Software through Pictures®
Unified Modeling Language

Millennium Edition

A Quick Tour of StP/UML

UD/UG/ST0000-10136/002

Aonix

Software through Pictures Unified Modeling Language
A Quick Tour of StP/UML
Millennium Edition
March 2002

Aonix® reserves the right to make changes in the specifications and other information contained in this publication without prior notice. In case of doubt, the reader should consult Aonix to determine whether any such changes have been made. 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.

World Headquarters
5040 Shoreham Place, Suite 100
San Diego, CA 92122
Phone: (800) 97-AONIX
Fax: (858) 824-0212
E-mail: info@aonix.com
http://www.aonix.com

Table of Contents

Welcome to Software through Pictures

Welcome to the Software through Pictures / Unified Modeling Language (StP / UML) guided tour. This brief tour takes you through an example StP system created in StP / UML, illustrating key StP features and comprehensive support of UML, including some of StP / UML's most powerful new Millennium Edition capabilities.

For the purposes of this tour, we assume that you have a copy of StP installed on your computer as well as a valid StP license installed. For installation information, please refer to *Installing StP for <platform>*.

For additional information, please consult the online documentation, visit our web site located at http: / / www.aonix.com, or call 1-800-97AONIX.

About StP/UML, Millennium Edition

Software through Pictures / Unified Modeling Language, Millennium Edition (StP / UML ME) provides system architects, analysts, designers, and developers with a powerful visual modeling and deployment environment, including tools to model, design, implement, and maintain your next-generation systems and applications.

It comprises product-specific modeling, reporting, and implementation applications that share a common architecture ("StP Core") built around a central database repository. StP supports the following UML diagram types:

Use case
Class
Sequence
Collaboration
State
Activity
Component
Deployment

as well class tables and state tables. For information on the latest features of StP, see the *Features Supplement*.

Software through Pictures Core

The common architecture, "StP core," provides a basic set of features and services that form the foundation for all product user-interface elements, including the StP desktop, product-specific diagram and table editors, and the StP core editors:

- Object annotation editor (OAE)—Allows you to add information to a model in the form of annotations containing specific values or textual descriptions
- Requirements table editor—Allows you to specify requirements for your project and allocate them to elements in your model

StP is built upon a customizable and extensible set of product templates and files that define and control the user interface and the behavior of StP.

StP Applications

StP applications include method-specific modeling and implementation tools based upon the common StP architecture.

StP/UML, the focus of this tour, is an StP application comprising UML-specific editors that allow you to create software models using the methods and notations specific to the Unified Modeling Language, as defined in the OMG UML 1.3 standards.

This tour covers some, but not all of the StP/UML Millennium Edition new features. For more information, see the *Features Supplement*.

Starting Software through Pictures

You are ready to start StP and begin exploring the example system, *uml_email*, which models an electronic mail system application.

Starting StP from UNIX

Before starting StP, you must source the appropriate *setup-stp.<shell_type>* shell script in the StP installation directory. For example, in a C-shell, source the *setup-stp.csh* file. Alternatively, enter the source command in your *.login* or *.cshrc* file so that it executes automatically, or create an alias containing the source command and execute the alias before starting StP. For more information about the UNIX *setup-stp* file, see *StP Administration*.

To start StP from the UNIX command line:

1. Source the *setup-stp* script, as needed.
2. Type:

   ```
   stp -product uml &
   ```

3. When the **Open System** dialog appears, double-click *uml_email* (or select it and click **Open**).

The StP desktop appears, with the name of the opened system in the desktop title bar. If you have problems starting StP, refer to *Installing StP for UNIX Platforms.*

Starting StP from Windows

To start StP from Windows:

1. From the Windows **Start** menu, choose **Programs** > **Aonix Software through Pictures** <*version*> > **StP/UML**.
2. When the **Open System** dialog appears, double click *uml_email* (or select it and click **Open**).

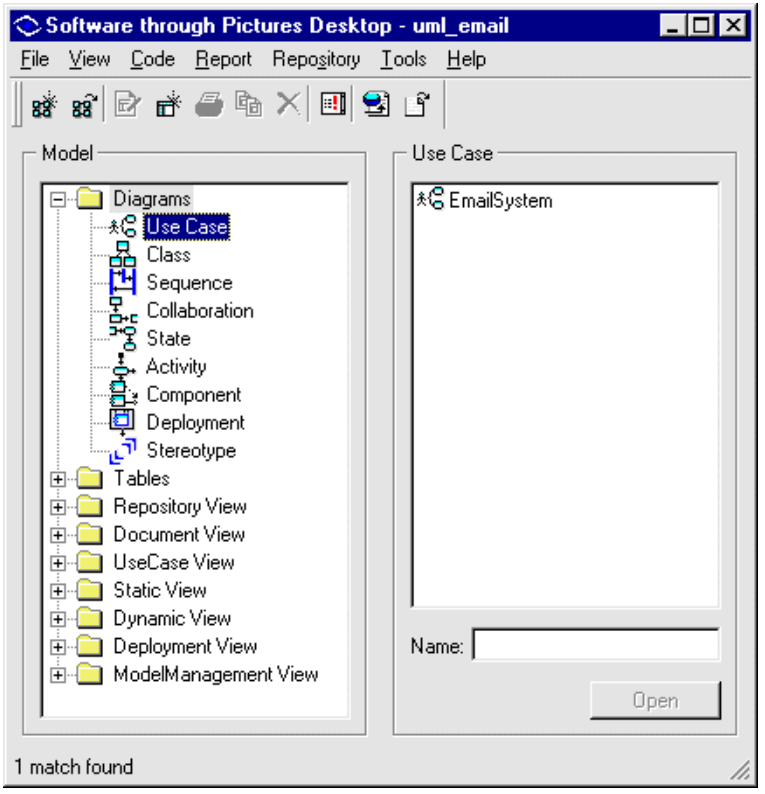
The StP desktop appears, with the name of the opened system in the desktop title bar. If you have problems starting StP, refer to *Installing StP for Windows Platforms*.

Opening a Diagram

From the StP desktop (see below), you can perform operations such as start editors, open diagrams, and generate code and documentation. For detailed information on desktop functions and operations, and on how to open, change, and create a system, refer to *Fundamentals of StP*.

To open a diagram from the example system *uml_email*:

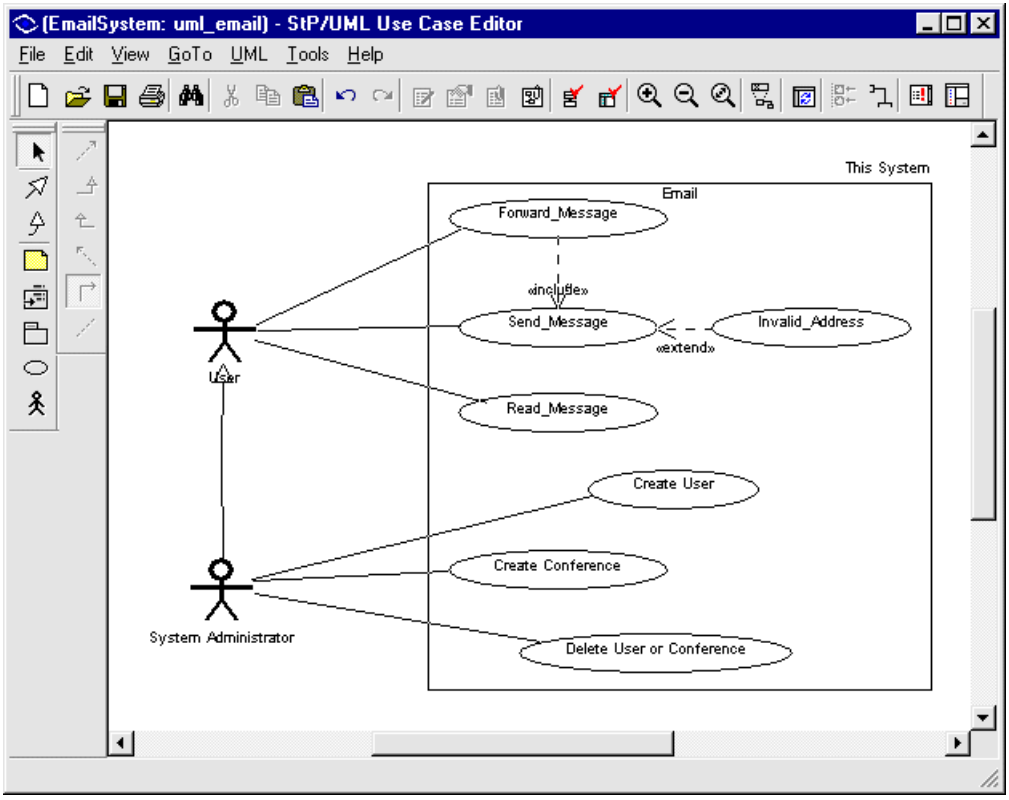
1. In the model pane (pane on the left side of the StP desktop), open the **Diagrams** category to see the StP / UML diagram types.



2. For this example, select the diagram type **Use Case**.

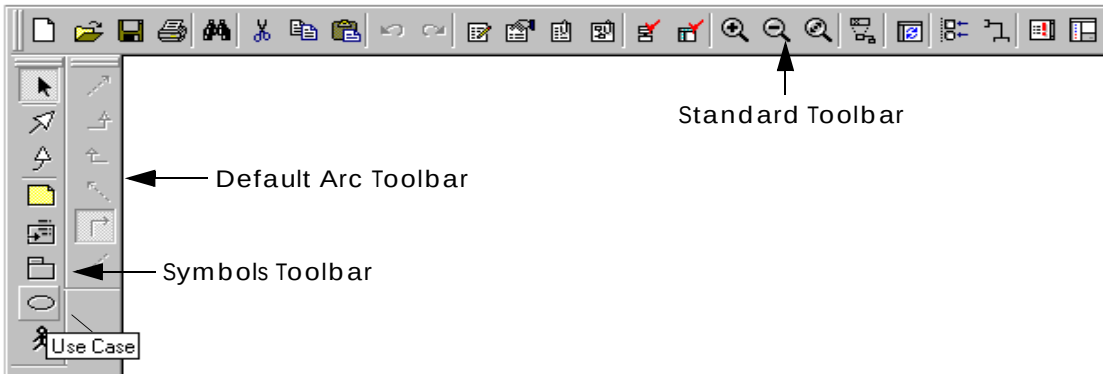
   A list of use case diagrams for the *uml_email* example system appears in the objects pane to the right of the model pane.

3. In the objects pane, double-click on *EmailSystem*. The use case editor opens.



Note the various areas of the editor window. Under the title bar is the set of menu functions (File, Edit, etc.). Below that are the standard toolbar icons for the editor; all diagram editors have the same set. Pass your cursor over each icon to see what each icon represents. On the left are icons for drawing diagrams; these vary according to editor.

Below is a screen showing the toolbar setup for the use case editor.



Standard Editing Operations

As you edit the diagrams in the example system, you may make mistakes or encounter instructions that require you to undo an operation or to cut, copy, paste, or delete elements in the diagram.

Undoing and Redoing an Operation

Using the following commands on the editor's **Edit** menu, or their keyboard shortcuts, you can:

- **Undo** (Control + Z) up to five previous operations
- **Redo** (Control + Y) up to five "undo" operations

Cutting, Copying, Pasting, and Deleting

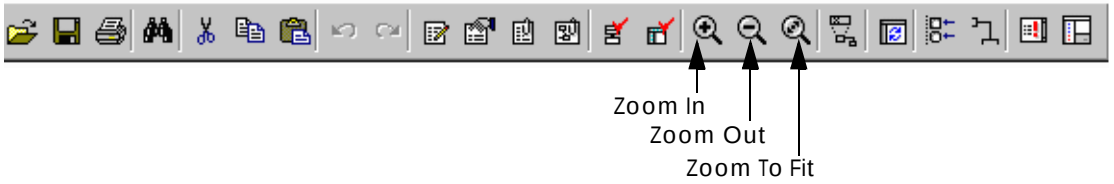
You can also cut, copy, paste, and delete elements in the diagram, using the appropriate commands on the **Edit** menu. Cutting or deleting an element that has links to other elements also removes the links.

Viewing and Aligning Parts of a Diagram

StP provides some helpful tools and commands for viewing and aligning parts of a diagram.

Using the Zoom Toolbar Buttons

With the zoom buttons on the toolbar, you can enlarge, shrink, and center a diagram or a portion of a diagram in the drawing area.



Try it:

1. Select a use case symbol in the use case diagram and click **Zoom To Fit**.

   StP enlarges and centers the selected part of the diagram.
2. To re-center and fit the entire diagram in the drawing area, click anywhere on the blank “canvas” of the drawing area (not on any symbol) and click **Zoom To Fit** again.

You can also resize the diagram by zooming in or out, using the **Zoom In** and **Zoom Out** toolbar buttons.

Using the Diagram Panner

StP provides the diagram panner for viewing very large diagrams and for viewing different areas of a zoom-enlarged diagram:

1. Once again, select a Use Case symbol in the diagram and click **Zoom To Fit** to enlarge a small area of the diagram.

2. From the **View** menu, choose **Show Panner**.

   A small window (the "panner") appears, displaying a miniaturized view of the current diagram, with a rectangular outline called the "bounding box" positioned over the selected symbol.

3. Use the mouse to drag the bounding box around the miniaturized diagram in the panner.

   StP repositions the diagram in the editor's drawing area accordingly.

4. Close the panner (click the **x** in the upper-right corner).

Aligning Objects

StP makes organizing your diagram easy by providing numerous ways to align objects and links automatically:

1. In the use case diagram, select the *Create User* use case and drag it out of position, so that it is no longer aligned with the adjacent use cases.
2. Hold down the Shift key and additionally select the two use cases, *Create Conference* and *Delete User or Conference*.
3. Click the **Align** toolbar button. 

   (Or, choose **Align** from the **Tools** menu.)

4. In the **Align** dialog box, make the following selections and click **OK**:
   - In the **Symbol Spacing** group, select vertical **Equal Distances**
   - In the **Symbol Alignment** group, select **Horiz. Center**

   StP realigns all objects.

   Note: To undo any results you do not like, choose **Undo** from the **Edit** menu, as needed, to restore the diagram to its previous state.

Editing a Use Case Diagram

The use case diagram shows the major capabilities of the system, as seen by the outside world. The large rectangle (with the words “This System” in the outside upper-right corner) represents the entire *Email* system.

Moving and Enlarging Elements in the Diagram

Before editing this diagram, make sure the *Email* system rectangle is large enough to accommodate a few more use cases:

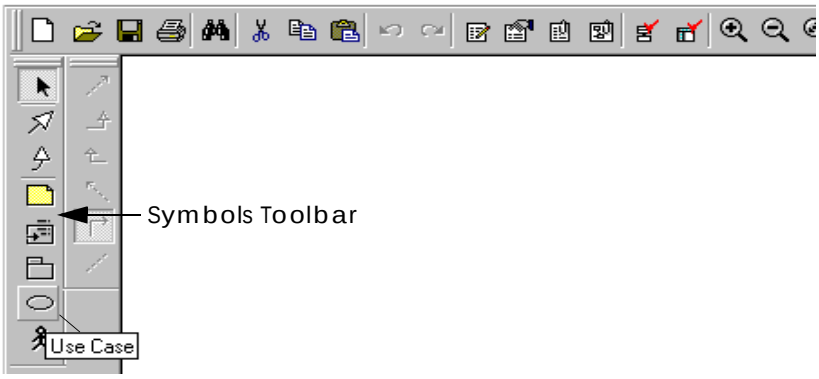
1. Click on the *Email* system symbol to display its “handles”; then drag one of the right-side handles to the right to enlarge the rectangle.

   All use cases should appear totally within the rectangle, and there should be extra room for inserting additional symbols. Actor symbols should remain outside the *Email* system symbol.
2. If needed, reposition any use case or other symbols, using the left mouse button to select and drag the symbol to the desired location.
3. To re-center and re-size the entire diagram, click anywhere in the blank area outside of the *Email* system rectangle, then click the **Zoom To Fit** button on the toolbar.

Adding a Single Use Case

To add another use case to this diagram:

1. With the left mouse button, select the Use Case symbol on the vertical Symbols toolbar to the left of the drawing area.



2. Click the left mouse button somewhere in the lower half of the *Email* system rectangle.

   A new use case symbol appears in the diagram, with an input text box for label entry.

3. In the text box, type a label for the use case and press Enter.

   Note: To create a multi-line label, press Control + Enter to insert a line break within the label.

Inserting Multiple Symbols

To add and label multiple instances of a symbol (for example, a use case):

1. Double-click the Use Case symbol on the Symbols toolbar to enter multiple insert mode.
2. Click the left mouse button in the drawing area repeatedly to create as many use cases as needed, labeling each one as created.

   Alternatively, press Control + Tab in a new use case's empty text label box (or choose **Edit > Choose Use Case Names**), select a name from the **Choose Name** dialog and click **OK**.

3. Click the Selection Tool on the Symbols toolbar to terminate multiple insert mode.

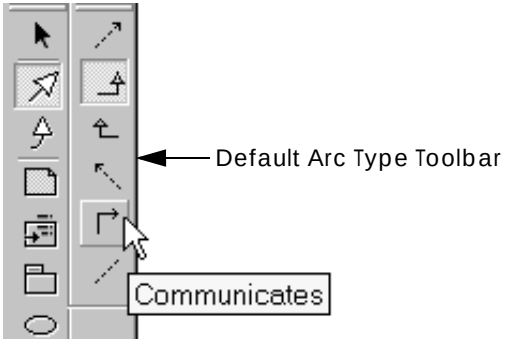
Note: To edit a completed label, select the symbol and press F2 (or double-click it) to display its text box, retype or edit the label and press Enter.

Linking Elements

To link the new use case to other elements in the diagram:

1. Select the Arc symbol on the Symbols toolbar.

2. Select the Communicates arc type on the Default Arc Type toolbar:



3. Click the *System Administrator* actor symbol in the diagram to attach one end of the arc.

4. Move the pointer to the new use case symbol and click again to attach the other end of the arc.
5. To draw multiple links without re-selecting the arc symbol each time, double-click the Arc symbol on the Symbols toolbar, draw the arcs, and click the Selection Tool or any other symbol on the Symbols toolbar to terminate multiple insert mode.

   Try it now. Draw an arc from an actor to a use case; then draw an arc between use cases.

   Note: Unless you specifically selected a new arc type from the Default Arc Type toolbar, StP uses either the previously set arc type or a default arc type for the particular link you are drawing.

Changing an Existing Link's Arc Type

To change the arc type of an existing link:

1. Select the use case-to-use case link you just drew in the diagram.
2. From the **Edit** menu choose **Replace**.
3. In the **Replace Arc Type** dialog box, select a different arc type and click **OK**.

The arc type of the selected link in the diagram changes to whatever you chose as the replacement arc type.

Allocating a Requirement

StP's capabilities allow you to allocate requirements to objects in the model and to track requirements that have been allocated in the system.

Let's allocate the requirement for the *CreateUser* use case:

1. In the *EmailSystem* use case diagram, select the *CreateUser* use case.
2. From the **GoTo** menu, choose **Allocate Requirements**.

   A previously created requirements table appears.
3. In the requirements table, select requirement *1.1 CreateUser*.

4. From the requirements table editor's **Tools** menu, choose **Allocate** > **Allocate**.

To check the allocated requirement:

1. In the requirements table, select *1.1 CreateUser* again.
2. From the **Tools** menu, choose **Query** > **Objects That Satisfy Selected Requirements**.
3. In the **Object Selector** dialog, select *UmlUseCase 'CreateUser'* and click **OK**.

   The use case to which the requirement has been allocated, in this case *CreateUser*, appears selected in the use case diagram.

To exit and optionally save the requirements table, choose **Exit** from the requirements table editor's **File** menu. A confirmation dialog asks if you want to save the table before exiting.

Modeling a Scenario in an Interaction Diagram

Given a particular task represented by a use case, a variety of scenarios could occur that the system must be able to handle. For example, the *Send_Message* task needs to handle scenarios in which a user attempts to send a message to either an invalid user or a valid user. Each of the different scenarios can be modeled in an interaction diagram.

There are two types of interaction diagrams:

- Sequence diagrams, which emphasize the order of message passing between interacting objects
- Collaboration diagrams, which emphasize the structure of the interaction

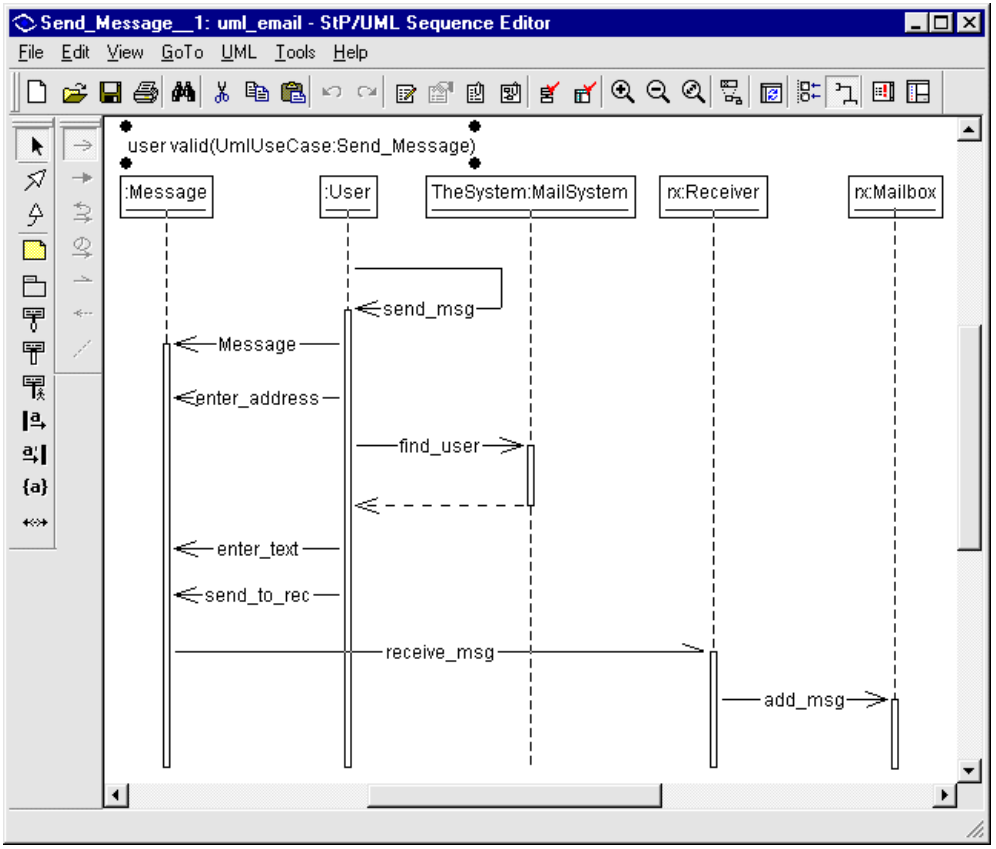
Editing a Scenario in a Sequence Diagram

In this tour, we'll explore the use of the sequence editor for modeling the scenario of sending a message to a valid user:

1. In the use case diagram, select the *Send_Message* use case.

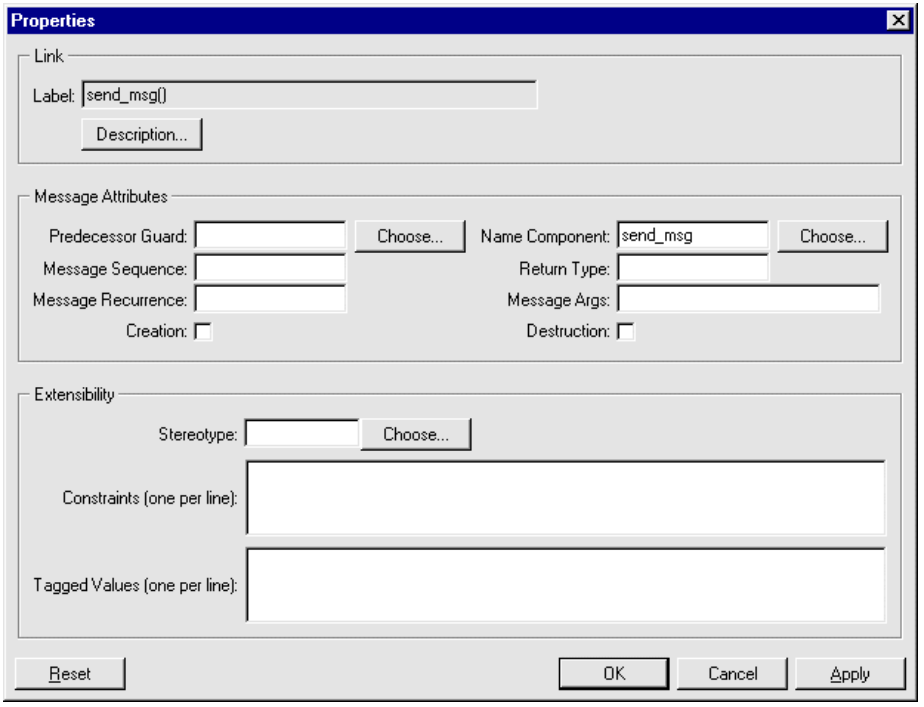
2. From the **GoTo** menu, choose **Scenario in Sequence Diagram**.
3. In the **Object Selector** dialog, double-click *uservalid* or select it and click **OK**. (You can also create a new scenario here by clicking on **<<Create New Scenario for Use Case>>**.)

   The *user valid(UMLUseCase:Send_Message)* scenario appears in the *Send_Message_1* sequence diagram in the sequence editor.



4. In the sequence diagram, select the *send_msg* message and choose **Properties** from the main menu or from the shortcut (right-click) menu. Or, you can click the **Properties** toolbar button.

5. Examine the properties sheet that appears and experiment with entering more detailed information for the selected message.



6. Click **OK** on the property sheet when done.
7. Click the **Refresh Display Marks** toolbar button (run your cursor over the buttons to see which one it is).

   Note: Some display marks may be turned off. To set the visibility of display marks, choose **Tools > Options** and select the **Display Marks** option or tab at the top of the dialog box that appears. In the dialog, set the display marks as desired and click **OK**.

8. Save the diagram (**File > Save**).

Viewing the Scenario in a Collaboration Diagram

To navigate from the scenario modeled in a sequence diagram to its sibling scenario in a collaboration diagram:

1. Select the *user valid (UMLUseCase:Send_Message)* scenario instance at the top of the sequence diagram.
2. From the **Go To** menu, choose **Sibling Use Case Scenario**.
3. In the **Object Selector** dialog that appears, select the *uservalid* UMLScenarioInstance in the collaboration diagram and click **OK**.

The collaboration diagram for the *uservalid* UMLScenarioInstance appears.

Note: Recall that you used a sequence diagram to model the details of the Send_Message use case ("Editing a Scenario in a Sequence Diagram" on page 19). You could also have used a collaboration diagram to do it. The use of the collaboration editor is similar to that of the sequence editor.

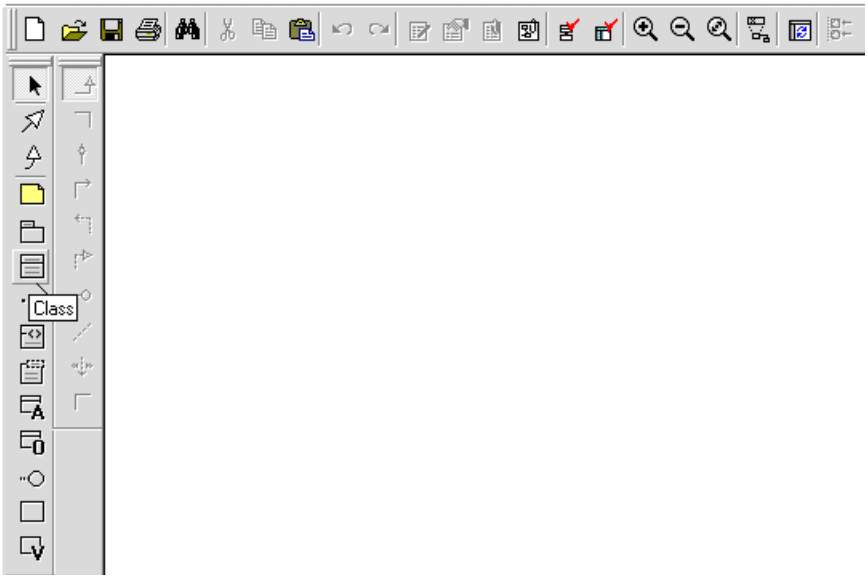
Using the Class Diagram and Table Editors

The class diagram shows the static relationship between the objects and / or classes in the system. The class table provides additional information about the classes in the class diagram.

Navigating to a Class Diagram

You can open a new class diagram by clicking on **Class** and right-clicking on **New** in the shortcut menu. You can open an existing class diagram from the StP desktop by clicking **Class** in the model pane and then

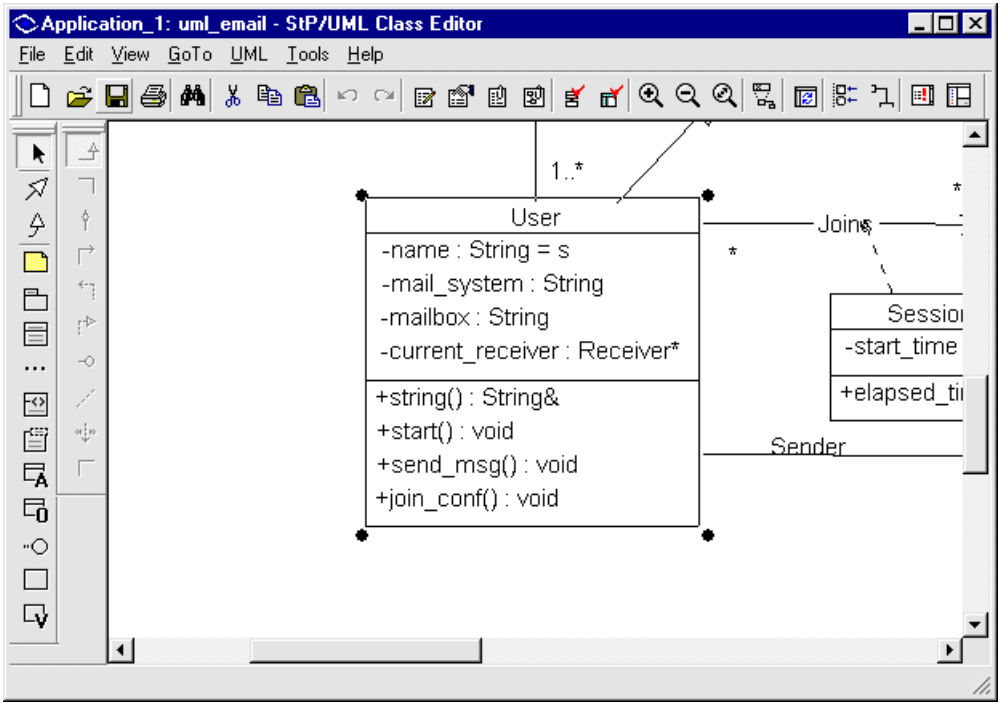
selecting the class from the objects pane. You create a new class by selecting the Class symbol from the Symbols toolbar and pasting it into the drawing area.



Here, we will be opening an existing class diagram by navigating to the diagram from the collaboration diagram opened in the previous step.

To navigate to a class diagram from the collaboration diagram:

1. In the collaboration diagram, select the *User* object in the drawing area.
2. From the **GoTo** menu, choose **Class Diagram Where Class Is Referenced**



Part of the class diagram appears in the editor's drawing area, with the *User* class selected. The rest of the diagram is not currently visible.

3. Click anywhere in a blank area of the drawing "canvas" (not on any object) and click the **Zoom To Fit** toolbar button to re-center and re-size the diagram.

Auto Drawing Classes

StP will automatically draw and construct classes for you from data already available in the repository.

1. Select the *Message* class symbol.

2. From the **UML** menu, choose **Auto Draw > Draw Associated Class(es) > into new diagram**.

   StP opens a new diagram and draws and labels class symbols for the *Message* class and its associated classes. No attributes or operations are shown.
3. Select the *Message* class in the new diagram.
4. From the **UML** menu, choose **Attributes and Operations > Construct Class from All Definitions**.

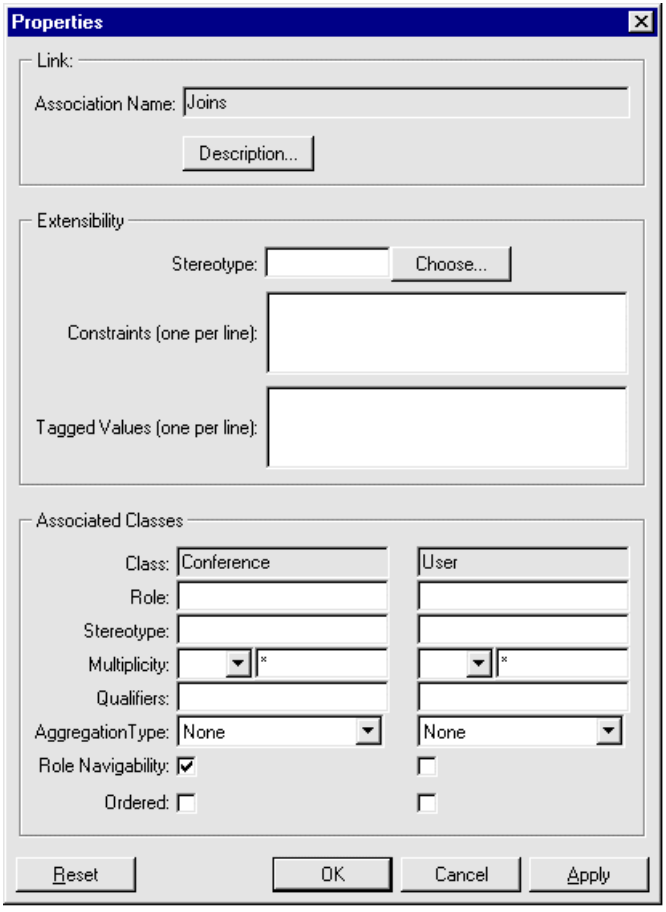
   StP reconstructs the *Message* class, with all of its members, from the relevant information in the repository.
5. Construct the *Mailbox* class from all definitions, repeating the instructions in the previous step.
6. Rearrange the objects as desired.

Note: If you need to resize a class to include its attributes and operations and are unable to drag a corner, select the class, choose **Current Symbol Options** from the shortcut (right click) menu, then make sure **Shrink around label** is turned off. Then resize the class symbol as needed.

Editing Role Names, Multiplicity, and Navigability

To enter or edit role names, choose predefined multiplicities, and represent bidirectional navigability of an association:

1. On the class diagram, select an association.
2. Display the association's property sheet, using the **Properties** toolbar button or the **Properties** command on the **Edit** or shortcut menu.



3. In the **Associated Classes** group, enter or edit role names in the **Role** fields, as desired.
4. Display the options list in a **Multiplicity** field for an associated class and either:
   - Select a predefined multiplicity from the list.
   - Manually type any allowable entry in the adjacent text entry field on the property sheet.
5. Set the **Role Navigability** options as desired:
   - For unidirectional navigability, clear the **Role Navigability** option for one of the association roles.

- To reestablish bidirectional navigability, set or unset role navigability on both ends of the association.

Note: By default, bidirectional associations have no arrowheads, but you can set a variable to use double arrowheads if you prefer (see the section on role navigability in the *Features Supplement*).

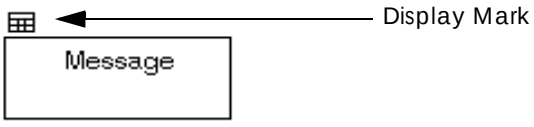
6. Click **OK**.

Viewing More Information About Classes

Various display marks appear on the diagram, each representing additional information, usually in the form of an assigned property or annotation for that object.

For example:

- Integers that compose a multiplicity display mark on an association arc represent the lower and upper bounds of the multiplicity.
- A table icon display mark indicates that a class table containing more information exists for that class object.



To view the contents of a class table:

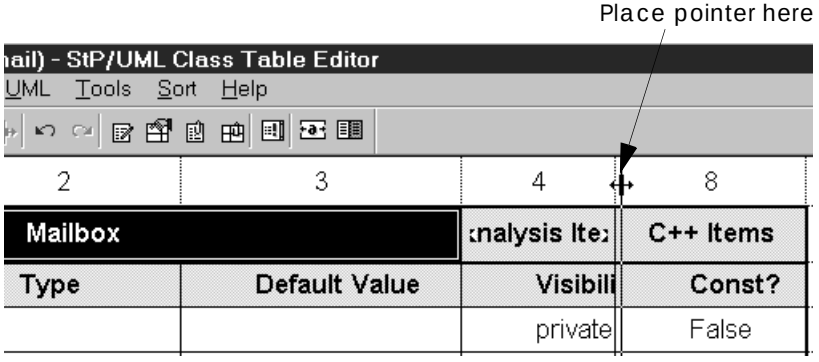
1. Select the *Mailbox* class and choose **Class Table** from the **GoTo** menu.

   A class table appears. It contains language-specific details of the class.
2. To widen one or more columns to fit the existing labels, select the column(s) by clicking on their column numbers (for example: columns 1, 2, 3) in the top row, while holding down the Shift key (for multiple columns); then do either of these:
   - From the **Table** menu choose **Adjust Scale To Fit Label.**
   - Click the **Fit Cell to Label** toolbar button.

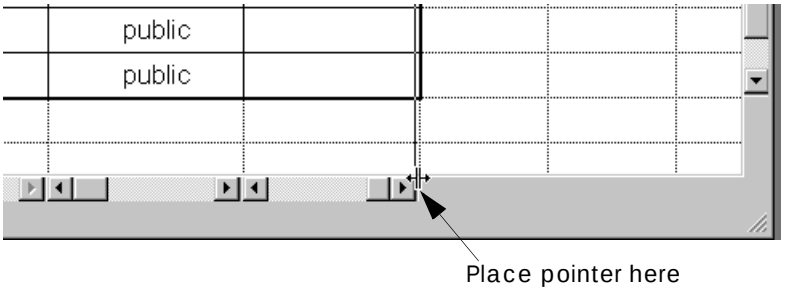
3. Drag the right side or lower right corner of the table editor to the right edge of your display to enlarge the editor window.
4. From the **View** menu, choose **Hide/Show**.

   The **Hide/Show Sections** dialog shows which rows and columns are currently accessible in the table (those that can be viewed directly or by scrolling).
5. In the **Hide/Show Sections** dialog, from the **Vertical Sections** pane, select *Row Numbering*, press Shift and select *C++ Implementation Items* (the first four entries should be highlighted); then click **OK**.

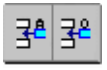
   A new vertical section (**Analysis Items**) appears in column 4.
6. To widen a single column, place the pointer on the column's right border in the top (0) row (the curser changes shape); then press the left mouse button and drag the column border to the right.



7. To make more columns visible in the **C++ Items** section, position the pointer as shown in the scroll bar below column 8, press the left mouse button and drag the column divider to the right, across two or three column divisions (it will "snap" from one divider to the next).



8. When done examining the table, choose **Exit** from the class table editor's **File** menu.

You can do many other things in the class table editor. One of the more important is adding attributes and operations to the class. To do this, use either the Add New Attribute or Add New Operation toolbar icons (left or right, respectively). 

Using the State Editor

The state diagram shows the potential states of an object and the events that transform the object from one state to another.

Navigating to a State Diagram

To examine a state diagram, we'll navigate to it from the class diagram:

1. On the class diagram, select the *Message* class.

2. From the **GoTo** menu, choose **State Diagram for Class**.

   The state diagram for the *Message* class appears.

   Note: You can browse through any diagram by positioning the pointer on various objects. The status area (lower-left corner of the editor window) reports the type of object the pointer is on.

Modeling Substates

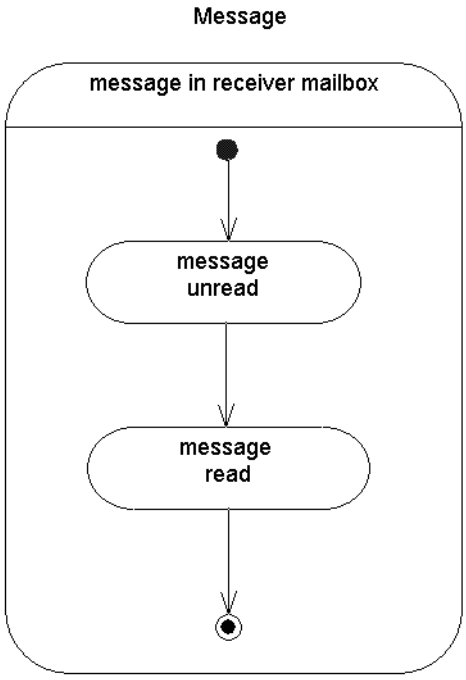
To model possible substates within a state, you can create a lower-level state decomposition diagram:

1. Select the *message in receiver mailbox* state.
2. From the **Go To** menu, choose **Refine**.
3. In the **Confirmation** dialog, click **OK**.

   StP creates a decomposition state diagram containing a composite state symbol, labeled *message in receiver mailbox*.
4. Drag the top or bottom handle on the composite state symbol in the drawing area to enlarge it.
5. On the Symbols toolbar, double-click the state symbol; then click inside the composite state symbol in the drawing area to insert and label the following substates:
   - *message unread*
   - *message read*
6. From the Symbols toolbar, select, insert, and optionally label initial and final state symbols within the composite state symbol.
7. Double-click the Arc symbol on the Symbols toolbar to begin multiple insert mode.
8. Select the State Transition Link arc type on the Default Arc Type toolbar.
9. Draw arcs between the objects in the diagram, clicking on each source and destination object in turn to draw the arc between them.

   Note: Terminate multiple insert mode by clicking the Selection Tool on the Symbols toolbar.

   Your decomposition diagram should look something like this:



10. Select the transition link between the initial state and the *message unread* substate and press F2 (or double-click the link) to select its label.
11. Type **`Request delivery`** and then press Control + Tab.

    The transition link's properties sheet appears, displaying the event portion of the label you just typed in the **Event** field.
12. To select an existing action from the repository, click **Choose** adjacent to the **Action List** field, select *send_to_rec ()* and click **OK** on the **Choose Action** dialog that appears; then click **OK** on the properties sheet.

    The label on the transition link in the diagram now appears as:

    `Request delivery/send_to_rec`

Creating and Managing Subsystems

StP model management, an StP/UML feature, supports distributed development of large, complex projects through subsystem partitioning and management. Each subsystem can be exported into a private workspace or configuration management system for independent development. The results can then be seamlessly integrated into the main model.

Defining a Subsystem

Partitioning into subsystems can be done at any time during the development process.

Defining a subsystem requires three major steps:

- Creating the subsystem in a use case or class diagram
- Drawing model elements in the subsystem
- Manually assigning diagrams that reference elements of a given subsystem to that subsystem

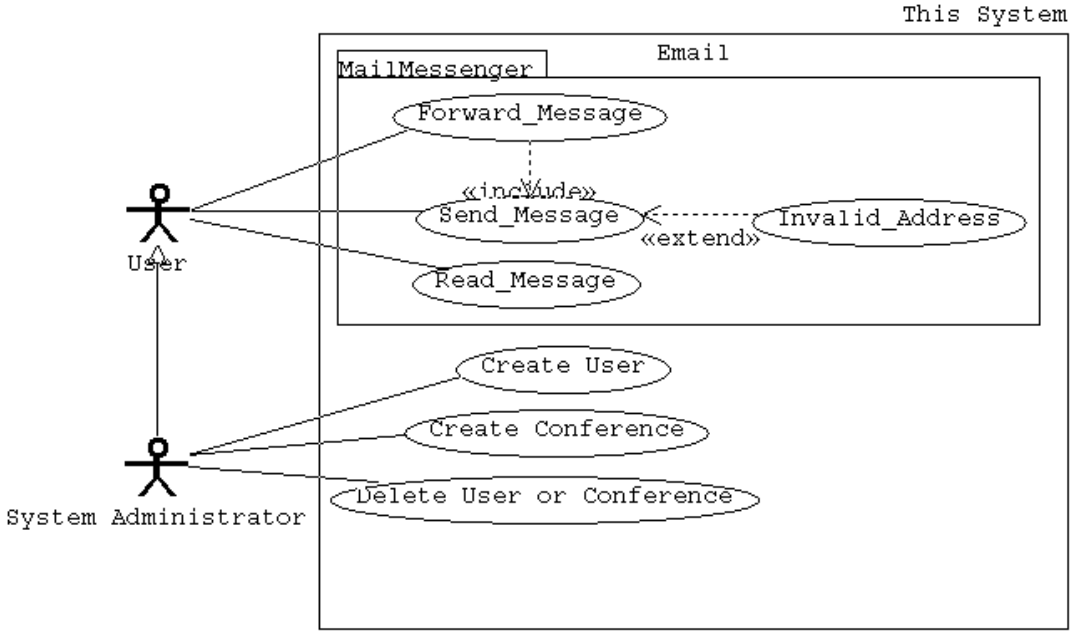
Drawing a Subsystem and its Elements

As an example, create a subsystem in the *EmailSystem* use case diagram:

1. From the desktop, click on **Use Case** and select *EmailSystem* from the objects pane.
2. Select a package symbol from the Symbols toolbar and insert it within the boundaries of the *Email* system rectangle, labeling it with the subsystem name, *MailMessenger.*
3. Enlarge the package symbol by dragging one of its corner handles and position the package so that it encompasses the following use cases:
   - *Forward_Message*
   - *Send_Message*
   - *Invalid_Address*

- *Read_Message*

If necessary, click on and drag use case symbols into or out of the package symbol. Your diagram should look something like this:



4. Select the *MailMessenger* package symbol and display its property sheet, using either **Edit > Properties** or the **Properties** toolbar button.
5. In the **Properties** sheet, click the **Choose** button adjacent to the **Stereotype** field.
6. From the pick list of predefined stereotypes that appears, select **subsystem** and click **OK** on the pick list dialog.
7. Click **OK** on the package symbol's **Properties** sheet.

8. Click the **Refresh Display Marks** toolbar button.

The *«subsystem»* stereotype display mark appears above the package symbol, in the diagram, identifying it as a subsystem.

Note: Other display marks may also appear on the diagram. For example, the rectangle display mark on the *Send_Message* use case indicates that a scenario for this use case exists in another diagram.

9. From the **File** menu, choose **Save**.

Viewing the Subsystem

Use the **ModelManagement View** category on the StP/UML desktop to examine your model's subsystem partitions and elements:

1. Re-display the StP/UML desktop (from the Use Case Editor's **View** menu, choose **Show Desktop**).
2. With the pointer on the **ModelManagement View** (last) category in the model pane, right-click the mouse and choose **Refresh tree** from the shortcut menu.
3. Open the **ModelManagement View** category by clicking its plus sign (+).

   At this point, the **ModelManagement View** for the example system contains only two entries—the *MailMessenger* subsystem you created and a folder labeled *Unassigned Diagrams*.
4. Click on the *MailMessenger* subsystem's plus sign (+) to display its contents.

   This level of the subsystem view should display the four use cases contained within the *MailMessenger* package symbol you created in the *EmailSystem* use case diagram.

5. Click on the plus sign (+) in front of the *Send_Message* entry in the *MailMessenger* subsystem view.

   Two instances of the *uservalid* scenario appear under *Send_Message* in the **Model Management View**.

   They were assigned by default to the same subsystem as their parent use case, but can be reassigned manually to another subsystem.

6. Select the first *uservalid* entry under *Send_Message*.

   The name of the diagram in which it occurs, *Send_Message_1*, appears in the Diagram References (right) pane.

Assigning Diagrams to a Subsystem

Elements belonging to a particular subsystem are referenced in various diagrams. You control which diagrams are assigned to a given subsystem. For example:

1. Select the *MailMessenger* subsystem in the **ModelManagement View**.

   Notice that the Assigned Diagrams (right) pane is empty.

2. Open the **Unassigned Diagrams** folder in the **Model Management View** and select the **Use Case** entry.
3. In the Unassigned Sequence Diagrams (right) pane, point to *EmailSystem*, right-click the mouse, and choose **Assign Files to Subsystem** from the shortcut menu.
4. In the **Assign File to Subsystem** dialog box that appears, select the *MailMessenger* subsystem and click **OK**.
5. Select the *MailMessenger* subsystem in the **ModelManagement View** again.

   Notice that the newly assigned diagram, *EmailSystem*, now appears in the Assigned Diagrams (right) pane.

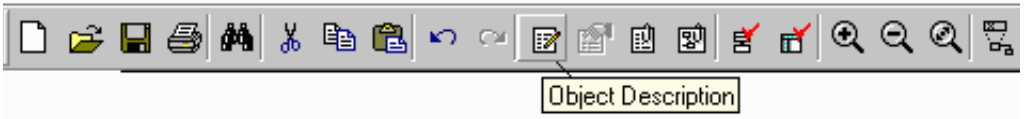
Adding Descriptions and External File Links

You can add a textual description to any object through the object annotation editor or by using StP's **Object Description** toolbar shortcut. You can also link an external file to an object.

Adding Object Descriptions Quickly

All StP/UML diagram editors and the class table editor provide quick and easy access to a simple text entry window or external editor in which you can enter a description for a selected object.

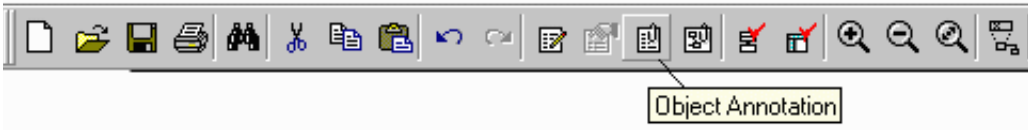
1. Select any State object in the diagram.
2. Click the Object Description toolbar button.



3. In the **Object Description** dialog (or supported external text editor, if one has been set up as the StP default editor), enter your description and click **OK**.

   The description is saved in an object annotation note.

   To display the annotation note, with the state object still selected, click the Object Annotation toolbar button.



In the object annotation editor (OAE) open the annotation folder and select the Object annotation. Your text appears in the Description text box.

Note: You can also store default descriptions that are loaded automatically when no other description has been entered for an object. For more information, see the section on OAE enhancements in Chapter 6 of the *Features Supplement*.

Linking External Files to StP Objects

With StP ME, you can link external files to an StP element. To try this, first create some text in an StP-supported external file editor or word processor.

To link the external file to an object:

1. Select an object, such as a state, in the state diagram.
2. Click the **Object Annotation** toolbar button.
3. In the object annotation editor (OAE), open the annotation folder and select the *Object* note.
4. From the **Edit** menu, choose **Add Item > External File**.

   An *External File* item appears beneath the *Object* note.
5. Right-click the mouse on the *External File* item and choose **Browse** from the shortcut menu that appears.
6. In the **Browse** dialog, locate the external file and click **Open**.
7. From the OAE **File** menu, choose **Save**.

To view the contents of the external file:

1. With the object selected in the diagram, click the **Object Annotation** toolbar button.
2. In the OAE, open the annotation folder and the *Object* note.
3. Right-click the *External File* item and choose **Open** from the shortcut menu that appears.

   Whenever possible, StP automatically opens the file in the correct application, based on the file extension.

Checking Syntax and Semantics

You can apply the following types of checks to your entire model or to individual diagrams:

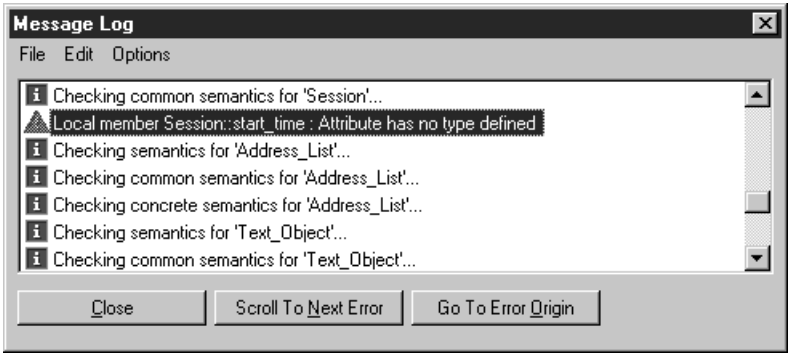
- Syntax check—Verifies that all diagrams are syntactically complete and correctly drawn; that is, all objects are properly labeled and all connection rules have been followed.
- Semantic check—Verifies that all objects are properly defined in the repository according to methodological constraints, and all elements are properly balanced.

In this tour, you will apply syntax and semantic checks to a class diagram from within the class editor:

1. Return to the class diagram named *Application_1* (by navigating from the *Message* state machine, reopening it from the StP desktop, or reactivating it, as needed).
2. From the class editor's **Tools** menu, choose **Check Syntax**. Or, click the Check Syntax toolbar icon. 

   A message in the editor window's status area (lower left) verifies that there are no syntax errors in the diagram.
3. From the class editor's **Tools** menu, choose **Check Semantics**. Or, click the Check Semantics toolbar icon. 

   A semantic check ensures consistency across the entire model of the modeling constructs present in the diagram being checked. This check will report some errors, visible in the message log.
4. In the message log, scroll to and select the error message for class *Session*, which is highlighted in the following figure; then click the **Go To Error Origin** button.



The *Session* class appears selected in the class diagram.

To correct the error, we'll define a type for the attribute *start_time*, belonging to class *Session*, in the class table.

5. Close the message log; then from the **GoTo** menu, choose **Class Table** to navigate from the diagram to the selected class's class table.
6. In the class table for *Session*, select the empty cell in the *Type* column for the attribute *start_time*.
7. Right-click the mouse in the selected cell and choose **Choose Names** from the shortcut menu that appears.
8. In the list of names that appears, scroll to and select *int*; then click **OK**.

   The *Session* class table should now look like this:

Session		
Attribute	**Type**	**Default Value**
start_time	int	
Operation	**Arguments**	**Return Type**
elapsed_time		

9. From the **File** menu, choose **Save**.
10. Close the class table and return to the class diagram; then rerun the **Check Semantics** command.

    This time the message log should report one less error.

Printing Diagrams and Reports

StP provides various printing and reporting capabilities, including:

- Printing any single diagram or table directly to a default printer
- Printing a diagram or table to a file that is formatted for viewing or printing in a supported publishing product or web browser
- Generating reports about your model

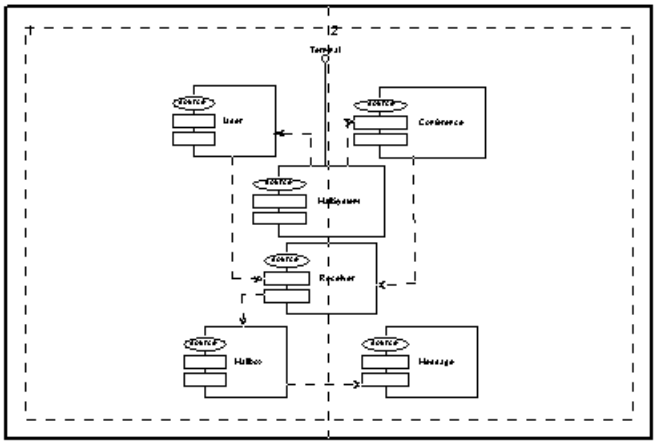
Printing a Diagram to a Default Printer

Let's print one of the component diagrams to your default printer:

1. Navigate to the StP desktop from any diagram by choosing **Show Desktop** from the **View** menu. Or, click the Show Desktop toolbar icon. 
2. Open the **Diagrams** category on the desktop and select **Component**.
3. In the objects pane, select *Application1* and click **Open**.

   The diagram appears in the component editor.
4. From the **View** menu, choose **Page Layout** to see how the diagram will print across multiple pages and where the page breaks will occur.

   Note: Initially, page layout view may not display entire pages. Click the **Zoom To Fit** toolbar button to display a complete view of the page layout for this diagram.



By default, StP chooses a page layout for the best fit, based on default or user-specified values for the following page setup options:

- Page orientation (in this example, the layout spans two portrait pages)
- Alignment
- Diagram scale and pagination grid specifications
- Paper size
- Margins

You can edit these values from the **Page Setup** dialog's **Appearance** or **Paper** options tabs. Let's try it.

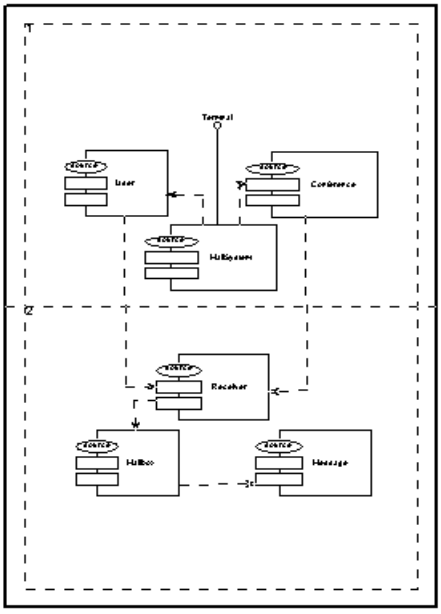
5. Choose **Page Setup** from the **File** menu and select the **Appearance** option or tab at the top of the dialog box; then select the **Landscape** orientation and click **OK**.

   StP adjusts the layout to display the diagram on landscape pages.

6. Reposition the various elements so they do not fall on a page break.

   Note: Use the Selection Tool on the Symbols toolbar to draw a box around several elements, or hold down the Shift key and select multiple elements; then drag them as a unit to reposition them.

StP adjusts the page layout accordingly, as shown here:



7. When satisfied with the layout, click the **Print** toolbar button.

   The **Print** dialog box appears. Do the following:

   - In the **Print range** group, specify which pages of the layout to print.
   - In the **Print What** group choose to print the **Entire diagram** (default) or **Selected elements**.

8. In the **Print** dialog box, click **Print** to print the diagram to your default printer.

Generating a Report

StP / UML allows you to generate various reports for your model. They can be tailored to your specific needs. To explore StP's reporting capabilities, do the following:

1. From the StP desktop, choose **Start Script Manager** from the **Report** menu.

2. In the Script Manager, open the **Product Scripts** category and then the **report** subcategory.

   At this point you can choose one of the following:

   - The **HTML_Report** format
   - One of the **UML_Report** formats: Formats **D** (German), **E** (English), and **F** (French) are available.

3. Select **HTML_Report**.
4. From the **File** menu, choose **Run Script**.
5. In the **Run Script** dialog, select **HTML**.
6. Click **OK**.
7. When the script has finished, close the Script Manager. Then choose **Report > Open HTML Report** from the StP desktop.
8. Select *index.html* from the **Open HTML Report** dialog. Assuming you have a browser available, the index page of the report opens. From there you can navigate to the information you wish to view.

The HTML output contains links to the model's diagrams. The output of **UML_Report** formats is sent to the *qrl_files* directory within your project directory; the output of an **HTML_Report** format is sent to the *html_files* directory within your project directory. Both types of HTML output contain the same basic information about your model, but the HTML report contains more links and provides navigation between diagram elements and their textual description. This navigation is available only when PostScript printing is enabled.

Note: Refer to "Publishing and Printing Enhancements" in Chapter 5 of the *Features Supplement* for information on obtaining a PostScript converter and on setting up your environment for printing.

Generating Code for Your Model

Once you finish the previous exercises, you might want to generate code for your model. Normally you would not attempt to generate code until late in project development, when you've worked out all the problems in your model. Nevertheless, StP allows you to generate code for your model incrementally. The code generated could theoretically be used as a starting point to create a full application.

Note: The example systems provided with the StP installation, including *uml_email*, are incomplete and purposely contain errors for users to correct.

StP generates code, including C++ code, by employing ACD (Architecture Component Development) templates. Generating code for a specific model is simple with ACD; it involves only two things:

- Choosing **Code >** <*language*> **> Generate** <*language*> **by ACD Template** from the StP desktop.
- Specifying an output directory.

The generated code is placed in <*specified_output_directory*>*/src_files*.

ACD is discussed in the *ACD Programming Guide*; ACD templates are discussed in *Using ACD Templates*.

Now It's Your Turn

This completes your introductory tour of StP/UML, Millennium Edition. You may exit all editors and the StP desktop, or continue exploring StP's capabilities on your own.

For more information about Aonix, please visit our web site at http://www.aonix.com.