**The lecture 14**

An event is a Stateflow object that triggers actions in a state machine or its environment. Stateflow defines a set of events that typically occur whenever a state machine executes. You can define other types of events that occur only during execution of a specific state machine or its environment.

To define an event:

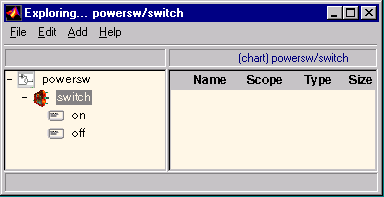
1Add a default definition of the event to the Stateflow data dictionary.

2Set the new event’s properties to values that reflect its intended usage.

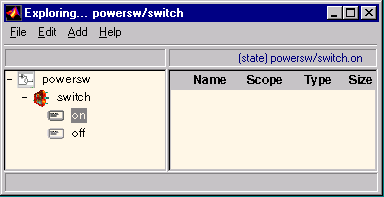
**Using the Explorer**

To use the Stateflow Explorer to define an event:

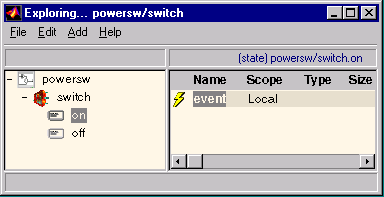
1. Select Explorefrom the Stateflow editor’s Toolsmenu. Stateflow opens the Explorer.



1. Select the object (state machine, chart, or state) in the Explorer’s object hierarchy pane where you want the new event to be visible.



1. Select **Event** from the Explorer’s **Add** menu. Stateflow adds a default definition for the new event in the data dictionary and displays an entry for the new event in the Explorer’s content pane.



**Defining Local Events**

A local event is an event that can occur anywhere in a state machine but is visible only in its parent (and its parent’s descendants). To define an event as local, set its Scope property to Local.

**Defining Input Events**

An input event occurs outside of a chart and is visible only in that chart. This type of event allows other Simulink blocks, including other Stateflow blocks, to notify a particular chart of events that occur outside it. To define an event as an input event, set its Scope property to Input from Simulink.

You can define multiple input events for a chart. The first time you define an input event for a chart, Stateflow adds a trigger port to the chart’s block.

External blocks can trigger the chart’s input events via a signal or vector of signals connected to the chart’s trigger port by associating input events with control signals. When defining input events for a chart, you must specify how control signals connected to the chart trigger the input events.

**Defining Output Events**

An output event is an event that occurs in a specific chart and is visible in specific blocks outside the chart. This type of event allows a chart to notify other blocks in a model of events that occur in the chart. To define an event as an output event, set its Scope property to Output to Simulink. You can define multiple output events for a given chart. Stateflow creates a chart output port for each output event that you define. Your model can use the output ports to trigger the output events in other Simulink blocks in the same model.

**Specifying Trigger Types**

A trigger type defines how control signals trigger input and output events associated with a chart. Trigger types fall into two categories: function call and edge. The basic difference between these two types is when receiving blocks are notified of their occurrence. Receiving blocks are notified of edge-triggered events only at the beginning of the next simulation time step, regardless of when the events occurred during the previous time step. By contrast, receiving blocks are notified of function-call-triggered events the moment the events occur, even if they occur in mid-step.

You set a chart’s trigger type by setting the Trigger property of any of the input or output events defined for the chart. If you want a chart to notify other blocks the moment an output event occurs, set the Trigger property of the output event to Function Call. The output event’s trigger type must be Either Edge. If a chart is connected to a block that outputs function-call events, you must specify the Trigger property of the receiving chart’s input events to Function Call, Stateflow changes all of the chart’s other input events to Function Call. If it is not critical that blocks be notified of events the moment they occur, you can define the events as edge-triggered. You can specify any of the falling types of edge triggers:

**Rising Edge.** A rising level on the control signal triggers the corresponding event.

**Falling Edge.** A falling level on the control signal triggers the event.

**Either Edge.** A change in the signal level triggers the event.

In all cases, the signal must cross 0 to constitute a valid trigger. For example, a change from -1 to 1 constitutes a valid rising edge, but not a change from 1 to 2.

**Defining Data**

A state machine can store and retrieve data that resides internally in its own workspace. It can also access data that resides externally in the Simulink model or application that embeds the state machine. When creating a Stateflow model, you must define any internal or external data referenced by the state machine’s actions.

To define an item of data:

**1** Add the item to the data dictionary.

**2** Set the new item’s properties.

**Adding Data to the Data Dictionary**

You can use either the Stateflow editor or Explorer to add data that is accessible only in a specific chart. You must use the Stateflow Explorer to add data that is accessible everywhere in a state machine or only in a specific state.

**Using the Stateflow Editor**

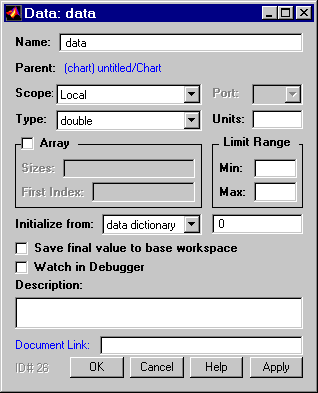
To use the Stateflow editor to add data:

1Select the data’s scope from the **Data** submenu of the Stateflow editor’s **Add** menu.Stateflow adds a default definition of the new item to the Stateflow datadictionary and displays a **Data** dialog that displays the new item’s defaultproperties.

2Use the **Data** dialog box to set the new item’s properties to reflect its intended usage.

**Data Dialog Box**

The Data dialog box allows you to set the properties of a dialog item.



**Name**

Name of the data item. A data name can be of any length and can consist of any alphanumeric and special character combination, with the exception of embedded spaces. The name cannot begin with a numeric character.

**Parent**

Parent of this data item. The parent determines the objects that can access it. Specifically, only the item’s parent and descendants of that parent can access the item. You specify the parent of a data item when you add the item to the data dictionary.

**Scope**

Scope of this data item. A data object’s scope specifies where it resides in memory relative to its parent. These are the options for the Scope property:

* **Local.** A local data object resides and is accessible only in a machine, chart, or state.
* **Input from Simulink.** This is a data item that is accessible in a Simulink chart block but resides in another Simulink block that may or may not be a chart block. The receiving chart block reads the value of the data item from an input port associated with the data item.
* **Output to Simulink.** This is a data item that resides in a chart block and is accessible in another block that may or may not be a chart block. The chart block outputs the value of the datumto an output port associated with the data item.
* **Type.** Data type of this data item, e.g., integer, double, etc.
* **Port.** Index of the port associated with this data item. This control applies only to input and output data.
* **Units.** Units, e.g., inches, centimeters, etc., represented by this data item. The value of this field is stored with the item in the state machine’s data dictionary.

**Defining Input Data**

Stateflow allows amodel to supply data to a chart via input ports on the chart’s block. Such data is called input data. To define an item of input data, add a default item to the Stateflow data dictionary as a child of the chart that will input the data. Set the new item’s Scope to **Input from Simulink**. Stateflow adds an input port to a chart for each item of input data that you define for the chart.

**Defining Output Data**

Output data is data that a chart supplies to other blocks via its output ports. To define an item of output data, add a default data item to the data dictionary as a child of the chart that supplies the item. Then, set the new item’s Scope property to Output to Simulink. Stateflow adds an output port to the chart for each item that it outputs.

