Statemate MAGNUM

I-Logix

3 Riverside Drive
Andover, MA 01810
U.S.A.

Tutorial

Copyright Notice and Proprietary Information

Copyright © 2001 by I-Logix Inc. — Printed in the United States of America.
3 Riverside Drive, Andover, MA 01810 U.S.A.
All rights reserved.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license. Statemate software contains proprietary information, as well as trade secrets of I-Logix Inc., and is protected under international copyright law. Reproduction, adaptation, or translation, in whole or in part, by any means — graphic, electronic or mechanical, including photocopying, recording, taping, or storage in an information retrieval system — of any part of this work covered by copyright is prohibited without prior written permission of the copyright owner, except as allowed under the copyright laws.

This product or products depicted herein may be protected by one or more U.S. or international patents or pending patents.

The information in this manual is subject to change without notice. I-Logix assumes no responsibility or liability for any errors contained herein or direct, indirect, special, incidental or consequential damages in conjunction with the furnishing, performance, or use of this material.

Restricted Rights Legend：. Use, duplication, or disclosure by the government is subject to restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 (October 1988) and FAR 52.227-19 (June 1987).

Trademarks

Rhapsody, Rhapsody in MicroC and Statemate MAGNUM are trademarks of I-Logix Inc. I-Logix and the I-Logix logo are trademarks of I-Logix Inc.

Microsoft, MS, and MS-DOS are registered trademarks, and Windows and Windows NT are trademarks of Microsoft Corporation.

X Window System is a trademark and product of the Massachusetts Institute of Technology.

Sun, SunOS, Solaris, OpenWindows, and NFS are trademarks or registered trademarks of Sun Microsystems, Inc.

Other products mentioned may be trademarks or registered trademarks of their respective companies.

Part No. 2251
Printed in the United States of America

Contents

Lesson 1. Written Requirements of a Clock 3

Lesson 2. Getting Started 4

2. 3. Creating the Project 5
2. 4. Opening the Project 6

Lesson 3. Creating the Activity Chart 7

3. 2. Creating an Activity Chart 8
3. 3. Drawing Activities 10
3. 4. Drawing Control Activities 12
3. 5. Drawing External Activities 14
3. 6. Drawing Flows 16

Lesson 4. Creating Statecharts 18

4. 3. Drawing States 19
4. 4. Drawing Transitions 21
4. 5. Associating the Control Activity with the Statechart 25
4. 7. Drawing States 27
4. 8. Drawing Transitions and Connectors 29

Lesson 5. Defining the Textual Elements 34

Lesson 6. Simulating the Model 40

6. 3. Setting Up a Monitor Window 41
6. 4. Stepping Through the Simulation 43

Lesson 7. Panel Creation and Simulation 47

7. 3. Drawing the Panel Elements 48

Lesson 8. Code Generation and Execution 57

Lesson 9. Creating Mini-Specs 60

Statemate MAGNUM Tutorial

This tutorial will introduce you to the basic tools of Statemate by guiding you through a series of exercises. The focus of these exercises is on the mechanics of the Statemate tool and is not intended to teach the design methodology of Statemate.

At the completion of this tutorial, you will have:

- Used the graphical editors
- Executed a simulation
- Generated C code

These exercises are meant to be used in conjunction with the ***Statemate Workshop***. Each chapter builds on the next, so it is advisable to do the exercises in the sequence presented.

Using Mouse Buttons

Statemate MAGNUM supports both the 3-button and the 2-button mouse (also referred to as a *PC mouse*). The default setting for Windows NT is a PC mouse, and the default setting for Solaris is a 3-button mouse.

Three-Button Mouse

In this tutorial, instructions are provided based on a 3-button mouse.

On the 3-button mouse, the

- **Left button** is called the **Select** button. Click this button once to select elements, icons and menu items and to place the starting point for transitions.

  When a procedure instructs you to click or select an item, do so using the left mouse button.
- **Middle button** is called the **Transfer** button. Click and hold this button to move elements by dragging. Click once to place an end point on a transition.
- **Right button** is called the **Menu** button. Hold down this button to display a pop-up menu. Click once to deselect a selected item.

PC Mouse (2-Button)

If you have a PC mouse with two buttons, set your Graphic Editor preferences to include PC Mouse (2-Button).

To change the mouse preference from 3-button to 2-button:

1. Open the **Options** menu on the Statemate Main window and select **Preferences Management** > **Graphic Editors**.
2. On the Graphic Editors Preferences window, scroll down to the **PC Mouse (2-Button)** parameter.
3. Set the value to **Yes**.

On the PC mouse, the

- **Left button** is called the **Select** button. When a procedure instructs you to click or select an item, do so using the left mouse button.

  This button functions as follows:
  - Click once to select elements, icons and menu items.
  - Click once to place the starting and ending points for transitions.
  - Click once on a boundary to select a complex box (activity that contains sub-activities).
  - Click and hold while moving the mouse to drag selected items.
- **Right button** is called the **Menu** button. Hold down this button to display a pop-up menu. Click once on an item to select it and display a pop-up menu for the item.

LESSON 1. WRITTEN REQUIREMENTS OF A CLOCK

1. 1. Clock Example Overview

The following example is of a simple clock. It has been constructed with functionality that will probably be familiar, so that you can concentrate on the mechanics of the Statemate tools, rather than the design problem itself.

1. 2. Clock Requirements

The main purpose of the clock shall be to display the current time of day, but it shall also have a mode that will allow the user to set the time of day. The user will be able to select which mode the clock will be operating in via a selection switch.

The clock will be in operation mode whenever power is present. If power is removed, the clock shall have a blank display until power returns. When power returns, the clock shall enter the set time mode.

In the set time mode, the clock will display the set time in hours and in minutes. The user will be able to increment the minutes by pressing the set minute button and will be able to increment the hours by pressing the set hours button.

In the current time mode, the clock will display the current time of day in hours and minutes. The minute value will increment every 60 seconds and the hour value will increment every 60 minutes.

LESSON 2. GETTING STARTED

2. 1. Introduction

In this chapter, you will

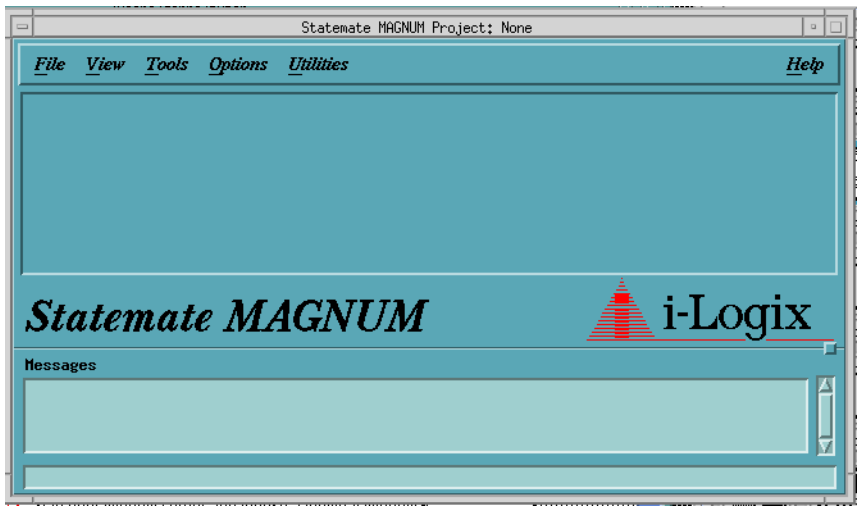
- Start Statemate
- Create a project
- Open the project

2. 2. Starting Statemate

1. Log on to your workstation.
2. Start your window server and invoke a terminal window.

   If you don't know how to log in or start the window system, ask your instructor or system administrator.
3. Type ***<path to Statemate software>*/run_stmm &** from the prompt in the terminal window.

   Result: The **Statemate Main** window appears.



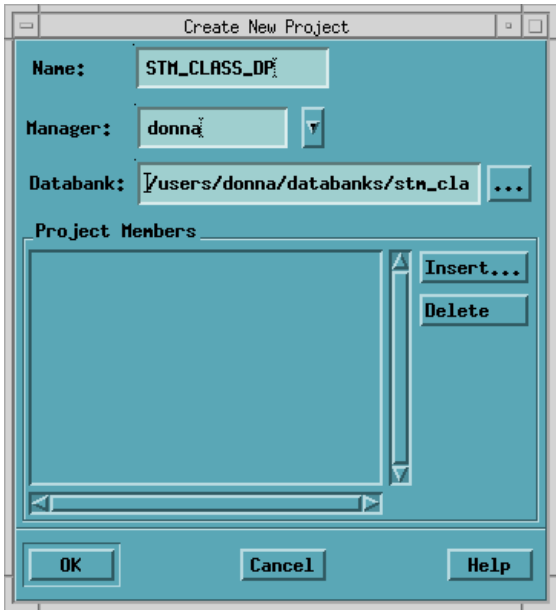
2. 3. Creating the Project

Typically, a Statemate project leader creates a project. Check your instructor to see if this is the case.

- If a project has been created for you, skip to the section titled ***Opening the Project***.
- If a project has not been created for you, follow the instructions below to create your own project.

1. Select the ***File*** option from the **Statemate Main** window with the left mouse button.
2. Select the ***New Project*** option.
3. Fill in the **Create New Project** form as follows:
   - **Name: STM_CLASS_*your-initials***
   - **Manager: *your login***
   - **Databank: *path to the databank directory***.

   Ask your instructor if you have any questions.



4. Select **OK**.

   Result: The **Open Project** form appears with the project you just created selected.

2. 4. Opening the Project

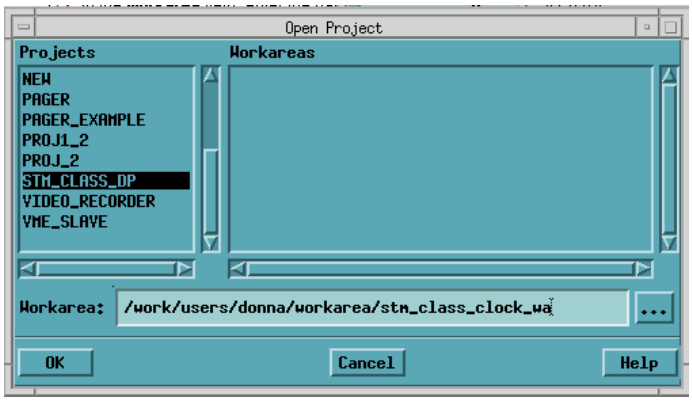
1. If the **Open Project** window is not open, select the ***File*** menu from the **Statemate Main** window, followed by the ***Open Project*** option

   Result: The **Open Project** form appears.

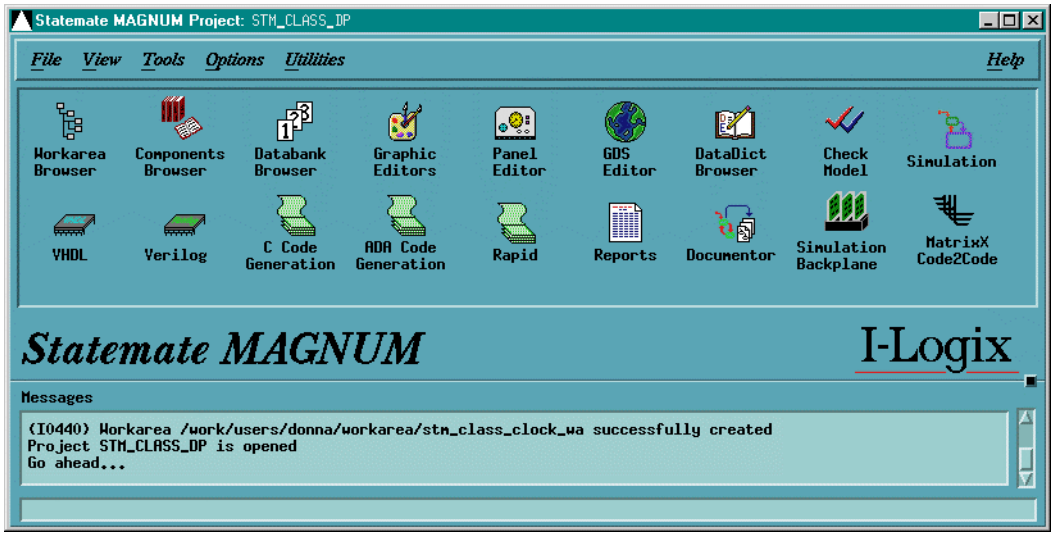
2. In the **Projects** listing on the left, select the project for this class.

3. In the **Workarea** field, enter the path to your workarea directory. Consult your instructor or system administrator for the appropriate workarea directory pathname.

4. Select **Yes** on the confirmation message dialog and **OK** on the **Open Project** form.



The project is now open and various icons appear in the **Statemate Main** window.



LESSON 3. CREATING THE ACTIVITY CHART

3. 1. Introduction

In this chapter, you create an Activity chart.

In class, you learned that activity charts represent the functional view of a system. You also learned that they consisted of the following elements:

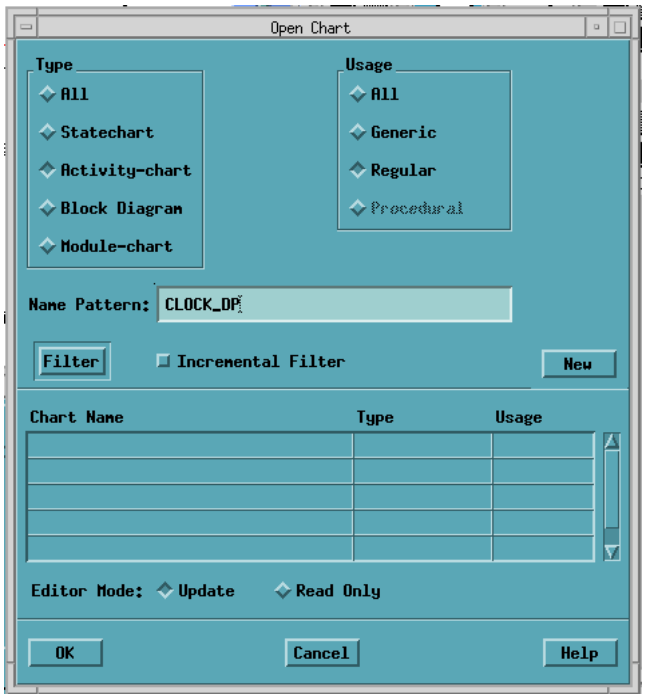
- **Activities**, which represent the functions in the system.
- **Control activities**, which represent the link to Statecharts. These Statecharts can control other activities or represent the activities behavior.
- **External activities**, which represent elements outside of the system (or current view) that the activities interface with.
- **Data flows**, which represent the exchange of data between activities.

While creating this activity chart, you will functionally decompose the clock into activities and you will show the external environment of the clock using external activities. You will also use data flows to represent the exchange of information between activities.

3. 2. Creating an Activity Chart

In this section, you will create an activity chart.

1. Click the **Graphic Editors** icon on the Statemate Main window.
2. Fill in the **Open Chart** form as follows:
   - **Type: Activity-chart**
   - **Usage: Regular**
   - **Name Pattern: CLOCK_<*your_initials*>**
3. Click **New**.



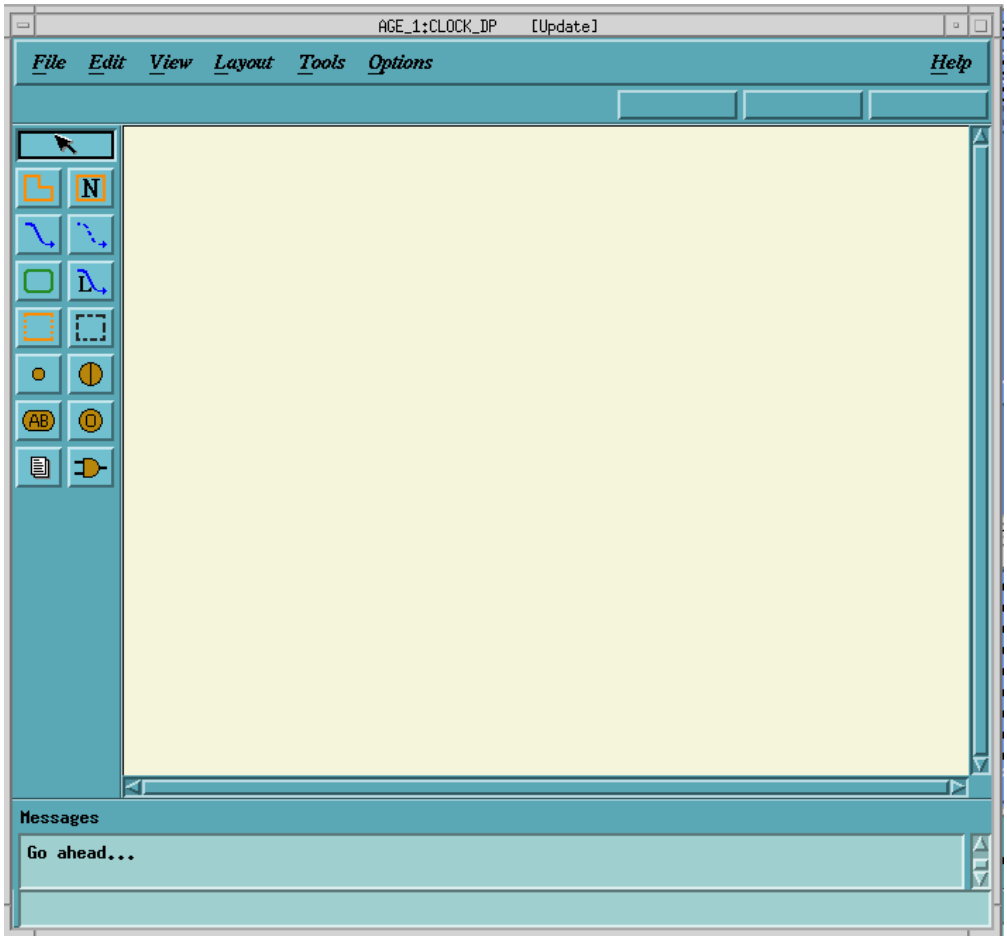
Result: An **Activity-chart Graphics Editor** (AGE) appears.

Notice that the top banner of the AGE provides some useful information:

- A process name
- The name of the chart being edited (separated by a colon (:) from the process name)
- The mode of the chart (either Update or Read Only)

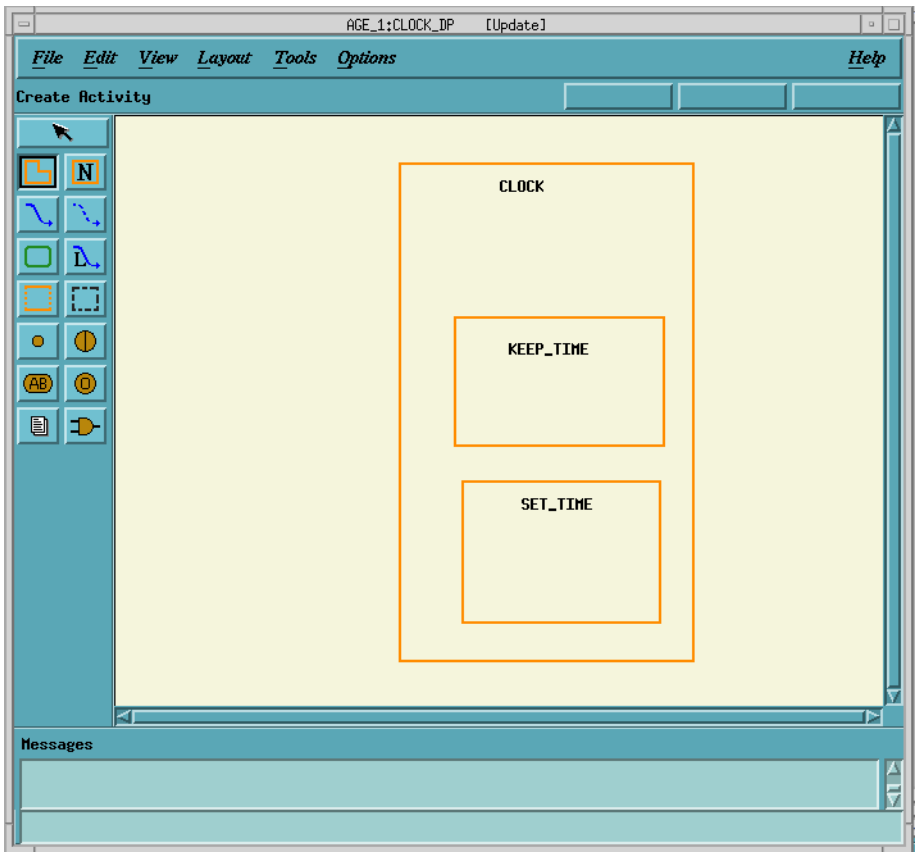
In this example, we are editing the chart named **CLOCK_<*your-initials*>** in update mode, and the process name is **AGE_1** (this

indicates the first AGE invoked from the current Statemate session).



3. 3. Drawing Activities

In this section, you will draw the activities for the clock system.



The first step is to draw the activity called **CLOCK**.

1. Select the **Create Internal Activity** icon from the graphic editor. Notice that the currently active command is indicated above the drawing icons.
2. Place the cursor at the desired location for the upper, left-hand corner of the activity **CLOCK** and hold the left mouse button down while dragging the mouse to the desired location of the lower, right-hand corner of the activity. A ghost image shows the activity outline. Release the mouse button and the activity is drawn.

The second step is to name the **CLOCK** activity.

1. While the **Create Activity** mode is still active, move your cursor inside the graphics drawing area.
2. Type the activity name, in this case, **clock**.

Result: The letters you are typing appear in the graphics drawing area.

3. Move the activity name and cursor to the appropriate location inside the activity's boundaries.
4. Click the left mouse button.

   Result: The name of the activity is changed to uppercase letters and is now associated with the activity that it resides in.

Note: Type the names and labels for boxes and arrows using lowercase letters. Conversion to uppercase letters, where appropriate, will be automatically performed by Statemate. Do *not* select the CAPS LOCK key before typing. Doing so will prevent you from making further selections.

The third step is to draw and name the remaining activities, **KEEP_TIME** and **SET_TIME**.

1. While the **Create Activity** mode is still active, move the cursor to the location of the upper, left-hand corner of the **KEEP_TIME** activity.
2. Press the left mouse button and hold it down while dragging the mouse to the desired location of the lower, right-hand corner of the activity. A ghost image shows the activity outline. Release the mouse button and the activity is drawn.
3. Type the activity name, in this case, **KEEP_TIME**.
4. Move the activity name and cursor inside the activity you just drew.
5. Click the left mouse button.

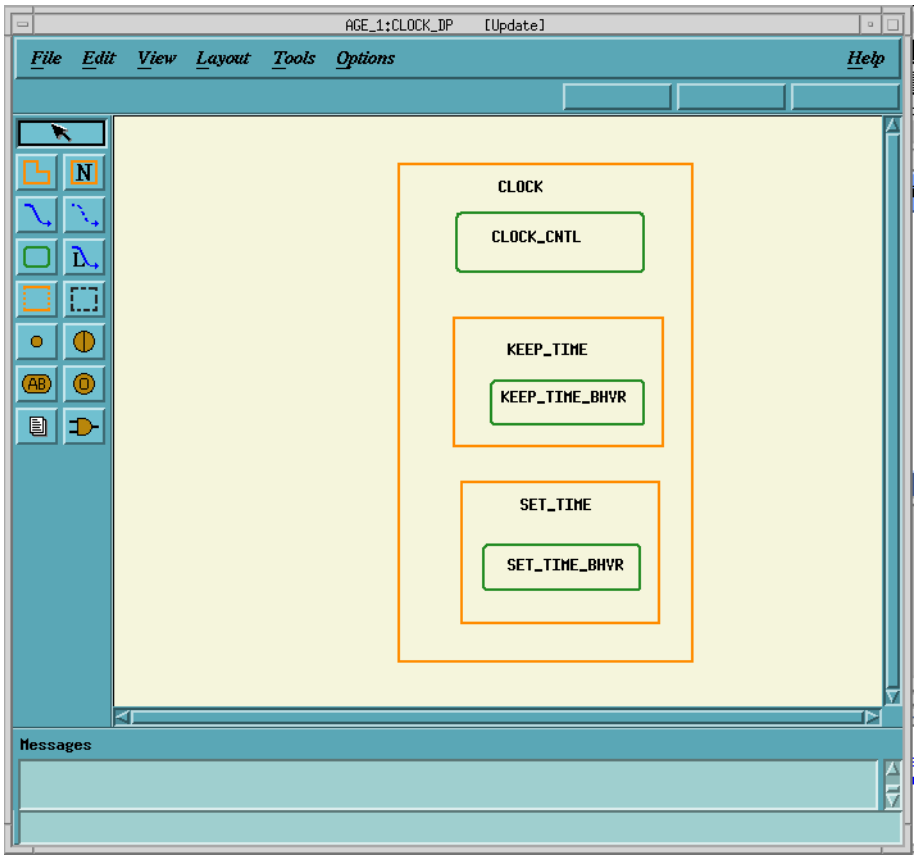
   Result: The name of the activity is now associated with the activity that it resides in.

6. Repeat step 1 through 5 above to draw and name the **SET_TIME** activity.

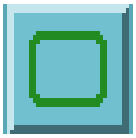
You have now entered the activities.

3. 4. Drawing Control Activities

In this section, you will draw and name control activities.



The first step is to draw the control activity called **CLOCK_CNTL**.



1. Select the **Create Control Activity** icon from the graphic editor. Notice that the currently active command is indicated above the drawing icons.
2. Place the cursor at the desired location for the upper, left-hand corner of the control activity **CLOCK_CNTL** and hold the left mouse button down while dragging the mouse to the desired location of the lower, right-hand corner of the control activity. A ghost image shows you the control activity outline. Release the mouse button and the control activity is drawn.

The second step is to name the **CLOCK_CNTL** control activity.

1. While the **Create Control Activity** mode is still active, move your cursor inside the graphics drawing area.

2. Type the control activity name, in this case **CLOCK_CNTL**. The letters you are typing appear in the graphics drawing area.
3. Now move the control activity name and cursor to the appropriate location inside the control activity's boundaries.
4. Click the left mouse button.

   Result: This associates the name with the control activity that it resides in.

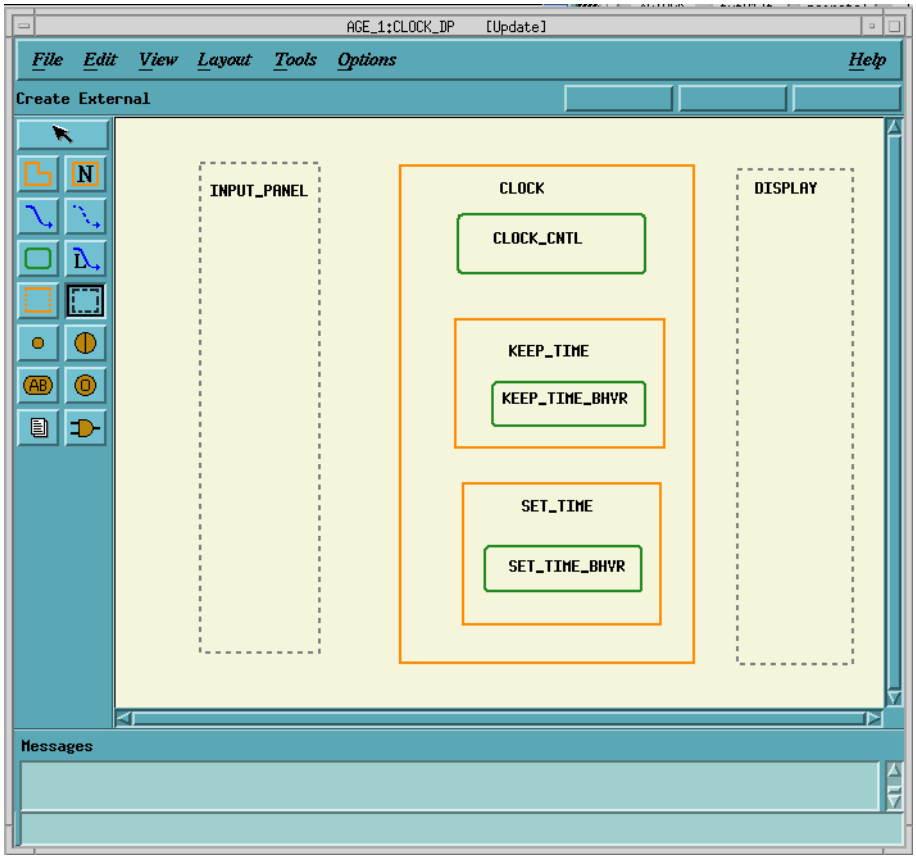
The third step is to draw and name the control activities for the **KEEP_TIME** and **SET_TIME** activities.

1. Draw the **KEEP_TIME_BHVR** control activity and name it as described above and shown in the figure.
2. Draw the **SET_TIME_BHVR** control activity and name it as described above and shown in the figure.

You have now entered the control activities.

3. 5. Drawing External Activities

In this section, you will draw and name external activities.



The first step is to draw the external activity called **INPUT_PANEL**.

1. Select the **Create External Activity** icon from the graphic editor. Notice that the currently active command is indicated above the drawing icons.
2. Place the cursor at the desired location for the upper, left-hand corner of the external activity **INPUT_PANEL** and hold the left mouse button down while dragging the mouse to the desired location of the lower, right-hand corner of the external activity. A ghost image shows the external activity outline. Release the mouse button and the external activity is drawn.

The second step is to name the **INPUT_PANEL** external activity.

1. While the **Create External Activity** mode is still active, move your cursor inside the graphics drawing area.
2. Type the external activity name, in this case, **INPUT_PANEL**. The letters you are typing appear in the graphics drawing area.
3. Now move the external activity name and cursor to the appropriate location inside the external activity's boundaries.
4. Click the left mouse button to associate the name with the external activity that it resides in.

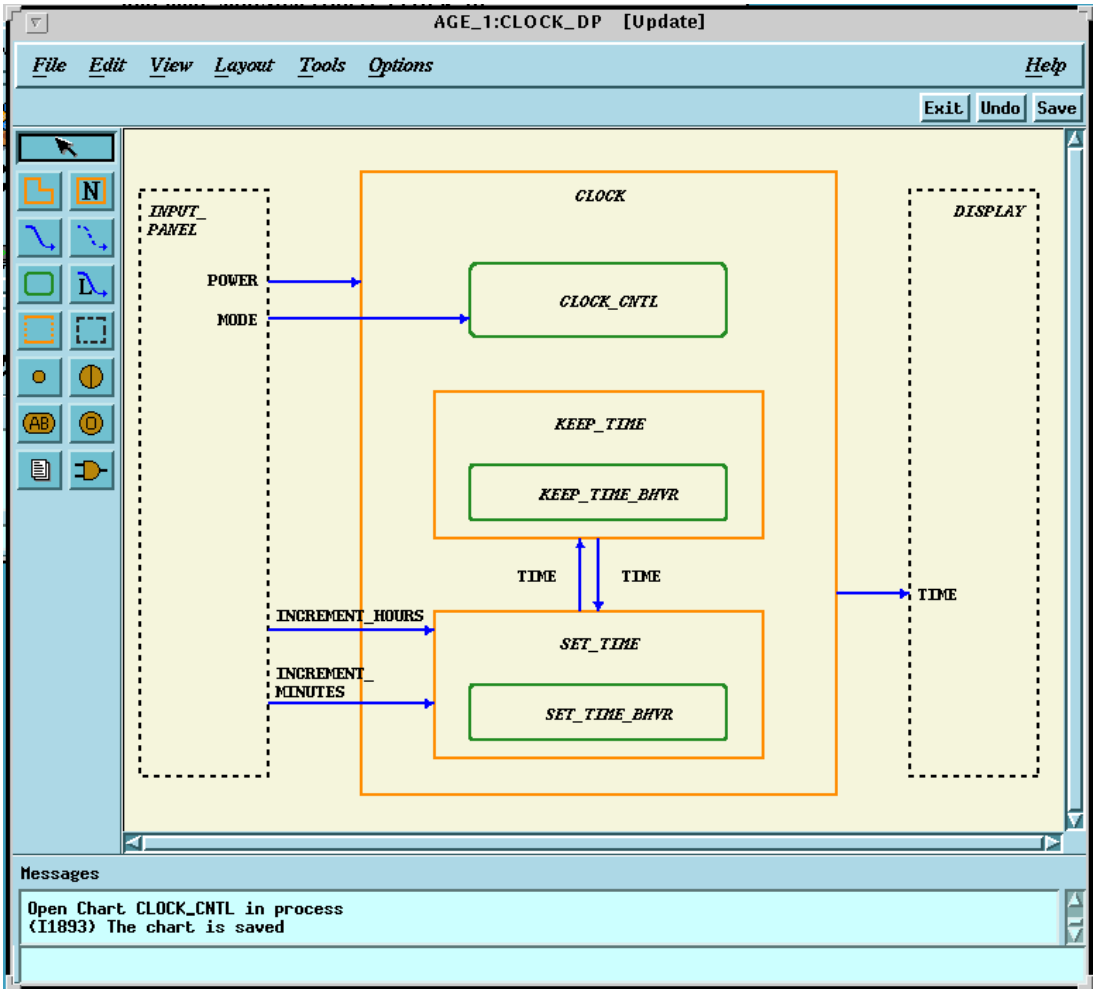
The third step is to draw and name the external activity called **DISPLAY**.

1. Draw the external activity, **DISPLAY**, as described above and shown in the figure.
2. Name the external activity, **DISPLAY**, as described above and shown in the figure.

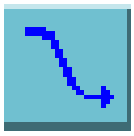
You have now entered the external activities.

3. 6. Drawing Flows

In this section, you will draw data flows.



The first step is to draw the **MODE** flow.



1. Select the **Create Data Flow** icon from the graphic editor. Notice that the currently active command is indicated above the drawing icons.
2. Locate the **INPUT_PANEL** external activity. This is the source activity.
3. Place the cursor on the edge of the box and click the left mouse button to enter the tail of the arrow.
4. Locate the **CLOCK_CNTL** control activity. This is the target activity.
5. Place the cursor on the edge of the box and click the middle mouse button to enter the arrow head.

The second step is to label the **MODE** data flow.

1. While the **Create Data Flow** mode is still active, move your cursor inside the graphics drawing area.
2. Type the data flow name, in this case, **MODE**. The letters you are typing appear in the graphics drawing area.
3. Now move the data flow name and cursor to the appropriate location with the '+' just touching the data flow.
4. Click the left mouse button to associate the name with the data flow that the '+' is touching.

The third step is to use the following table and the procedure outlined above to label all the data flows in the Activity-chart.

Note: To create multi-line labels, enter a backslash (\), followed by the RETURN key to indicate continuation of the label onto the next line.

SOURCE	TARGET	LABEL
INPUT_PANEL	CLOCK	POWER
INPUT_PANEL	SET_TIME	INCREMENT_HOURS
INPUT_PANEL	SET_TIME	INCREMENT_MINUTES
SET_TIME	KEEP_TIME	TIME
KEEP_TIME	SET_TIME	TIME
CLOCK	DISPLAY	TIME

LESSON 4. CREATING STATECHARTS

4. 1. Introduction

In this chapter, you create three Statecharts. The first represents the control for the clock. The other two represent the behavior of the **SET_TIME** and **KEEP_TIME** activities.

4. 2. Creating a Statechart Using the Open Chart Form

In this section, you will create a Statechart.

1. Select the **Graphic Editor**s icon from the **Statemate Main** window.
2. Fill in the **Open Chart** form as follows:
   - **Type:Statechart**
   - **Usage:Regular**
   - **Name Pattern:CLOCK_CNTL**
3. Select the **New** option.

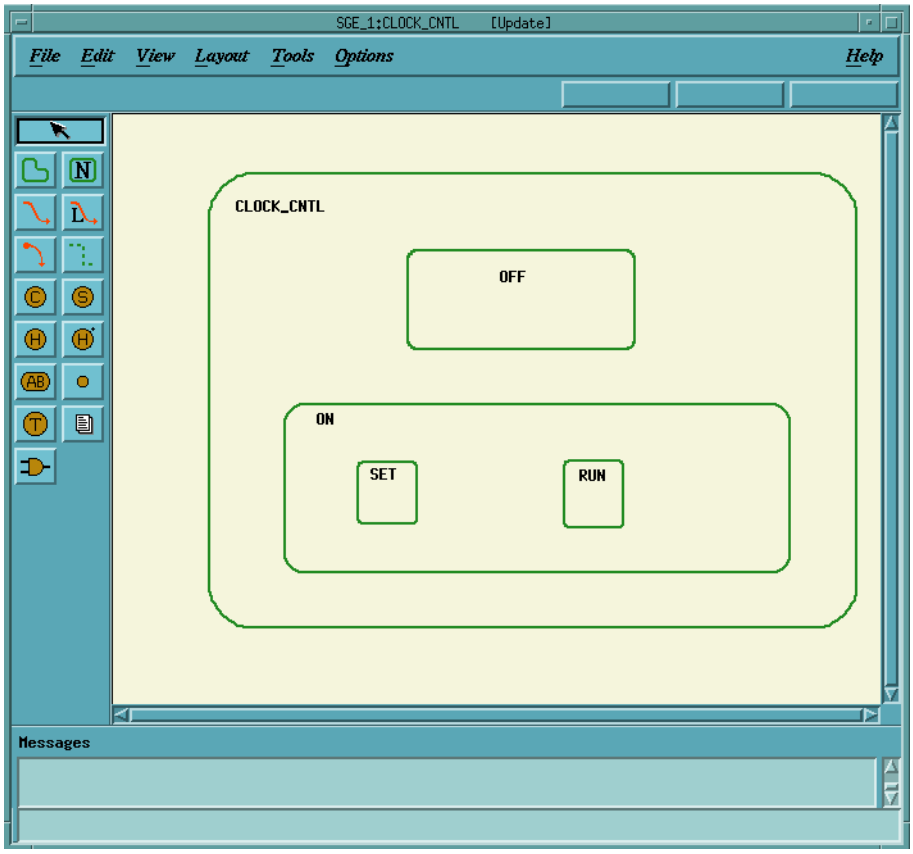
A **Statechart Graphics Editor** appears. Notice that the top banner of the SGE displays

- The process name
- The name of the chart being edited (separated by a colon (:) from the process name)
- The mode of the chart (either Update or Read Only)

In this example, we are editing the chart named **CLOCK_CNTL** in update mode, and the process name is SGE_1 (this indicates the first SGE invoked from the current Statemate session).

4. 3. Drawing States

In this section, you will draw the states for **CLOCK_CNTL**.



The first step is to draw the state called **CLOCK_CNTL**.

1. Select the **Create State** icon from the graphic editor. Notice that the currently active command is indicated above the drawing icons.
2. Place the cursor at the desired location for the upper, left-hand corner of the state **CLOCK_CNTL** and hold the left mouse button down while dragging the mouse to the desired location of the lower, right-hand corner of the state. A ghost image shows you the state outline. Release the mouse button and the state is drawn.

The second step is to name the **CLOCK_CNTL** state.

1. While the **Create State** mode is still active, move your cursor inside the graphics drawing area.

2. Type the state name, in this case, **CLOCK_CNTL**. The letters you are typing appear in the graphics drawing area.
3. Now move the name and cursor to the appropriate location inside the state's boundaries.
4. Click the left mouse button to associate the name with the state that it resides in.

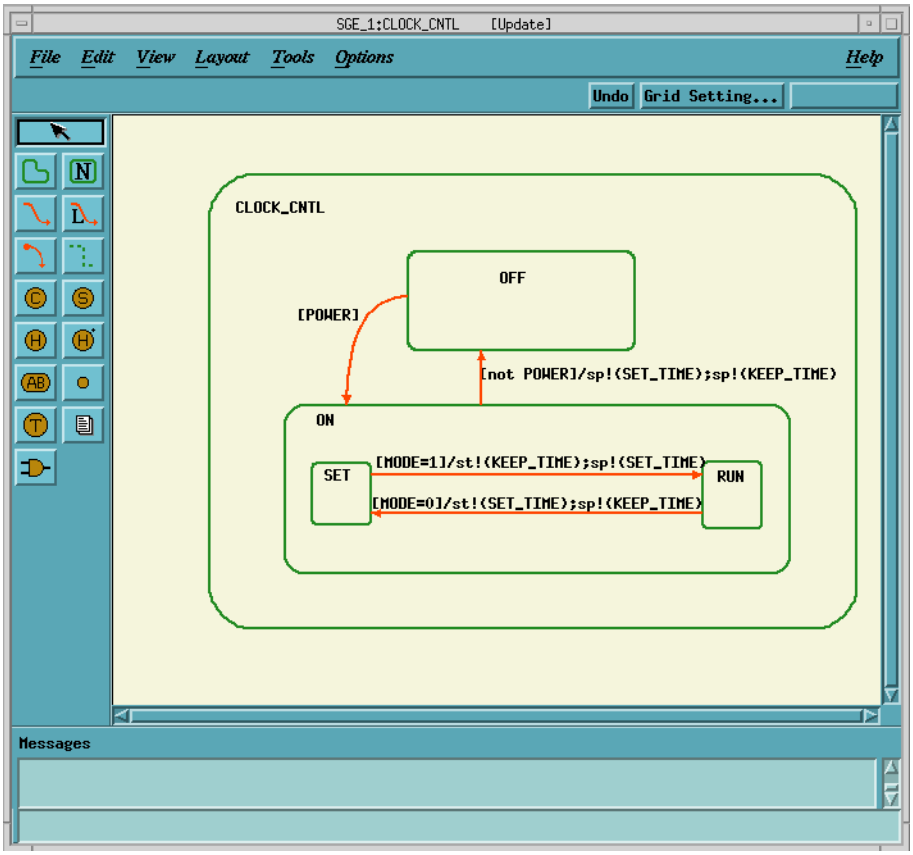
The third step is to draw and name the remaining states: **OFF**, **ON**, **SET** and **RUN**.

1. While the **Create State** mode is still active, move the cursor to the location of the upper, left-hand corner of the **OFF** state.
2. Press the left mouse button and hold it down while dragging the mouse to the desired location of the lower, right-hand corner of the state. A ghost image shows the state outline. Release the mouse button and the state is drawn.
3. Type in the state name, in this case, **OFF**.
4. Move the state name and cursor inside the state you just drew.
5. Click the left mouse button to associate the name with the state that it resides in.
6. Repeat steps 1 through 5 above to draw and name the states **ON**, **SET** and **RUN**.

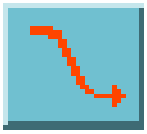
You have now entered the states for the **CLOCK_CNTL** Statechart.

4. 4. Drawing Transitions

In this section, you will draw the transitions between states.



The first step is to draw the transition from the **OFF** state to the **ON** state.



1. Select the **Create Transition** icon from the graphic editor. Notice that the currently active command is indicated above the drawing icons.
2. Locate the **OFF** state. This is the source state.
3. Place the cursor on the edge of the **OFF** state and click the left mouse button to enter the tail of the arrow.
4. Locate the **ON** state. This is the target state.
5. Place the cursor on the edge of the box and click the middle mouse button to enter the arrow head.

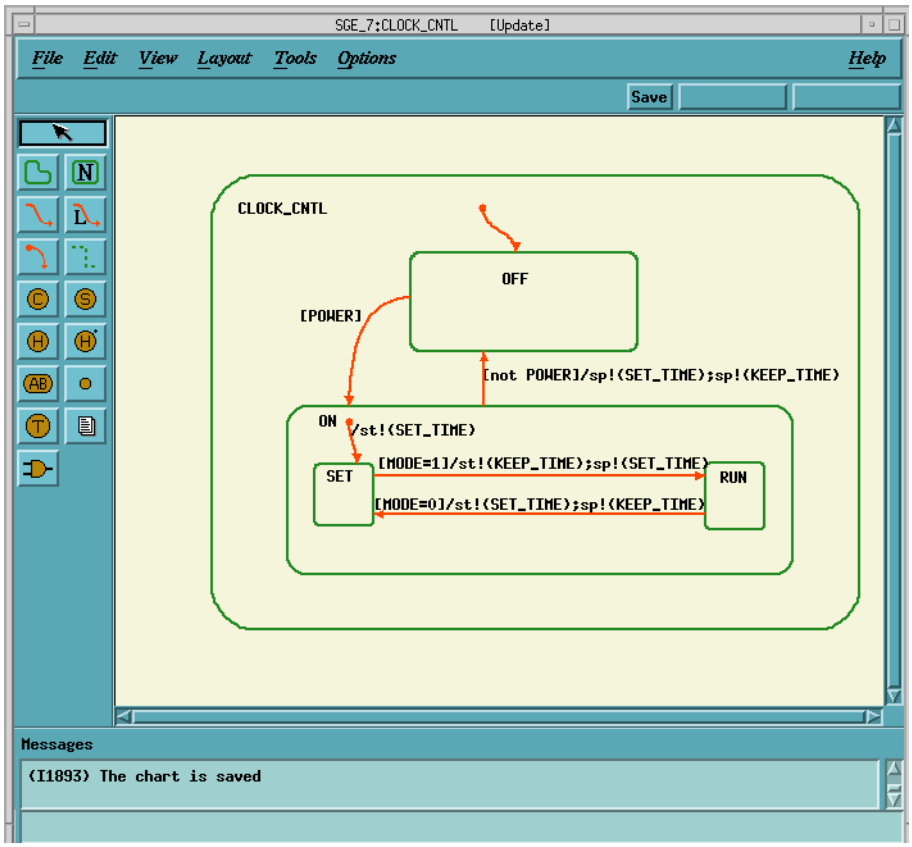
The second step is to label the transition from the **OFF** state to the **ON** state.

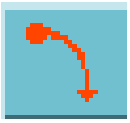
1. While the **Create Transition** mode is still active, move your cursor inside the graphics drawing area.
2. Type the label, in this case, **[POWER]**. The letters you are typing appear in the graphics drawing area.
3. Now move the label and cursor to the appropriate location with the '+' just touching the transition.
4. Click the left mouse button to associate the name with the transition that the '+' is touching.

The third step is to use table below and the procedure outlined above to label all the transitions in the Statechart.

SOURCE	TARGET	LABEL
ON	OFF	[not POWER]/sp!(KEEP_TIME);sp!(SET_TIME)
SET	RUN	[MODE=1]/st!(KEEP_TIME);sp!(SET_TIME)
RUN	SET	[MODE=0]/st!(SET_TIME);sp!(KEEP_TIME)

The fourth step is to enter the default transitions, one into the **OFF** state and the other into the **SET** state.





1. Select the **Create Default Transition** icon from the graphic editor. Notice that the currently active command is indicated above the drawing icons.
2. Locate the **OFF** state. This is the target state.
3. Locate a place just outside of the **OFF** state. This is where the source of the default transition be. Click the left mouse button to enter the arrow tail.
4. Place the cursor on the edge of the **OFF** state and click the middle mouse button to enter the arrow head.
5. Locate the **SET** state. This is the target state.
6. Locate a place just outside of the **SET** state to be the source of the default transition. Click the left mouse button to enter the arrow tail.
7. Place the cursor on the edge of the **SET** state and click the middle mouse button to enter the arrow head.

The fifth step is to label the default transition to the **SET** state.

1. While the **Create Default Transition** mode is still active, move your cursor inside the graphics drawing area.
2. Type the label, in this case, **/st!(SET_TIME)**. The letters appear in the graphics drawing area.
3. Now move the label and cursor to the appropriate location with the '+' just touching the default transition.
4. Click the left mouse button to associate the label with the default transition that the '+' is touching.

The **CLOCK_CNTL** Statechart is now complete.

4. 5. Associating the Control Activity with the Statechart

Next, make a link between the control activity and the Statechart that represents it. Do this by placing an "@" symbol in front of the control activity name in the Activity-chart and making sure the name of the control activity matches the name of the Statechart.

The first step is to place the "@" symbol in front of the control activity name to create the link.

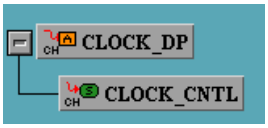
1. Open the **Workarea Browser** by returning to the **Statemate Main** window and selecting the **Workarea Browser** Icon.
2. In the **Workarea Browser**, locate the **CLOCK_*your-initials*** Activity-chart. Double click on this chart's icon with the left mouse button to open the chart.
3. Locate the **CLOCK_CNTL** control activity.
4. Double click on the **CLOCK_CNTL** control activity name. causes the name to be highlighted in black with a cursor appearing.
5. Click the left mouse button at the beginning of the name and then move the cursor to the left of the name.
6. Type the @ symbol.

   Result: The name now appears as **@CLOCK_CNTL**.
7. Click the middle mouse button away from the selected text field to terminate edit text mode.

The last step is to verify that the link was correctly made between the control activity and the Statechart.

1. In the **Workarea Browser**, select the ***View*** pull-down menu, followed by the ***Refresh*** option.

   Result: This updates the **Workarea Browser**.



2. Verify that a line appears in the **Workarea Browser** showing that the **CLOCK_CNTL** Statechart is connected to the **CLOCK_*your-initials*** Activity-chart.

If this is not the case, verify that the name of the Statechart matches the name of the control-activity.

You have now completed the link between the control activity and the Statechart.

4. 6. Creating a Statechart Using Create Sub-Chart

In this section, you create the **KEEP_TIME_BHVR** Statechart and its states using the **Create Sub-Chart** command.

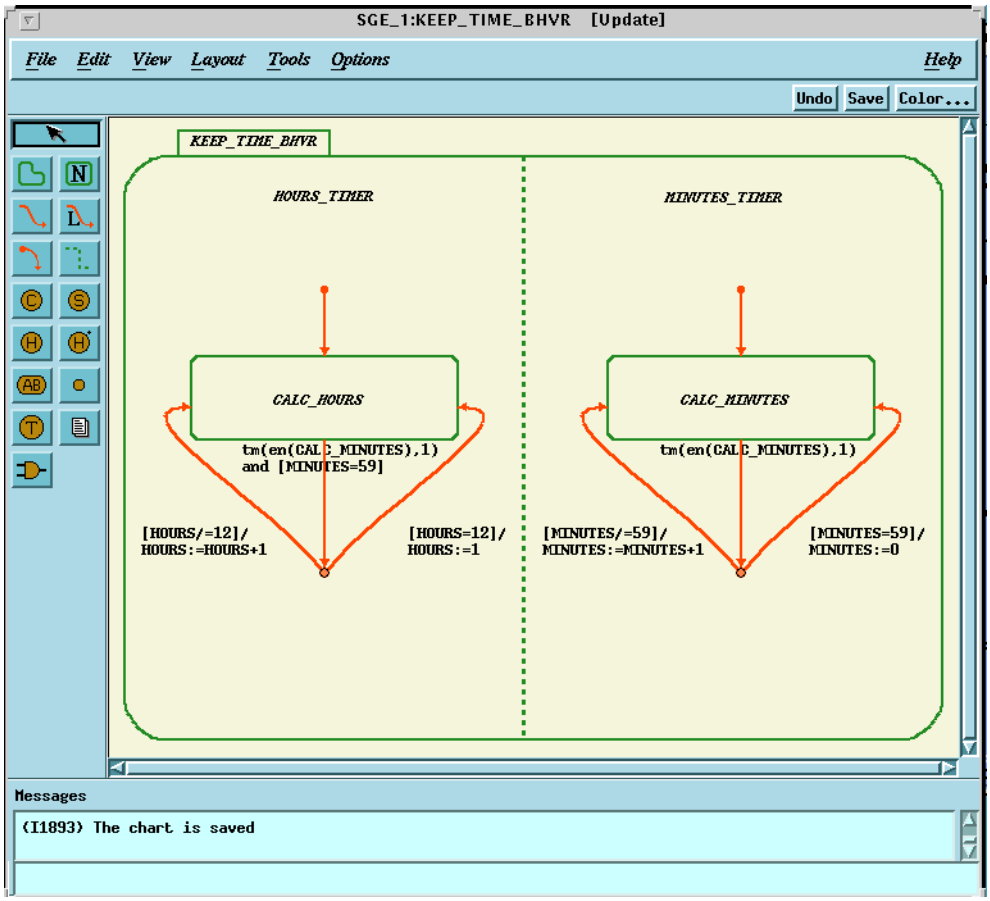
The first step is to create the **KEEP_TIME_BHVR** Statechart.

1. If the Activity-chart **CLOCK_*your-initials*** is not open, open it using the **Workarea Browser** (as described in section 4.5).
2. In the Activity-chart, select the control activity named **KEEP_TIME_BHVR**.
3. From the ***File*** menu, select ***Create Sub-Chart***.

This creates a Statechart called **KEEP_TIME_BHVR** and places the "**@**" symbol in front of the control activity named **KEEP_TIME_BHVR**.

4. 7. Drawing States

In this section, you will draw the states for the **KEEP_TIME_BHVR** Statechart.



The first step is to reformat the state called **KEEP_TIME_BHVR**.

1. Select the **KEEP_TIME_BHVR** state.
2. Select one of the 8 handles with your middle mouse button, and while keeping the button pressed, reshape the state. Continue with this process until the state fills the drawing area within the graphics editor.
3. Select the **KEEP_TIME_BHVR** state name with the middle mouse button, and while keeping the button pressed, move the state name to the upper left-hand corner of the state.

The second step is to create and name the and-states, **HOURS** and **MINUTES**.

1. Select the **Create And-Line** icon.

2. Move your cursor to a point half way across the top horizontal border of the state named **KEEP_TIME_BHVR**.
3. Draw a line to the bottom border of the state.

4. Select the **Name Existing State** icon.
5. Move your cursor inside the graphics drawing area.
6. Type the state name, in this case, **HOURS_TIMER**. The letters you are typing appear in the graphics drawing area.
7. Now move the state name and cursor to the appropriate location inside the left concurrent state within the **KEEP_TIME_BHVR** state.
8. Click the left mouse button to associate the name with the state that it resides in.
9. Repeat steps 5 through 8 to name the other concurrent state, **MINUTES_TIMER**.

The third step is to draw and name the remaining states: **CALC_HOURS** and **CALC_MINUTES**.

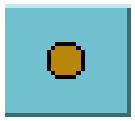
1. Select the **Create State** icon, move the cursor to the location of the upper, left-hand corner of the **CALC_HOURS** state.
2. Press the left mouse button and hold it down while dragging the mouse to the desired location of the lower, right-hand corner of the state. A ghost image shows the state outline. Release the mouse button and the state is drawn.
3. Type the state name, in this case, **CALC_HOURS**.
4. Move the state name and cursor inside the state you just drew.
5. Click the left mouse button to associate the name with the state that it resides in.
6. Repeat steps 1 through 5 above to draw and name the state **CALC_MINUTES**.

You have now entered the states for the **KEEP_TIME_BHVR** Statechart.

4. 8. Drawing Transitions and Connectors

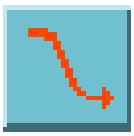
In this section, you will draw connectors, draw the transitions between states, and draw transitions between states and connectors.

The first step is to draw the junction connectors within the **HOURS_TIMER** and **MINUTES_TIMER** states.



1. Select the **Create Junction** icon from the graphic editor. Notice that the currently active command is indicated above the drawing icons.
2. Place the cursor at the desired location within the **HOURS_TIMER** timer state for the junction connector and click the left mouse button. This draws the junction connector.
3. Place the cursor at the desired location within the **MINUTES_TIMER** timer state for the junction connector and click the left mouse button.

The second step is to draw the transition from the **CALC_HOURS** state to the junction connector.



1. Select the **Create Transition** icon from the graphic editor. Notice that the currently active command is indicated above the drawing icons.
2. Locate the **CALC_HOURS** state. This is the source state.
3. Place the cursor on the edge of the **CALC_HOURS** state and click the left mouse button to enter the tail of the arrow.
4. Locate the junction connector. This is the target connector.
5. Place the cursor in the center of the connector, then click the middle mouse button to enter the arrow head.

The third step is to label the transition from the **CALC_HOURS** state to the junction connector.

1. While the **Create Transition** mode is still active, move your cursor inside the graphics drawing area.
2. Type the label **[MINUTES=59] and tm(en(CALC_MINUTES),1)**. The letters you are typing appear in the graphics drawing area.
3. Now move the label and cursor to the appropriate location with the '+' just touching the transition.

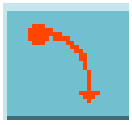
4. Click the left mouse button to associate the name with the transition that the '+' is touching.

The fourth step is to use table below and the procedure outlined above to label all the transitions in the Statechart.

Note: To give the transition its desired shape, create intermediate points by clicking the left mouse button while drawing the transition.

SOURCE	TARGET	LABEL
Junction Connector	CALC_HOURS	[HOURS/=12]/ HOURS:=HOURS+1
Junction Connector	CALC_HOURS	[HOURS=12]/HOURS:=1
CALC_MINUTES	Junction Connector	tm(en(CALC_MINUTES),1)
Junction Connector	CALC_MINUTES	[MINUTES/=59]/ MINUTES:=MINUTES+1
Junction Connector	CALC_MINUTES	[MINUTES=59]/MINUTES:=0

The fifth step is to enter the default transitions, one into the **CALC_HOURS** state and the other into the **CALC_MINUTES** state.



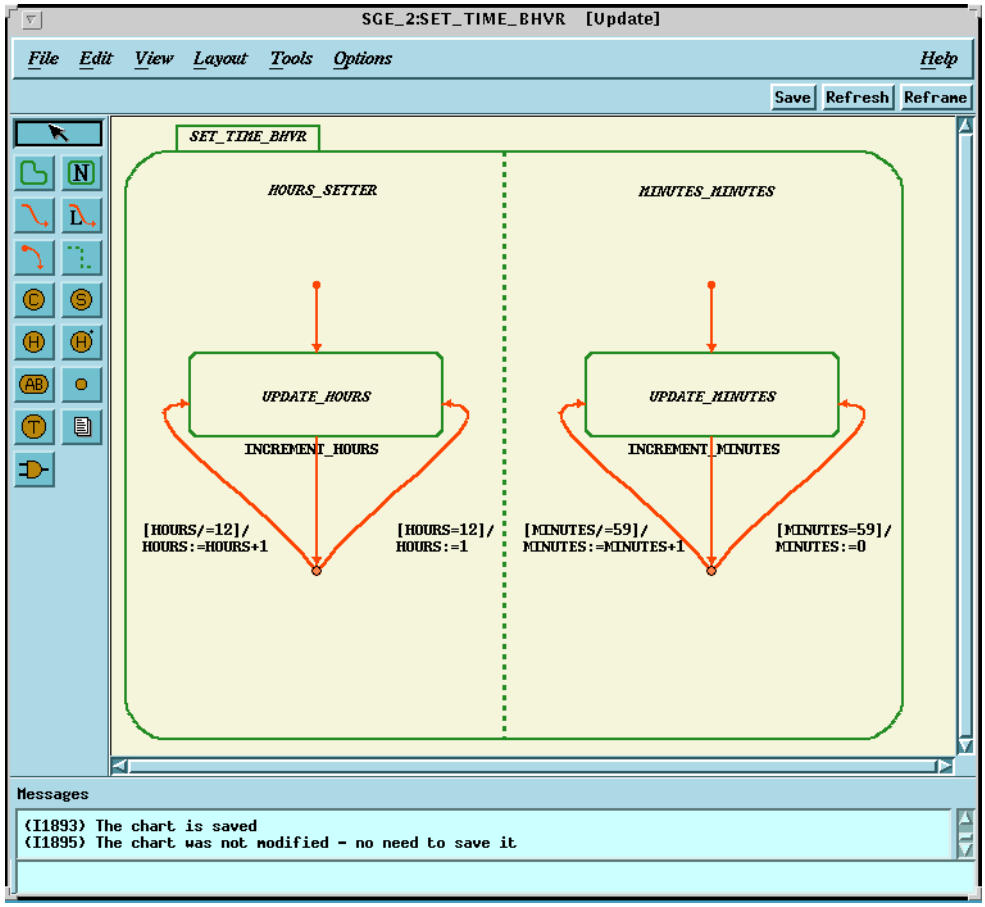
1. Select the **Create Default Transition** icon from the graphic editor. Notice that the currently active command is indicated above the drawing icons.
2. Locate the **CALC_HOURS** state. This is the target state.
3. Locate a place just outside of the **CALC_HOURS** state to be the source of the default transition. Click the left mouse button to enter the arrow tail.
4. Place the cursor on the edge of the **CALC_HOURS** state, then click the middle mouse button to enter the arrow head.
5. Locate the **CALC_MINUTES** state. This is the target state.
6. Locate a place just outside of the **CALC_MINUTES** state to be the source of the default transition. Click the left mouse button to enter the arrow tail.
7. Place the cursor on the edge of the **CALC_MINUTES** state, then click the center mouse button to enter the arrow head.

The **KEEP_TIME_BHVR** Statechart is now complete. Save the chart by selecting the File pulldown menu, followed by the Save option.

4. 9. Creating a Statechart Using the Copy Command

In this section, you create the **SET_TIME_BHVR** Statechart using the **Copy** command.

Notice that the behavior of both the **KEEP_TIME_BHVR** and **SET_TIME_BHVR** control activities takes on the same structure, so we make a copy of the **KEEP_TIME_BHVR** Statechart and then modify it to create the **SET_TIME_BHVR** Statechart.



The first step is to copy the **KEEP_TIME_BHVR** Statechart to create the **SET_TIME_BHVR** Statechart.

1. If the **Workarea Browse**r is not open, select the **Workarea Browser** icon from the **Statemate Main** window.

2. Select the **KEEP_TIME_BHVR** Statechart icon from the model tree.
3. Select ***Edit*** > ***Copy***.

   Result: The **Copy Statechart** dialog appears with **KEEP_TIME_BHVR** entered in the source name field.
4. Enter the name **SET_TIME_BHVR** into the target name field.
5. Select **OK**.

   Result: The **SET_TIME_BHVR** Statechart is created.

The second step is to place the "**@**" symbol in front of the control activity name to create the link between the control activity and the Statechart. We also must verify that the name of the Statechart we just created, **SET_TIME_BHVR**, is the same as that of the control activity it will be linked to.

1. In the **Workarea Browser**, locate the **CLOCK_*your-initials*** Activity-chart. Double-click on this chart's icon to open the chart.
2. Locate the **SET_TIME_BHVR** control activity.
3. Double-click on the **SET_TIME_BHVR** control activity name. This causes the name to be highlighted in black with a cursor appearing.
4. Click the left mouse button at the beginning of the control activity name. This moves the cursor to the left of the **SET_TIME_BHVR** name.
5. Type the symbol **@**. The name now appears as **@SET_TIME_BHV**R.
6. Click the middle mouse button away from the selected text field to terminate **Edit Text** mode.

The third step is to verify that the link was made between the control activity and the Statechart.

1. In the **Workarea Browser**, select the ***View*** pulldown menu, followed by the ***Refresh*** option to update the display.
2. Verify that a line appears in the **Workarea Browser** showing that the **SET_TIME_BHVR** Statechart is connected to the **CLOCK_*your-initials*** Activity-chart.

If this is not the case, verify that the name of the Statechart matches the name of the control activity.

The fourth step is to modify the names and labels in the **SET_TIME_BHVR** Statechart.

1. Open the **SET_TIME_BHVR** Statechart from the **Workarea Browse**r by double-clicking on the **SET_TIME_BHVR** Statechart icon from the model tree.
2. Double-click the state name **KEEP_TIME_BHVR** with the left mouse button. This causes the name to be highlighted in black with a cursor appearing.
3. Edit the state name from **KEEP_TIME_BHVR** to **SET_TIME_BHVR**.
4. Select the next state name or transition label to be changed by clicking the left mouse button on it.
5. Follow the table below, along with the above procedure, to make the rest of the text changes.
6. Click the middle mouse button away from the selected text field to terminate the edit text mode.

DESCRIPTION	OLD NAME OR LABEL	NEW NAME OR LABEL
State Name	HOURS_TIMER	HOURS_SETTER
State Name	MINUTES_TIMER	MINUTES_SETTER
State Name	CALC_HOURS	UPDATE_HOURS
State Name	CALC_MINUTES	UPDATE_MINUTES
Label	[MINUTE=59] and tm(en(CALC_MINUTES),1)	INCREMENT_HOURS
Label	tm(en(CALC_MINUTES,1)	INCREMENT_MINUTES

7. Save your chart using ***File>Save***.

You have now completed the **SET_TIME_BHVR** Statechart.

Note: For more information on Statecharts, see

Statemate MAGNUM Reference Manual, Chapters 3 and 4.

Statemate MAGNUM User Guide, Chapter 6.

Modeling Reactive Systems with Statecharts, The Statemate Approach, by Michal Politi and David Harel.

LESSON 5. DEFINING THE TEXTUAL ELEMENTS

5. 1. Introduction

In this chapter, you will define textual elements in your design.

In class, you learned that textual elements can be the following:

- **Events**—An edge trigger or impulse
- **Conditions**—A level trigger or Boolean
- **Data-Items**—Integer, real, string, bit, bit-array, record, union or user-defined type

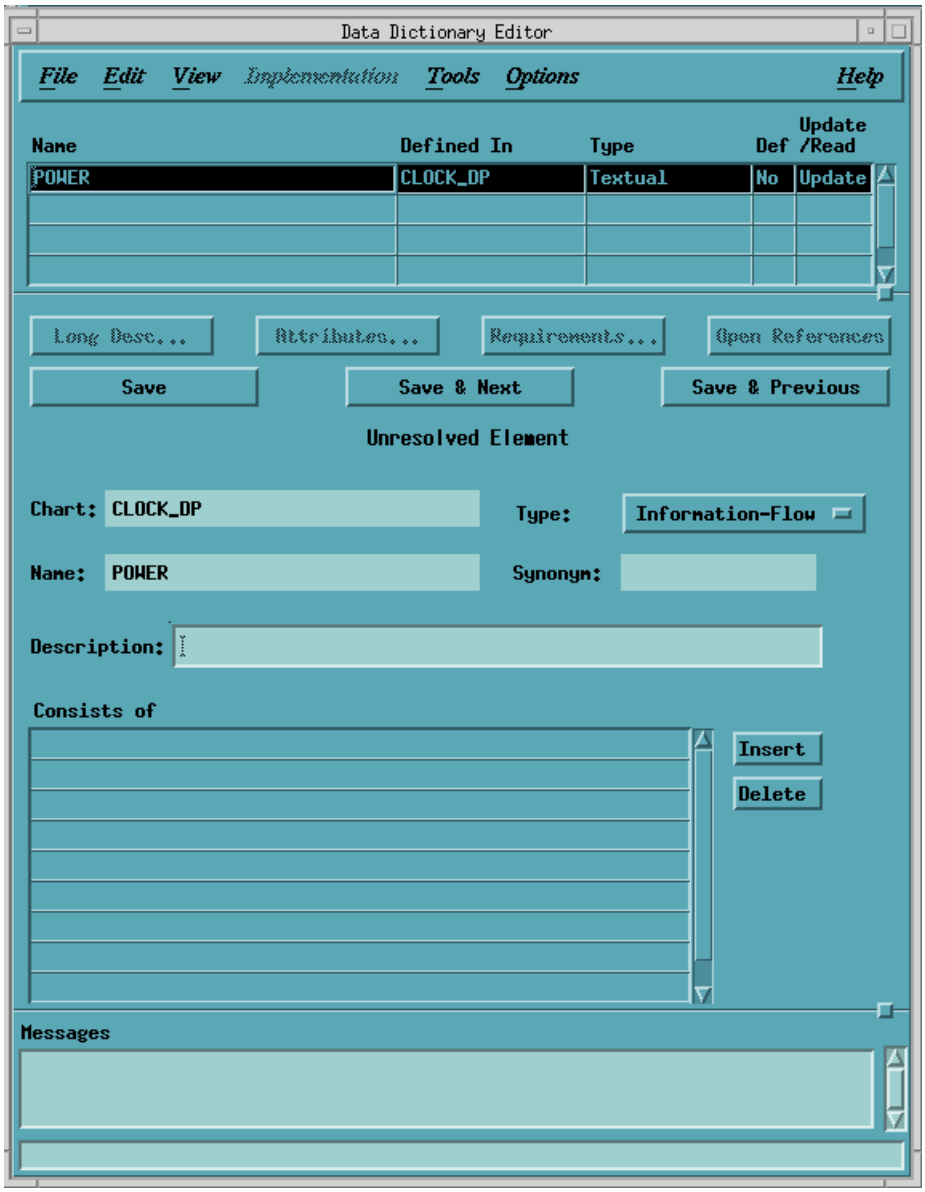
You also learned that in Activity-charts you can have information-flows. An information-flow is a container used to group textual elements together.

5. 2. Defining Elements from the GE

In this section, you will define an element directly from the **Graphics Editor**.

1. In the **CLOCK_*your-initials*** Activity-chart, select the **POWER** label.
2. Select ***Tools>Data Dictionary***.

   Result: A **Data Dictionary Editor** opens, with the **POWER** element selected.



3. Define this element as a condition by selecting the **Type** menu button in the lower portion of the form and from the list of types, select **Condition**.
4. Select the **Save** button to save this definition.
5. Select ***File>Exit*** to close the **Data Dictionary** window.

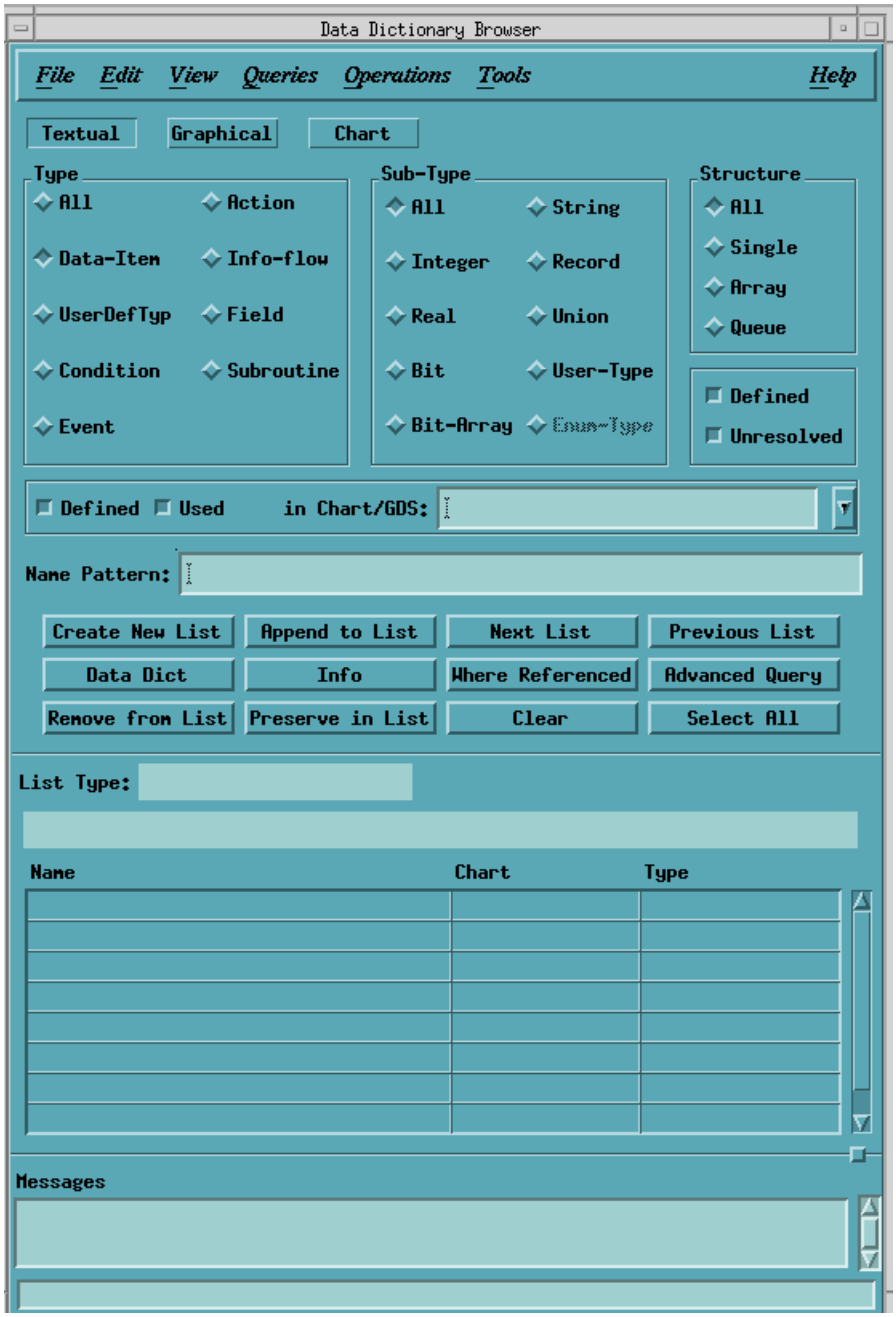
You have just defined the element **POWER** to be a condition.

5. 3. Defining Elements Using the Data Dictionary Browser

In this section, you will define an element from the **Data Dictionary Browser**.

1. Select the **Data Dictionary Browser** icon from the **Statemate Main** window.

   Result: RESULT: A **Data Dictionary Browser** window is displayed.



2. In the **Data Dictionary Browser** window, make the following selections:
   - **Textual**
   - **Type - All**
   - **Structure - All**
   - **Unresolved** (deselect **Defined**)
   - **Used in Chart/GDS: CLOCK_your-initials**
   - **Name Pattern: TIME**
3. Select the **Create New List** button.

   Result: The **TIME** element is displayed in the list area.
4. Select the **Select All** button to highlight the element in the list.
5. Select the **Data Dict** button to open a **Data Dictionary Editor** window with a list containing the **TIME** element. The element is highlighted and the form containing information about this element is displayed.
6. Verify or enter the following information about the **TIME** element in the **Data Dictionary Editor** window.
   - **Chart: CLOCK_*your-initials***
   - **Name: TIME**
   - **Type: Information-flow**
   - Add to the **Consists of** field the following elements:

     HOURS

     MINUTES
7. Select the **Save** button.
8. Select ***File > Exit***.

You have now defined an element using the **Data Dictionary Browser** and **Editor**.

5. 4. Defining All the Unresolved Elements in a Model

You now use the **Data Dictionary Browser** to define all of the unresolved elements in your model.

1. Select the **Clear** button in the **Data Dictionary Browser** window to clear your list.
2. In the **Data Dictionary Browser** window, make the following selections:

- **Textual**
- **Type - All**
- **Structure - All**
- **Unresolved**
- **Used in Chart/GDS: CLOCK_y*our-initials***
- Clear the **Name Pattern:** field.

3. Select the **Create New List** button.

   Result: The **HOURS**, **INCREMENT_HOURS**, **INCREMENT_MINUTES**, **MINUTES** and **MODE** elements are displayed in the list area.

4. Select the **Select All** button to highlight all the elements in the list.

5. Select the **Data Dict** button to open a **Data Dictionary Editor** with a list containing the elements listed above. The **HOURS** element is highlighted and the form containing information about this element is displayed.

6. To define the element **HOURS**, enter the following information in the **Data Dictionary Editor** window.

   - **Chart: CLOCK_your-initials**
   - **Name: HOURS**
   - **Type: Data-Item**
   - **Structure: Single**
   - **Data type: Integer**
   - **Min: 1**
   - **Max: 12**
   - **Usage: Variable**

7. Select the **Save & Next** button to save the **HOURS** element and select the **INCREMENT_HOURS** element.

8. To define the element **INCREMENT_HOURS**, enter the following information in the **Data Dictionary Editor** window.

   - **Chart: CLOCK_your-initials**
   - **Name: INCREMENT_HOURS**
   - **Type: Event**

9. Select the **Save & Next** button to save the **INCREMENT_HOURS** element and select the **INCREMENT_MINUTES** element.

10. To define the **INCREMENT_MINUTES** element, enter the following information in the **Data Dictionary Editor** window:

- **Chart: CLOCK_your-initials**
- **Name: INCREMENT_MINUTES**
- **Type: Event**
- **Structure: Single**

11. Select the **Save & Next** button to save the **INCREMENT_MINUTES** element and select the **MINUTES** element.
12. To define the **MINUTES** element, enter the following information in the **Data Dictionary Editor** window.
    - **Chart: CLOCK_your-initials**
    - **Name: MINUTES**
    - **Type: Data-Item**
    - **Structure: Single**
    - **Data type: Integer**
    - **Min: 0**
    - **Max: 59**
    - **Usage: Variable**
13. Select the **Save & Next** button to save the **MINUTES** element and select the **MODE** element.
14. To define the **MODE** element, enter the following information in the **Data Dictionary Editor** window.
    - **Chart: CLOCK_your-initials**
    - **Name: MODE**
    - **Type: Data-Item**
    - **Structure: Single**
    - **Data type: Integer**
    - **Usage: Variable**
15. Select the **Save** button to save the **MODE** element.
16. Select ***File > Exit***.

You have now defined all the textual elements in your model.

Note: For more information on the Data Dictionary, see

Statemate MAGNUM Reference Manual, Chapter 8.
Statemate MAGNUM User Guide, Chapter 7.

LESSON 6. SIMULATING THE MODEL

6. 1. Introduction

In this chapter, you will perform an interactive simulation of the model.

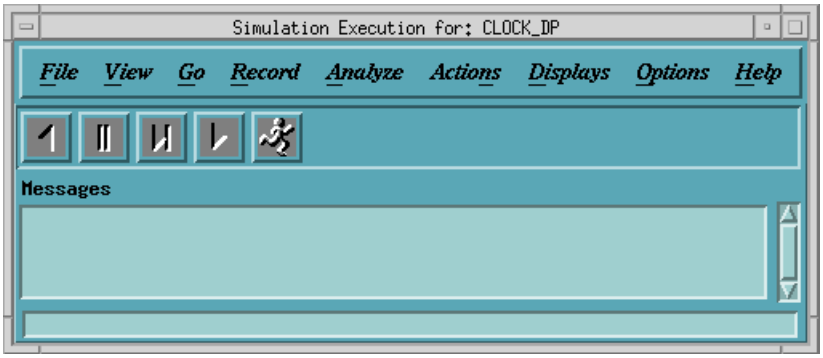
In class, you learned that you can simulate your model to perform various *what if* scenarios. When simulating, some of the things you can do are:

- Take steps through the model using the **GoStep** command.
- Verify which activities are active and which states the system is in by animating the graphics.
- Examine the value of elements and change the value of inputs using a monitor.

6. 2. Invoking Simulation from the GE

In this section, you will start the simulator from an Activity-chart graphics editor.

1. If the **CLOCK_*your-initials*** Activity-chart is not open, open it using the **Workarea Browser**.
2. Select the **CLOCK** activity.
3. Select ***Tools*** > ***Simulation*** to open the **Simulation Execution** main window.

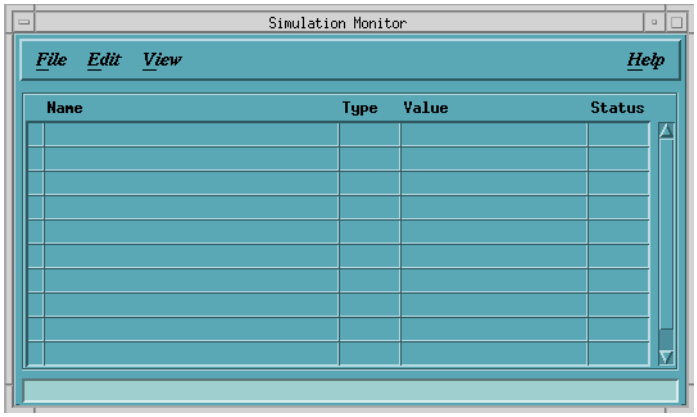


You have now invoked the simulator.

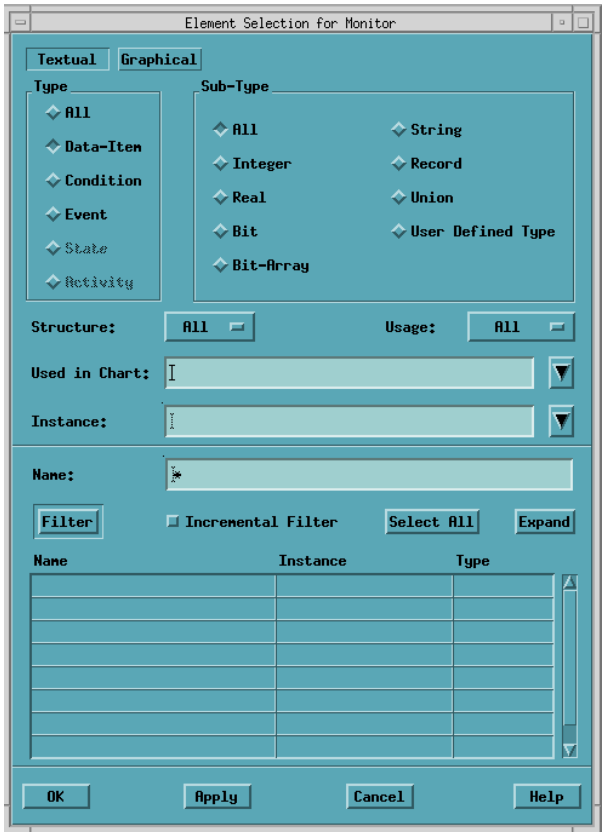
6. 3. Setting Up a Monitor Window

In this section, you will set up a monitor window.

1. Select ***Displays*** > ***Monitors*** to open an empty **Simulation Monitor** window.



2. Select ***Edit > Add*** to open the **Element Selection for Monitor** window.



3. Fill in the options/fields for the **Element Selection for Monitor** window as follows:
   - Make sure the **Textual** button is selected and the **Graphical** button is not selected.
   - For **Type**, select **All**.
   - For **Structure**:, select **All**.
   - For **Usage**:, select **All**.
   - The **Used in Chart**: field should be blank.
   - The **Instance**: field should be blank.
   - The **Name**: field should contain the symbol "*".
4. Select the **Filter** button.

   Result: The following signals are displayed in the listing area of the window:

   - **INCREMENT_HOURS**
   - **INCREMENT_MINUTES**
   - **HOURS**
   - **MINUTES**
   - **MODE**
   - **POWER**

5. Select the **Select All** button.

   Result: All the elements in the listing area are highlighted.

6. Select **OK**.

   Result: The **Element Selection for Monitor** window closes and the signals listed above appear in the **Simulation Monitor** window.

You have now set up a monitor window.

6. 4. Stepping Through the Simulation

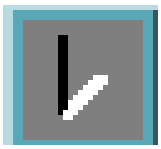
In this section, you organize your screen for viewing a simulation and then step through the simulation.

The first step is to organize your viewing area.

1. Open the following charts:
   - **CLOCK_your-initials** (Activity-chart)
   - **CLOCK_CNTL** (Statechart)
   - **KEEP_TIME_BHVR** (Statechart)
   - **SET_TIME_BHVR** (Statechart)
2. Resize all four charts so that you can simultaneously view all the charts and the **Simulation Execution** and **Simulation Monitor** windows.
3. In each of the graphic editors, select ***Options > Enable Scale Text***.

The second step is to simulate the model.

1. Verify that the **CLOCK** activity has been highlighted, but the **KEEP_TIME** and **SET_TIME** activities are not highlighted. This indicates that the **CLOCK** activity is active and the **KEEP_TIME** and **SET_TIME** activities are inactive.
2. Verify that none of the states in any of the three Statecharts are highlighted. This indicates that the system is currently in none of the states.



3. Take a step by selecting the **Go Step** icon from the **Simulation Execution** window.

   Result: The **OFF** state and the default transition to the **OFF** state in the **CLOCK_CNTL** Statechart become highlighted. This indicates that the system is transitioning into the **OFF** state.
4. Take another step by selecting the **Go Step** icon again.

   Result: Just the **OFF** state is highlighted now. This indicates that the system is in the **OFF** state. The default transition to the **OFF** state in the **CLOCK_CNTL** Statechart becomes highlighted. This indicates that the system is transitioning into the **OFF** state.

5. In the **Simulation Monitor** window, select the value cell associated with the **POWER** condition.

   Result: The value of the **POWER** condition toggles from **False** to **True**.

6. Take another step by selecting **Go Step**.

   Result: In the **CLOCK_CNTL** Statechart, the **SET** state is highlighted, along with the default transition into it. This indicates that the system is now entering the **SET** state. Notice also that the transition from the **OFF** state to the **ON** state is highlighted, indicating that the system is leaving the **OFF** state and transitioning to the **ON** state. Also notice that the **SET_TIME** activity in the **CLOCK_*your-initials*** Activity-chart is highlighted, indicating that it has become active.

7. Take another step by selecting **Go Step**.

   Result: Just the **SET** state in the **CLOCK_CNTL** Statechart is highlighted now, indicating that the system is in the **SET** state. In the **SET_TIME_BHVR** Statechart, notice that the **UPDATE_HOURS** and **UPDATE_MINUTES** states are highlighted, along with the default transitions to them. This indicates that the system is transitioning to the **UPDATE_HOURS** and **UPDATE_MINUTES** states.

8. In the **Simulation Monitor** window, select the value cell associated with the **INCREMENT_HOURS** event.

   Result: The value of the **INCREMENT_HOUR** event is generated. This is indicated by an **X** being displayed in the value cell.

9. Set the value for the **HOURS** data-item in the **Simulation Monitor** window to **1**.

10. Take a step by selecting **Go Step**.

    Result: The transition from the **UPDATED_HOURS** state to the junction connector and back to the **UPDATE_HOURS** state becomes highlighted.

11. Verify in the **Simulation Monitor** window that the value of the data-item **HOURS** has changed from **1** to **2** and that the **INCREMENT_HOURS** event no longer exists (indicated by the value cell associated with the **INCREMENT_HOURS** event becoming blank).

12. Set the value for the **MODE** data-item in the **Simulation Monitor** window to **1**.
13. Take a step by selecting **Go Step**.

    Result: In the **CLOCK_CNTL** Statechart, the **RUN** state is highlighted, along with the transition from the **SET** state to the **RUN** state. This indicates that the system is now entering the **RUN** state and exiting the **SET** state. Also notice that the **KEEP_TIME** activity in the **CLOCK_your-initials** Activity-chart is highlighted, indicating that it has become active and that the **SET_TIME** activity is no longer highlighted, indicating that it has become inactive.
14. Take a step by selecting the **Go Step** icon.

    Result: Just the **RUN** state in the **CLOCK_CNTL** Statechart is highlighted now, indicating that the system is now in the **RUN** state. In the **KEEP_TIME_BHVR** Statechart, notice that the **CALC_HOURS** and **CALC_MINUTES** states are highlighted, along with the default transitions to them. This indicates that the system is transitioning to the **CALC_HOURS** and **CALC_MINUTES** states.
15. Take a step by selecting the ***GoNext*** option from the ***Go*** pull-down.

    Result: Time is advanced by the time delay in the timeout event.
16. Take a step by selecting the **Go Step** icon.

    Result: This results in the transition from the **CALC_MINUTES** state to the junction connector and back to the **CALC_MINUTES** state to be highlighted.
17. Verify in the **Simulation Monitor** window that the value of the data-item **MINUTES** has changed from **0** to **1**.
18. Take a step by selecting ***Go*** > ***GoNext***.

    Result: Time is advanced by the time delay in the timeout event.
19. Take a step by selecting the **Go Step** icon.

    Result: The transition from the **CALC_MINUTES** state to the junction connector and back to the **CALC_MINUTES** state becomes highlighted.

20. Verify in the **Simulation Monitor** window that the value of the data-item **MINUTES** has changed from **1** to **2**.
21. In the **Simulation Monitor window**, select the value cell associated with the **POWER** condition.

    Result: The value of the **POWER** condition toggles from **True** to **False**.
22. Take a step by selecting the **Go Step** icon.

    Result: The transition from the **ON** state to the **OFF** state in the **CLOCK_CNTL** Statechart becomes highlighted. Notice also that both the **KEEP_TIME** and **SET_TIME** activities are not highlighted, indicating that they are both inactive.
23. Take some time now to experiment with some of your own simulation scenarios.

You have now performed a simulation of your model.

Note: For more information on simulation, see

Statemate Simulator Reference Manual

LESSON 7. PANEL CREATION AND SIMULATION

7. 1. Introduction

In this chapter, you will build a panel for your design.

7. 2. Creating a Panel

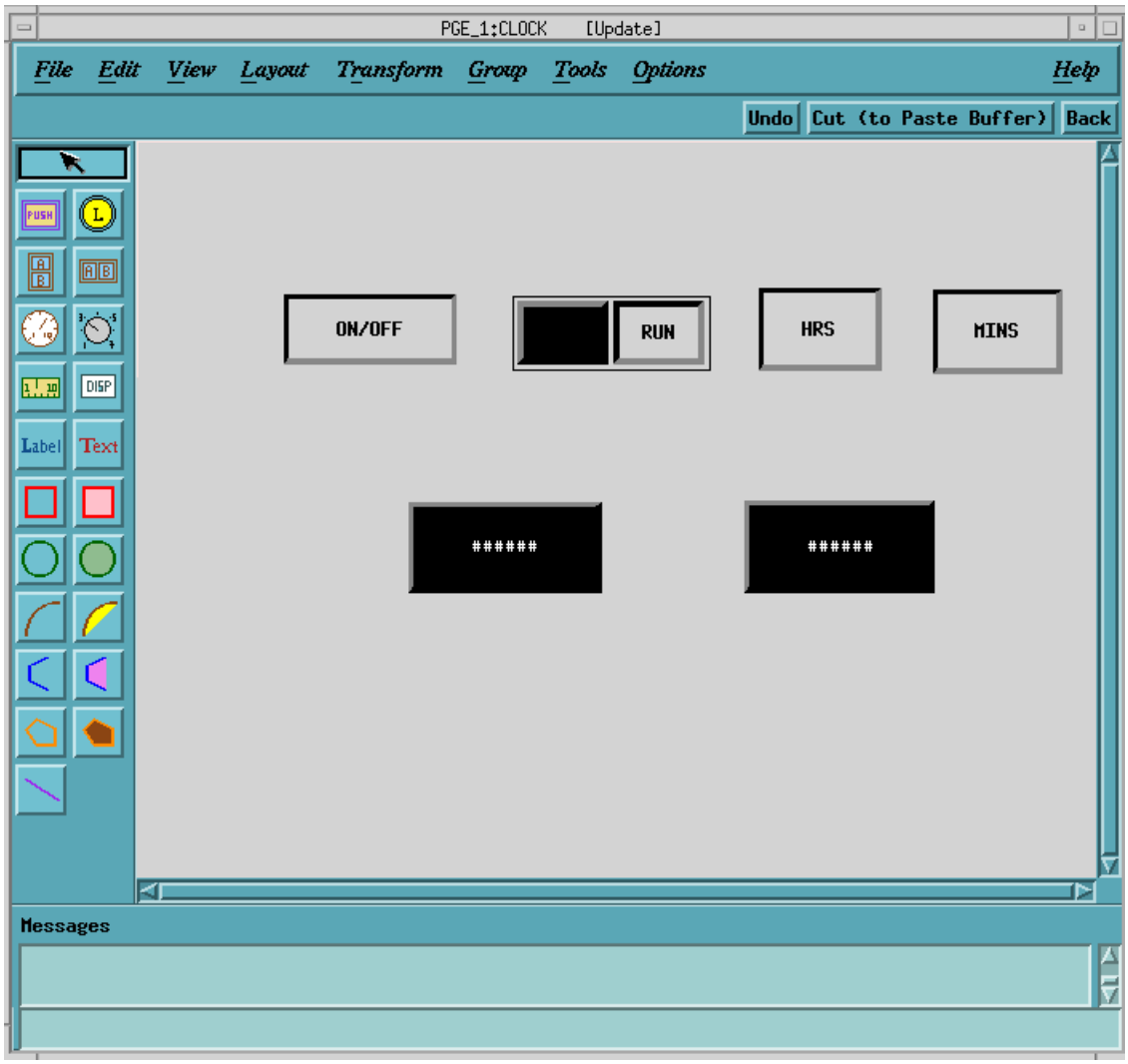
In this section, you will create a panel.

1. Select the **Panel Editor** icon from the **Statemate Main** window.
2. Fill in the **Open Panel** form, as follows:
   - **Panel Name:CLOCK**
   - **Editor Mode:Update**
3. Select **OK**.

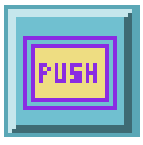
   Result: A **Panels Editor** window appears.

7. 3. Drawing the Panel Elements

You now will draw elements of a panel.



The first step is to draw the **POWER (ON/OFF)** push button.

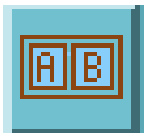


1. Select the **Create Push Button Interactor** icon from the panel editor. Notice that the currently active command is indicated above the drawing icons.
2. Place the cursor at the desired location for the upper, left-hand corner of the **ON/OFF** push button and hold the left mouse button down while dragging the mouse to the desired location of the lower, right-hand corner of the push button. A ghost image shows the push button outline. Release the mouse button and the push button is drawn.

The second step is label the **ON/OFF** push button.

1. Select the **Label Existing Interactor** icon.
2. Type the push button label, in this case, **ON/OFF**. The letters you are typing appear in the graphics drawing area. (For uppercase letters, press the SHIFT key.)
3. Now move the label and cursor to the appropriate location inside the push button's boundaries.
4. Click the left mouse button to associate the label with the push button that it resides in.

The third step is to draw the **MODE (SET/RUN)** radio button.



1. Select the **Create Horizontal Multi-Choice Interactor** icon.
2. Place the cursor at the desired location for the upper, left-hand corner of the radio button and hold the left mouse button down while dragging the mouse to the desired location of the lower, right-hand corner of the radio button. Release the mouse button and a dialog box appears.
3. In the dialog box, set the number of buttons field to **2.**
4. Select **OK.**

   Result: The radio button is drawn with two buttons.

The fourth step is to label the SET/RUN radio button.

1. Select the **Label Existing Interactor** icon.
2. Type one of the radio button choice labels, in this case, **SET**.
3. Now move the label and cursor to the appropriate location inside the radio button choice boundaries.
4. Click the left mouse button to associate the label with the radio button choice that it resides in.
5. Type the other radio button choice label, in this case, **RUN**.
6. Now move the label and cursor to the appropriate location inside the radio button choice boundaries.
7. Click the left mouse button to associate the label with the radio button choice that it resides in.

The fifth step is to draw the **INCREMENT_HOURS (HRS)** push button.

1. Select the **Create Push Button** Interactor icon.
2. Place the cursor at the desired location for the upper, left-hand corner of the push button and hold the left mouse button down while dragging the mouse to the desired location of the lower, right-hand corner of the push button. Release the mouse button.

The sixth step is label the **HRS** push button.

1. Select the **Label Existing Interactor** icon.
2. Type the push button label, in this case, **HRS**.
3. Now move the label and cursor to the appropriate location inside the push button's boundaries.
4. Click the left mouse button to associate the label with the push button that it resides in.

The seventh step is to draw the **INCREMENT_MINUTES (MINS)** push button.

1. Select the **Create Push Button Interactor** icon.
2. Place the cursor at the desired location for the upper, left-hand corner of the push button and hold the left mouse button down while dragging the mouse to the desired location of the lower, right-hand corner of the push button. Release the mouse button.

The eighth step is label the **MINS** push button.

1. Select the **Label Existing Interactor** icon.
2. Type the push button label, in this case, **MINS**.
3. Now move the label and cursor to the appropriate location inside the push buttons boundaries.
4. Click the left mouse button to associate the label with the push button that it resides in.

The ninth step is to draw the **HOURS** display.

1. Select the **Create Textual Display Interactor** icon.
2. Place the cursor at the desired location for the upper, left-hand corner of the display and hold the left mouse button down while dragging the mouse to the desired location of

the lower, right-hand corner of the display. Release the mouse button.

The tenth step is to draw the **MINUTES** display.

1. Select the **Create Textual Display Interactor** icon.
2. Place the cursor at the desired location for the upper, left-hand corner of the display and hold the left mouse button down while dragging the mouse to the desired location of the lower, right-hand corner of the display. Release the mouse button.
3. Select ***File>Save***.

You have now drawn your panel elements.

7. 4. Setting Element Properties and Binding the Elements

You now will set panel element properties and bind the panel elements to model elements.

The first step is to set the properties for the **ON/OFF** push button.

1. Select the **ON/OFF** push button.
2. Select ***Tools>Properties*** to open a **Bindings/Properties** window, with the properties of the **ON/OF**F push button displayed.
3. In the lower section of the window, select the **Name** property.
4. Enter **POWER** into the field associated with the **Name** property.

5. Select the **Apply** button.
6. Select the **Fill Color** property.
7. Select the **Choose** button to open a **Color Viewer for PGE** window.
8. In the **Color Viewer for PGE** window, select the color **GREEN4**.
9. Select **OK** in the **Color Viewer for PGE** window.

10. Select the **Apply** button in the **Bindings/Properties** window.

The second step is to bind the **ON/OFF** push button.

1. In the **Bindings/Properties** window, change the **Properties** selection to **Bindings** to display the bindings for the push button.
2. Change the element type to **Condition** by clicking on the **Element type** field (currently displaying **Event**) with the left mouse button. This results in a pop-up menu being displayed.
3. In the pop-up menu, select the **Condition** option.
4. Change the **Button type** to toggle by clicking on the element type field (currently displaying **Flash**) with the left mouse button. This results in a pop-up menu being displayed.
5. In the pop-up menu, select the **Toggle** option.
6. Enter the element **POWER** in the **Controlled by** field.
7. Select **OK**.

The third step is to set the properties for the **SET/RUN** radio button.

1. Select the **SET/RUN** radio button.
2. Select ***Tools>Properties*** to display the **Bindings/ Properties** window with the properties of the **SET/RUN** radio button displayed.
3. Select the **Name** property.
4. Enter **MODE** into the field associated with the **Name** property.
5. Select **Apply**.
6. Select the **Foreground Color** property.
7. Select the **Choose** button to open a **Color Viewer for PGE** window.
8. In the **Color Viewer for PGE** window, select the color **RED3**.
9. Select the **OK** button in the **Color Viewer for PGE** window.
10. Select **Apply** in the **Bindings/Properties** window.

The fourth step is to bind the **SET/RUN** push button.

1. In the **Bindings/Properties** window, change the **Properties** selection to **Bindings**.
2. Ensure the value of **SET** is **0**.
3. Ensure the value of **RUN** is **1**.
4. Enter the element **MODE** in the **Controlled by** field.
5. Select **OK**.

The fifth step is to set the properties for the **HRS** push button.

1. Select the **HRS** push button.
2. Select ***Tools>Properties*** to display the **Bindings/ Properties** window with the properties of the HRS push button displayed.
3. Select the **Name** property.
4. Enter **INC_HRS** into the field associated with the Name property.
5. Select **Apply**.

The sixth step is to bind the **HRS** push button.

1. In the **Bindings/Properties** window, change the **Properties** selection to **Bindings**.
2. Verify that the **Element type** is **Event**.
3. Verify that the **Button type** is **Flash**.
4. Enter the element **INCREMENT_HOURS** in the **Controlled by** field.
5. Select **OK**.

The seventh step is to set the properties for the **MINS** push button.

1. Select the **MINS** push button.
2. Select ***Tools>Properties*** to display the **Bindings/ Properties** window with the properties of the **MINS** push button displayed.
3. Select the **Name** property.
4. Enter **INC_MINS** into the field associated with the **Name** property.
5. Select **Apply**.

The eighth step is to bind the **MINS** push button.

1. In the **Bindings/Properties** window, change the properties selection to bindings.
2. Verify that the **Element type** is **Event**.
3. Verify that the **Button type** is **Flash**.
4. Enter the element **INCREMENT_MINUTES** in the **Controlled by** field.
5. Select **OK**.

The ninth step is to set the properties for the **HOURS** display.

1. Select the **HOURS** display.
2. Select ***Tools>Properties*** to open the **Bindings/Properties** window with the properties of the **HOURS** display displayed.
3. Select the **Name** property.
4. Enter **HOURS** into the field associated with the **Name** property.
5. Select **Apply**.

The tenth step is to bind the **HOURS** display.

1. In the **Bindings/Properties** window, change the properties selection to bindings.
2. Change the **Element type** to **Integer**.
3. Verify that the **Format** is **Decimal**.
4. Change the **Field length** to **2**.
5. Change the **Fraction part** to **0**.
6. Enter the element **HOURS** in the **Controlled by** field.
7. Select **OK**.

The eleventh step is to set the properties for the **MINUTES** display.

1. Select the **MINUTES** display.
2. Select ***Tools>Properties*** to open the **Bindings/Properties** window with the properties of the **MINUTES** display displayed.
3. Select the **Name** property.
4. Enter **MINUTES** into the field associated with the **Name** property.
5. Select **Apply**.

The last step is to bind the **MINUTES** display.

1. In the **Bindings/Properties** window, change the **Properties** selection to **Bindings**.
2. Change the **Element type** to **Integer**.
3. Verify that the **Format** is **Decimal**.
4. Change the **Field length** to **2**.
5. Change the **Fraction part** to **0**.
6. Enter the element **MINUTES** in the **Controlled by** field.
7. Select **OK**.

You have now set the properties for the panel elements and made your panel bindings.

7. 5. Simulating the Panel

In this section, you simulate your design much like you did earlier, except instead of a monitor window, you use your panel to generate inputs and observe your outputs.

The first step is to start the Simulator from an Activity-chart graphics editor.

1. If the **CLOCK_*your-initials*** Activity-chart is not open, open it using the **Workarea Browser**.
2. Select the **CLOCK** activity.
3. Select ***Tools>Simulation*** to open the **Simulation Execution** main window.

The second step is to include the panel within this simulation environment.

1. Select ***Displays>Panels*** from the **Simulation Execution** main window to display a list of all the panels in the workarea.
2. Select the panel named **CLOCK**.
3. Select **OK** to display the panel.

The third step is to organize your display for viewing a simulation.

1. Verify that your Activity-chart and all the Statecharts are open.

2. Arrange your screen so that you can easily view all the charts, the panel, and the Simulation Execution main window. If you have questions on how to do this, refer back to Chapter 5.

The fourth step is executing the simulation.

1. Put the simulation into auto run mode (continually advancing) by selecting the **Auto Run** icon from the **Simulation Execution** window.
2. Use the various buttons on the clock panel to change your inputs and verify your outputs using the displays.

You have now simulated your model using a panel.

Note: For more information on the Panel Graphics Editor, see

Statemate MAGNUM Reference Manual, Chapter 3 and Chapter 5.

Statemate MAGNUM User Guide, Chapter 10.

LESSON 8. CODE GENERATION AND EXECUTION

8. 1. Introduction

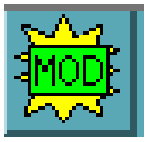
In this chapter, you will

- Create a code generation profile
- Use the profile to generate code
- How to run the code

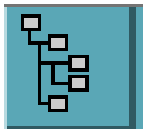
8. 2. Creating the Code Generation Profile

You first create the code generation profile.

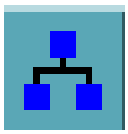
1. Select the **C Code Generation** icon from the **Statemate Main** window to open a **SW Code Generation Profile** window.
2. Select ***File>New Profile*** to display the **New Profile** window.
3. Enter **CLOCK_C_CODE** for the **Profile Name**.
4. Select **OK**.



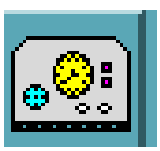
5. Select the **Create a New (Empty) SW Module in Profile** icon from the **SW Code Generation Profile** window to display a **Module Name** window.
6. Enter **CLOCK_MODULE** for the **Module name**.
7. Select **OK**.



8. Select the **Connect this Profile Editor to Workarea Browser** icon from the **SW Code Generation Profile** window to display the **Workarea Browser**.
9. In the **SW Code Generation Profile** window, select the module **CLOCK_MODULE**.
10. In the **Workarea Browser**, select the Activity-chart **CLOCK_*your-initials***.



11. Select the **Add Selected Chart to Selected Profile** icon from the **SW Code Generation Profile** window to add the chart **CLOCK_*your-initials***, along with all its descendant charts, to the simulation scope of this profile.
12. In the **Workarea Browser,** select the panel named **CLOCK**.



13. Select the **Add Selected Panel to Selected Module** icon from the **SW Code Generation Profile** window to add the **CLOCK** panel to the simulation scope of this profile.

14. Select ***File>Save***.

You have now created a code generation profile.

8. 3. Generating the Code

In this section, you will generate code.

1. From the **SW Code Generation Profile** window, select ***Compile>Generate Code*** to display the **Output Directory** window.
2. Select **OK** in the **Output Directory** window to open the **Code Generation** window.
3. Select **Dismiss** in the **Code Generation** window.

You have now created a code generation profile.

8. 4. Running the Code

In this section, you will execute the code you have generated.

1. From the **SW Code Generation Profile** window, select ***Compile>Make Code*** to open an **Output Directory** window.
2. Select **OK** in the **Output Directory** window to open the **Make Code** window.
3. Wait for the **"press Enter to quit prompt"** to appear.
4. Press RETURN to exit the **Make Code** window.
5. From the **SW Code Generation Profile** window, select ***Compile>Run Code*** to open the **Output Directory** window.
6. Select **OK** in the **Output Directory** window to open a **Debugger of Generated Code** window.
7. At the `pdb>` prompt, type **go**.
8. Press RETURN.

   Result: The code starts executing.
9. Use the panel to toggle inputs and verify your outputs.
10. Press <CONTROL>**c** to stop the program.
11. Enter **quit** at the `pdb>` prompt and **N** to the **Do you want to review?** prompt.

    Result: This exits the debugger.

You have now executed the code.

Note: For more information on the code generators, see

Statemate Code Generator Manuals:
Code Generator User's Guide
HDL Code Generator Reference Manual

LESSON 9. CREATING MINI-SPECS

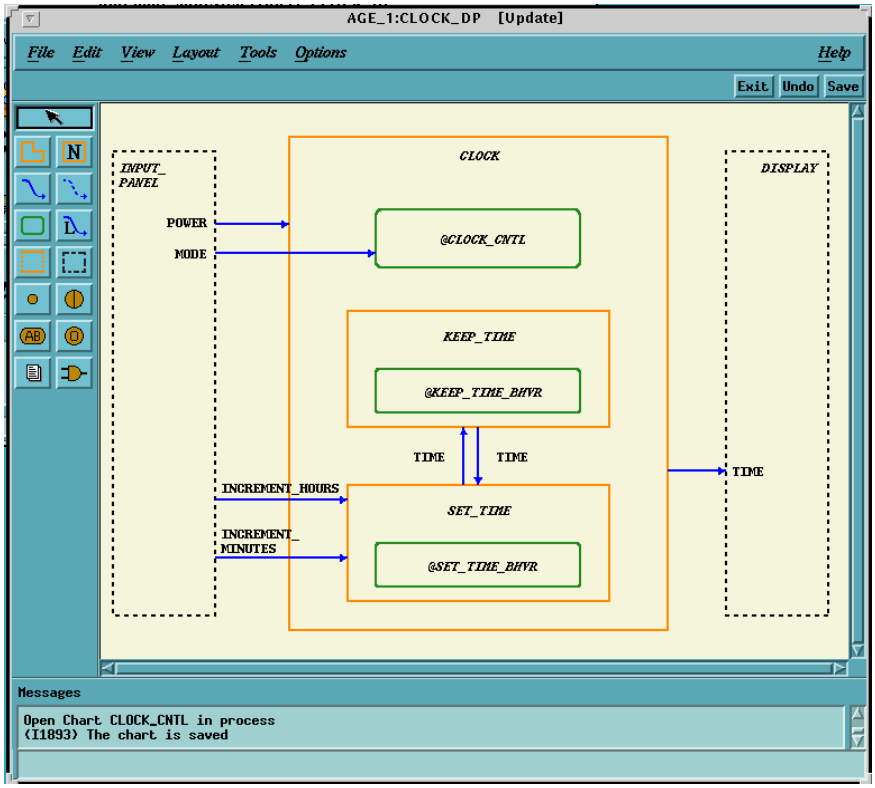
9.1 Introduction

We have seen how statecharts can be used to describe the behavior of an activity. In this chapter we will modify the model and describe the behavior of the activity **SET_TIME** using a mini-spec.

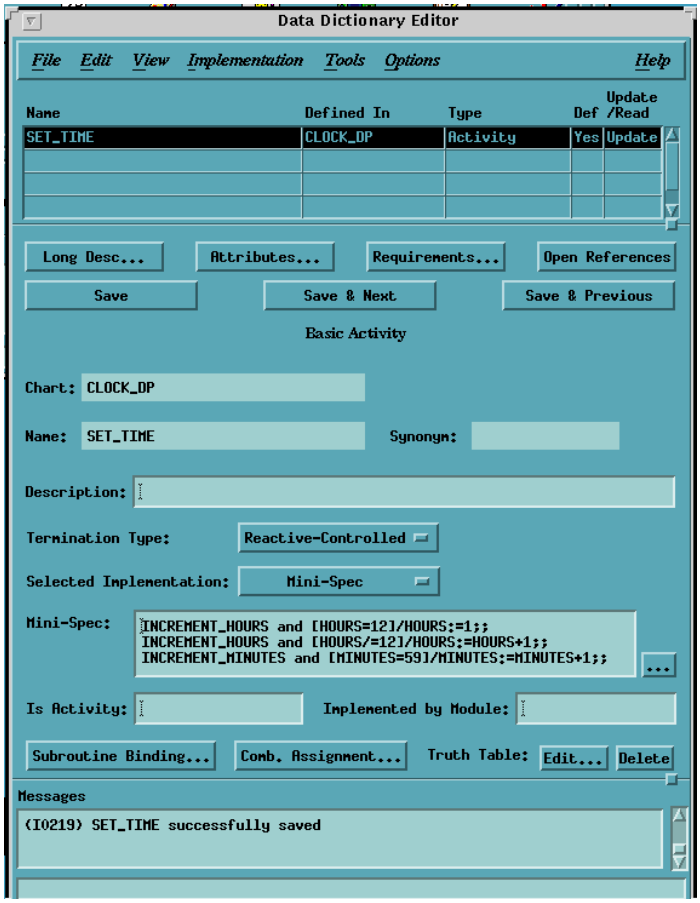
9.2 Entering a Mini-Spec

The follow steps will describe how to enter a mini-spec to the clock activity-chart.

1. Open the **CLOCK** activity-chart.
2. From the workarea browser, select activity chart.
3. Select Edit > Selected.
4. Select the control activity called **@SET_TIME_BHVR** and delete it, disassociating the statechart with this activity.
5. Select the activity called **SET_TIME**.



6. Open the Data Dictionary by selecting Tools > Data Dictionary.
7. For selected implementation, select mini-spec.



8. Within the mini-spec text box enter:

```
INCREMENT_HOURS AND [HOURS=12]/HOURS:=1;;
INCREMENT_HOURS AND [HOURS/=12]/HOURS:=HOURS+1;;
INCREMENT_MINUTES AND [MINUTES=59]/MINUTES:=0;;
INCREMENT_MINUTES AND [MINUTES/=59]/MINUTES:=MINUTES+1;
```

Alternate Mini-spec:

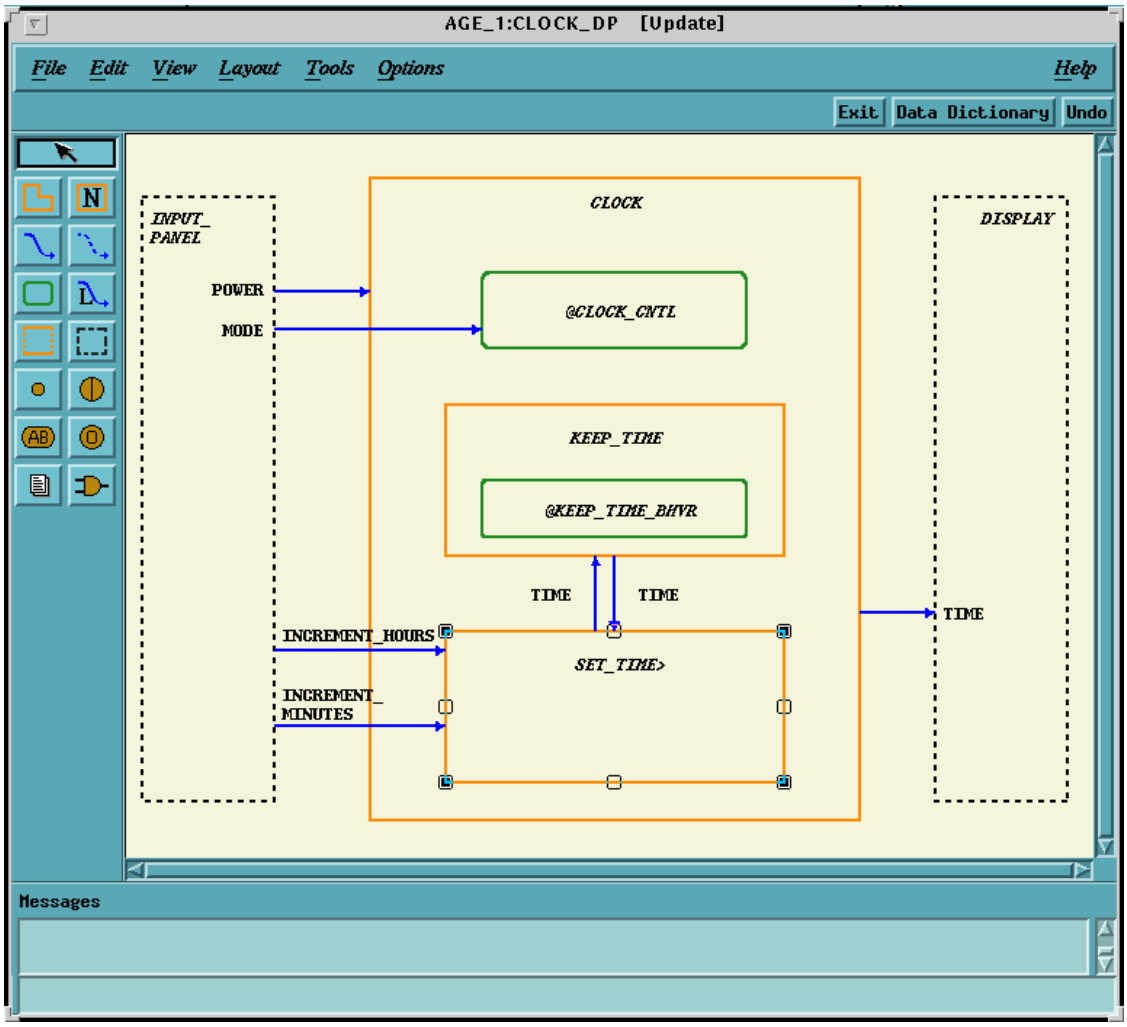
```
INCREMENT_HOURS/if HOURS=12 then
   HOURS:=1;
else
   HOURS:=HOURS+1;
end if;;

INCREMENT_MINUTES/if MINUTES=59 then
   MINUTES:=0;
else
   MINUTES:=MINUTES+1;
```

```
end if
```

9. Select *Save > File Exit*.

   Result: A carrot (>) is added following the element SET_TIME. This signifies that information has been entered into the Data Dictionary for this element.



You can now simulate the model and generate code using this new methods and verify that it works that same as the previous model.