameters tab, where you should enter a Project Title, Analyst Name (that's you), and Project Description. You might also need to modify the selection of which statistics will be collected and reported, depending on how you have your defaults set in *Tools > Options > Project Parameters* (we want Entities, Resources, Queues, and Processes). The completed dialog was shown in Figure 3-15 in Section 3.3.11.

The other tab that you need to edit is Replication Parameters. These edits were also discussed in Section 3.3.11, and were shown in Figure 3-16 there.

3.4.13 Establishing Named Views

To set up a Named View for your model, first pan and zoom to the scene you want to remember, then *View > Named Views* (the button or or type **?**), press *Add*, then pick a name and maybe a hot key (case-sensitive). If you want to change this view's definition later, press *Edit* instead once you've panned and zoomed to the new view you want to associate with this name and possible hot key; to delete it from the list, select it there and press *Delete*. We discussed using Named Views in Section 3.2.3.

At first blush, setting up Named Views might seem like a frill, but trust us—you'll want some of these when your models grow.

3.5 Case Study: Specialized Serial Processing vs. Generalized Parallel Processing

A classic operational question is whether to have specialized or generalized workers when processing involves multiple different tasks; a related question is how processing-time variability affects the decision. Based on a paper by Harrison and Loch (1995)[3], we can investigate this using only the Arena modules introduced in Model 3-1. Arguments can be made for both specialized and generalized work, depending on the setting, and experiments have been conducted on real systems, sometimes with disappointing results. A careful simulation study might help avoid such disappointments. Consider a loan-application office (as did Harrison and Loch), where applications arrive with exponentially distributed interarrival times with mean 1.25 hours; the first application arrives at time zero. Processing each application requires four steps: first a credit check (this takes time but everyone passes), then preparing the loan covenant, then pricing the loan, and finally disbursement of funds. For each application, the steps have to be done in that order. The time for each step is exponentially distributed with mean 1 hour, independent of the other steps and of the arrival process. Initially, the system is empty and idle, and we'll run it for 160 hours (about a work month); see Chapter 5 for other kinds of stopping rules based on entity counts and other conditions. Output performance

[3] Thanks to Professor Jim Morris of the University of Wisconsin-Madison for recommending this paper to us.

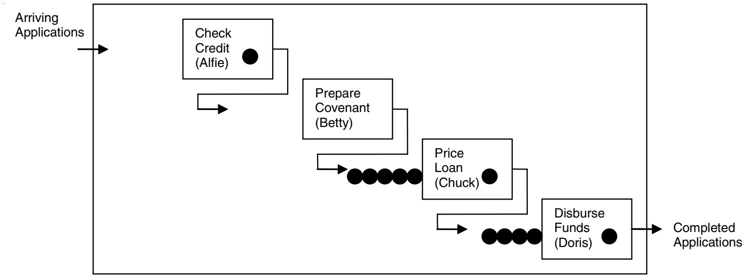


Figure 3-25. Specialized Serial Processing

measures include the average and maximum total number of applications in process, and the average and maximum total time, from entry to exit, that applications spend in the system, as well as their time waiting for the next processing step to begin. There are four employees available (Alfie, Betty, Chuck, and Doris), all equally qualified for any of the four steps, and the question is how best to deploy them.

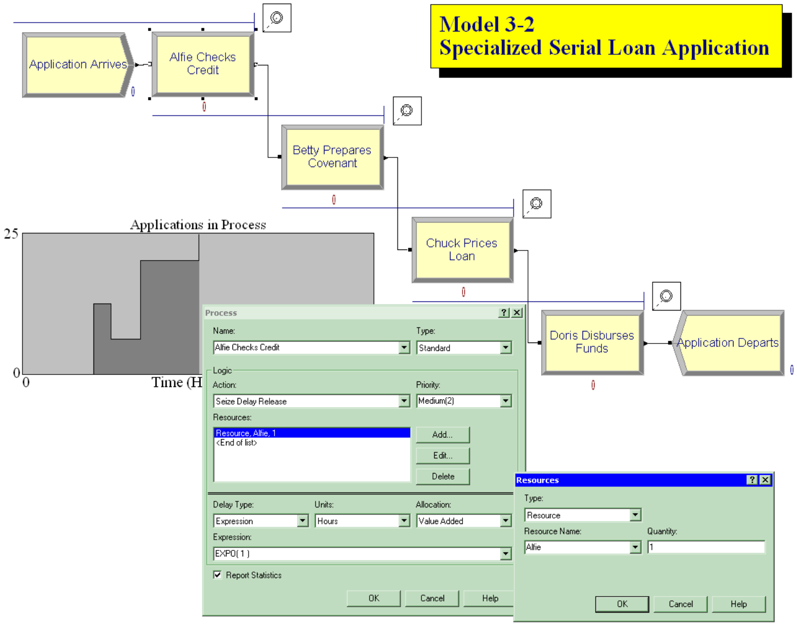
3.5.1 Model 3-2: Serial Processing – Specialized Separated Work

A first thought might be to specialize the employees and assign, say, Alfie to check credit for all applications, Betty to prepare all covenants, Chuck to price all loans, and Doris to disburse all funds. So each application would first have to go through Alfie, then Betty, then Chuck, and finally Doris. A layout might look like Figure 3-25, where there are currently 12 applications (the filled circles) in process, Alfie is busy but with no more applications in queue, Betty is idle (so of course with no queue), Chuck is busy with five more in queue, and Doris is busy with four more in queue. All queues are FIFO. Though Betty might disagree, it's too bad that, under these operational rules, she can't lend a hand to Chuck or Doris right now.

An Arena simulation of this, Model 3-2, is fairly straightforward, and essentially just involves adding three Process modules to Model 3-1, along with three more Resources. The completed model is in Figure 3-26, with the first Process module and its Resources dialog open.

Since the modules are so similar to those in Model 3-1, we'll just highlight the differences, and will trust you to browse through the model:

- The Create module is the same as in Model 3-1, except for the module Name, the name for the Entity Type, the Value of the mean of the exponential interarrival times, and the time Units (now Hours).
- The four Process modules have, as before, the Action of `Seize Delay Release`, but of course have different names for the modules and Resources. Since exponential process times are not among the Delay Type choices, instead choose `Expression` there, and then use the Expression Builder (go to "`Random Distributions`") to fill in the Expression field, as seen in Figure 3-26. The time Units here are also Hours.



Process

Name: Alfie Checks Credit

Type: Standard

Logic

Action: Seize Delay Release

Priority: Medium(2)

Resources:

Resource, Alfie, 1
<End of list>

Add... Edit... Delete

Delay Type: Expression

Units: Hours

Allocation: Value Added

Expression: EXPO(1)

☑ Report Statistics

OK Cancel Help

Resources

Type: Resource

Resource Name: Alfie

Quantity: 1

OK Cancel Help

Figure 3-26. Completed Model 3-2 with the First Process Module and its Resources Dialog

- Other than its Name, the Dispose module is the same as in Model 3-1.
- The default Entity Picture (Entity data module, `Picture.Report`) is okay here since the entities are, well, reports of the loan applications.
- The default Resource animations are almost okay, but we made the Idle picture also have a white fill rather than the default green (double-click on the Resource animation icon to get to the Resource Picture Placement window, then double-click on the Idle and Busy icons in turn and use the Fill Color tool). Remember to select in the Identifier field the correct Resource name for each of the four cases. Also, check the Seize Area box.
- The Queue and Resource data modules are automatically filled in correctly if the Process modules were set up first. The defaults for everything (e.g., the Type and Capacity of each of the four Resources) work in this model.
- The dynamic Plot traces the total number of applications in process, which is the Arena Expression `EntitiesWIP(Application)`, and can be found in the Expression Builder under `Basic Process Variables`, then `Entity`, and finally `Number in Process`. “WIP” is an acronym for work in process. The other entries in the Plot dialog are similar those in Model 3-1, adjusted of course for the different time frame and maximum y-axis value.
- In *Run > Setup*, in the Project Parameters tab check the Processes box in Statistics Collection, and enter some documentation elsewhere; in the Replication

Parameters tab enter 160 for the Replication Length and make sure the Time Units and Base Time Units are both Hours.

- After some initial runs and noting the maximum queue lengths, we lengthened their animations a bit out to the left to accommodate.

Run this model and look at the Category Overview report to find, among other things:

- The average and maximum total number of applications in process were, respectively, 12.3931 and 21 (via Loan Application → Entity → Other → WIP).
- The average and maximum total time, from entry to exit, that applications spent in the system were, respectively, 16.0831 hours and 27.2089 hours (Loan Application → Entity → Time → Total Time). Note that this counts only those applications that had left the system when the simulation stopped, not those that were still in process at that time (since, of course, their total time in system had not yet been completed).
- The average and maximum total time that applications spend waiting for the next processing step to begin were, respectively, 11.9841 hours and 22.2732 hours (Loan Application → Entity → Time → Wait Time). This includes only time "wasted" waiting in queue, and not "value-added" time spent undergoing processing at the four steps, so is a good measure of system inefficiency. Arena counts in this statistic the total waiting time in the four queues only for those applications that completed all four steps and exited the system; waiting times in the individual queues are under Loan Application → Queue → Time and, as noted for Model 3-1, may include more entities than those included for the overall system-wide average and maximum.
- During the 160 hours, 117 applications were completed (Loan Application → Entity → Other → Number Out), a measure of productivity.
- Alfie, Betty, Chuck, and Doris were busy, respectively, 82.33%, 70.34%, 80.44%, and 80.80% of the time.
- Alfie, Betty, Chuck, and Doris processed, respectively, 128, 128, 122, and 117 applications (Loan Application → Process → Other → Number Out). In this model, all applications visit these people in this order, so these numbers in this order must decrease or stay the same. (Under what conditions would they all be the same? Is that possible in this model?)

The main inefficiency in this system is the waiting in queue that the applications must endure, and there are four different places where that could happen. Also, there is the possibility, as in Figure 3-25, that applications could be queued at some stations while other employees are idle, which seems like a waste of resources.

3.5.2 Model 3-3: Parallel Processing – Generalized Integrated Work

Would it be better to "generalize" or "integrate" the work so that each employee completely processes all four steps of one application at a time, and with a single queue of applications "feeding" the group? From the applications' viewpoint, this would present just one opportunity for wasting time waiting in queue, but then again the

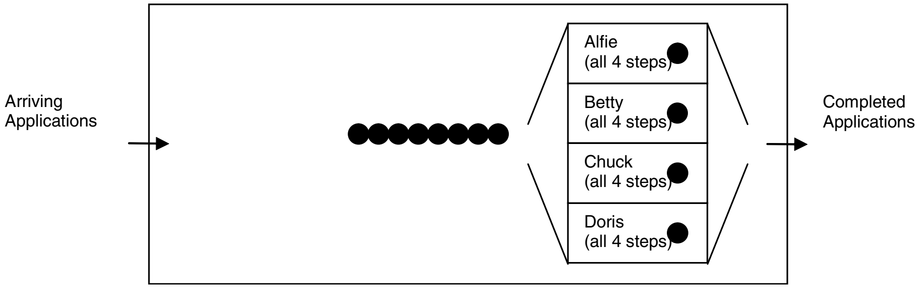
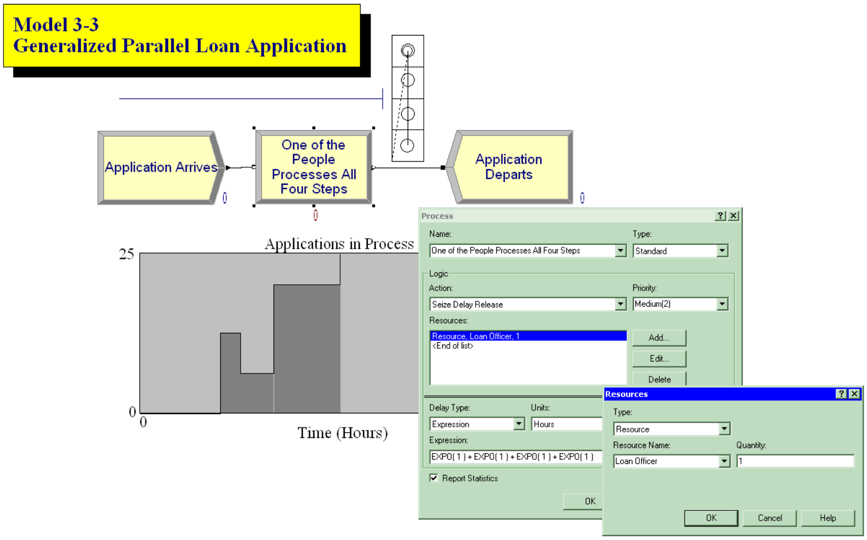


Figure 3-27. Generalized Parallel Processing

processing would be longer. Figure 3-27 shows how this would work, where each employee processes all four steps before moving on to the next application in queue. Like the specialized serial processing snapshot in Figure 3-25, there are currently 12 applications in process, but note that the total number in queue is now eight rather than nine, since Betty is now busy (we don't know how she feels about that, but this looks like better service).

The Arena model for this, Model 3-3, is actually simpler than Model 3-2, and is shown in its completed form in Figure 3-28 with the (sole) Process module and its Resources dialog showing. The Create and Dispose modules are identical to Model 3-2, as is the dynamic Plot and the *Run* > *Setup* dialog (except for labeling).



Process

Name: One of the People Processes All Four Steps
Type: Standard

Logic
Action: Seize Delay Release
Priority: Medium(2)
Resources:
Resource, Loan Officer, 1
<End of list>

Delay Type: Expression
Units: Hours
Expression: EXPO(1) + EXPO(1) + EXPO(1) + EXPO(1)

☑ Report Statistics

Resources

Type: Resource
Resource Name: Loan Officer
Quantity: 1

Figure 3-28. Completed Model 3-3 with the Process Module and its Resources Dialog

The main change is that the four Process modules in Model 3-2, each of which represented a single employee, are replaced by a single Process module representing all four employees. We replaced the four single-unit resources in Model 3-2 with a single four-unit resource here, which we named `Loan Officer` (look at the Resource data module in Model 3-3). Note that, in the Resources dialog of the Process module shown in Figure 3-28, the Quantity field is still 1, representing the number of units of the `Loan Officer` resource that are seized and released by each `Application` entity; this is *not* where we specify that there are four units of this resource in existence, but rather in the Capacity column of the Resource data module.

And since each employee must now complete all four tasks in tandem before taking the next application out of the queue, we must make the Delay time longer in the Process module. Each of the four steps requires `EXPO(1)` hours to complete, so we simply add four of these up in the Expression field in the Delay Type area of the Process module, as seen in Figure 3-28, each representing one of the four processing steps (it's okay to do arithmetic inside an Expression field anywhere in Arena). Now we're guessing that, if you're reading a book like this, you probably did well in seventh-grade algebra, so you know that adding up four of the same thing is the same as multiplying that thing by 4, so you may be wondering why we didn't save ourselves some typing and enter something like `4*EXPO(1)` instead. Well, the problem with that is that it would make just a single "draw" from the exponential distribution and multiply that single resulting number by 4, meaning in our case that each of the four steps took exactly the same amount of time, not what we want. So, ugly as it is, `EXPO(1) + EXPO(1) + EXPO(1) + EXPO(1)` is what we need since this makes four separate, independent draws from the exponential distribution to represent four separate, independent times for the four steps.

Finally, a change in the Resource animation would make things a little more realistic (but still not completely right), though would not be necessary for the model or its results to be correct. First, change the icon by double-clicking on the Resource animation and then the Idle icon to get to the Resource Picture Placement window, and duplicate the white square three times and line them all up vertically. Then select and copy this whole thing, go back (just close the window), open the Busy icon, and paste it in to replace what's already there. Back out in the model window, double-click on the two concentric circles representing the Seize Area to get to the Seize Area dialog, select Points, and then Add three times to get three more related seize areas for this multi-unit resource; close the Seize Area dialog and drag these around to where they belong inside the boxes (you may need to turn off Snap to Grid to get them just right). In the animation, then, you'll see as many entity icons in the Resource animation as there are entities in service at any time. What makes even this not completely right is that the icons for the entities in service will always shove themselves toward the original Seize Area (the double concentric circles) rather than staying where they "belong," if you view the boxes from top to bottom as being for Alfie, Betty, Chuck, and Doris (you can really get this right by using Resource Sets on single-unit resources, rather than a single multi-unit resource, as discussed in Model 5-2 in Chapter 5).

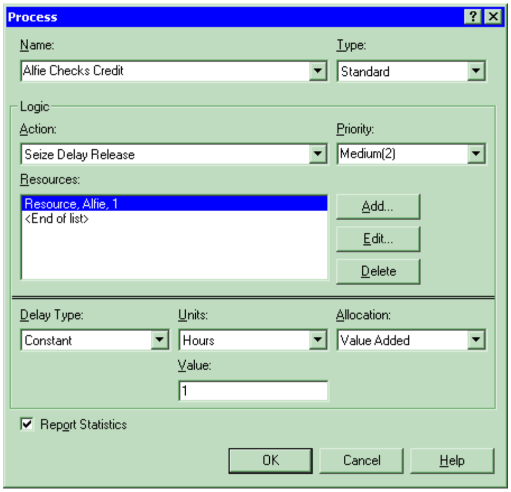
This parallel, integrated-work configuration appears to provide better service than the serial, specialized-work setup. The average and maximum total number of applications in process here are, respectively, 4.6118 and 10, compared to 12.3931 and 21 for the serial configuration. The average and maximum total time in system of applications here are, respectively, 5.3842 and 13.7262 hours, down from 16.0831 and 27.2089 hours for the serial model. The average and maximum time wasted waiting in queue here are, respectively, 1.3282 and 6.8231 hours, down markedly from 11.9841 and 22.2732 hours for the serial setup. Productivity went up, from 117 to 135 applications completed, though such improvement is of course limited by the arrival rate. All these improvements are due to less idle time among the employees; the utilization of the `Loan Officer` resource, defined as the time-average number of units busy (which came out to be 3.48) divided by 4, was 87%, compared to the average of 78.48% over the four individual employees in the serial configuration. We just never have a situation like that of Figure 3-25 where jobs are queued yet resources are idle. Of course, these comparisons are from just one replication of each of Models 3-2 and 3-3, so we're on thin statistical ice, as discussed in Section 2.6 (see Exercise 6-19).

3.5.3 Models 3-4 and 3-5: The Effect of Task-Time Variability

Harrison and Loch (1995) pose a further question about the comparison between the above two layouts—will the second one always be a lot better than the first one, as seemed to happen above? While many other aspects of the situation could speak to this question, they looked at variability of the task times.

Each of the four tasks for an application has an average duration of one hour, but these durations have an exponential probability distribution around that mean, so there is variability in the task time; let's get an idea of how much variability. For example, the probability that a task can be done in under ten minutes (1/6 hour) is $F(10) = 1 - e^{-1/6} \cong 0.15$, where F denotes the cumulative distribution function of an exponential random variable with mean one hour (see Appendix C for a refresher on probability and statistics, and Appendix D for definition of the exponential and other probability distributions). But some tasks will take a long time; the probability that a task will take more than two hours is $1 - F(2) = e^{-2} \cong 0.14$, about the same as the 0.15 probability of doing a task in less than ten minutes. Put another way, about 15% of the tasks are very quick (under ten minutes), but about the same number (14%) take a very long time (over two hours). So there's quite a lot of variation in task times with their exponential distribution, and in particular the long task times contribute significantly to congestion and queueing delays, and thus to inefficiency. In the serial configuration of Model 3-2, when just one of these long task times pops up, it creates a big backup at that station (maybe that's what's happening to Chuck in Figure 3-25). In the parallel configuration of Model 3-2, a long task time will also increase congestion, but probably not by as much as in the serial configuration since it's unlikely to be accompanied by similarly long times for the other three tasks for that application, which somewhat dampens its effect.

But what if there were less variability among task times, even while holding the mean task time to the same one hour? Let's take that to the extreme and look at what happens if there's *no* variability, i.e., each task time takes *exactly* one hour (and thus each application will require *exactly* four hours of processing time in total). Maybe we could move toward this with better support for the tasks, either computer-based, clerical, or via better pre-screening. Since it's not under our control in reality, we'll keep the arrival

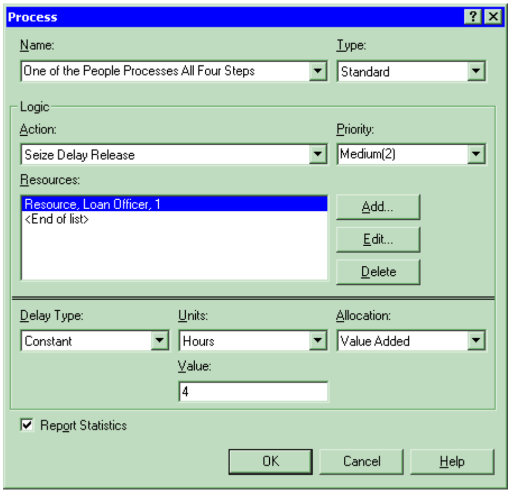


Name: Alfie Checks Credit
Type: Standard
Logic
Action: Seize Delay Release
Priority: Medium(2)
Resources:
Resource, Alfie, 1
<End of list>
Delay Type: Constant
Units: Hours
Allocation: Value Added
Value: 1
☑ Report Statistics

Figure 3-29. Completed Process Module for Alfie in Model 3-4 with Constant Task Times

process the same; i.e., with interarrival times that have an exponential distribution with mean 1.25 hours, so this model still has some stochastic (random) input.

Now how do the serial and parallel configurations compare? We modified Model 3-2 to get Model 3-4, where each of the four Process modules was modified to make the corresponding task time exactly one hour. This was very easy to do; in each of the Process module dialogs just change Delay Type to Constant in the pull-down menu, set Value to 1, and make sure the Units are still Hours. Figure 3-29 shows this for Alfie, and the other three work the same way.



Name: One of the People Processes All Four Steps
Type: Standard
Logic
Action: Seize Delay Release
Priority: Medium(2)
Resources:
Resource, Loan Officer, 1
<End of list>
Delay Type: Constant
Units: Hours
Allocation: Value Added
Value: 4
☑ Report Statistics

Figure 3-30. Completed Process Module for Model 3-5 with Constant Task Times

Similarly, we created Model 3-5 from Model 3-3 by just getting rid of the ugly `EXPO(1) + EXPO(1) + EXPO(1) + EXPO(1)` in the Process module for the Expression Delay type, changing the Delay Type to Constant, and entering a Value of `4` Hours; Figure 3-30 shows the completed Process module dialogue.

Table 3-1 summarizes the results from the constant-service-time Models 3-4 and 3-5 (bottom half) and repeats those from the exponential-service-times Models 3-1 and 3-2 (top half), rounding to two decimals. For the constant-service models (3-4 and 3-5) there might still be some improvement in going from the serial to parallel operation, but only a very slight one (and might not even be statistically significant, per Exercise 6-19). Evidently, when service times have little or no variability there's not much difference between the specialized serial and generalized parallel work designs.

Of course, you could raise some questions, like:

- In the parallel integrated-work model, wouldn't you expect these generalized people to be at least a little less efficient than if they specialized in doing just one thing? Our Model 3-3 doesn't allow for this possibility, but Exercises 3-13 and 6-20 do.
- When dealing with human resources, don't they need a little idle time, maybe via scheduled breaks, to catch their breath and stay fresh? Resource Schedules, which can be used to model such breaks, will be discussed in Chapters 4 and 5.
- Are the improvements from going from the serial to parallel designs statistically significant, or could they be just spurious, the result of mere sampling fluctuation (after all, we made just a single replication of each configuration)? This is taken up in Chapter 6, and in Exercises 6-19 and 6-20.

3.6 More on Menus, Toolbars, Drawing, and Printing

In this section, we'll briefly mention some miscellaneous information that we haven't covered on the Arena menus and toolbars, and mention a few more things about its drawing and printing capabilities. As we go through the examples in later chapters, we'll give more detail on these things as needed.

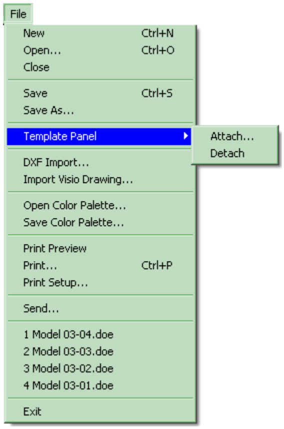
3.6.1 Menus

Here we give a quick overview of what's in the Arena menus. Some options in some menus will be dimmed (meaning that you can't select them) if they don't apply in your particular situation or status at the moment. For more information about menu entries,

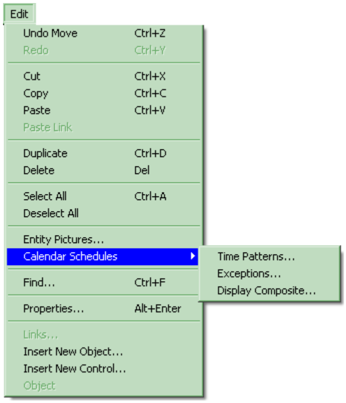
Table 3-1. Summary Results from All Four Scenarios of the Loan-Processing Model

	Model	Total WIP		Total Time in System		Total Waiting Time		Number Processed	Avg. Utilization
		Avg.	Max.	Avg.	Max.	Avg.	Max.		
Expo service	3-2 (serial)	12.39	21	16.08	27.21	11.98	22.27	117	0.78
	3-3 (parallel)	4.61	10	5.38	13.73	1.33	6.82	135	0.87
Constant service	3-4 (serial)	3.49	12	5.32	11.38	1.32	7.38	102	0.65
	3-5 (parallel)	3.17	11	4.81	10.05	0.81	6.05	102	0.66

remember that you can click ? and then use that to click on any menu item (dimmed or not) to get complete documentation on it, including hyperlinks to related topics.

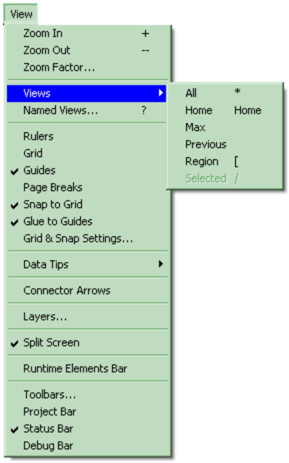


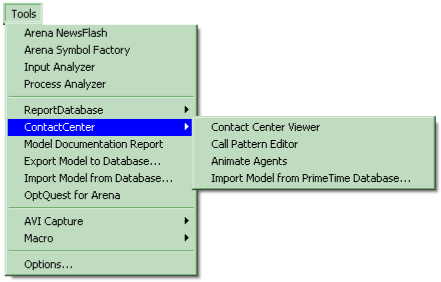
File menu. This is where you create new Arena model files, open existing ones, close windows, and save your models. This is also where you attach and detach Project Bar panels. You can also import CAD drawings from AutoCAD® (and from other CAD programs in standard DXF format) for use as Arena "backdrops" and, in some cases, active elements (like paths for wire-guided vehicles) to allow you to use existing detailed drawings of facilities. Another type of graphics file you can import is a Microsoft® Visio® drawing file in *.vsd* format. If you change the colors Arena uses, you can save them as a color palette (you can do some of this with Windows® as well); you can also open previously saved color palettes. The Arena printing functions are accessible from this menu. The Send command allows you to send mail from within Arena and attaches any active model to your message. Arena remembers the most recent documents, and you can open them quickly. The Exit command is one of the ways to quit Arena.



Edit menu. Here you'll find the usual options as applied to objects in Arena models. You can Undo previous actions or Redo your Undos. You can Cut or Copy a selected object (or group of objects) to the Clipboard for placement elsewhere in the current model, to other models, or in some cases, to other applications. Paste allows you to insert the Clipboard contents into a model, and Paste Link creates an OLE link to the source document that's currently in the Clipboard. Duplicate makes a copy of what's selected and places it nearby in the current model, and Delete permanently removes whatever you have selected. You can Select All objects in a model as well as Deselect All. With Entity Pictures you can change what's in the list presented in the Entity data module, as well as the appearance of those pictures; you can copy pictures into this list from Arena's picture libraries (*.plb* files). Calendar Schedules allows you to describe complex time patterns with hierarchies (weeks are made of days, which are made of shifts, and so on), define exceptions like holidays and vacations, and show the net effect of all this in a composite view. Arena's Find function searches all modules and animation objects in the active model for a text string with the usual control over whole-word searches and case sensitivity (this can come in handy for finding entries that you didn't think you had but about which you're getting some error message). You can display additional object Properties, such as its unique object tag. If you have links in your model to other files, such as a spreadsheet or sound file, Links tells you about them and allows you to modify them. Insert New Object lets you make placements from other applications, like graphics and multimedia, and Insert New Control allows insertion of a VBA®/ActiveX® control. Object lets you edit something you've brought into the model from another application.

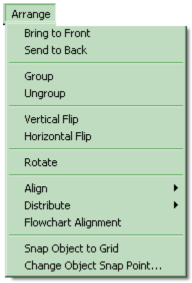
View menu. From this menu, you can control how your model appears on the screen, as well as which toolbars you want to display. Zooming lets you view the model from different "altitudes" so you can see the big picture or smaller sections in more detail. The Zoom Factor allows you to set how much you zoom in or out each time. Views (whose submenu is shown) offers certain "canned" views of your model; and Named Views lets you define, change, and use your own views. Rulers, Grid, Guides, Glue to Guides, and Snap to Grid are useful if you want to line things up geographically; Grid & Snap Settings gives you control over the spacing for Grid and Snap. Page Breaks displays your model indicating how it will be broken into pages for printing. Data Tips allows you options for displaying object properties in tool tips if you let the mouse hover over that object. Connector Arrows lets you put arrowheads on connectors to visualize the direction of entity flow. Layers lets you control what kinds of objects show up during the edit or run mode. If Split Screen is checked (toggled on), your model window will show both the flowchart and spreadsheet views simultaneously. Runtime Elements Bar, if checked, displays a window to allow you to choose what is displayed during execution. Toolbars is one way you can designate which sets of buttons are displayed on your screen (see Section 3.6.2), Project Bar toggles for whether the Project Bar (which hosts the panel) is visible, and the Status Bar entry lets you decide whether you want to see the horizontal bar at the very bottom of the screen, which tells you what's going on and indicates the world coordinates of the mouse pointer in the Arena workspace. Debug Bar, if checked, shows a window of debugging tools during a run.



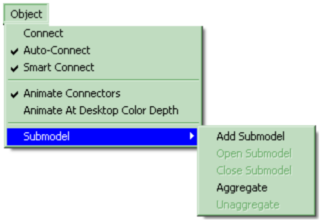


Tools menu. Arena News Flash is an Internet-based news feed for updates, etc. Arena comes not only with the modeling capability with which we've spent our time so far, but also contains a suite of related tools, possibly depending on what you've licensed. The Arena Symbol Factory provides a large collection of graphics objects in many categories from which you can create graphical symbols for animation of things like entities and resources. The Input Analyzer fits probability distributions to your observed real-world data for specifying model inputs (see Section 4.6.4). The Process Analyzer organizes efficient ways for you to make multiple simulation runs, which might represent different model configurations, and keep track of the results; it also helps you carry out proper statistical analyses of your simulation's results, such as a reliable way to select the best from among several different model configurations. Another application with additional statistical capabilities, called the Output Analyzer, comes with Arena but must be launched separately (it's in the Arena 10.0 folder). We'll show you how to use the Process Analyzer and the Output Analyzer in Sections 6.5 and 6.4, respectively. Report Database exports summary statistics from a run to a CSV (Comma-Separated Values) file to be read into a spreadsheet for post-processing analysis. ContactCenter (shown with its submenu open) provides special functions to model contact/call centers. Model Documentation Report produces a compact but complete set of documentation, including run conditions, modules used, and any submodels. Export Model to Database allows you to save the details of your model to an Access or Excel database; Import Model from Database allows you to bring in those details from such a database to construct or update a model quickly. OptQuest for Arena is an application that decides how to change model inputs that you select and then runs a sequence of simulations to search for a combination of these inputs that optimizes (maximizes or minimizes) an output performance measure that you designate; we'll give an example of its use in Section 6.6. AVI Capture enables recording many actions, including editing and animation, into a video (*.avi*) file for playback. The Macro item provides the tools to record and run Visual Basic (VB) macros and to open the VB Editor for writing custom logic. These VB topics are discussed in Section 10.2. Finally, the Options item lets you change and customize a lot of how Arena works and looks to suit your needs (or tastes).

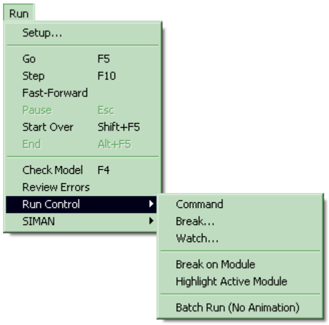
Arrange menu. The items here pertain to the position of modeling modules and graphics objects; some apply only to graphics objects. Bring to Front and Send to Back position the selected object(s) on the top and bottom, respectively, of a "stack" of objects that may overlap. Group and Ungroup, respectively, put together and subsequently take apart objects logically, without affecting their physical appearance; Grouping is useful if you want to move or copy a complex picture built from many individual objects. The Flip entries invert the selected object(s) around a line in the indicated direction, and Rotate spins the selection clockwise 90°. Align lines up the selected objects along their top, bottom, left, or right edges. Distribute arranges the selected objects evenly in either the horizontal or vertical direction, and Flowchart Alignment arranges the selected flowchart modules evenly both vertically and horizontally. Snap Object to Grid forces the selection to align to the underlying grid of points, and Change Object Snap Point lets you alter the exact point on the selected object that gets snapped.



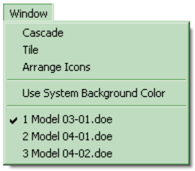
Object menu. These items relate to the model's logical structure, its flowchart modules, and the connections between logical pieces of the model. Connect changes the pointer to cross hairs and lets you establish graphically a connection between modules for entities to follow; selecting it twice sets up a Connect "session" for repeated use (right-click or hit *Esc* to end the session). Auto-Connect is a toggle that allows you automatically to connect a newly placed module to one that's already selected. Smart Connect causes newly added connections to be drawn in horizontal/vertical segments instead of possibly diagonal lines, unless you make intermediate clicks if you're placing the Connection by hand. Animate Connectors causes entities to show up as they move along Connections between flowchart modules, even if this movement is occurring instantly in the simulation. Animate At Desktop Color Depth, if checked, runs the animation with all the colors on your desktop, which could slow down the simulation; if not checked, the animation runs at 8-bit color depth with no slowdown. Submodel, whose submenu is shown, lets you define and manage hierarchical submodels.

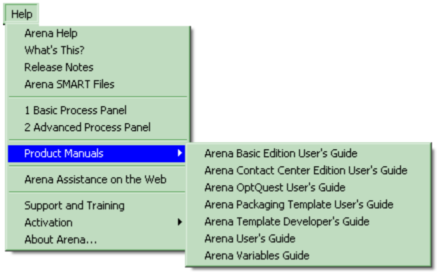


Run menu. This menu contains the *Run > Setup* dialog boxes that we discussed in Sections 3.3.11 and 3.4.12, which control the manner in which the current model will be run (including possibly its run length). It also contains entries for running the simulation in different ways, as well as several options to watch the execution, check it (and view any errors), and to set up and control how the run goes and is displayed on your screen. We'll describe these capabilities further in Section 3.8. You can also access the code that Arena actually generates in the underlying SIMAN simulation language.



Window menu. If you have several models open at once, you can arrange them physically in an overlapping Cascade, or in a non-overlapping Tile arrangement. If you have several models minimized (via the ▬ button in each window), select Arrange Icons to organize them. The Use System Background Color entry causes this model to use whatever background color is selected at the Windows® operating-system level rather than what is set internal to Arena; to return your model to Arena's internal color, select this item again. (This menu item changes/toggles between "System" and "Custom" each time you select it.) Finally, you can activate a model that's already open by selecting it at the bottom of the menu.





Help menu. This is one of several ways to access Arena's online Help system (see Section 3.7 for more help on Help). If you select Arena Help, you'll get to the table of Contents, an Index, and a Search utility for getting to the topic you want. What's This changes the pointer to ↖? that you can then use to click on a menu entry or toolbar button to find out a little bit about it. Release Notes has information on recent changes and system requirements. Arena SMART Files produces a subject-based index to a couple of hundred little Arena models that illustrate a wide variety of specific modeling techniques. Next on this menu, you'll find general help topics on whatever modeling panels you have attached at the moment. Product Manuals takes you to detailed documents on the various components of the Arena software and related products. Arena Assistance on the Web takes you directly to a Web site with the latest help and software information (of course, you have to be online for this to work). The last three entries on this menu provide information on support and training; the copy protection and activation procedures used for commercial and other versions of Arena; and detailed version information.

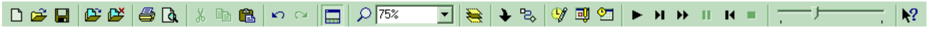
3.6.2 Toolbars

Arena has several *toolbars* with groups of buttons and drop-down list boxes to facilitate quick access to common activities. Some of these buttons are just faster ways to get at menu items (discussed above), and some represent the only way to do something.

Select the menu option *View > Toolbars* (or right-click in a toolbar) to choose which toolbars will be displayed. As in many applications, you can tear off toolbars and float them in the interior of the Arena window as palettes, or dock them to an edge (if you want it near the edge but not docked, hold down the *Ctrl* key while approaching the shoreline). You won't have to set your toolbar configuration every time you use Arena as it will remember your last configuration. You can also have different configurations for when you're editing your model, when the simulation is running, and when various other Arena window types are active (such as the Picture Editor), and again, Arena will remember what each was.

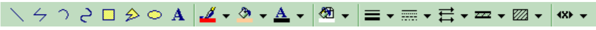
Customize how toolbars are displayed via *View > Toolbars > Customize* or right-clicking in a toolbar, then selecting Customize, and then Customize again. We'll mention each toolbar in turn below, but you have the option to rearrange what buttons are on which toolbars by this customization capability.

- While you could choose to hide the *Standard* toolbar, it's a little hard to see how you could do much without it:



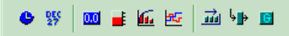
It starts with buttons to create a New model, Open an existing one, and Save the active model, as on the File menu; also from that menu are buttons to Attach a panel or Detach the visible one, and to Print and do a Print Preview. From the Edit menu are Cut, Copy, and Paste, as well as Undo and Redo. Next is the Toggle Split Screen button for a split model window, and then the magnifying glass to View a Region that you select with it in the flowchart view of the model window at the closest possible zoom. You can choose a Zoom percent from a drop-down list. The Layers button lets you control which types of objects show up in the flowchart view of the model window in both edit and run mode. You can add a Submodel and Connect flowchart modules. You can create and manage calendar time patterns and their exceptions with Edit Time Patterns and Edit Exceptions. Display Composite View lets you manage the capacity and efficiency data associated with specific system elements (like scheduling a resource to be available for two shifts instead of one). The next six buttons are run controls, and will be discussed in Section 3.8. The slider bar allows you to speed up or slow down the animation during a run. The Standard toolbar ends with the context-sensitive Help button; click on it (note the **?** that gets added to the mouse pointer), then click on a toolbar button or a menu command or a module in the Project Bar to learn about it.

- Buttons on the *Draw* toolbar have no corresponding menu options, so drawing can be done only by toolbar access:



This is how you can draw static Lines, Polylines, Arcs of ellipse boundaries, Bézier Curves, Boxes, Polygons, and Ellipses to dress up your model, as well as add Text to annotate it. There are also controls for changing the color of Lines (including borders of shapes), Fill of objects, Text, and the Background of the flowchart view of the model window. You can alter the Line Width and Style (thickness as well as whether it's there or not, and dash patterns), and put arrowheads on lines. Line Patterns provides different pre-drawn line appearances, like roads, tracks, and belts. Fill Pattern provides hatch patterns for shapes. Show Dimensions lets you display the sizes of shapes and lines for precise drawing. You've probably used draw features in other applications, so Arena's capabilities will be familiar. By far, the best way to familiarize yourself with these things is to open up a "test" model window and just try them out; see Section 3.6.3 for more on drawing.

- The *Animate* toolbar contains capabilities to allow you to animate your model or enhance the animation that is inherent in some Arena modules:



Typically, click one of these buttons, enter a dialog box to describe exactly what you want, then place the animation in your model. There are a lot of different capabilities here, and we'll illustrate most of them as we progress through building models in later chapters (we've already used Plot and Resource). For now, hover your mouse above each button to show the Tooltip with its name.

- The *Integration* toolbar contains buttons related to Arena's Module Data Transfer wizard and VBA (the Visual Basic Editor and VBA Design Mode button):

Chapter 10 discusses the use of VBA, which augments Arena's standard modeling features with a complete Visual Basic programming interface.

- The *View* toolbar has buttons to control how you view the flowchart view of the model window:

From here you can Manage Named Views, Zoom In and Out, or choose to View All the model or View Previous. You can also reveal the Rulers and Grid, create Guides, show page breaks for printing, and Snap new objects to the Grid, as well as Glue objects to Guides.

- The *Arrange* toolbar corresponds closely to the Arrange menu:

You can Bring a selected object to the Front or Send it to the Back. A selection of multiple drawing objects can be made into a logical Group and Ungrouped later. Drawing objects can also be Flipped around a Vertical or Horizontal line on their midpoint or Rotated clockwise 90°. You can also Align the selected objects on their Top, Bottom, Left, or Right edges, as well as Space them evenly either horizontally or vertically. Finally, you can Snap selected objects to the Grid.

- The *Run Interaction* toolbar has buttons corresponding to the Check Model, Command, Break, Watch, and Break on Module entries from the Run menu (see Section 3.8 for more). The last button corresponds to Animate Connectors from the Object menu:

- The *Record Macro* toolbar has buttons to Start/Pause/Resume as well as Stop recording a VB macro. A macro is a series of Visual Basic statements stored in a subroutine in a Visual Basic module (macros are discussed in Chapter 10):

- The *Animate Transfer* toolbar gives you tools to add animation objects to your model:

  These include Storage, Seize, Parking, Transporter, Station, Intersection, Route, Segment, Distance, Network, and Promote Path. We'll discuss these capabilities in subsequent chapters as we develop models whose animations can benefit from them.

3.6.3 Drawing

The *Draw* toolbar, mentioned in Section 3.6.2, has a variety of shapes, text tools, and control features to allow you to enhance the model by placing static (no participation in the simulation or animation) objects in the model window to help document things or to make the animation seem more "real" by adding items like walls, aisles, and potted plants. This isn't intended to be a complete, full-featured CAD or artwork capability, but it usually proves adequate; of course, you can always import graphics from other packages, as mentioned in Section 3.4.11. Arena's drawing tools work a lot like other drawing packages, so we'll just point out what's there and let you play with things to get used to them:

- *Line,* ╲: Click once on this button, changing the mouse pointer to cross hairs, then click where you want the line to start and again where you want it to end. To constrain the line to be vertical or horizontal or on a 45° angle, hold down the *Shift* key while moving to the end of the line.
- *Polyline,* ↯: This lets you draw a jagged line with an unlimited number of points. After selecting this button, click where you want to start and then again for each new point; double-click for the endpoint. Hold down the *Shift* key during a segment to constrain it to vertical, horizontal, or 45°.
- *Arc,* ⌒: Draw part of the border of an ellipse. Click first for the center of the ellipse, then move the mouse and follow the bounding outline, clicking again when it's the size and shape you want (hold down the *Shift* key to constrain the ellipse to be a circle). At this point, the mouse pointer becomes the end of a line emanating from the ellipse's center; click to define one end of the arc, then again for the other end. To edit the arc later, select it and use the lines to change what part of the arc is shown and use the disconnected handle to change the ellipse size or shape.
- *Bézier Curve,* ∽: These have become popular due to their ability to assume a lot of different shapes yet maintain their smoothness and inherent beauty. Click for one endpoint, then make intermediate clicks (up to 30) for the interior "attractor" points; double-click to place the other endpoint. Holding down the *Shift* key while moving to the next point causes the (invisible) lines connecting them to be horizontal, vertical, or at 45°. To change the curvature, select the curve and drag the interior attractor points around; dragging the endpoints anchors the curve to different places. Move the curve by dragging it directly.

- *Box,* □: Click first for one corner, then again for the opposite corner. Hold down the *Shift* key to constrain it to a square. This object, like the next two, has a border regarded as a "line" for color and style, as well as a "fill" for color or pattern choices.
- *Polygon,* ⬠: Click for the first point, then move to new locations to click the others; double-click for the final point, which you want to be connected back to the first one. Hold down the *Shift* key to force line segments to be horizontal, vertical, or at 45°. This object has a line border and a fill like a box.
- *Ellipse,* ⬭: First click for the center, move the mouse and follow the bounding outline to the size and shape you want, and finally click again. Hold the *Shift* key to force it to a circle. This object has a line border and a fill like a box.
- *Text,* **A**: This is how you add annotation to your model to label things or provide documentation. Clicking the button brings up a dialog box where you type in your text; use *Ctrl+Enter* to go to a new line and *Ctrl+Tab* for a tab. The Font button lets you change the font, style, and size. Closing this dialog box changes the mouse pointer to cross hairs, which you click where you want to position the northwest corner of your text. Use the underline to move, resize, or reorient the text to a different angle (hold the *Shift* key to constrain the angle to horizontal, vertical, or 45°).
- *Line Color,* 🖌 ▾: If a line object (a line, polyline, arc, Bézier curve, or the border of a shape) is selected, clicking the paintbrush part of the button changes that object to the color underlining the paintbrush. Clicking the drop-down arrow changes the selected line object to the color you then select from the palette, and changes the underline color to this as well. New line objects will be in the underline color. Arena will remember this line color not only for future line objects in this window, but also for new windows and future Arena sessions, until you change it again.
- *Fill Color,* 🪣 ▾: This operates on the interior of a shape (box, polygon, or ellipse) just as Line Color operates on line objects.
- *Text Color,* A ▾: This operates on Text drawing objects just as Line Color operates on line objects.
- *Window Background Color,* 🖼 ▾: This sets the background color of the flowchart view of the model window to the color you select from the palette.
- *Line Width,* ≡ ▾: Operates on the width of line objects just as Line Color operates on their color.
- *Line Style,* ┄ ▾: Provides dash patterns for lines. The No Line option makes the line invisible but it is still logically there (this might make sense for a border of a shape).
- *Arrow Style,* ⇄ ▾: Provides various arrowheads for lines.
- *Line Pattern,* ▬ ▾: Provides different pre-drawn line appearances, like roads, tracks, and belts.
- *Fill Pattern,* ▨ ▾: Operates on the pattern for the interior of a shape just as Fill Color operates on its color.

- *Show Dimensions,* «x» ▾: Shows sizes of shapes and lengths of lines and line segments, for precise drawing.

3.6.4 Printing

All or parts of the flowchart view of the active model window can be printed directly from Arena (in color, if you have it). *File > Print Preview* (or 🔍) lets you preview what's coming, *File > Print* (or 🖨 or *Ctrl+P*) lets you print it, and *File > Print Setup* lets you select a print driver or do a setup of your printer.

If your model is big, the print will extend across several pages. And if you have Named Views, you'll get a print of the current view, followed by a separate print of each named view. If you don't want all this, use Print Preview to see what's on which page, then selectively print only the pages you want. The menu option *View > Page Breaks* will show in the flowchart view where the pages will end if the model is printed.

As an alternative to printing directly from Arena, remember that you can get to a view you like (maybe even during a pause in the animation), press the *Prnt Scrn* (Print Screen) key, switch over to your word-processing document, and paste the shot in where you want it. Then you can print this document when you'd like. You could also paste the Prnt Scrn image into a paint program and perhaps crop it or otherwise clean it up (in fact, that's exactly how we got all the pieces of the Arena window into this book).

3.7 Help!

Arena has an extensive and comprehensive Help system to serve as a reference, guide you through various operations, and supply examples of modeling facets as well as complete projects. The help system is carefully integrated and provides extensive hyperlinks to other areas to aid in getting the information you need quickly and easily. There are several different ways to access the help system, which we'll describe briefly in this section. However, you may find that the best way to learn about help is just to get in and start exploring.

At any time, you can access the full help system from the *Help* menu. The contents of this menu were described above, at the end of Section 3.6.1.

The ↖? button invokes *context-sensitive help*. To use it, just click the button, then click on whatever you're curious about—a toolbar button, a command from a menu, a module in the Project Bar—and you can get to the information you need via a visual path.

Most Arena dialog boxes have a *Help* button that you can press. This is a good way to get direct information on what that part of the software is about, what your options are, how relevant things are defined, related concepts (via hyperlinks to other parts of the Help system), and examples. You can also use the **?** button at the top of the dialog box to access What's This? help information on individual items in a dialog box. Simply click the **?** and then click on the selected item.

In case you forget what a particular button does, you can let your pointer stay motionless on it for a second or two; a little boxed *Tooltip* will appear to remind you what it is.

SMART files, via the Help menu, are subject-categorized Arena models (a couple of hundred) that illustrate, via small working models, a wide variety of Arena modeling capabilities.

Online help is available in the form of technical notes and updates, at http://support.rockwellautomation.com. You can set up your own account there (it's free) to download the latest in such items.

In the Examples folder inside the Arena 10.0 folder are several detailed Example models that you can open, browse, copy, edit (to be safe, edit a copy), and run. If you have an academic or other limited version of Arena, you may not be able to run your models to completion due to the limit on the number of concurrent entities. If you open a large model (one that exceeds the academic version limits), Arena will enter a runtime mode. Runtime mode allows you to make minor changes and run the model, but does not allow significant changes like adding and deleting modules. The Example models illustrate different aspects of building and studying models. The Help topic[4] "Example Models" has a description of what's in each one.

3.8 More on Running Models

Usually you'll just want to run your model to completion as you have it set up, but there are times when you might like to control how the run is done. Entries from the *Run* menu, as well as corresponding buttons from the *Standard* and *Run Interaction* toolbars, let you do this. (See Section 4.5 for details and examples of using these capabilities.)

- *Run > Setup* gives you access to many options for how runs are made for the current model, like deciding whether to see the animation and perhaps running in full-screen mode. These selections and specifications are saved with the current model rather than going into global effect. Click the *Help* button inside any *Run > Setup* dialog box and browse through the hyperlinked topics to get familiar with what the (numerous) possibilities are.
- *Run > Go* (or ▶ from the *Standard* toolbar or the F5 function key) just does it (or resumes it after a pause). If you've made changes to the model since the last check (see below), it gets checked before it's run.
- *Run > Step* (or ▶❙ or F10) executes the model one action at a time so you can see in detail what's going on. This gets really boring so is useful primarily as a debugging or demonstration tool. As with the *Go* button, use of *Step* causes the model to be checked if Arena detects changes since the last check was performed.
- *Run > Fast-Forward* (or ▶▶) disables the animation and executes the run at a much faster rate. You can pause at any time during the run to view the animation. *Fast-Forward* causes the model to be checked if Arena detects changes since the last check was performed.
- *Run > Pause* (or ❚❚ or *Esc*) interrupts the run so you can look at something. Press ▶, ▶❙, or ▶▶ to resume it.

[4] To go to a specific Help topic, select *Help > Arena Help > Index*, type the topic name into the first region, and then open (double-click, or single-click and then click Display) the index topic in the second region below.

- *Run > Start Over* (or ⏮ or *Shift+F5*) goes back to the beginning of the simulation and reruns the model. *Start Over* causes the model to be checked if Arena detects changes since the last check was performed.
- While Arena is running your model, it's in what's called run mode, and most of the model-building tools are disabled. So when the run is over, you need to select *Run > End* (or ■ or *Alt+F5*) to get out of run mode and enable the modeling tools again. If you've paused your run before it terminated, this will kill it at that point.
- Use *Run > Check Model* (or ✓ or *F4*) to "compile" your model without running it. If Arena detects errors at this stage, you're told about them (gently, of course) in an Errors/Warnings window; the buttons at the bottom of this window help you find the problem (for example, by selecting the offending module in the flowchart view of the model window).
- *Run > Review Errors* recalls the most recent Errors/Warnings window containing whatever Arena found wrong during that check.
- *Run > Run Control > Command* (or 🖥) gets you to an interactive command-line window that allows control over a lot of how the run is done—like interrupts and altering values. This also checks the model, if required, and starts the run if it's not yet started.
- *Run > Run Control > Breakpoints* (or ✋) lets you set times or conditions to interrupt the model in order to check on or illustrate something.
- *Run > Run Control > Watch* (or 👓) establishes a window in which you can observe the value of a variable or expression as the run progresses. *Run > Setup > Run Control* lets you determine whether this is concurrent with the run or only when the Watch window is the active window (the latter will speed things up).
- *Run > Run Control > Break on Module* (or ✋) either sets or clears a break put on the selected module. A break on a module halts execution when an entity starts or resumes execution of the logic for the module.
- *Run > Run Control > Highlight Active Module* causes a flowchart module to be highlighted when it is being executed, which provides a visual indication of the action during the animation.
- If *Run > Run Control > Batch Run (No Animation)* is checked, the model will be run without any animation. This is even faster than Fast-Forward, and is usually used when the project is in production mode to produce adequate statistics for precise analysis.
- *Run > SIMAN* allows you to view or write to a file the model file (*.mod*) and experiment file (*.exp*) for the code in the underlying SIMAN simulation language that your Arena model generates. To understand this, you'll of course need to know something about SIMAN (see Pegden, Shannon, and Sadowski, 1995).

3.9 Summary and Forecast

After this guided tour through Arena, you should be a pretty knowledgeable (and tired) tourist. In Chapters 4 through 9, we'll set you out to explore on your own, but with a pretty detailed road map from us. In those chapters, you'll build a sequence of progressively more complex models that illustrate many of Arena's modeling capabilities and sometimes require you to perform a few creative modeling stunts. We'll also integrate at the end of Chapter 4, throughout Chapter 6, at the end of Chapter 7, and in Chapter 12 some information on the statistical aspects of doing simulation studies with Arena, on both the input and output sides.

3.10 Exercises

3-1 Make five replications of Model 3-1 by just asking for them in *Run > Setup > Replication Parameters*. Look at the output and note how the performance measures vary across replications, confirming Table 2-4. To see the results for each of the five replications individually, you'll need to open the Category by Replication report; the confidence-interval half-widths can be seen in the Category Overview report, however.

3-2 Implement the double-time arrival modification to Model 3-1 discussed in Section 2.6.3 by opening the Create module and changing the Value `5` to `2.5` for the mean of the exponential distribution for Time Between Arrivals (don't forget to click *OK*, rather than *Cancel*, if you want this change to happen). Make five replications and compare the results to what we got in the hand simulation (see Figure 2-4). To see the results for each of the five replications individually, you'll need to open the Category by Replication report.

3-3 Lengthen the run in Model 3-1 to 12 hours for a more interesting show. If you want the plots to be complete, you'll have to open them and extend the Time Range (mind the units!), as well as possibly the Maximum value for the *y* axis in the Number in Queue plot. You might also need to increase # History Points in the Number in Queue Plot. Make just one replication.

3-4 Implement the change to Model 3-1 described in Exercise 2-4 from Chapter 2. Open the Process module, change the Delay Type to `Expression`, and enter the appropriate Expression (use the Expression Builder if you like). Run the model for 20 minutes and check against your hand-simulation results. If they're different, what might be the explanation (assuming you did everything right)?

Next, try running this for 24 hours and watch the queue-length plot (for both plots, change the Time Range to `1440` and the # History Points to `1000`, and in the queue-length plot, change the Maximum for the *y*-axis to `50`). To allow more room in the queue animation, click on the line for the queue and drag its left end to the left. What's happening? Why?

3-5 Implement the additional statistic-collection function described in Exercise 2-1, and add a plot that tracks the total number of parts in the system (also called *work in process*, abbreviated as WIP) over time. Note that at any given point in simulated time, WIP is the number in queue plus the number in service; you might also use the Expression Builder and check the Help topic EntitiesWIP Variable.

3-6 Modify Model 3-1 with all of the following changes:

- Add a second machine to which all parts go immediately after exiting the first machine for a separate kind of processing (for example, the first machine is drilling and the second machine is washing). Processing times at the second machine are the same as for the first machine. Gather all the statistics as before, plus the time in queue, queue length, and utilization at the second machine.

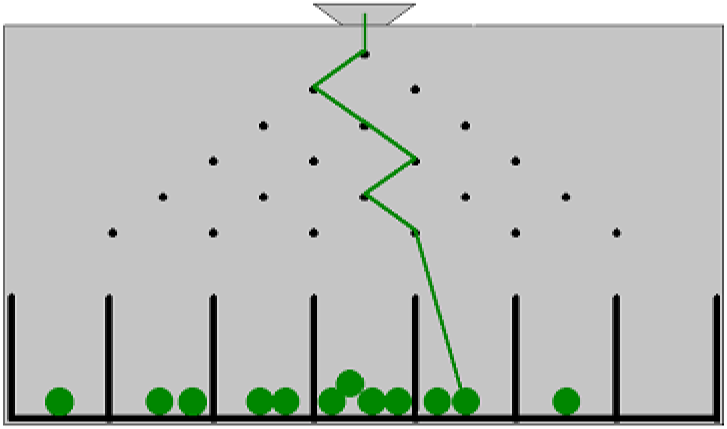
- Immediately after the second machine, there's a pass/fail inspection that takes a constant 5 minutes to carry out and has an 80% chance of a passing result; queueing is possible at inspection, and the queue is first-in, first-out. All parts exit the system regardless of whether they pass the test. Count the number that fail and the number that pass, and gather statistics on the time in queue, queue length, and utilization at the inspection center. (HINT: Try the Decide flowchart module.)
- Include plots to track the queue length and number busy at all three stations. Configure them as needed.
- Run the simulation for 480 minutes instead of 20 minutes.

3-7 In Exercise 3-6, suppose that parts that fail inspection after being washed are sent back and re-washed, instead of leaving; such re-washed parts must then undergo the same inspection, and have the same probability of failing (as improbable as that might seem). There's no limit on how many times a given part might have to loop back through the washer. Run this model under the same conditions as Exercise 3-6, and compare the results for the time in queue, queue length, and utilization at the inspection center. Of course, this time there's no need to count the number of parts that fail and pass, since they all eventually pass (or do they?). You may have to allow for more room in some of the queue animations and plots' *y*-axes.

3-8 In Exercise 3-7, suppose the inspection can result in one of three outcomes: pass (probability 0.80, as before), fail (probability 0.09), and re-wash (probability 0.11). Failures leave immediately, and re-washes loop back to the washer. The above probabilities hold for each part undergoing inspection, regardless of its past history. Count the number that eventually fail and the number that eventually pass, and gather statistics on the time in queue, queue length, and utilization at the inspection center. (HINT: Explore the Decide flowchart module and contemplate a higher-dimensional coin to flip.)

3-9 In Model 3-1, suppose that instead of having a single source of parts, there are three sources of arrival, one for each of three different kinds of parts that arrive: Blue (as before), Green, and Red. For each color of arriving part, interarrival times are exponentially distributed with a mean of 15 minutes. Run the simulation for 480 minutes, and compute the same performance measures as for Model 3-1. Once the parts are in the system, they retain their correct color (for the animation) but are not differentiated for collection of statistics on time in queue, queue length, or utilization (that is, they're lumped together for purposes of processing and statistics collection on these output performance measures); however, collect statistics separately by part color for total time in system. Processing times at the drilling center are the same as in Model 3-1 and are the same regardless of the color of the part.

3-10 In science museums, you'll often find what's called a *probability board* (also known as a *quincunx*):



This is like a big, shallow, tilted baking pan with a slot at the midpoint of the top edge through which marbles roll, one at a time, from a reservoir outside the board; let's say the reservoir has *k* marbles in it. Just below the slot is a fixed peg, which each incoming marble hits and causes the marble to roll left or right off of; assume that you've tilted the board so that there's an equal chance that the marble will roll left vs. right (interpret "left" and "right" from your viewpoint as you look at the board from in front of it, which is the opposite from the marbles' viewpoint as they dive down the board nose first on their stomachs). Below this peg is a row of two pegs, parallel to the top edge of the board but offset horizontally from the first peg so that the two pegs in this second row are diagonally arranged below the first peg, as in the picture. Assume that the board's tilt angle, the peg spacing, the marbles' mass, and the gravitational field of the host planet are just right so that each marble will next hit exactly one of the two pegs in the second row (which peg it hits is determined by whether it rolled left or right off of the first peg). The marble will next roll left or right off of whichever peg it hits in the second row (again, assume a 50-50 chance of rolling left vs. right). The next parallel row of pegs has three pegs in it, again offset so that each marble will hit exactly one of them and roll left or right, again with equal probabilities. This continues through the last row; let's say that the number of rows is *n* so that the last row has *n* pegs in it ($n = 6$ in the picture, counting the first peg at the top as its own row). After rolling off of a peg in the last row, the marble will land in exactly one of $n + 1$ bins arranged diagonally under the last row of pegs. Create an Arena simulation model to simulate a probability board with $n = 6$ rows of pegs and $k = 1000$ marbles in the reservoir. Animate the marbles bouncing down the board, and also animate the number of marbles accumulating in the bins at the bottom with Level animation objects from the *Animate* toolbar. In addition, count the number of marbles that land in each of the 7 bins. The proportion of marbles landing in each bin estimates the probabilities of what distribution? What if somebody opens a window to the left of the board and a wind comes in to blow the marbles toward the right as they roll, so that there's a 75% (rather than 50%) chance that they'll roll to the right off of each peg?

3-11 In Exercise 3-9, as part of the processing of parts, an inspection is included (there's no extra processing time required for the inspection, and during the inspection, the part entity continues to occupy the resource). Each part passes this inspection with probability 0.93, regardless of its color. If it passes, it just leaves the system, as before. If it fails, it must loop back and be re-processed, going to the end of the queue if there is one and taking an amount of time for re-processing that is an independent draw from the same processing-time distribution; parts undergoing re-processing must still be inspected (with the same pass probability) and can fail multiple times (there's no limit on the number of failures for a given part) before finally passing and leaving. Adjust the plots if needed so they show the entire curve. Compare the average time of parts in the processing queue, the average length of the queue, the resource utilization, and the average time in system by part color with what you got from Exercise 3-9 ... does it make sense? (HINT: Explore the Decide module, with "Type" chosen to be "2-way by Chance.")

3-12 In Exercise 3-11, after (finally) passing inspection, parts' paint need to be touched up, so they are sent to one of three separate touch-up-paint booths, one for each color, with each part being directed to the booth for its color (of course); each touch-up booth has its own FIFO queue and its own (single) server. Touch-up times are TRIA(3, 9, 18) minutes, regardless of color. Add a plot that tracks, on a single set of axes, the queue lengths of each of the touch-up booths (with appropriate colors for each of the three curves). Collect the same output statistics as for Exercise 3-9, in addition to the time in queue, number in queue, and utilization for each touch-up booth. (HINT: Explore the Decide module some more, this time with "Type" chosen to be "N-way by Condition" and then "Add" conditions based on Entity Type.)

3-13 In Model 3-3, time studies showed that moving to this integrated work entailed an average increase of 18% in the time it takes to complete each of the four tasks to process an application since employees are no longer specialized in just one task, as they were in Model 3-2. Model this as increasing each service time by 18% from what it was in Model 3-3. If this happens, is the generalized parallel integrated-work scheme still advisable, in comparison with the specialized serial organization of Model 3-2?

3-14 Five identical machines operate independently in a small shop. Each machine is up (i.e., works) for between six and ten hours (uniformly distributed) and then breaks down. There are two repair technicians available, and it takes one technician between one and three hours (uniformly distributed) to fix a machine; only one technician can be assigned to work on a broken machine even if the other technician is idle. If more than two machines are broken down at a given time, they form a (virtual) FIFO "repair" queue and wait for the first available technician. A technician works on a broken machine until it is fixed, regardless of what else is happening in the system. All uptimes and downtimes are independent of each other. Starting with all machines at the beginning of an "up" time, simulate this for 160 hours and observe the time-average number of machines that are down (in repair or in queue for repair), as well as the utilization of the repair technicians as a group. Animate the machines when they're either undergoing repair or in queue for a repair technician, and plot the total number of machines down (in repair plus in queue) over time. (HINT: Think of the machines as “customers” and the repair technicians as “servers” and note that there are always five machines floating around in the model and they never leave.)