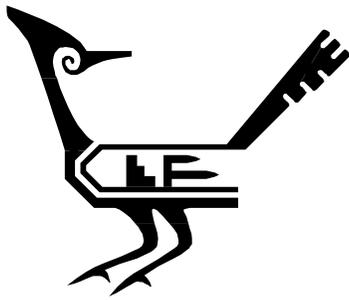


Quick Start

1



© 1993-2003 by Performance Engineering Services Division, L&S Computer Technology, Inc., #110 PO Box 9802, Austin, TX 78766. All rights reserved worldwide. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the express written permission of Performance Engineering Services.

SPE-ED is a trademark of Performance Engineering Services.

Welcome to *SPE•ED* and thank you for selecting it to help you apply the Software Performance Engineering techniques. *SPE•ED* is a “methodology companion” for the Software Performance Engineering techniques described in the book, *Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software* by Connie U. Smith and Lloyd G. Williams. The book is referenced in sections of this user guide as “the SPE book.” The references explain the SPE techniques to help you understand the connection between the techniques and the tool.

This chapter first explains *SPE•ED* terminology and some general concepts you need to operate it. Then, it walks you through three demonstrations that illustrate how to use *SPE•ED* to create and evaluate software and system models. You will learn the most about *SPE•ED* in the shortest time if you follow the detailed steps in the demos then experiment with it later.

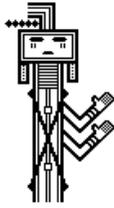


If You Refuse to Read the User Guide

SPE•ED was created specifically to support the SPE methods — it is not merely a modeling tool or a drawing package. It has a streamlined interface for creating execution graph models—it does not work like an ordinary drawing package.

Please scan this *SPE•ED* Quick Start chapter. It explains the streamlined interface and it walks you through all the basic concepts and commands that you need to know to start using *SPE•ED* effectively. It takes you through the creation and solution of a software model, explains the overhead matrix, and creates and solves a system model. Chapters 3 through 8 are primarily for reference — refer to them if you have a specific question about the tool. Chapters 9, 10, and the Appendix describe how to use *SPE•ED* to evaluate systems. Chapter 10 describes features for modeling distributed systems and how to use them.

The Appendix is an overview of the SPE methods. Start there if you are unfamiliar with SPE. The tool is specifically designed to support them and you will find it much easier to use if you are familiar with the methods.



SPE·ED Basics

What You Should Know

You should be familiar with the Software Performance Engineering methods as described in the SPE book. An overview of them is in the *SPE Basics* appendix. Pertinent references are included in the user guide when you should understand the SPE concepts in order to use the tool effectively. This background is essential to the effective use of *SPE·ED*.

User Guide Notation

SPE·ED commands that appear on a pull-down menu or an on-screen menu are distinguished from surrounding text in this user guide. This sentence illustrates by referring to the Quit command on the File menu. Buttons on dialog boxes are distinguished from commands with their font and style. For example, the Cancel button is on most dialogs, and the OK on-screen command is not the same as the OK button that appears on some dialog boxes. Notice the groups of on-screen commands on the Software model screen under the headings “Display:” and “Specify:”. The user guide refers to these commands with the following notation: Display: Software mod and Specify: Names. Thus, the on-screen Display: Overhead command is not the same as the Specify: Overhead command.

Model Composition

Models are contained in *projects*. Each project contains one or more related *scenarios*; each scenario is a software execution model. Scenarios consist of a hierarchy (or tree) of subgraphs. Each subgraph defines the *processing steps* for the software execution structure with nodes and arcs. Each subgraph in the tree is said to be a subgraph of its parent node.

Each project contains one system execution model. Each scenario in the project defines the software execution characteristics for a workload scenario to be analyzed. See section 7.2 in the SPE book for details on performance scenario selection and data requirements.

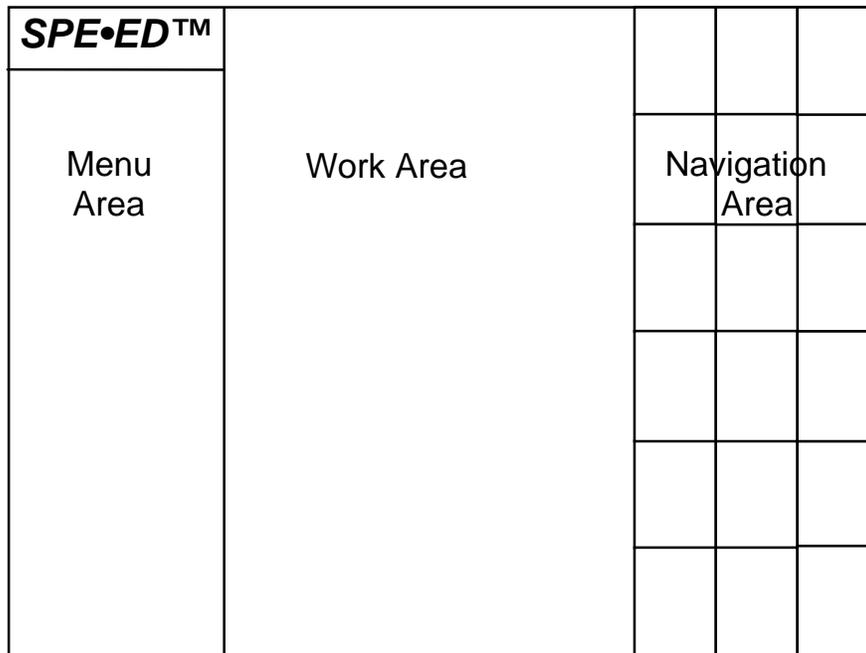
SPE•ED Startup

Start *SPE•ED* by double-clicking on the application. If you are unable to launch, or if *SPE•ED* is unable to open the *SPE•DB* database, refer to the Installation Instructions to resolve the set-up problems before you continue.

You may leave *SPE•ED* anytime during the demonstration sessions by choosing Exit from the File menu. An auto-save feature (described later) displays the save dialog box if you have made changes since the last save. Don't save the changes so it will be easy for others to use the demos.

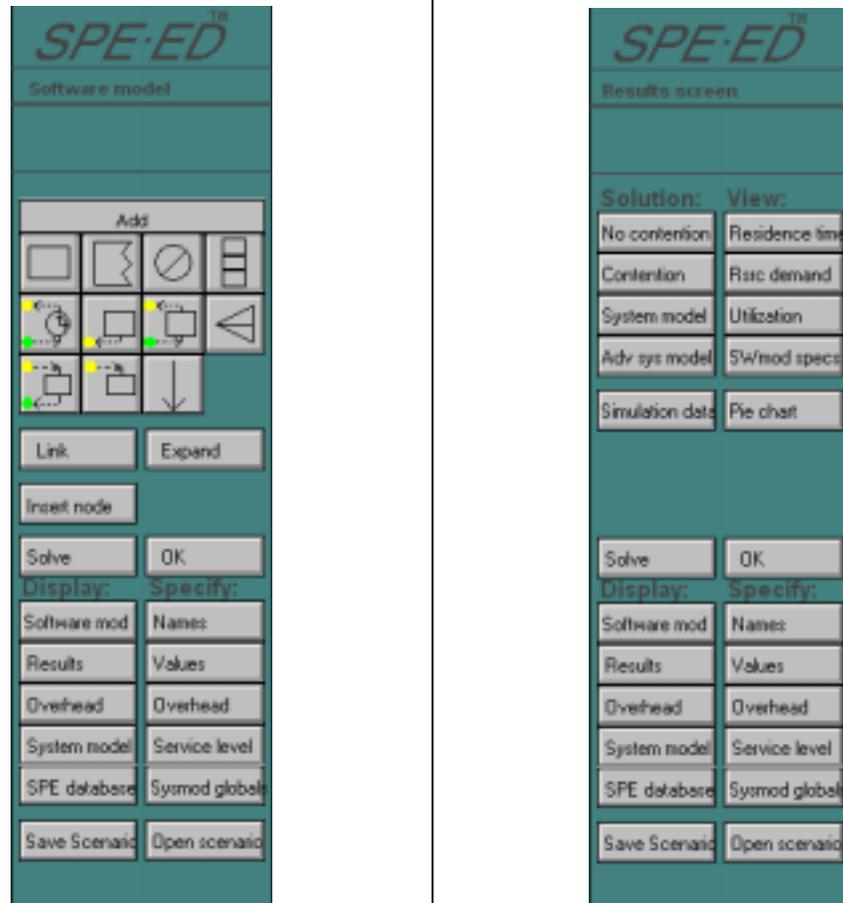
Basic Screen Layout

SPE•ED screens share the same general layout. Most of the screen is designated as the Work Area, the area that contains the model information. To the right of the Work Area, the Navigation Area contains miniatures of up to 27 submodels in the current hierarchy (depending on the size of your screen). The on-screen Menu Area on the left contains a button for each frequently-used command.



Command Selection

Select on-screen commands by positioning the cursor on top of the appropriate button in the Menu Area and pressing the mouse button. The selected command is highlighted. There are two on-screen menus (shown in the following figure). The menu on the left appears in the Menu Area when the Software model screen is displayed; the menu on the right appears when any other screen is displayed. The Software model menu has buttons at the top for specifying processing steps. Buttons in the lower section are for displaying screens, specifying model parameters, and other miscellaneous commands. The Results menu has buttons at the top for selecting the solution methods and results for viewing. The same buttons appear in the lower section.



SPE-ED also uses standard pull-down menus. There are nine menus in the menu bar.

Select the Open scenario command from the on-screen menu. A dialog box appears with the list of scenarios in the database; it will be described later. Select the 'Intro scenario' by double-clicking on its name in the list, or clicking on the name then clicking the Open button. The scenario appears in the Work Area of the Software model screen.



Command Termination

Many of the commands do a specific task then terminate by themselves (e.g., Open scenario). Some of the commands stay active until you terminate them by selecting another command. For example: you select the Add nodes command (on the Edit menu) once and continue to add nodes until you select another command.

The OK command is convenient for terminating these multi-step commands. If OK is selected when no command is active, nothing happens, so it is also convenient to use if you are not sure what is in progress.

Navigation

Each graph in the hierarchy is pictured in a separate box in the Navigation Area. The Navigation Area provides a world-view of the model and allows direct access to any graph in the hierarchy. The subgraphs in the Navigation Area are drawn in different colors. The parent node is drawn in the same color as its subgraph to show the correspondence between parents and their subgraphs.

The Current Subgraph is the one that appears in the Work Area of the Software model screen. To change the Current Subgraph, position the cursor over the desired box in the Navigation Area and press the mouse button. An OK command is automatically issued so if a command is active, it is terminated. The subgraph in the Work Area is then replaced by the new selection from the Navigation Area, which becomes the Current Subgraph. The subgraph previously in the Work Area is re-drawn in the navigation area incorporating any changes made.

Position the mouse cursor over the top-middle Navigation box and press the mouse button. The Current Subgraph changes to the subgraph for 'Parent node' and shows its two processing steps. Return to the original subgraph by selecting the top-left Navigation box. A similar technique navigates through results. It is described later.

Definition of Processing Steps

SPE•ED defines processing steps with execution graph nodes and arcs.

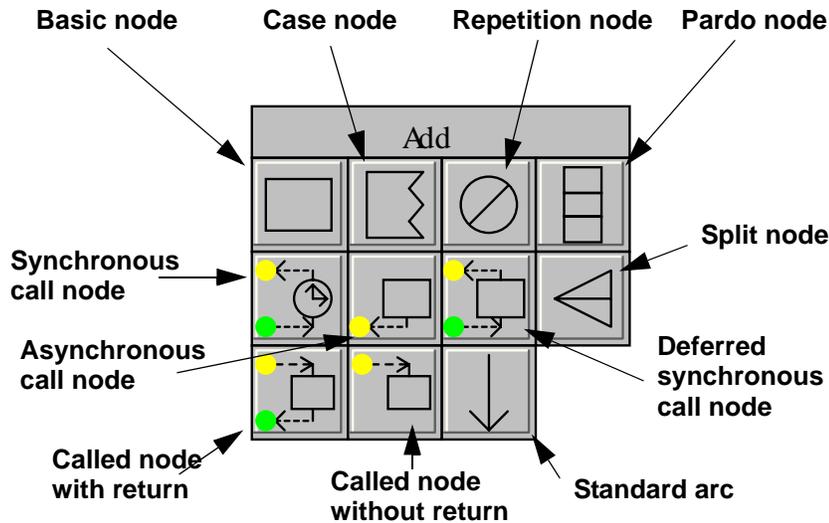
- Basic node** - represents a component at the lowest level of detail appropriate for the current design or implementation stage.
- Case node** - represents conditional execution of components, each with a probability of execution.
- Repetition node** - shows which components are repeated; the node's repetition factor specifies the number of times the component repeats.
- Pardo node** - represents parallel execution paths, each with a probability of being initiated.
- Standard arc** - represents flow of control from one node to the next.
- Synchronization** - 3 nodes show synchronous, asynchronous, and deferred synchronous calls. 2 nodes show called nodes and distinguish whether or not they send a reply. These nodes are covered in detail in chapter 7.
- Split node** - shows the initiation of concurrent processes that need not join.

Two additional nodes are generated as models are created:

- Expanded node** - shows that processing details are expanded in a subgraph at the next level of detail.

Link node - represents a processing step whose execution requirements are specified in a previously saved subgraph.

The demos in this chapter use basic nodes, repetition nodes, expanded nodes and standard arcs. The other nodes are covered in the *Software Model* and *Distributed Systems* chapters



Model Creation with *SPE•ED*

SPE•ED has a streamlined interface for creating execution graph models – it does not work like an ordinary drawing package.

- Nodes are automatically connected with arcs as they are defined.
- Combinations of nodes and arcs are created for *compound nodes*: case, repetition, pardo, and split nodes. The default combinations are described in the next section.
- Most nodes in execution graphs are basic nodes, so after each node placement, the Basic node symbol becomes active.
- When commands affect nodes or arcs, such as Delete, Expand, Insert node, Link, and Connect nodes, select the command first, then select the appropriate node(s) or arc(s). The Delete and Add nodes commands let you do multiple deletions or connections without re-selecting the command.

To take advantage of these features, first define all the processing steps in a subgraph in the order they execute; use the default version of the compound nodes then modify the default combinations if

necessary; and finally specify all the names at once. The specific steps for these actions are described later in this chapter.

Structured Drawing

SPE•ED automatically creates combinations of nodes and arcs for case, repetition, pardo, and split nodes. When you place a case node, two basic nodes are attached. When you place a repetition node, one basic node is placed in the loop. Pardo and split nodes attach two basic nodes.

The generated nodes can be expanded to describe additional processing steps, deleted if not needed, or additional nodes can be attached if more nodes are needed. The procedure for modifying the execution graph structures is described later. Nodes are automatically connected with arcs in the sequence they are defined.

SPE•ED Screens

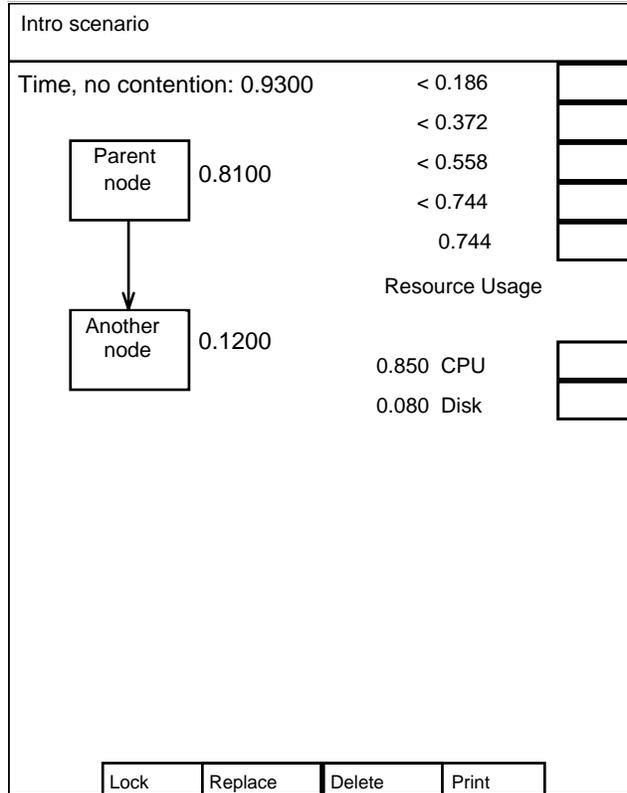
Five screens show model information:

- | | | |
|----------------|---|---|
| Software model | - | shows the processing steps defined for the current scenario. |
| Results | - | shows combinations of results for the current scenario. |
| Overhead | - | specifies the computer resource requirements for software model specifications. These specifications are usually created by performance specialists and reused by all projects that execute on the same facility. |
| System model | - | shows the scenarios in the project, the computer facilities, allocations of scenarios to facilities, and the results of the system execution model solution. |
| SPE Database | - | provides query and viewing capabilities to locate, peruse, and export information about models and results. |

The screens are selected with the five on-screen commands under 'Display:' or from the pull-down Display menu.

Note that "tabs" were not available when *SPE•ED* was originally created. If they had been, the Display screens would be tabs – you would switch between screens by clicking on an appropriate tab. A future version will convert the display commands to tabs.

Select the Solve command. The Software model screen is replaced by the Results screen. The name of the current screen appears in the Menu Area under the *SPE•ED* title. You can change screens anytime by selecting the screen you wish to see. For example, select Display: Software mod. The previous Software model screen replaces the results; select Display: Results to return. The other screens are described later in the chapter.



Anytime a screen in the Work Area replaces the screen you want to see, select the appropriate command under 'Display' to return to a previous screen.

Working-Set Screens

Three of the screens, Results, Overhead, and System Model, use split-screens to increase the amount of information displayed in the Work Area. For example, the Results screen can display up to four sets of specified values or statistical results in any combination you wish.

If you request more information after the Work Area is full, *SPE•ED* uses a simple algorithm to select a quadrant for replacement. A set of commands at the bottom of each working-set screen lets you control the placement decisions:

Lock	-	keeps the screen visible.
Replace	-	replaces the screen if another exists, but does not delete it.
Delete	-	deletes the screen and replaces it if another is hidden.
Print	-	prints the contents of the screen (see <i>Printing</i> for more information)

Select Display: Results then select View: Rsrc demand. The Results screen splits into quadrants and two sets of results are shown. The solutions and results are described later. Select Display: Software mod; after the Software model screen appears select Display: Results. The Results screen returns with the same contents it had when it was replaced. Position the mouse cursor over the Delete box at the bottom of the Resource demand quadrant and press the mouse button. The corresponding quadrant is removed. Only one quadrant remains, so the “Time, no Contention” quadrant appears in the large screen. Refer to the section on *Controlling Working-Set Screens* for more information.

SPE Database

SPE•ED keeps a database of models and results as you conduct studies for convenient model retrieval, performance tracking throughout development, trend analysis, activity reports, model and performance-study archives, model reuse, model merging, and audit trails. SPE database access is through screens and menus in the SPE Database window (see *Using the SPE Database Retrieval Screens*).

The data is captured in two ways:

1. An auto-save feature displays a dialog box when it thinks you might have completed a performance study. For example, after you create a scenario, solve it, then select a command that may change the processing steps or resource requirements, the auto-save prompt appears.
2. You may explicitly select the Save scenario or Save system model commands.

Select the Close project command from the File menu. The save dialog box prompts for a description that will help you locate this study later. It also gives you the option to Replace last version of the model, or Don't save. These help you manage the size of the growing SPE Database when you know that you will not need a version of the model in the future (e.g. if

it contains an error). Select Don't save now so future users can see the original scenario. The SPE Database keeps information about your models and studies.



A Simple SPE Study

Demo 1 Overview

To begin, we will create and solve a simple example. The execution graph and software resource requests are shown here. If *SPE•ED* is not running, follow the startup steps in the previous section.

The demonstrations focus on how to create the models and produce results rather than how to interpret the results for SPE studies. *Review the results in the demonstrations to understand what they tell you.*

Open a Project

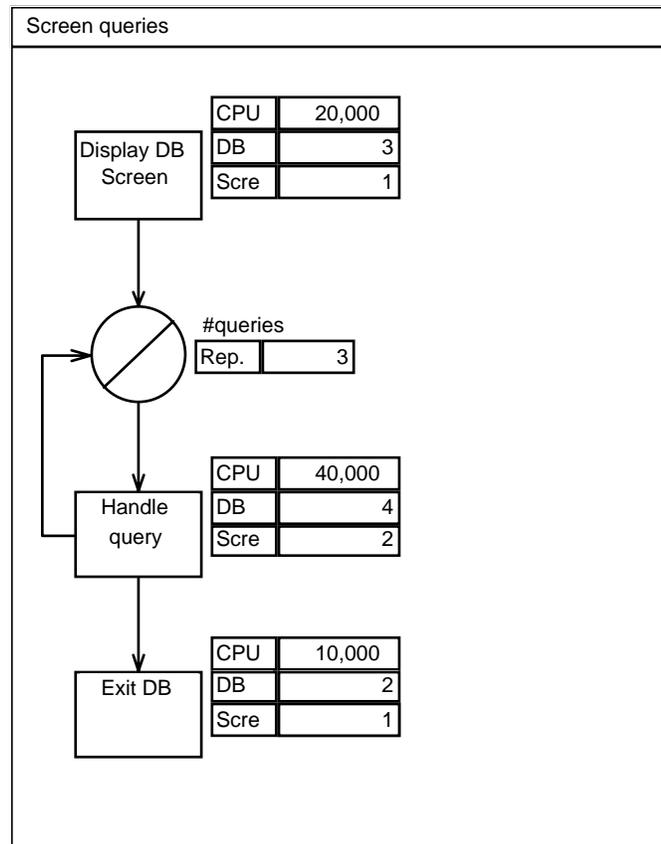
Model information is stored in *projects*. To simplify this first example, we will use the 'Manual Demo 1' project. Choose Open project from the File menu. You should see 'Manual Demo 1' on the list of current projects. Select the project by clicking on its name in the list.

The dialog box also has a Cancel button to terminate the Open project command. The dialog can also be used to create a project by typing a new name in the Project name box and selecting New. Names can be up to 32 characters, and can include spaces. Quotes are not necessary, if entered they become part of the project name. If you want to change the name later, you can do so in the Projects screen of the SPE database.

The Same scenario selection box is empty when the feature is off (the default). If you click in the box an "x" appears and the feature is on. This feature opens the scenario in the project that was most recently open.

We want to create the scenario, so leave the box empty and select the Open button.¹

-
1. Some users have found that they responded to a dialog box before they read the instructions and discovered that they chose a different button than the demo specifies. This is usually reversible. You can select the OK command then re-select the command that triggered the dialog and make a different choice.



Create the Scenario

You have a choice in creating a new scenario:

1. You can start by assigning a name to the new scenario by selecting **New** from the File menu. This displays a dialog box showing existing scenario names (in this project) and lets you enter the new name (up to 32 characters).
2. You can also start by describing the processing steps and naming the scenario later. (The auto-save feature asks for a name if you have not named the scenario before saving.)

Let's proceed with option (2) and define the processing steps next. (If you already selected **New** or **Open scenario**, select the **Cancel** button.)

Define Processing Steps

The first node in the execution graph is a basic node. Position the cursor over the symbol for the Basic node in the on-screen commands and click the mouse button. Position the cursor in the Work Area where you want the node and click the mouse button again. The basic node is drawn on the screen. *SPE•ED* centers nodes on the closest grid intersection to keep the execution graphs aligned. Tip: *start your graph in the upper-left section of the Work Area to maximize the screen area available for your graph and to avoid the sections of the screen that will display the results legend (described later).*

Next we want a repetition node, so position the cursor over the symbol for the repetition node, click the mouse button, then position the cursor where you want the repetition node and click the mouse button again. The node is drawn and is automatically connected to the previous node with a standard arc. A basic node is automatically placed in the loop and connected with arcs. Most nodes in execution graphs are basic nodes, so after each node placement, the Basic node symbol becomes active.

Position the cursor in the Work Area where you want the last basic node and click again to create it. Terminate the Basic node command by selecting the OK command.

Next, we will enter the scenario specifications. The five on-screen buttons under the Specify: heading are in the order (top to bottom) that you will likely want to enter the specifications: Names, Values, Overhead, Service level, and Sysmod globals.

Specify Names

Select the Specify: Names command. The nodes have a light highlight color to show that they can be edited. The cursor changes to the I-beam cursor (in the textedit areas of the screen) to remind you that you are in the process of specifying and the specifications have not yet been saved. Specify the names using standard text-editing conventions.

If you entered a scenario name in the “New scenario” dialog box it appears at the top of the screen with the insertion point at the end of the text. If you did not, the default name ‘Scenario’ appears and is highlighted so it is easily replaced.

Type a unique scenario name with your initials and a description such as ‘CS-Screen queries.’ You can move to the processing step names by either pressing Enter or by positioning the I-beam cursor inside a processing-step node and clicking the mouse button. You can select any node with the mouse. The Enter key moves to the next node in sequence, the up-arrow moves to the previous node in sequence. Names can be up to 32 characters long (you may not be able to see all 32 characters inside the small boxes). Text automatically wraps to the next line in a node as in a word-processing document. The standard Cut, Copy, Paste, and Clear commands also work in the usual manner. You can either select them from the Edit

menu or with the standard command key equivalents. (Sorry, Undo is not yet implemented).

Enter the names shown in the original figure (in the *Demo 1 Overview* section). Select the OK command to save them. The cursor changes to the arrow, and the normal background color is restored in the nodes. Your scenario picture should match the execution graph in the *Demo 1 Overview* section.

Specify Values

Next, select the Specify: Values command. The “Select spec template” dialog box appears. The specification templates contain the names of the software resources that the execution steps require.² The list in the scroll box shows the name(s) of the specification templates defined for the Project. The “Software resources” box below the list shows the names of the resources in the highlighted template. Select the ‘Query specs’ template then select OK.



A template appears by each basic node with the names of the software resources. The I-beam cursor returns, the highlight color is in the templates. The insertion point is in the first box of the template for the first node in the graph.

2. If you are unfamiliar with software resource requirements refer to the *SPE Basics* Appendix or sections 2.1, 4.4, and 7.2.4 in the SPE book for more information.

Type the numbers from the example into the templates then select OK to save them. The same text-edit conventions apply for traversing the cells, Edit menu commands, and correcting mistakes. If you select OK then notice an error, select the Specify: Values command, select the cell with the error, correct it, then select OK.

The values are saved and the templates are hidden. You have the option of showing the templates and values with the Show values command on the Software model pull-down menu. You can toggle Show values on and off by selecting it again. Your model values should look like the picture shown.

Solve the Model

This example presents a typical situation – a performance specialist has already specified the computer resource requirements for the software resource requests. A later example covers these overhead specifications in more detail.

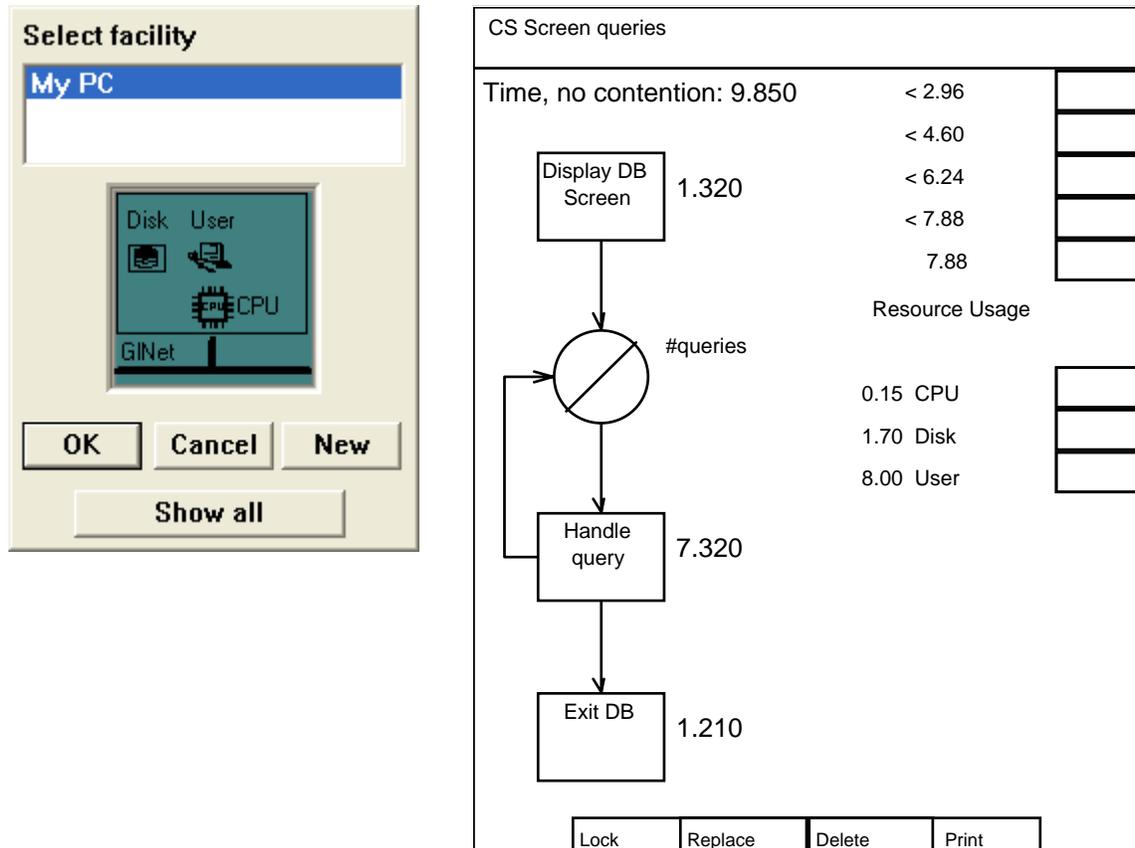
Select the Solve command. The “Select facility” dialog box appears. The facility templates contain the computer system devices that the software requires to execute. The solution results depend on the computer system devices you use to execute the scenario. (If you are unfamiliar with computer resource requirements, refer to the *SPE Basics* Appendix or sections 2.2.7, 4.4, or 7.2.5 in the SPE book.)

The list in the scroll box shows the name(s) of the facility templates defined for the project. The picture below the list shows the highlighted facility. Select the facility named ‘My PC’ then select OK.

The software execution model is solved with the No contention solution (the default). This solution calculates optimistic results for one user of the scenario executing on the selected facility with no contention from other scenarios. The screen changes to the Results screen and the Residence time results are shown. Your results should match those in the figure. They are explained later in *View Results*.

Select the View: Swmod specs command to check the model parameters. The results screen splits into quadrants. The top-left contains the previous view, the top-right contains the values we specified previously. Select the Delete Working-set screen command, at the bottom of the Software model specs top-right quadrant. The view is deleted and the large screen reappears.

You can also view specifications by selecting the Software mod on-screen command under Display. This returns to the software model screen where you can review or change the model. If you change it, select Solve to return to the results screen, otherwise select Display: Results.

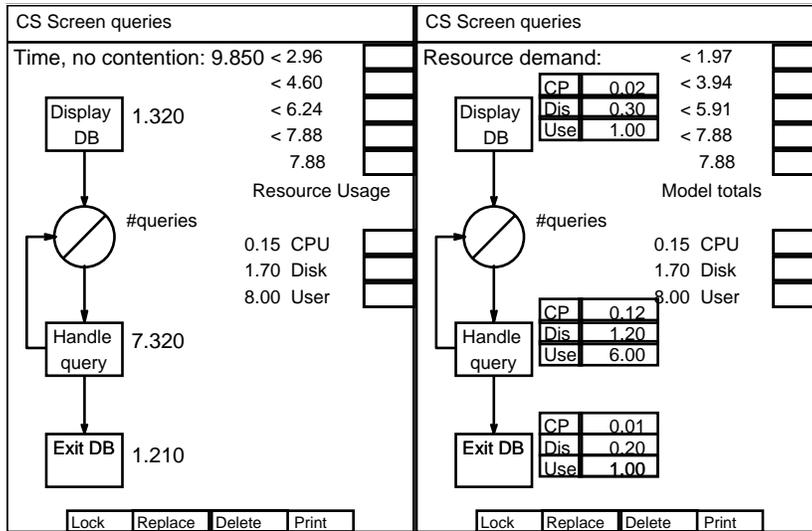


If your Solve or View selections seem to be ignored, it may be because *SPE·ED* will not re-solve a model it thinks is the same as the model it just solved. Refer to *Model Results Displayed on the Screen* in the *Solutions and Results* chapter for tips on avoiding this problem.

View Results

The residence time result shown is the time required for each step and the overall time (when there is no contention for devices). It is thus a lower bound on response time. Note that the time shown for 'Handle query' is the total for all loop repetitions.

The color bar at the top-right corner of the quadrant shows the legend to the colors displayed inside each node. The colors distinguish nodes with low residence times from those with higher values. This highlights potential problems and focuses attention on areas where the greatest improvement may be made.



Below the color bar legend is the total residence time for each device in the facility, and the corresponding colors. This indicates which devices cause the longest delays.

Select the View: Rsrc demand command. The results screen again splits into quadrants. The top-left contains the previous view, the top-right contains the resource demand results. The templates beside each node show the devices in the facility and the total time spent at each device (for each node). The template values are colored according to the color bar legend. This shows how the residence time for each step is divided among the devices and highlights potential problem areas.

The View: Utilization command has no meaning for the no-contention solution. This solution cannot determine actual device utilization because the frequency of scenario executions is excluded from the calculations.

The next section modifies this model. There is no need to save these results unless you want to continue the demo at a later time.



Another SPE Study

Next, we will create and evaluate the subgraph for the 'Handle query' node. This illustrates the steps you would use to study the software design as it evolves. In this example we have designed some details of the processing for 'Handle query'.

Create a Subgraph

Select the Display: Software mod command. After the Software model screen appears, select the Expand command. The auto-save feature displays the "Save scenario" dialog box and gives you several options for saving the model and results. It automatically assigns the scenario name you entered when you specified names. It asks for a description that will help you locate this model later. Descriptions can be up to 40 characters and can include spaces. Quotes are not necessary. Text automatically wraps inside the box.

An 'Enter' triggers the Replace last button and overwrites the last model saved. Other options on the dialog box let you Cancel the command that triggered the save (Expand in this example). Baseline adds a new version of the model and preserves the previous version for later retrieval. Don't save postpones the save.

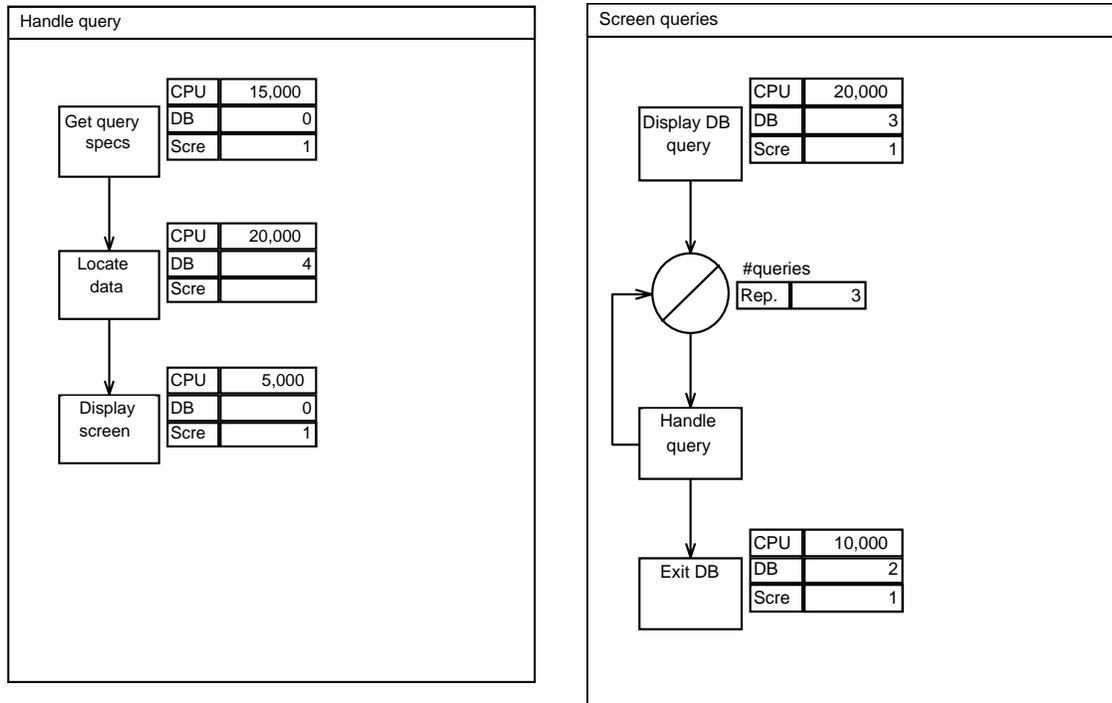
Enter a description and select the Replace last button. The Expand command resumes and displays a message reminding you to click on the basic node you wish to expand. Acknowledge the message, then position the cursor over the 'Handle query' node and click the mouse button. (The 'Enter' key is a quick way to close this dialog box)

A miniature of the top-level processing steps is copied to the top-left Navigation box. The color of the expanded node in it matches the new color of the top bar of the Work Area. The Basic node add command is activated. You could either select a different node to add, or position the cursor where you want a basic node and click.

Draw the 'Handle query' subgraph shown in the following figure in the Work Area. It should require only three mouse clicks in the Work Area. Then select OK.

Enter Specifications

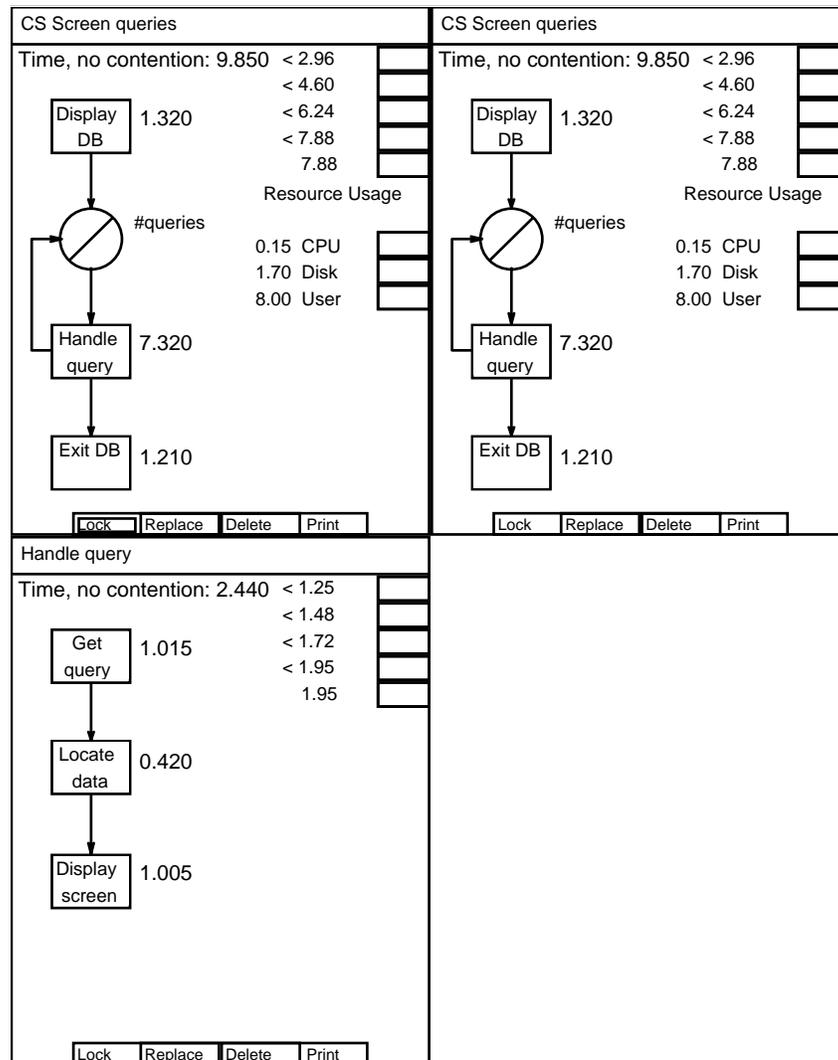
Select the Specify: Names command and enter the names shown in the figure. Note that the default name of the subgraph is the name of the node we expanded, 'Handle query.'



Select **OK**, then select **Specify: Values**. *SPE-ED* assumes that you want the same software specification template for subgraphs as for the parent node. You could select a different one by using the steps described in the *Change Specification Templates* section (see the *Software Model* chapter).

The specification template is displayed and the cursor is in the first cell. Enter the values shown in the figure and select **OK**. Note that leaving a cell blank is equivalent to typing a zero.

Make the top-level of the processing steps the Current Subgraph by positioning the cursor over the top-left Navigation box and clicking the mouse button. The new subgraph is copied to its Navigation box. Select the Show values command from the Software model pull-down menu. (We are using this command frequently for demonstration purposes. You are unlikely to need it this often for regular use, so it is not an on-screen button.) Note that the specification template for the 'Handle query' node no longer appears. Its values are now calculated from the subgraph.



View Combinations of Results

Select the **Display: Results** command. The previous results appear (they may be in a different location). Click on the **Lock** box at the bottom of the quadrant containing "Time, no contention." This will let us compare the results before and after the change. Select the **View: Residence time** command and compare the results before and after. The results are identical. Click the **Navigation** box for the new subgraph to see its residence time. Each 'Handle

query' takes 2.44 sec. The higher level model shows the total for all three repetitions. You can use these steps to view any combination of results for any subgraph in the model by choosing the View command and the subgraph you wish to see.



Contention Solutions

Next, we will examine the effect of multiple users of the scenario on the response time and computer usage. The contention solutions are solved analytically so you see the results quickly. Use them to rule out problems before you use the system model and its simulation solution to evaluate the combined effect of multiple users of multiple scenarios. System models are introduced later in this chapter.

Specify Service Level

Select the Solution: Contention command. The “Service level” dialog box appears because we have not yet specified workload intensity. We could have selected the Specify: Service level command at any earlier point in this example.

The scenario name appears at the top of the box. Below the name is the response-time performance objective. Type 20 (sec.) then press the tab key. This specifies the objective for the end-to-end response time for all queries (as opposed to the time for each user screen). Refer to Section 7.2.2 in the SPE book for a discussion of performance objectives.

Workload intensity is specified as an arrival rate (jobs per second) for open models, or the number of users and the “think time” between job submission for closed models.³ Type an arrival rate of 0.5 jobs per second, make sure the radio button for arrival rate is highlighted, then select the OK button.

The model is solved and the current view of the results is displayed. If the View: Residence time was not displayed, select it to compare the solutions for the three versions of the model.

3. If you are unfamiliar with open and closed models, refer to section 6.2.3 in the SPE book for an overview of contention models. Page 147 describes open versus closed models.

Screen queries

Performance goal
Response time:
20 sec.

Workload intensity

Arrival rate
0.5 jobs per sec.

Number of users
1 Users

60 Think time

OK Cancel

Compare Results to Performance Objectives

Note that the value you enter for the performance objective on the “Service level” dialog box controls the values that appear on the results color bar legend. The value <20 is used for the yellow results, any results that exceed the performance objective will be red, and the other colors are proportionally lower values.

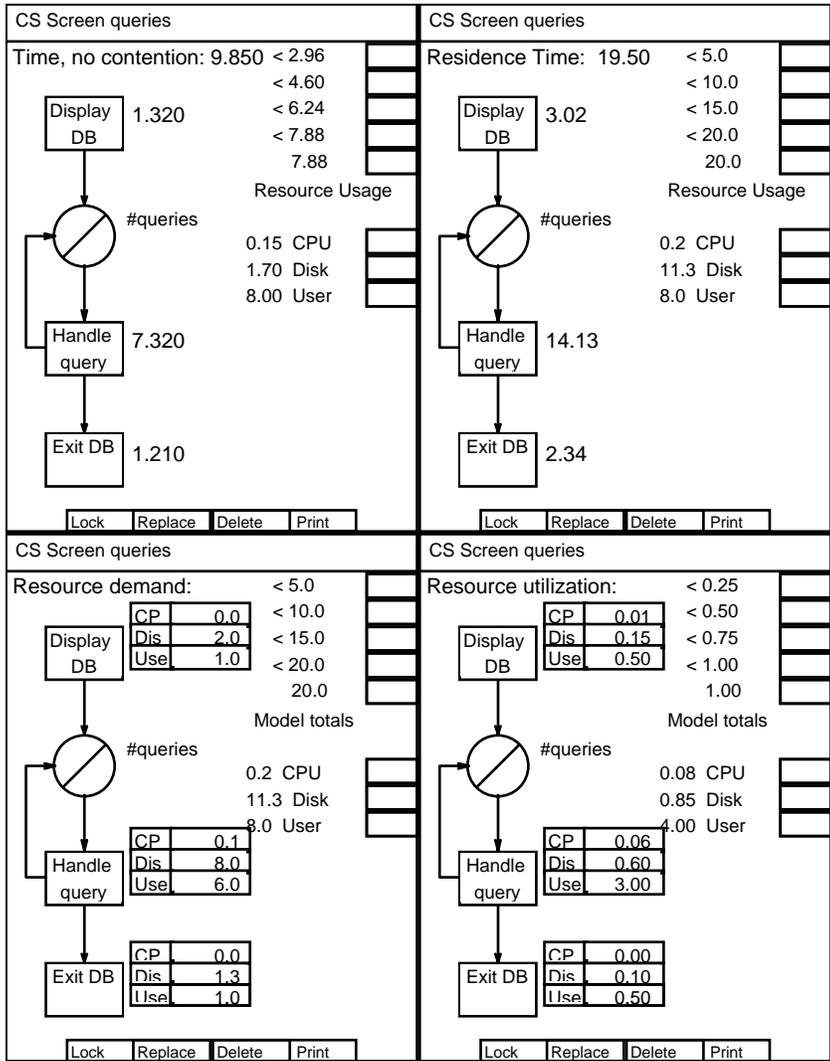
The View: Residence time shows the average elapsed time for each node in each level of the model, and the overall average time for the scenario. The elapsed time includes the time spent in device queues waiting for service. Note that the contention delays are due to other Screen query scenarios, but do not yet include contention with other workloads.

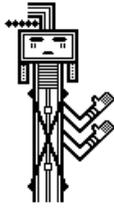
View: Rsrc Demand shows the average elapsed time for each device and each processing step in the graph. The overall average elapsed time for each device is in the summary “Model totals” below the color bar legend.

View: Utilization shows the overall average device utilization for all users and the portion due to each processing step. The color bar legend for utilization is always 0-1.0.

Note that the total utilization for the “user” device is 400%. This indicates that one user display device could not support a Screen query scenario arrival rate of 0.5 jobs per second.

This hypothetical application would require at least 4 user devices to generate scenario requests at that rate.





Demo 1 Summary

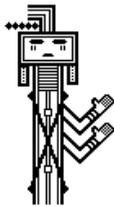
This concludes the first scenario walkthrough. You have seen how to open a project, create a scenario and “draw” the processing steps. We entered the names for the steps, selected the software specification template and specified values for the resource requirements. We solved the model, viewed the results, and used the working-set screen controls.

The second part of the first scenario demonstrated how to modify a scenario, expand nodes, and navigate through the model hierarchy. We also solved the model and combined various views of the results and navigation box selections.

The last part of the first scenario demonstrated service level specifications for performance objectives and workload intensity and results for the Contention model solution. We then compared the results for the software execution model with No contention to the simple Contention model with multiple users of the same scenario.

Next we will examine the process that performance specialists use to specify the computer device overhead for software resource requests. Then we will create and evaluate a system execution model. We will use different projects for the other demonstrations. If you would like to explore the commands for viewing combinations of results, solutions, or navigation box selections you can use this scenario as a test before you proceed to the other examples.

Before you leave this project select the Display: Overhead command to see the simple technique for specifying overhead described in the next section.



Overhead Specifications

First we review the easiest technique for specifying overhead and the reasons that you may not want to use the simple technique. Then we will examine another technique that makes software resource specification easier.

If you are restarting the demo, start *SPE•ED* by double-clicking on the application; then Open project and open 'Manual Demo 1' and select Display: Overhead.

The Overhead Matrix

The overhead matrix shows the devices in the 'My PC' Facility. The device pictures are across the top, and their names are at the top of each column. The device on the far right, GNet, is a special device that represents a "Global Network" shared by other facilities and other scenarios. It is described in the *Overhead* chapter and in the *Distributed Systems* chapter. It is not used in this example (its quantity is set to 0).

The second row specifies how many devices of each type are in the facility. For example if you have 20 disk devices, you have one disk device column and specify 20 for its Quantity row. The third row is a comment that describes the service units that you enter when you specify values for software processing steps. You can tailor this to your environment. For example, CPU may be specified in seconds of service, or as in this example instructions executed.

Software spec template:

Facility template:



Devices	CPU	Disk	User	GNet
Quantity	1	1	1	0
Service units	Instr.	I/Os	Screens	Msgs
CPU	1			
DB		1		
Screen			1	
Service time	1e-006	0.1	1	0.1

The next five rows are for the software resources in the software specification template, Query specs (it uses only three of five). The software resource names in the first column are the titles you see when you Specify: Values for the nodes in the software model.

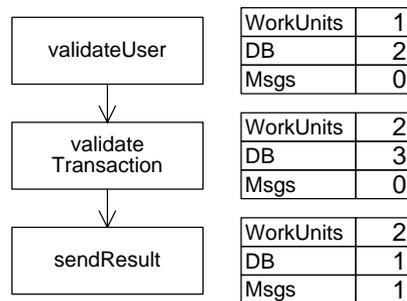
The last row of the overhead matrix specifies the service time for the devices. The time for each CPU instruction is 0.000001; each Disk I/O takes 0.1 sec., and each User screen takes 1 sec.

The values in the center section of the matrix correlate software resource requests with computer device usage. The 1 in the CPU row under the CPU device means that the instructions specified in the software model execute on the CPU device of the 'My PC' facility. Each DB software request sends one I/O to the Disk device. Each screen makes one visit to the user device.

This version of the overhead specifications is simple but it may be more difficult to specify software resource values. For example, DB values must specify the number of physical I/Os, and CPU values must incorporate the overhead of DB requests. The next example shows a more complex overhead matrix that simplifies the specification of software resource values.

Demo 2 Overview

We will create and solve the ATM example described in Section 4.4 in the SPE book. The execution graph and software resource requests are shown here. We will follow the steps in the book for specifying overhead values and compare the results produced by *SPE•ED* to the SPE book's results.



Demo Project

Start *SPE•ED* by double-clicking on its application icon. If you are continuing from the previous example, select Close project from the File menu to clear the previous model, or pro-

ceed to the next step which will close the previous project for you. Don't save the changes to the screen query scenario.

Choose Open project from the File menu. Select the 'Manual Demo 2' project by double-clicking on its name in the list.

Scenario Definition

You should already be familiar with the steps for creating a new scenario, drawing the processing steps, and specifying names and values. If you would like to skip this step, select the Open scenario command and open the 'Withdrawal Demo' scenario. The Open scenario dialog box has the same options and buttons as the Open project dialog. If you would like to practice, create a new scenario that matches the figure. Refer to the earlier section, *A Simple SPE Study*, if you want a reminder of how to create the model.

The Overhead Matrix

Select Specify: Overhead. If the "Select facility" dialog box appears double-click on the ATM Facility. This overhead matrix shows the devices in the ATM Facility and the software resources in the Overhead Demo software specification template. Each row specifies the computer resource requirements from each device that are required for each software resource request. Note the 2 in the DB row under the Disk Column. This means that the 1 DB access we specified for the 'sendResult' node in the software model sends 2 physical I/Os to the Disk device. The 1 DB access also sends 500K instructions to the CPU, and 0 Messages to the Network.

The row for the Msgs software resource shows that each Msg software resource request sends 10K instructions to the CPU, 2 Physical I/Os to the Disk device, and 1 Message to the Network. The 'sendResult' node has 1 Msg so it causes a total of 10K CPU instructions, 2 Physical I/Os, and 1 Network message.

The values shown in the matrix come from the SPE book, Table 4.1

The last row in the overhead matrix specifies the service time for the devices. Each KInstructions takes 0.00001 second, each Disk physical I/O takes 0.02 second, and each Network Message takes 0.01 seconds.

Select the Solve command. The result is the Residence time for the No contention solution for each step and overall. You could view combinations of results as before, but the results for this subgraph are not particularly interesting.

Select the Specify: Overhead command. Change all the service time values across the bottom row to 1, then select OK

Software spec template:

Overhead Demo

Facility template:

ATM Facility



Devices	CPU	Disk
Quantity	1	1
Service units	KInstr.	Phys I/O

Network
1
Msgs

WorkUnit	20	
DB	500	2
Msgs	10	2

1

Service time	1e-005	0.02
--------------	--------	------

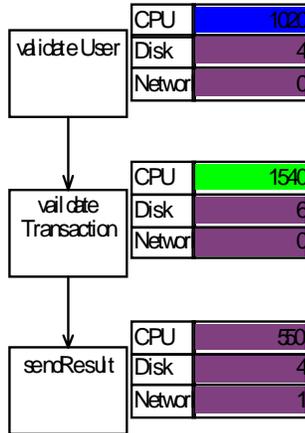
0.01

A dialog box appears because you changed an overhead template that may be used by other scenarios in the project. You have the option of propagating the changes to all scenarios in the project (the OK selection), Create a new facility with the new specifications for this scenario only, or Cancel the OK command. You will typically propagate overhead changes when all scenarios execute on the same facility. For this example, select Create so the original values will be preserved for future users of the Manual Demos. **SPE•ED** assigns a new facility name for you. You can change it later with the Specify: **Overhead** command. Acknowledge the new name message then select View: **Rsrc demand**.

These results show the number of requests for each computer system device for each node, and the model totals. Your results should match those shown in Table 4-2 in the SPE book and shown in the following figure. (Note that I removed the performance objective so that the colors would not be all red). If your results don't match, select View: **SWmod specs** or Display: **Software mod** to verify that the specifications are correct, and Display: **Overhead** to verify that the matrix is correct and the service times are all 1.

This is a good time to review Section 7.2 in the SPE book to confirm that you understand the relationship between software resource requests and computer resource requests.

Resource demand:



< 625
< 1250
< 1875
< 2500
≥ 2500



Model totals
3110 CPU
14 Disk
1 Network



System Execution Models

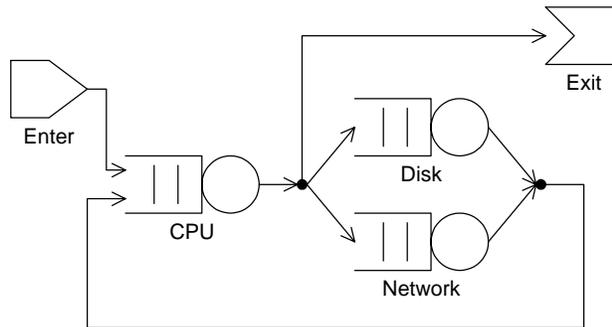
This example shows how to create and solve a system execution model to evaluate the effect of multiple scenarios executing on a computer facility. The computer facility consists of one CPU, one Disk, and a Network as shown in the following figure.

Scenario Definition

Two scenarios are already defined in the 'Manual Demo 3' project. If you have another project open, select the Close project command from the File menu. The auto save feature checks for changes that need to be saved – don't save them. Select Open project and double-click on "Manual Demo 3." Next select Open scenario. This project has two scenarios. Each scenario will be a workload in the system execution model. Choose the "Teller balance check" scenario.

Notice that this scenario has only one processing-step node. Select Show values from the Software model menu. The values are the total for the execution graph. This is a convenient technique to use for very early life cycle feasibility studies. It is also convenient when you can measure resource requirements for other work that will execute on the computer facil-

ity but do not wish to examine the processing details of the software model for that other work.



Mbd summary	WrkUnit	12
	DB	2
	Msgs	2

The overhead specifications are already in the database; select the Display: Overhead command to see them (it is the same matrix shown in the previous Overhead Demo example). This is where the number of devices and their service times are specified.⁴

To specify the workload intensity, 1 transaction per second, select the Specify: Service level command. The dialog box shows that performance objective is 0.5 second for the entire Teller balance check, and the workload intensity specifies an arrival rate of 1 arrival per second (it is an open model).

Select Solve to see the software model results. Then select the Solution: Contention command to see the effect of 1 'Teller balance check' transaction per second (and no other work on the facility). Select View: Utilization. The disk has the highest utilization, but it is only 16% busy. Select View: Residence Time.

4. If you would like more information on the overhead matrix refer to the *SPE Basics* Appendix or Section 7.2.5 in the *SPE* book

Define Other Workloads

Next, examine the other model workload. Select Open scenario and open 'authorizeTransaction.' The processing steps in the figure are already specified for this scenario. This scenario is covered in section 6.3 in the SPE book. It was also covered in Manual Demo 2 in the previous section.

Note that there is one scenario in the project for each workload in the system execution model. You specify the scenarios by elaborating their processing steps, or by summarizing them with a single node. You should be familiar with *SPE•EDs* mechanisms for creating these scenarios. They are reviewed in the following paragraphs. Skip to *Create the System Execution Model* if you don't need the review.

For each scenario you “draw” the processing steps, Specify: Names, and Specify: Values for software resource requirements. You either use predefined overhead specifications or create them with Specify: Overhead. Overhead specifications define the devices in the computer facility, their quantity and service time for each device visit, and the computer device usage for each software resource request (e.g. path lengths).

You specify the workload intensity for each scenario with the Specify: Service level command. You also use this command to specify the performance objective for the color bar legends. Solve the software model (No contention then Contention solutions), examine the results and correct any performance problems before proceeding to the System Execution Model.

Create the System Execution Model

All the data you need to create the default system execution model is in the SPE database. Select the Display: System model command. *SPE•ED* uses the database to build a default model configuration. It displays the model and a dialog box reminding you to select Solve when the model is ready. This gives you an opportunity to make changes before you solve the model. Acknowledge the message, but do not select Solve yet.

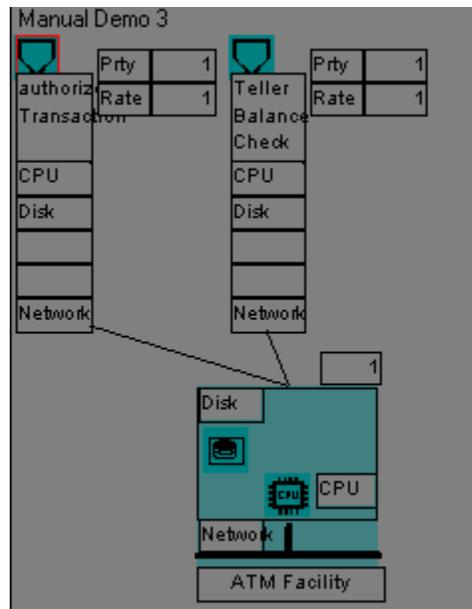
Each scenario in the database is shown across the top of the screen, each facility is across the bottom and lines show the facility you selected for each scenario. If you have a scenario open, it becomes the *focus scenario* and it is highlighted in the system model.

Open and closed scenarios are distinguished with icons and by the specification template beside the workload name (as shown in the following table. Below each scenario, a tem-

plate lists the names of the devices in the selected facility. This template will be used to display the system model results for each device.)

Workload Type	Icon	Template Contents
Open		Priority, Arrival rate
Closed		Priority, Number of users, Think time

The facility pictures across the bottom show the devices defined in the overhead matrix. The small box above the facility shows the number of identically configured facilities in the network; this example has one. Models with multiple facilities are explained in the *System Model* chapter.



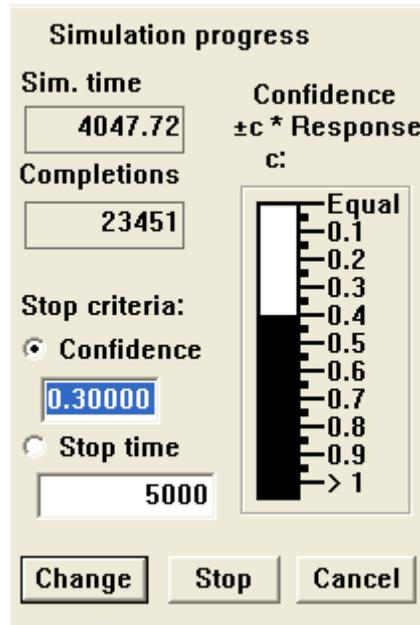
Global Parameters

Select the **Specify: Sysmod globals** command. This command lets you set priorities for scenarios and vary their workload intensity (arrival rate, or number of users and their think time) to compare system model results. Priorities are integer values; larger values are higher pri-

orities than small ones. The workload intensity values originated in the Specify: Service level command for each scenario. Changes made on this screen are stored with the system model in the SPE database, but *do not affect the original scenario specifications*. Make sure the values for priority and arrival rate for both scenarios are set to 1 then select OK. (Note that arrival rates do not have to be the same, but they are in this example).

Simulation Progress

Select the Solve command. The simulation progress dialog box displays the simulated time, number of completions and the current confidence interval. Suppose you run the simulation for a system model with one scenario and see a response time of 2.0 seconds and a confidence value of 0.3. This means that with 70% probability, the true mean is between 1.4 and 2.6 seconds. The *Simulation Progress* section in the *Solutions and Results* chapter discusses desirable confidence values.



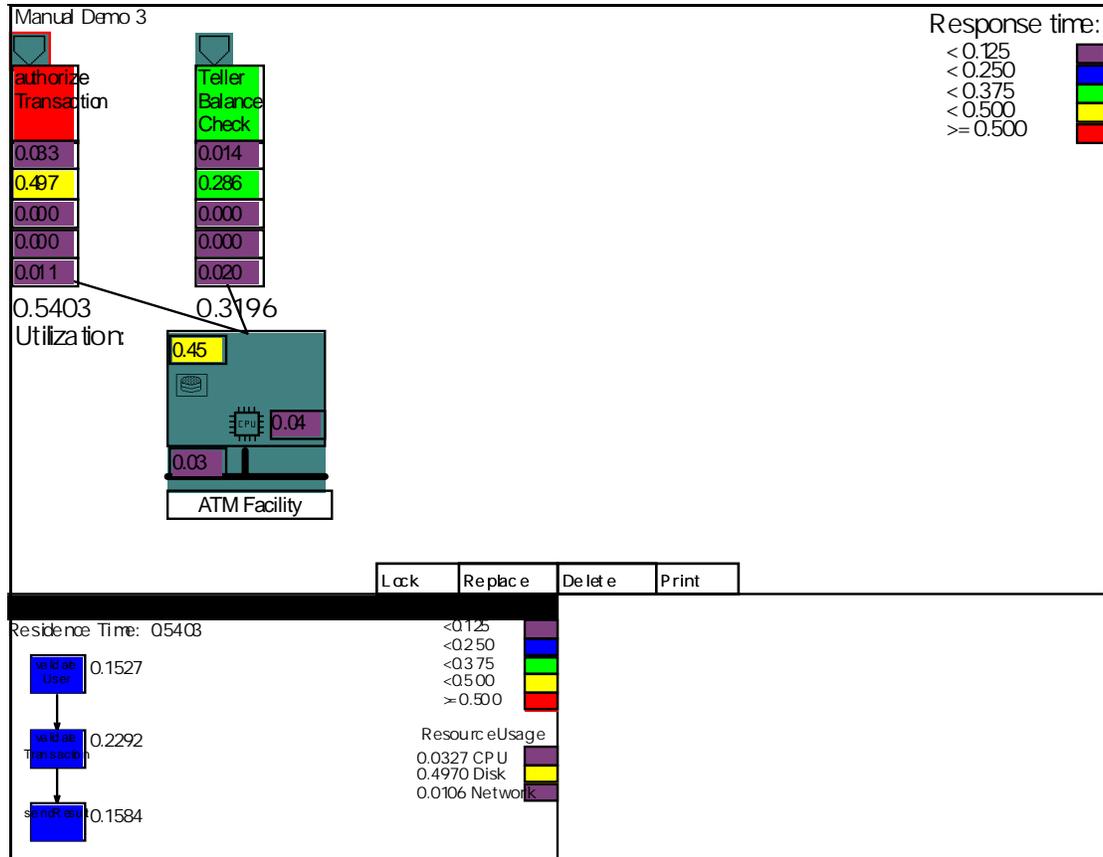
The image shows a 'Simulation progress' dialog box with the following fields and controls:

- Sim. time:** 4047.72
- Completions:** 23451
- Stop criteria:**
 - Confidence: 0.30000
 - Stop time: 5000
- Confidence $\pm c$ * Response c:** A vertical scale from 0.1 to > 1, with a black bar indicating the current confidence level at 0.3.
- Buttons:** Change, Stop, Cancel

The default stop criteria for the simulation is 0.3 confidence or 5000 seconds simulated time if the confidence has not yet been achieved. You have the option of stopping the simulation at any time with the Stop button, or overriding the stop time or the confidence interval. Click in the box you want to set, enter your value then click on the Change button.

Model Results

The results for the system model are displayed in the working-set screen at the top and the details for processing steps of the focus scenario are at the bottom. Your results should match the picture. They may vary slightly due to the simulation solution method. Make sure the View: Residence time was selected.



In the top section of the screen, the response time for each scenario appears below the scenario, and the corresponding color is inside the scenario name rectangle. The color bar legend is based on the focus scenario's performance objectives. The values and their corresponding color in each device template show how much of the total response time was spent at each device.

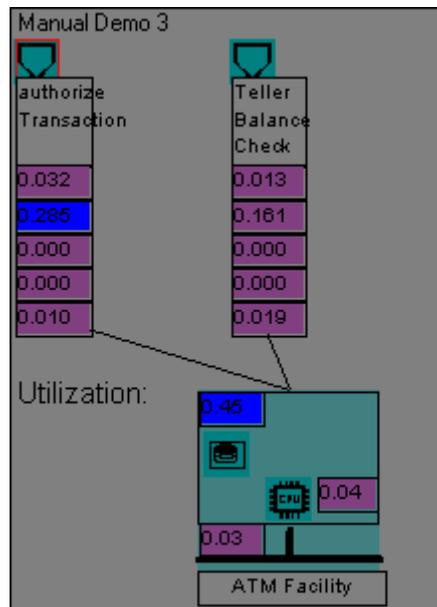
The values for the devices in the facility show the average utilization of each device. When there are multiple servers (the Quantity specification in the Overhead matrix is greater

than one), the utilization shown in the facility picture is the average for an individual server. The total time in the device template for the scenario, however, is the total time for all the servers.

The values for the results replace the names of the devices. You can see the names by clicking Replace at the bottom of the working-set screen until the system model specification screen appears. The top section of the system model screen shows only the latest result. To review earlier pictures, use the Replace working-set command at the bottom of the screen.

In the lower part of the screen, the response time results for the scenario processing steps have the same interpretation as they had on the Results screen. The difference is that the system execution model is solved with a simulation solver and produces results that quantify the overall contention of all scenarios. You can view combinations of results for various Navigation boxes as before.

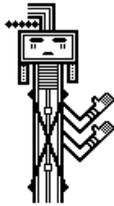
Select View: Utilization. The results in the top-screen system model show the portion of the utilization of each device attributed to each scenario. For example, the Disk device is 45% busy, 28.5% of the usage is the 'authorizeTransaction' scenario, 16.1% is 'Teller balance check'. The sum of the values for the disk device row across scenarios is the total shown for the disk in the facility picture (rounded). The utilization shown is the average for each server when a device quantity is greater than one.



Resource demand results for the system model in the top screen will be identical to the response time results. The scenario processing-step results below will show the demand for each step. They are not shown here.

You can experiment with this model. Try assigning a higher priority to authorizeTransaction, or increasing its arrival rate to 2 sessions per second and decreasing 'Teller Balance Check' to 0.5 sessions per second. Be sure to restore the original specifications to all ones for the next user.

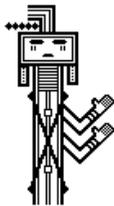
Select Display: Software model, change the number of DBs for the 'ValidateUser' step to 3, and then Solve the model again. Try other combinations to see how to interpret the model results and how to study alternatives. Don't save these changes.



SPE Database Backups

Be sure to back up the database files in the *SPE•ED* folder, spe-db0, spe-db1, spe-db2, spe-db3, regularly. They will contain evolutionary versions of your SPE models throughout each SPE project. To back up the files, copy the 4 spe-db files to another disk. (It is not necessary to save the spe-dbl file; it is a temporary file created by the database routines as needed.)

If you ever need to restore the SPE database from the backup, copy all four files to the active *SPE•ED* folder. This restores the entire database to the state of the last backup — changes since the backup are lost.



Summary

You have finished the *SPE•ED Quickstart*. In the first project you created a new scenario, Screen query, and examined its performance. You used the following steps to create and solve the first software execution model:

- Open a project
- Start a new scenario by selecting New, Open scenario, or by proceeding and naming it later
- Define the processing steps with nodes and arcs
- Specify names for the processing steps
- Select a software specification template to guide the specification of resource requirements
- Specify values for the software resources for each step
- Select a facility that represents the computer system execution environment
- Solve the software model (the default solution has No contention for computer system resources)
- Examine combinations of residence time, resource demand, and device utilization for the processing steps with working-set window controls.

We saw one technique for executing these steps for a simple model. Later chapters will describe additional features for defining and modifying software execution models, and cover the other types of nodes for defining processing steps.

The second example in the first project demonstrated how to add information to the models to reflect an evolving software design. The Expand command replaces a basic node with an expanded node and places the processing steps for the expansion – the submodel– in another Navigation box. You can view any of the submodels by clicking on the desired Navigation box.

To explore combinations of results for submodels, you first select the desired result to View then select the submodel's Navigation box. The working-set controls on the result screens let you Lock a screen, Delete it, or Replace it without deleting it.

The third example in the first project covered solutions that assess the effect of contention delays due to multiple users of the scenario. To quantify contention delays, Specify: Service levels and enter either the arrival rate or the number of users and their think time. Then select Solution: Contention and view the analytic results for residence time, resource demand (total time at each computer device including contention delay), and device utilization.

To compare the results to the performance objective, select Specify: Service level and enter the response time objective. The objective is used to set the color bar legend for results to highlight potential problem areas with high intensity colors.

The solutions depend on the overhead for each software resource request. Overhead values typically are specified by performance specialists, stored in the database, and reused for all models that execute in the environment.

You specify overhead for each software specification template and facility template combination. The software resource names on the left of the matrix appear as prompts when you Specify: Values for software processing steps. The device names, quantity, and service times define the characteristics of the system execution model. The values inside the matrix specify the service requirements from each device for each software resource request.

You saw two methods for specifying overhead. The first simplified the overhead matrix but required users to specify computer device requirements for software processing steps. The second method specified path lengths in the matrix to simplify the software value specification. Later sections will cover techniques for defining facilities and changing devices in facilities.

The last project demonstrated the relationship between projects, scenarios and the system execution model. Each scenario defines a workload in the system execution model. You define the scenarios by repeating the steps described in the first two projects, or by creating a scenario with one basic node that specifies the total resource requirements for the workload. The workload intensity for each scenario is specified in the service level dialog box.

SPE•ED creates the default system execution model. Each scenario in the project is a workload, each facility in the project appears, and the assignment of scenarios to facilities is shown. *SPE•ED* uses the overhead specifications and the software model solution for each scenario to derive the computer service requirements for each facility and device.

System model solutions show the combined device usage by all scenarios and each scenario's portion. Colors highlight potential problems. You can combine the system model results in the top portion of the screen with results for processing steps in the focus scenario. You can vary the global parameters for scenario priorities and workload intensity and compare the solution results. Later sections will describe how to create custom system execution models, modify scenario allocation to facilities, and other techniques for evaluating the effect of scenario contention.

