COMP 3111: Software Engineering

Domain Modeling and Use-case Modeling Using Rational Software Modeler[1] Tutorial and Lab Notes

IBM Rational Software Development Platform (SDP) is an Eclipse-based common development environment for object-oriented system development using the Unified Modeling Language (UML). One component of Rational SDP is Rational Software Modeler (RSM), which is a visual modeling tool for constructing UML models. In RSM, UML models are stored as files (also called modeling resources) that reside within projects. RSM provides the following types of modeling files[2]:

- *Blank Package* – A modeling file that is not based upon a model template. It has no special content other than a single "Main" (freeform) diagram. It can be used to represent any type of model. In particular, it can represent a domain model (i.e., a class diagram).
- *Use-Case Package* – A modeling file containing content specifically for specifying use cases.
- *Analysis Package* – A modeling file containing content for specifying use-case realizations–analysis.
- *Design Package* – A modeling file for specifying use-case realizations–design.
- *Deployment Package* – A modeling file for specifying the deployment of modeling elements.

Table 1 shows the correspondence between Unified Process (UP) models and RSM package types.

UP Model	RSM Package Type
Domain Model	Blank Package used as Domain Model
Use-Case Model	(Blank) Use-Case Package
Analysis Model	Blank/Simplified Analysis Package
Design Model	Blank/Simplified Design Package
Implementation Model	Implementation projects (e.g., VB, C# project)
Deployment Model	Blank/Simplified Deployment Package

Table 1: Correspondence between UP Models and RSM Package Types.

CREATING CLASS DIAGRAMS

A class diagram is a graphical presentation used to document a system's static structure in terms of classes, attributes, operations, associations and generalizations. It can be used to show data aspects (e.g., a domain model) or functional aspects of a system (e.g., classes that collaborate to accomplish a use case). We will demonstrate how to create the domain model class diagram for the ASU application shown in Appendix A. Although we will specifically be creating a domain model class diagram, the method for creating other kinds of class diagrams is similar.

Creating a Domain Model

To create a domain model for the ASU application, do the following.

1. Create a folder called "ASUModels" in your personal directory.
2. Start Rational SDP.
3. Create a new UML project by selecting "File→New→Project …" from the main menu as shown in Figure 1 or by pressing "Alt+Shift+N" and selecting "Project" from the popup window.
4. In the `Select a Wizard` page of the `New Project` window (Figure 2(a)) select "UML Project" inside the "Modeling" folder and click the "Next >" button.
5. In the `Create Model Project` page of the `UML Modeling Project` window (Figure 2(b)), enter "ASU Student Registration" in the *Project name* field, uncheck "Use default location", browse to the "ASUModels" folder and click the "Next >" button.
6. In the `Create Model` page of the `UML Modeling Project` window (Figure 2(c)), select "General" in the Categories (left) pane, select "Blank Package" in the Templates (right) pane, enter "Domain Model" in the *File name* field [3] and click the "Next >" button.

[1] These notes describe the version of Rational Software Modeler installed in Lab 4. A newer version of Rational Software Modeler called simply Rational Modeler can be downloaded for free at http://www.ibm.com/developerworks/downloads/r/modeler/.

[2] While RSM will not prevent a model file from containing inappropriate content, it is highly recommended that you treat model files as being typed and only place appropriate content in them.

[3] You can give your model any meaningful name. In this case, since we are creating a domain model, "Domain Model" is the most appropriate name for the model.

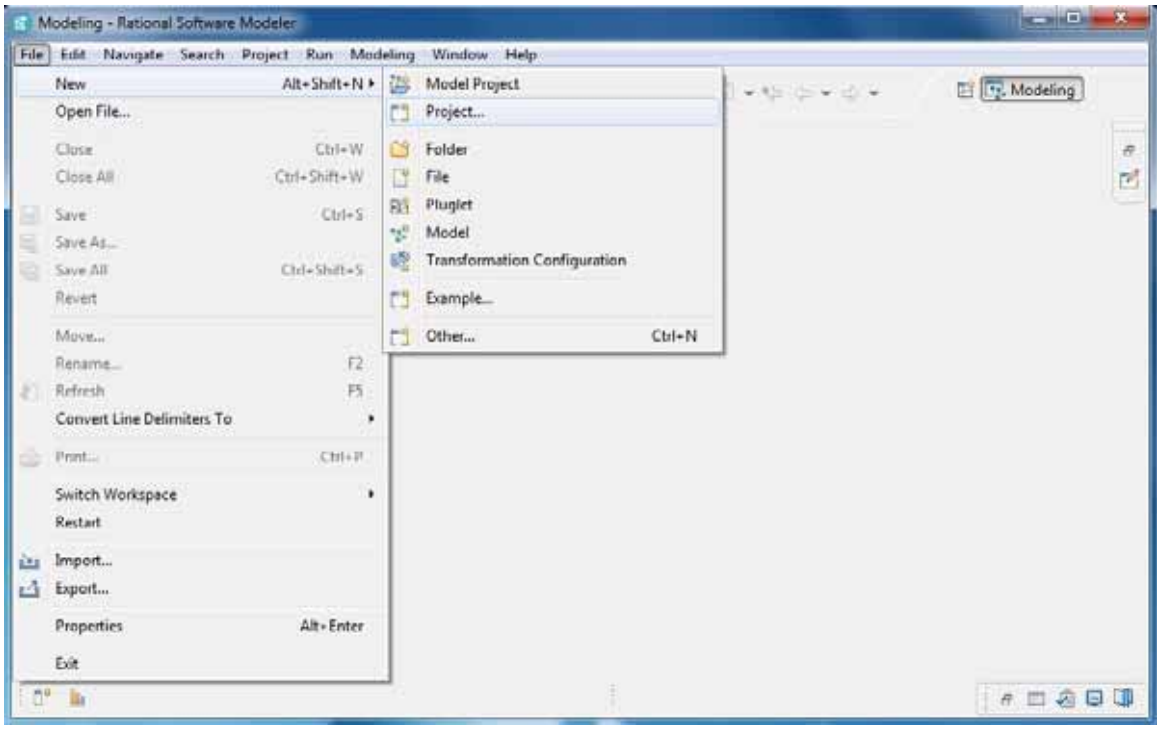


Figure 1: Creating a new project in RSM.

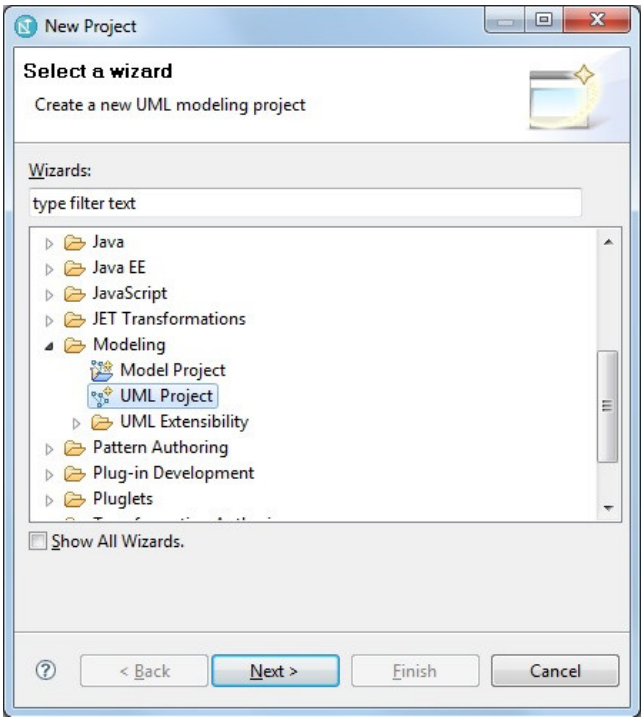
7. In the `Package Details` page of the `UML Modeling Project` window (Figure 2(d)), select the "Model" radio button under *Package type*, select "Class Diagram" from the *Default diagram type* dropdown list and click the "Finish" button[4].
8. If necessary, close the Start Page to see the RSM workspace shown in Figure 3.

The RSM workspace shown in Figure 3 contains several modeling views (tabbed panes):

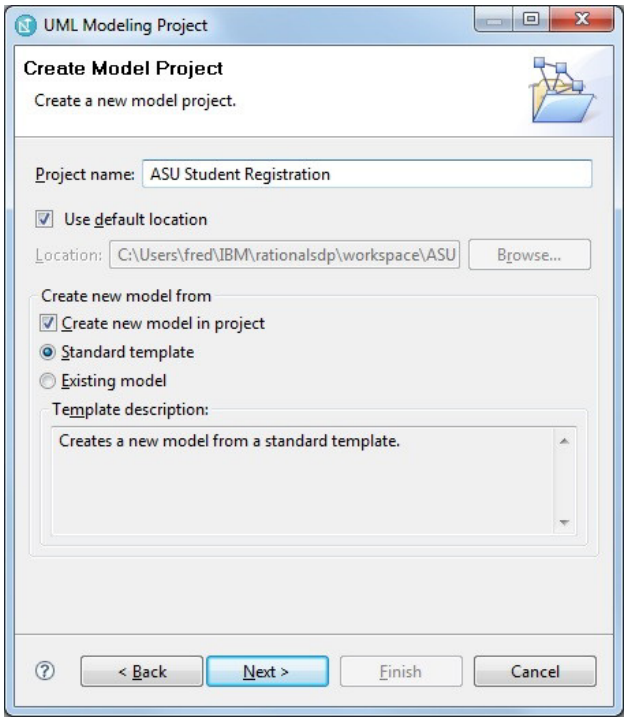
- *Project Explorer* – functions like Microsoft Windows Explorer showing the files that have been created within a project.
- *UML Editor* – provides a design surface for creating UML models (represented by the tabs labelled *Domain Model.emx* and *Main* in Figure 3). The `Main` tab is where you create UML models. It has a Palette attached to its right side that acts like a toolbox containing the various modeling elements needed to create UML diagrams.
- *Properties* – allows you to specify the properties of modeling elements.
- *Tasks* – displays a list of "To Do" tasks.
- *Console* – provides feedback from RSM.
- *Bookmarks* – stores any bookmarks you have created.
- *Outline* – gives you a thumbnail view of the model in the `Main` tab and allows you to see what part of the model is currently displayed in the `Main` tab as well as quickly change the part of the model that is displayed in the `Main` tab by moving a highlighted rectangle representing the current view in the `Main` tab.
- *Inheritance Explorer* – allows you to explore the inheritance structure of a model element.

Close any windows you do not need, such as the `Welcome` window, and resize the workspace so that there is room to create the domain model class diagram in the `Main` tab.

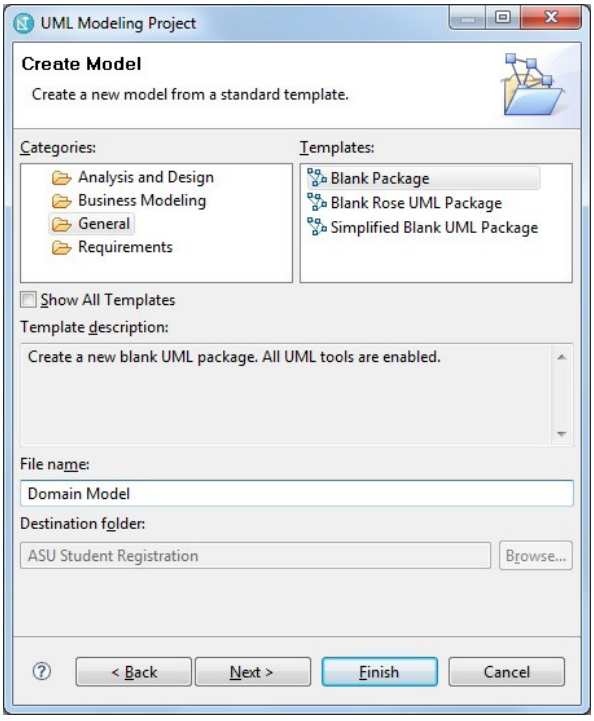
[4] There are three more pages `Model Capabilities`, `Add the project to Modeling Working Sets` and `Reference Projects` on which various options can be set. We will use the default settings for these pages.



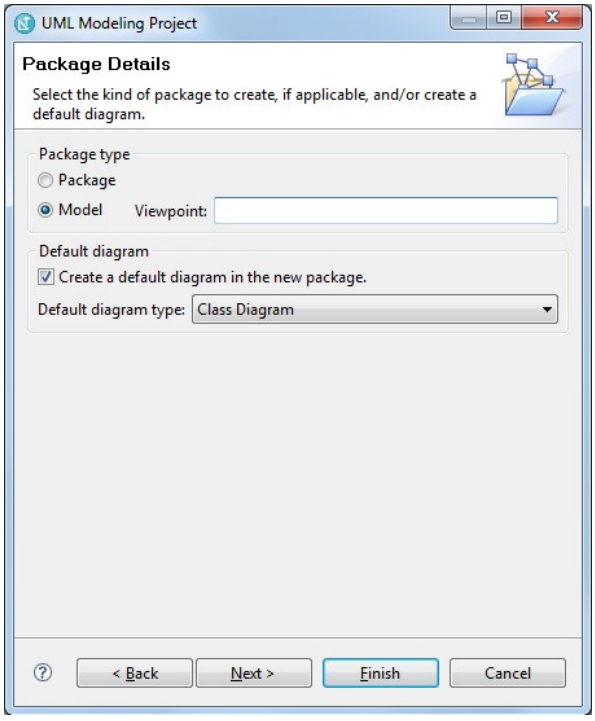
(a) Select the project type.



(b) Name the project.



(c) Select the UML package template.



(d) Select the package type and diagram type.

Figure 2: Creating a new (domain model) class diagram.

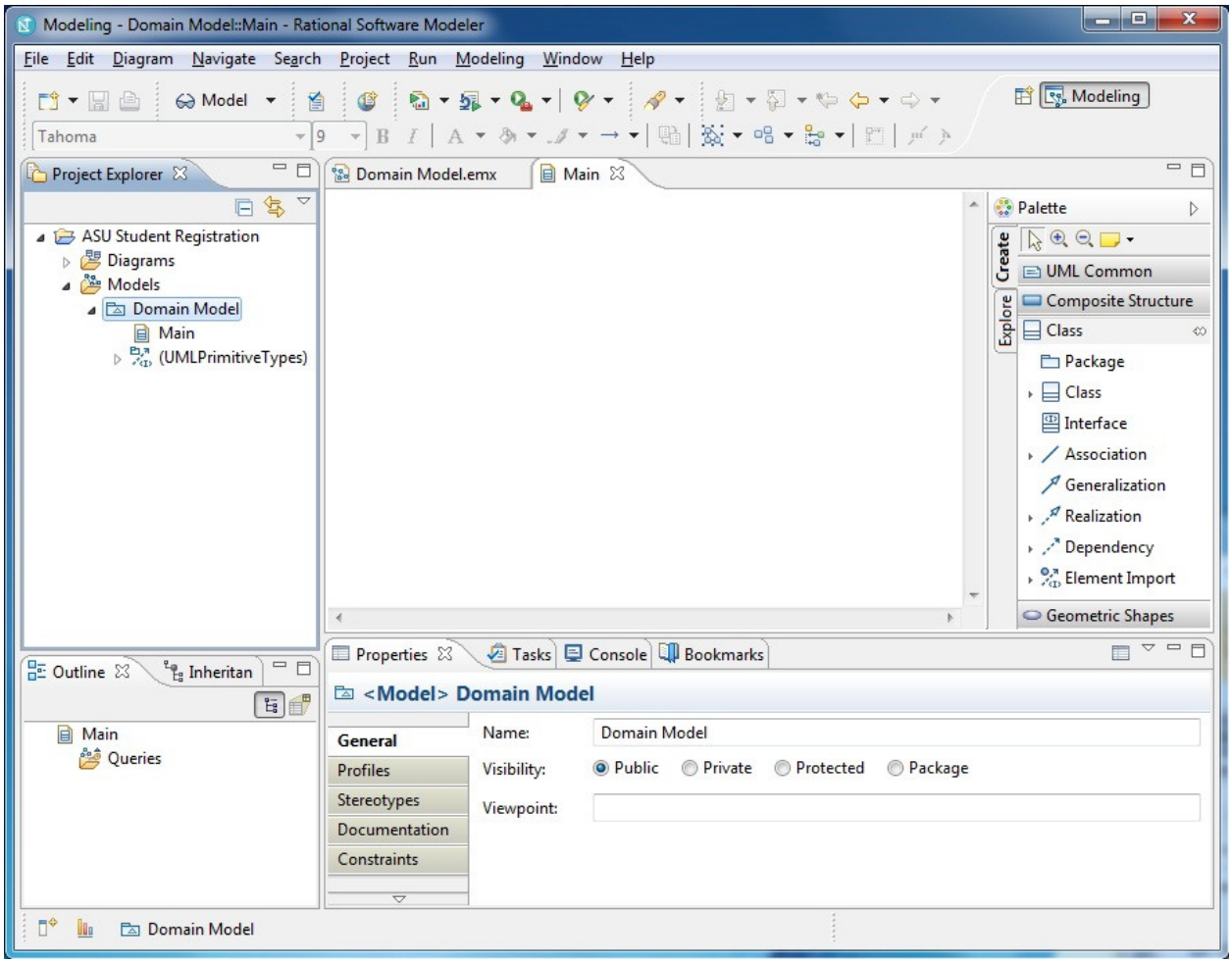


Figure 3: RSM workspace.

Creating a Class

1. In the Class drawer of the Palette attached to the right side of the `Main` tab of the design surface (see Figure 5), click the "Class" item and then click anywhere in the design surface or double-click the "Class" item or right-click in the design surface and select "Add UML→Class" or right-click the Domain Model node[5] in the Project Explorer and select "Add UML→Class"[6]. A class called "Class1" is placed in the design surface.
2. While the new class is still selected, enter the name of the class[7] (e.g., "Student" as shown in Figure 4).

Figure 4: Class "Student" added to the design surface.

In a similar way you can create all the classes shown in Appendix A in the design surface.

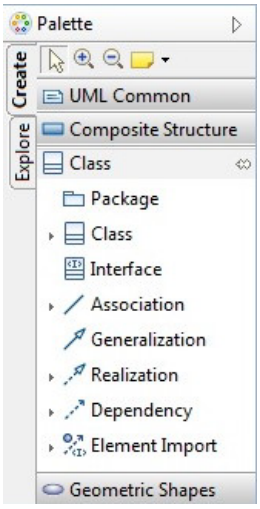


Figure 5: Class drawer of the Palette.

[5] If you create a class by right-clicking the Domain Model node in the Project Explorer, the class is created in the Project Explorer, but does not appear in the design surface. To place the class onto the design surface, simply drag it from the Project Explorer onto the design surface.

[6] It is also possible to do this and many other tasks via a context-sensitive drawing action bar that appears in the design surface periodically. The one for class diagrams is shown here  .

[7] You can also double-click on the name compartment of a class to edit its name.

Documenting a Class

Classes are documented in the `Properties` window[8], which appears below the design surface by default. To document a class do the following.

1. Select the class in the design surface (e.g., Student).
2. Select the Documentation tab in the `Properties` window and enter the documentation for the class as shown in Figure 6.

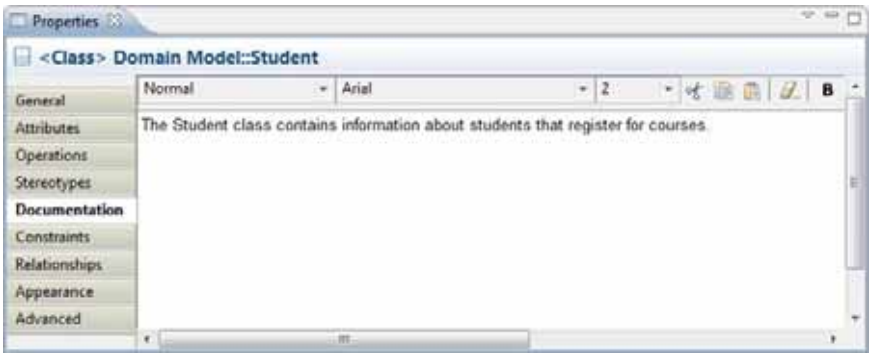


Figure 6: Documenting a class.

Changing the Display Appearance of a Class

The appearance of a class in a class diagram can be changed using the Appearance tab of the `Properties` window. The font and color of the class, parent style, stereotype decoration and visibility of compartments and compartment titles can be changed as shown in Figure 7. In particular, by default, the stereotype decoration of a class is displayed in the name compartment of a class. This can be changed in the `Properties` window of the class as shown in Figure 7 so that only the class name is displayed.

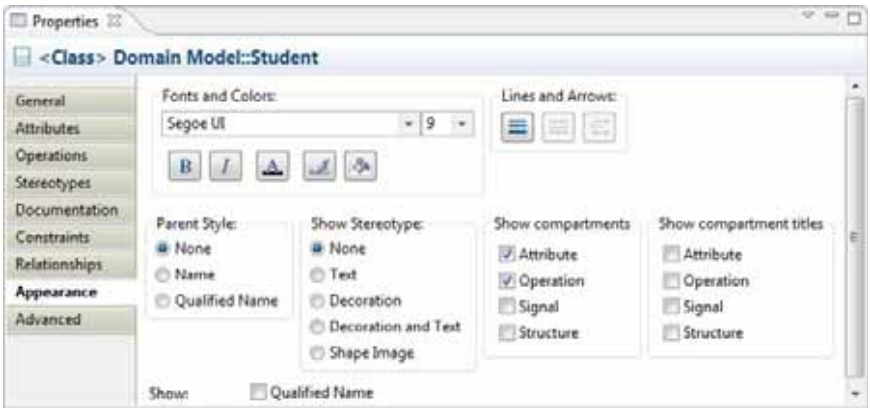


Figure 7: Changing the display appearance of a class.

Defining/Editing an Attribute for a Class

To define an attribute for a class do the following.

1. Right-click on the class in the design surface or right-click the class in the Project Explorer.
2. Select "Add UML→Attribute" from the pop-up menu. A new attribute called "Attribute1" is placed in the attribute compartment of the class.
3. Edit the attribute name to the desired name and type of the attribute.

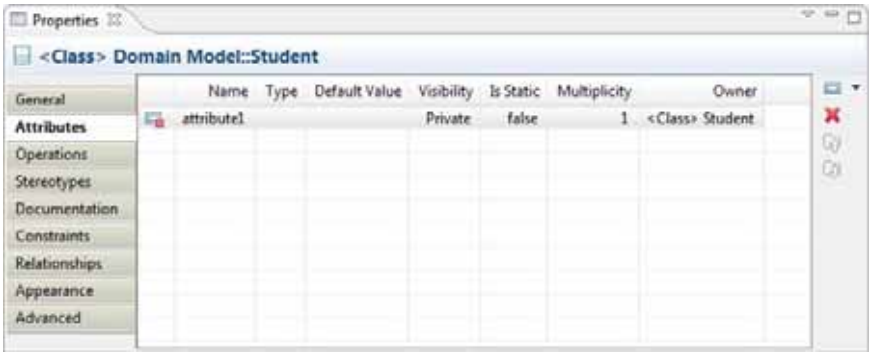


Figure 8: Defining attributes of classes using the `Properties` window.

Alternatively, attributes can be defined for a class and their properties edited using the `Properties` window.

1. Select the class in the design surface, open the `Properties` window for the class and select the Attributes tab in the `Properties` window.
2. To add a new attribute, click on the double rectangle on the right side of the `Properties` window or right-click in the `Properties` window and select "Insert New Attribute" from the pop-up menu. A new row with values filled in for the Name, Visibility, Is Static, Multiplicity and Owner properties appears as shown in Figure 8.

[8] The `Properties` window can be opened by selecting "Window→Show View→Properties" from the main menu or right-clicking in the design surface and selecting "Show Properties View" from the pop-up menu.

3. You can edit the properties of an attribute as follows.
   - *Name* – Select the name and type a new name.
   - *Type* – Click in the field, click on the ellipse button and select a type[9] from the `Select Element for Type` window as shown in Figure 9[10].
   - *Visibility* – Click in the field and select either public, private, protected or package in the dropdown list.
   - *Is Static*[11] – Click in the field and select either true or false in the dropdown list.
   - *Default Value* – Click in the field and select the default value type desired from the dropdown list.

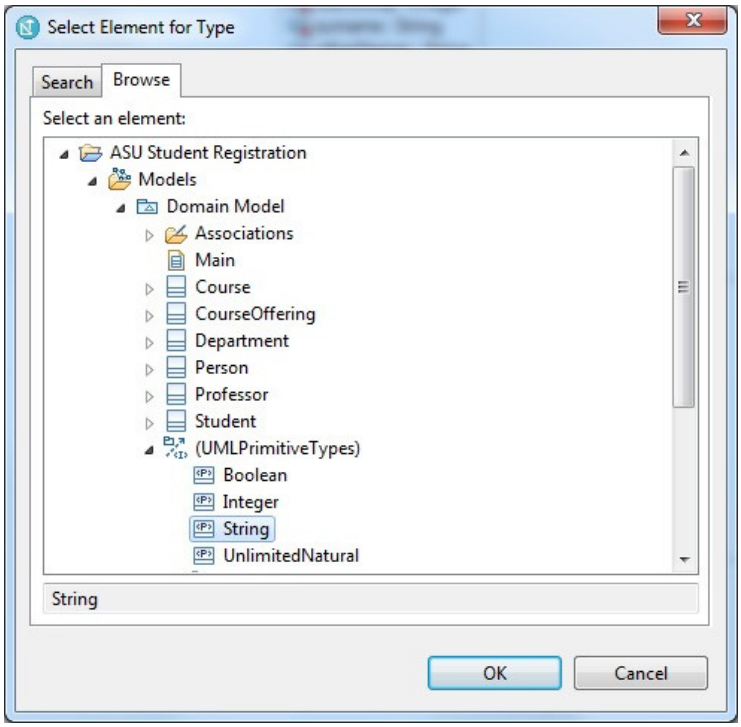


Figure 9: `Select Element for Type` window.

Defining/Editing an Operation for a Class

To define an operation for a class do the following.

1. Right-click on the class in the design surface or right-click the class in the Project Explorer.
2. Select "Add UML→ Operation" from the pop-up menu. A new operation called "Operation1" is placed in the operation compartment of the class.
3. Edit the operation name to the desired name of the operation.

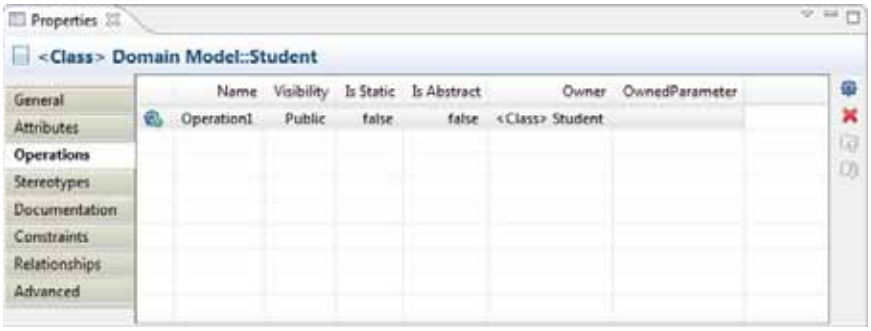


Figure 10: Defining operations of classes using the `Properties` window.

Alternatively, operations can be defined and their properties edited using the `Properties` window.

1. Select the class in the design surface, open the `Properties` window for the class and select the Operations tab in the `Properties` window.
2. To add a new operation, click on the cogwheel symbol on the right side of the `Properties` window or right-click in the `Properties` window and select "Insert New Operation" from the pop-up menu. A new row appears with values filled in for the Name, Visibility, Is Static, Is Abstract and Owner properties of the operation as shown in Figure 10.
3. You can edit the properties of an operation as follows.
   - *Name* – Select the name and type a new name.
   - Visibility – Click in the field and select either public, private, protected or package in the dropdown list.
   - *Is Static*[12] – Click in the field and select either true or false in the dropdown list.
   - *Is Abstract* – Click in the field and select either true or false in the dropdown list.

[9] The easiest way to select a type that is already defined is to type the first few characters of its name in the Search box of the Search tab in the `Select Element for Type` window.

[10] You can define a new primitive type (e.g., Decimal) in your model by right-clicking on a package node (e.g., the Domain Model node) in the Project Explorer and selecting "Add UML→Primitive Type" from the pop-up menu. The new primitive type is added to your model and is available thereafter in the `Browse` tab of the `Select Element for Type` window. Note that all classes in a model are also available as types as shown in Figure 9.

[11] A static attribute has class scope and is underlined in a class diagram.

[12] A static operation has class scope and is underlined in a class diagram.

- *OwnedParameter* – Click in the field, click on the ellipse and add parameters in the `Properties` window that appears by right-clicking in the right side of the window and selecting "Insert New Parameter" from the pop-up menu. Edit the properties of the parameter as necessary.

Creating an Association

To create an association do the following.
1. In the Class drawer of the Palette (see Figure 5) select the "Association" item[13].
2. Click on one of the classes to be related in the design surface (e.g., Professor) and, while holding down the mouse button, drag the cursor to the other class (e.g., Department) as shown in Figure 11[14].
3. Enter a name for the association in the textbox that appears near the middle of the association line (e.g., Appoints)[15] as shown in Figure 11.

Figure 11: Creating an association between two classes.

Alternatively, you can name an association using the `Properties` window.
1. Select the association in the design surface, open the `Properties` window and select the General tab.
2. Edit the Label field of the `Properties` window (see Figure 15) to the desired name for the association[16].

Documenting an Association

As for a class, an association is documented in the `Properties` window. To document an association do the following.
1. Select the association in the design surface (e.g., Appoints).
2. Select the Documentation tab in the `Properties` window and enter the documentation for the association as shown in Figure 12.

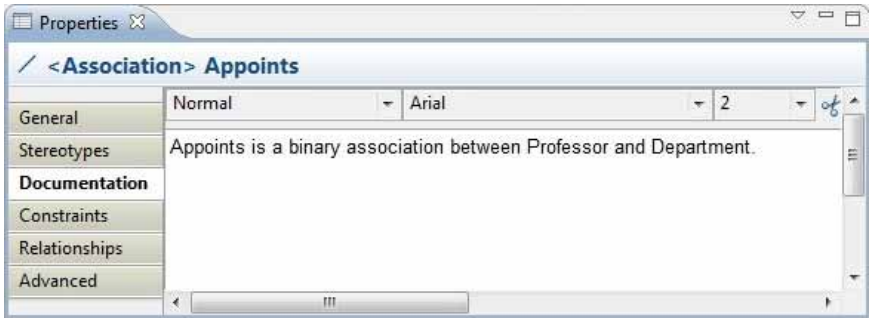


Figure 12: Documenting an association.

Editing the Role Name Properties of an Association

By default, role names with private visibility are assigned to both ends of an association as shown in Figure 11. To edit the properties of an association role name do the following.
1. Select the role name in the design surface, open the `Properties` window and select the General tab.
2. Edit the desired property (e.g., Name) in the `Properties` window as shown in Figure 13.

If you do not want a role name to be displayed, you can delete it from the diagram by right-clicking on the role name in the design surface and selecting "Delete from Diagram" in the pop-up menu[17]. Note that once you delete a role name from a diagram, there is no way to display the role name again should you decide to change your mind. You will have to delete the association from the model and recreate it to display its role names.

[13] Alternatively, to create an association, you can click a class and a set of arrows will appear on the border of the class near to where you click. Click and drag an arrow to the class to be related. A pop-up menu appears with a list of possible relationships. Select the type of relationship that you want to create from the list. In this case you would select "Create New Association" from the list.
[14] You can adjust the angle and position of the association line by selecting it at any point and dragging the line.
[15] You can change the position of any of the association adornments (i.e., name, role name and multiplicity) by selecting and dragging it with the mouse. For finer positioning, hold down the Alt key while dragging a model element.
[16] You can also keep the `Properties` window open and simply select an association to view and change its properties.
[17] Note that you need to select "Delete from Diagram" and not "Delete from Model". If you select "Delete from Model", the entire association is deleted from your model. The "Delete from Diagram" option only deletes the role name from the diagram, but it is still present in the model.

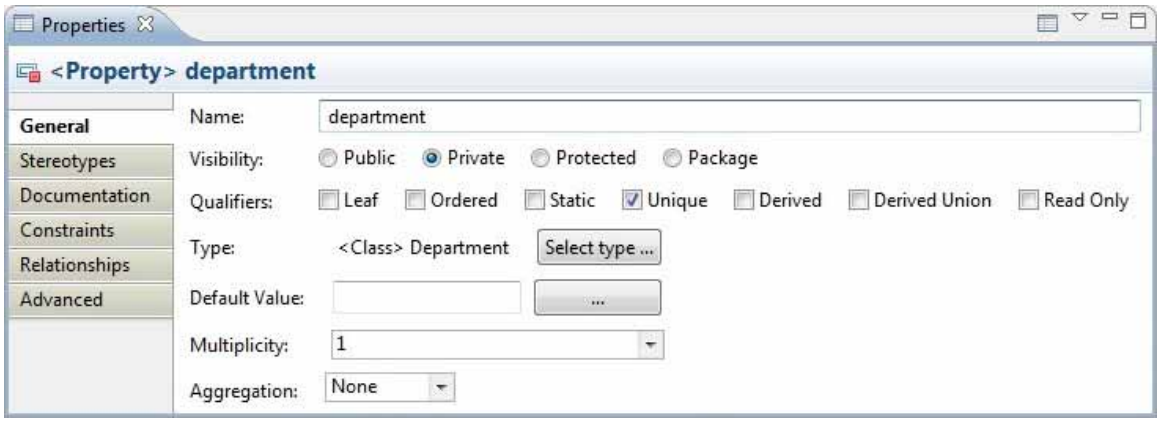


Figure 13: Editing association role name properties.

Editing the Multiplicity of an Association

By default, the multiplicity for an association is set to 1 for both ends as shown in Figure 11. There are several ways to edit the multiplicity.

1. Double-click on the multiplicity in the design surface.
2. Type a new value for the multiplicity.

Alternatively, you can

1. Select the association in the design surface, open the `Properties` window and select the General tab.
2. Edit the multiplicity directly in the Multiplicity field shown in Figure 15 for each end of the association to the desired value.

Alternatively, you can

1. Select the multiplicity in the design surface, open the `Properties` window and select the General tab.
2. Select the desired multiplicity from the dropdown list of the Multiplicity field shown in Figure 15.

Note that this option only allows you to select from a fixed list of multiplicities (i.e., (none), *, 0..1, 1 and 1..*). If the multiplicity that is needed is not in this list, then you will need to use one of the previous methods to edit the multiplicity.

Creating a Composition Association

To create a composition association do the following.

1. In the Class drawer of the Palette, expand the “Association” item and select “Composition Association” from the expanded list[20].
2. Click on the composite class (i.e., the class playing the role of the "whole") in the design surface (e.g., Course) and drag the cursor to the component class (i.e., the class playing the role of the "part") in the design surface (e.g., CourseOffering).
3. Edit the properties of the composition association (i.e., role names, multiplicity and navigability[18]) to the desired values as shown in Figure 14.

Figure 14: Creating a composition association.

[18] By default, composition and aggregation associations are navigable in only one direction from the “whole” (e.g., Course) to the “part” (e.g., CourseOffering). Hence, a navigability adornment (i.e., an arrow head) appears at the “part” (e.g., CourseOffering) end of the association. If the association is navigable in both directions, you need to check the “Is navigable” checkbox at the “whole” (e.g., Course) end of the association in the General tab of the `Properties` window. When you do so, you may also need to again set the Type field of the association to be composition/aggregation in the General tab of the `Properties` window.

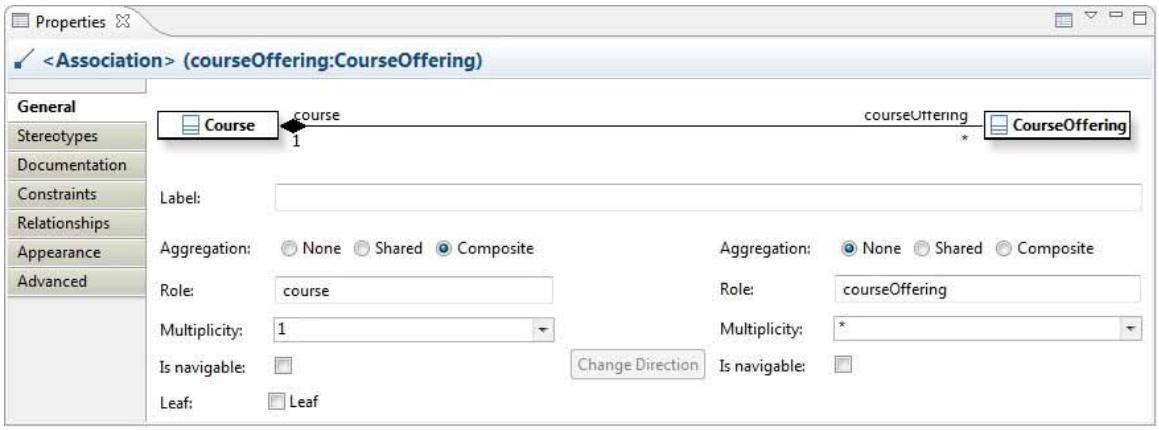


Figure 15: Changing the association type in the `Properties` window.

Creating an Aggregation Association

To create an aggregation association:

1. In the Class drawer of the Palette, expand the "Association" item and select "Aggregation Association" from the expanded list [20].
2. Click on the aggregate class (i.e., the class playing the role of the "whole") in the design surface and drag the cursor to the component class (i.e., the class playing the role of the "part").
3. Edit the properties of the aggregation association (i.e., role names, multiplicity and navigability) to the desired values.

It is also possible to create a composition association or an aggregation association by changing the type of an existing association. To do so, select the association in the design surface, open its `Properties` window and select the General tab as shown in Figure 15. For the class that plays the role of the "whole" in the composition or aggregation association, select either the radio button "Composite" for a composition association or the radio button "Shared" for an aggregation association.

Creating an Association Class

To create an association class[19] do the following.

1. In the Class drawer of the Palette, expand the "Association" item and select "Association Class" from the expanded list[20].
2. Click on one of the classes to be related in the design surface (e.g., Student) and drag the cursor to the other class to be related (e.g., CourseOffering) as shown in Figure 16.
3. Edit the name of the association class to the desired name either directly in the design surface or using the `Properties` window for the association class.

You can edit the properties of an association class in a way similar to that for an association. To view the properties of an association class, select either the association or the association class in the design surface[21].

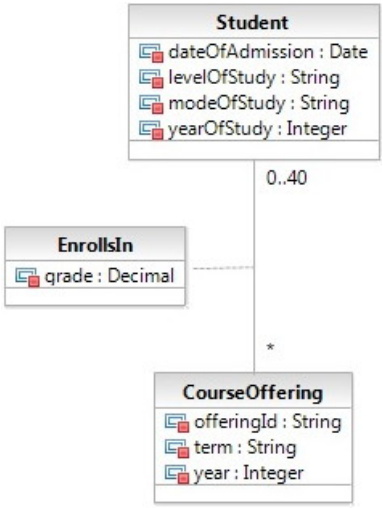


Figure 16: Creating an association class.

[19] The two classes that are to be related must already have been created in the design surface.
[20] You can also use the method discussed in footnote 13.
[21] Although supported in the UML, it is not possible to relate an association class to a class in the version of RSM described in these notes.

Creating a Unary Association

To create a unary association do the following.

1. In the Class drawer of the Palette select the "Association" item[20].
2. Click on the class in the design surface.
3. Enter a name for the association in the textbox that appears near the middle of the association line.
4. Edit the properties of the association (i.e., role names, multiplicity and navigability) to the desired values.

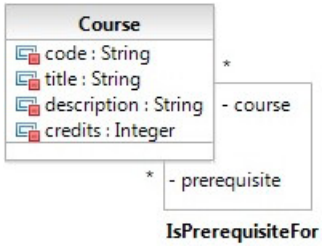


Figure 17: Creating a unary association.

Creating a Ternary and Higher Order Association

It is not possible to create ternary or higher order associations in the version of RSM described in these notes.

Creating a Generalization

To create a generalization, do the following *for each subclass* that is part of the generalization.

1. In the Class drawer of the Palette, select the "Generalization" item[20].
2. Click on the subclass (e.g., Student) and drag the generalization line to the superclass (e.g., Person) as shown in Figure 18.
3. To create only one generalization arrowhead as shown in Figure 18 for a generalization with multiple subclasses, drag the second and subsequent generalization lines to the existing generalization line rather than to the superclass.
4. To adjust the angle and position of the inheritance line to that shown in Figure 18, select it at any point and drag it.

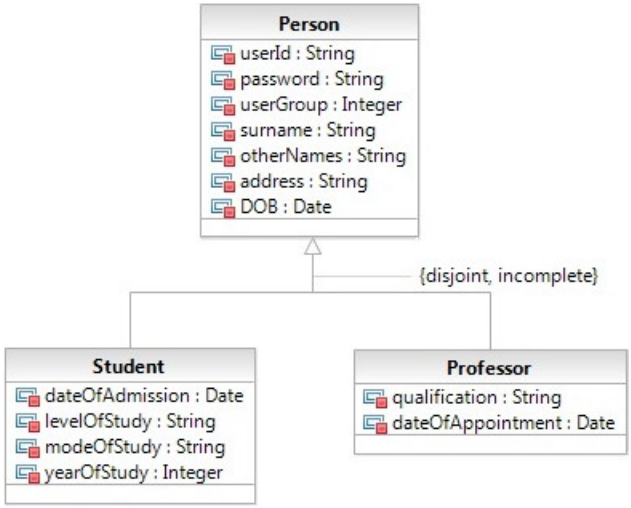


Figure 18: Creating a generalization.

Creating a Constraint/Comment

To create a constraint or comment, do the following.

1. In the UML Common drawer of the Palette, select the "Constraint" item or select the "Comment" item and click anywhere in the design surface or right-click in the design surface and select "Add UML→Constraint" or select "Add UML→Comment" from the pop-up menu. A note appears with a textbox highlighted.
2. Type the constraint (e.g., disjoint, complete) or comment as shown in Figure 19.

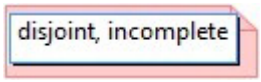


Figure 19: Creating a constraint.

To attach a constraint or comment to a model element[22] do the following.

1. In the UML Common drawer of the Palette, expand the "Constraint" item and select "Constraint Attachment" from the expanded list or select "Comment Attachment" from the expanded list under the "Comment" item.
2. Click on the constraint or comment in the design surface and drag the cursor to the model element to which the constraint or comment applies. A dashed line will indicate the model element to which the constraint or comment is attached as shown in Figure 18.

Note that constraints and comments can only be attached to certain model elements. The cursor will change to only an arrow for model elements to which a constraint or comment can be attached. You can change the background and border colour of the constraint or comment in the `Properties` window (e.g., to make the containing note invisible as shown in Figure 18).

[22] You can also right-click on the generalization and select "Add UML→Constraint" from the pop-up menu to simultaneously create the constraint and attach it to the generalization. You may need to adjust the attachment point of the constraint to where you want it by dragging the constraint attachment line while holding down the Alt key. However, exact positioning of the constraint line may not be easy!

CREATING USE-CASE DIAGRAMS

We will demonstrate how to create the use-case diagram shown in Appendix B. The Rational Software Modeler (RSM) modeling guidelines (see Appendix C) recommend that, in addition to the actors and use cases, a use-case model contain:

- A "main" diagram at the model root that depicts the other packages of the model and supports drill-down into those packages and their respective "main" diagrams.
- In each use-case package, a diagram that depicts the package's use cases, any relationships among them, and the actors that participate in them. (If the number of use cases is large, more than one diagram may be appropriate.)
- A description of each use case's main and alternate flows in its Documentation field.
- Optionally, a use-case model can also contain an activity diagram that reflects the overall activity flows of the use case when the complexity of a use case warrants it.

Creating a Use-case Model

To create a use-case model do the following.

1. In the Project Explorer, right-click on the node "ASU Course Registration System" and select "New→UML Model" from the pop-up menu or select "File→New→UML Model" from the main menu.
2. In the first `UML Model – Create Model` window, leave the "Standard template" radio button selected and click the "Next" button.
3. In the second `UML Model – Create Model` window do the following.
   a. Select "Requirements" in the Categories (left) pane.
   b. In the Templates (right) pane, select either "Blank Use Case Package" if you want to only create a use-case diagram or, as shown in Figure 20, select "Use Case Package" if you want to create additional diagrams for a use case and structure your use-case model[23].
   c. Enter a file name for the model (e.g., Use Case Model).
   d. Click the "Finish" button.

As shown in Figure 21, several packages and files intended to provide guidance in specifying a use-case model are created in the `Project Explorer`[24]. These packages and the "Instructions" file can be deleted from your use-case model *once you are finished specifying it*.

The "«perspective» Overviews" package stores diagrams depicting cross-cutting views of a use-case model and initially contains three files[25].

- *Overviews* – contains links for navigating to other diagrams in the "«perspective» Overviews" package.
- *Actors Overview* – a diagram that depicts the actors in your use-case model and their generalization relationships.
- *Context Diagram* – a diagram that depicts the most important use cases of your system. The use cases should be placed within a rectangle that represents the boundary of the system. The actors that perform the use cases should be placed outside of the rectangle.

The "Use-Case Building Blocks" package provides two templates for constructing a use-case model:

- ${functional.area} – a template for defining the functional areas of your use-case model (see the section Creating a Functional Area below).
- ${use.case} – a template for defining the use cases of your use-case model (see the section Creating a Use Case below).

The "Versatile Actors" package contains actors that participate in multiple use cases that span functional areas. Finally, the "${project} Use Case Overview"[26] file contains links to the Versatile Actors package and the functional area packages of your use-case model.

[23] For the course project you should select "Blank Use Case Package".
[24] While there is no standard way to internally organize the model elements (i.e., actors and use cases) in a use-case model in terms of the files in which they should be placed, some RSM guidelines for this can be found in Appendix C.
[25] You can add additional diagrams to your model by right-clicking any of the packages in the `Project Explorer` and choosing "Add Diagram" from the pop-up menu.
[26] You should replace the string "${project}" with the name of your project.

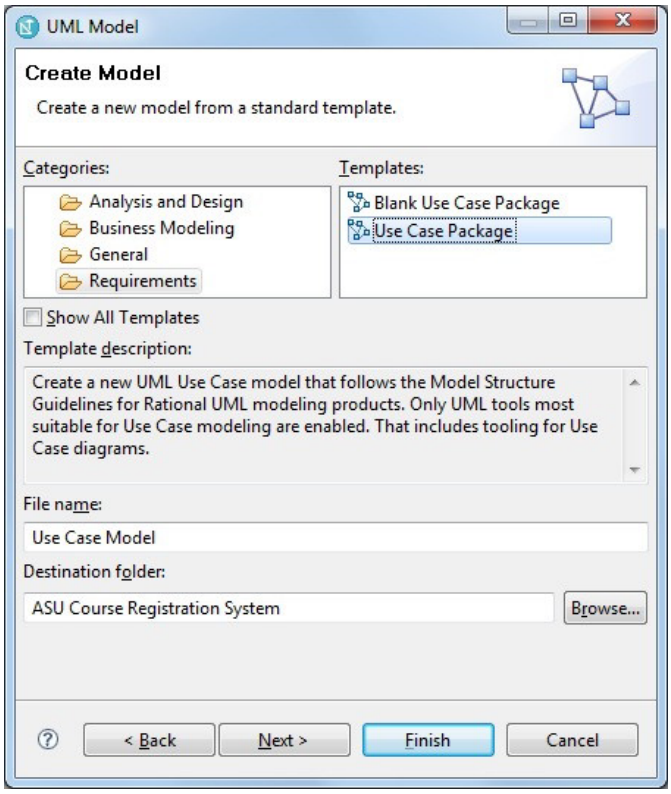


Figure 20: Creating a use-case model.

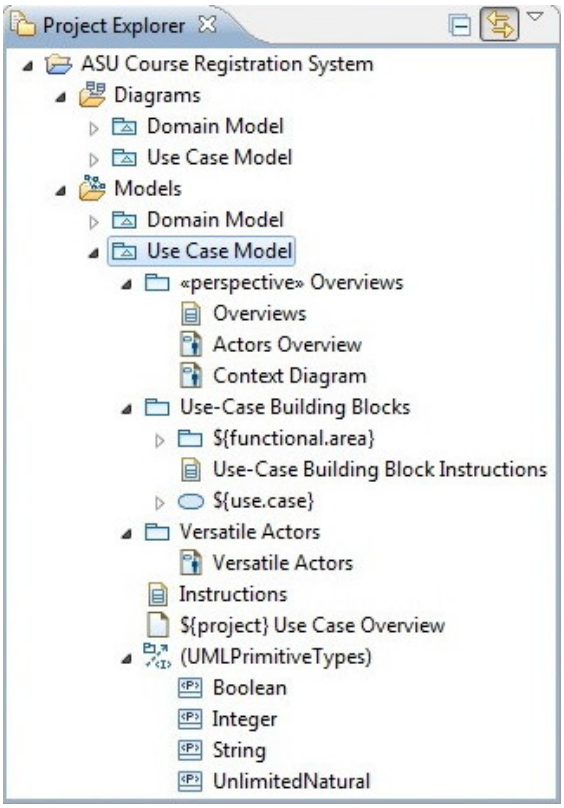


Figure 21: Use-case Model files created by RSM.

Creating a Functional Area[27]

Depending on the size of your application, it may be a good idea to partition the use-case model into several functional areas to aid understanding and to be able to more easily divide up the development work. RSM allows you to define functional areas in which you can define and group your use cases. To illustrate the use of functional areas, we will define three functional areas for the "ASU Course Registration System" use-case model: "Information Management", "Course Registration" and "Course Management".

To define a new functional area:

1. Select the node "${functional.area}" under the "Use-Case Building Blocks" node and Ctrl-drag it to the "Use Case Model" node in the `Project Explorer`[28].
2. Replace the name of the newly created functional area node by right-clicking it and selecting "Find/Replace …" from the pop-up menu[29].
3. Enter "${functional.area}" in the "Find what:" field and the name of the functional area (e.g., "Information Management") in the "Replace with:" field. (See Figure 26 for an example of this renaming method.)
4. Click the "Replace All" button.

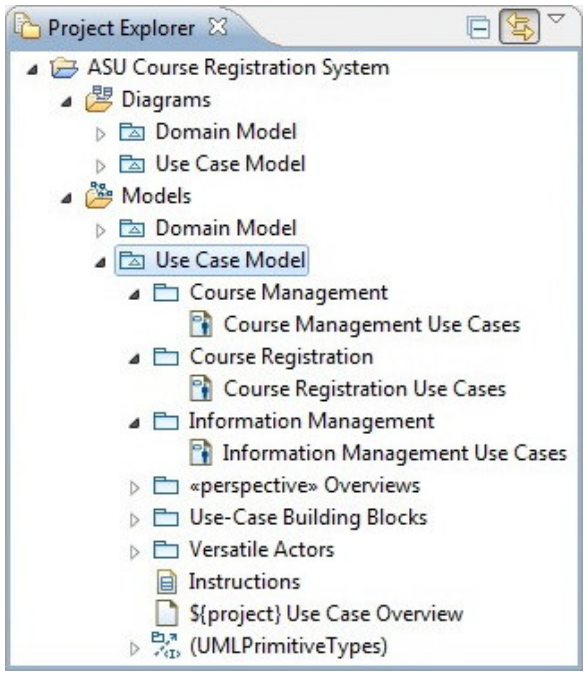


Figure 22: Creating functional areas.

Figure 22 shows the three newly created functional areas in the `Project Explorer`.

[27] A functional area can be a department of an organization that caries out a specific task or "function" for the organization (e.g., marketing, sales, accounting, etc.). You do not need to create functional areas for the course project.

[28] You may first need to double-click on the "${functional.area}" node to open its instruction page before RSM will allow you to Ctrl-drag the "${functional.area}" node to the Use Case Model node.

[29] If you select "Rename" from the pop-up menu, then only the top-level functional area node shown in Figure 22 will be renamed. The nodes under the top-level functional area node will not be renamed.

Creating an Actor

Generally, an actor should be stored within the same functional area as the use case(s) with which it interacts. However, actors that interact with multiple use cases that span functional areas should be stored in the "Versatile Actors" package[30].

To create an actor in a functional area[31] do the following.

1. Double-click on the functional area node in the `Project Explorer` (e.g., "Course Registration").
2. In the Use Case Diagram drawer of the Palette, double-click the "Actor" drawer or select the "Actor" drawer and click anywhere in the design surface or right-click in the design surface (or right-click the functional area node in the `Project Explorer`) and select "Add UML→Actor". A new actor called Actor1 is placed in the design surface[32].
3. Edit the Actor1 name to the desired name of the actor (e.g., Student) as shown in Figure 23.

Figure 23: Creating an actor.

Documenting an Actor

An actor can be documented in the `Properties` window[33], which appears below the design surface by default. To document an actor:

1. Select the actor in the design surface (e.g., Student).
2. Select the Documentation tab in the `Properties` window and enter the documentation for the actor as shown in Figure 24.

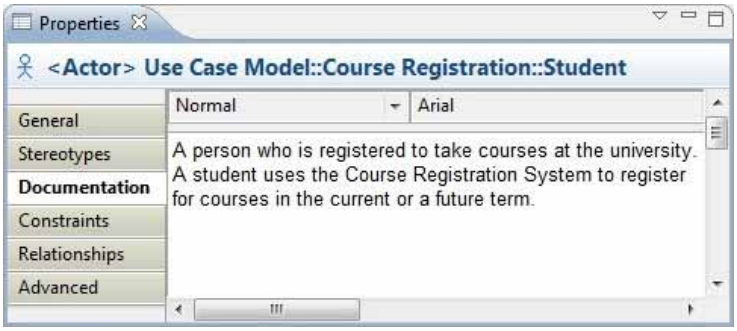


Figure 24: Documenting actors.

Notice that the actor appears in the `Project Explorer` under the "Course Registration" functional area node as shown in Figure 25.

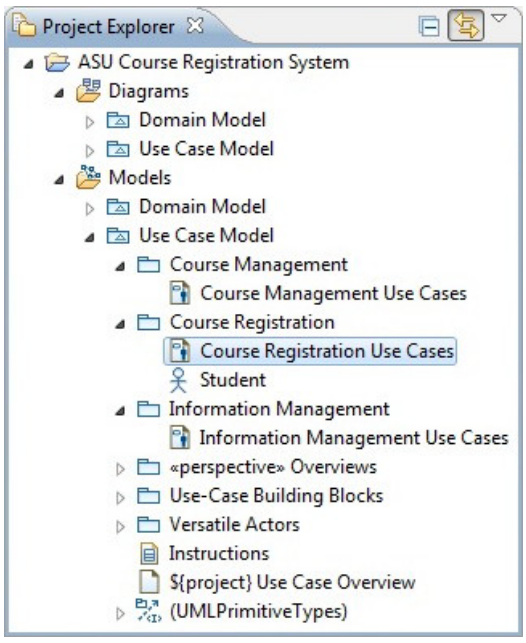


Figure 25: `Project Explorer` showing Student actor.

Creating a Generalization Relationship Among Actors

Since actors are classes in the UML, they can be related by generalization relationships. To create a generalization relationship among actors[34] do the following.

1. In the Use Case Diagram drawer of the Palette, select the "Generalization" item, which can be found in the drop-down list for the «include» relationship.
2. Click on the specialized actor and drag the Generalization relationship to the generalized actor.

[30] If later you realize that an actor participates in multiple use cases that span functional areas, you can use the `Project Explorer` to move the actor to the "Versatile Actors" package.

[31] Versatile actors are created in a similar way by first clicking on the "Versatile Actors" node in the `Project Explorer`.

[32] You should delete the note in the design surface.

[33] The `Properties` window can be opened by selecting "Window→Show View→Properties" from the main menu or right-clicking in the design surface and selecting "Show Properties View" from the pop-up menu.

[34] Alternatively, to create a generalization relationship, you can click on an actor and a set of arrows will appear on the border of the actor near to where you click. Click and drag an arrow to the actor to be related. A pop-up menu appears with a list of possible relationships. Select the type of relationship that you want to create from the list. In this case you would select "Create New Generalization" from the list.

Creating a Use Case

A use case should be created within its functional area. To create a use case:

1. In the `Project Explorer`, select and Ctrl-drag the node "${use.case}" from the "Use-Case Building Blocks" package to the desired functional area package (e.g., "Course Registration").
2. Rename the newly created use-case node by right-clicking it and selecting "Find/Replace…" from the pop-up menu[35].
3. Enter "${use.case}" in the "Find what:" field and the name of the use case (e.g., "Register For Courses") in the "Replace with:" field as shown in Figure 26.
4. Click the "Replace All" button.

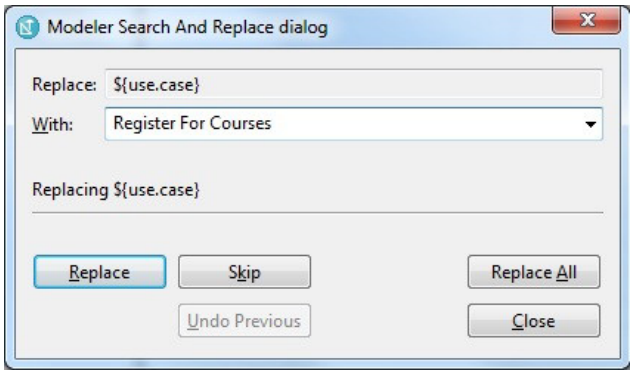


Figure 26: Editing the name of a use-case template.

An Activity Diagram node, which is used to construct an activity diagram for the use case, is created under the top-level use-case node when a "${use.case}" template is used to create a use case as shown in Figure 27.

It is also possible to create a use case from the Use Case Diagram drawer of the Palette or by right-clicking in the design surface (or on the functional area node in the `Project Explorer`) and selecting "Add UML→Use Case" from the pop-up menu. However, these methods will only create the top-level use case node shown in Figure 27; they will not create the activity diagram node shown under the top-level use-case node. This node will have to be added manually, if required.

So far we have created a use case in the `Project Explorer`, but the use case does not yet appear in the use-case diagram for the functional area. To place the use case in the use-case diagram, simply drag it from the `Project Explorer` onto the design surface as shown in Figure 28.

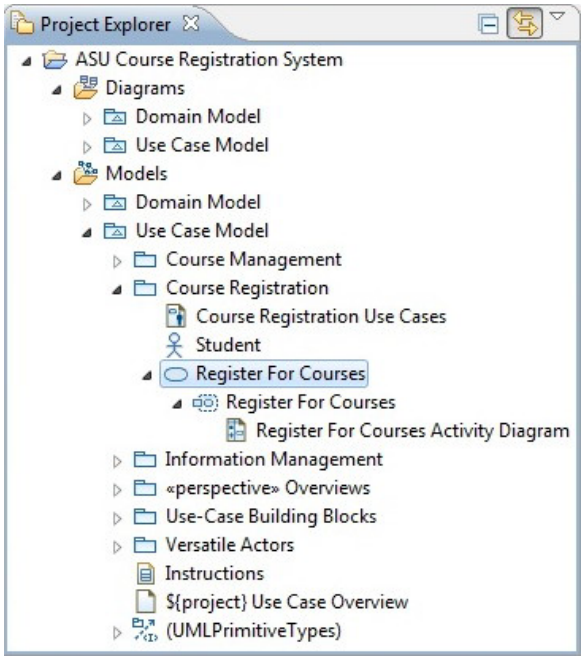


Figure 27: Files for a newly created use case.

Figure 28: Placing a use case onto a use-case diagram.

[35] If you select "Rename" from the pop-up menu, then only the top-level use-case node shown in Figure 27 will be renamed. The nodes under the top-level use-case node will not be renamed and will need to be renamed manually.

Documenting a Use Case

A use case can be documented in the `Properties` window. To document a use case do the following.

1. Select the use case in the design surface, open the `Properties` window and select the Documentation tab.
2. Enter a brief description for the use case as shown in Figure 29.

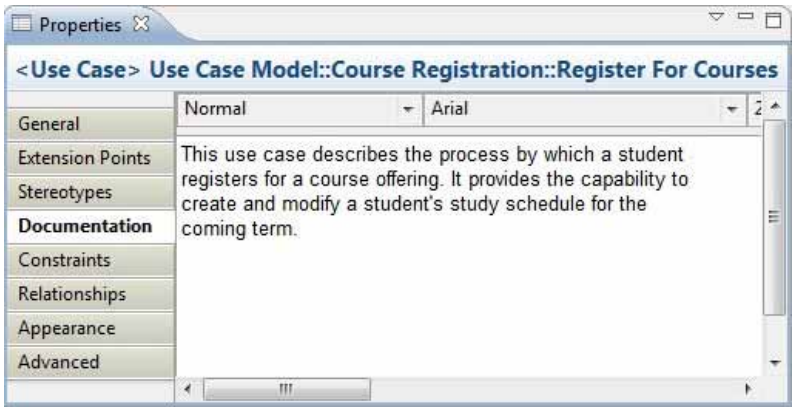


Figure 29: Documenting a use case.

Since there is no fixed format for the flow of events of a use case, no specific field is provided in the `Properties` window to type a use case's flow of events. While the flow of events can be documented in the Documentation tab in the `Properties` window, it is recommended that a use case's flow of events be documented instead in a separate document, such as a Word document, where it can be more easily edited and formatted[36].

Creating a «communication» Association between an Actor and a Use Case

A «communication» association implies that there is an interaction between the connected model elements. In a use-case model, it is an association between a use case and an actor. To create a «communication» association[37] do the following.

1. In the Use Case Diagram drawer of the Palette, select the "Association" item[38].
2. Click on an actor (e.g., "Student") (or use case (e.g., "Register For Courses")) and drag the association line to the desired use case (or actor).

You can omit entering a name for the association and you can delete any multiplicity adornments as shown in Figure 30.

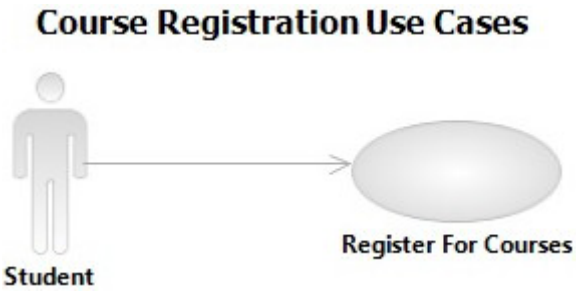


Figure 30: Creating a «communication» association.

Creating a Generalization Relationship among Use Cases

Since use cases are classes in the UML, they can be related by generalization relationships[39]. To create a generalization relationship among use cases[40] do the following.

1. In the Use Case Diagram drawer of the Palette, select the "Generalization" item.
2. Click on the specialized use case and drag the Generalization relationship to the general use case.

Creating an Overview Diagram

To create the overview diagrams (e.g., Actors Overview or Context Diagram), simply drag the required model elements (e.g., actors and/or use cases) from the `Project Explorer` onto the overview diagram. Any relationships that already exist among model elements will also be shown on the overview diagrams (e.g., «communication» associations among actors and use cases). If any additional relationships are required (e.g., actor generalizations), they can be added directly on the overview diagrams.

---

[36] A reference to this document can be included in the use case's Documentation tab.
[37] Alternatively to create a «communication» association, you can click a use case or actor and a set of arrows will appear on the border of the use case or actor near to where you click. Click and drag an arrow to the use case or actor to be related.
[38] Select "Directed association" if you want to show that the actor starts the use case.
[39] It is not recommended that you use generalization relationships among use cases in the course project.
[40] You can also use the method discussed in footnote 34.

GENERATING DOCUMENTATION

To generate the documentation for a model select the model's package in the `Project Explorer` and choose "Modeling→Publish→Web…" in the main menu as shown in Figure 31. The `Publish to Web` window shown in Figure 32 appears for you to specify options for the HTML report and where it should be stored.

Figure 31: Generating documentation.

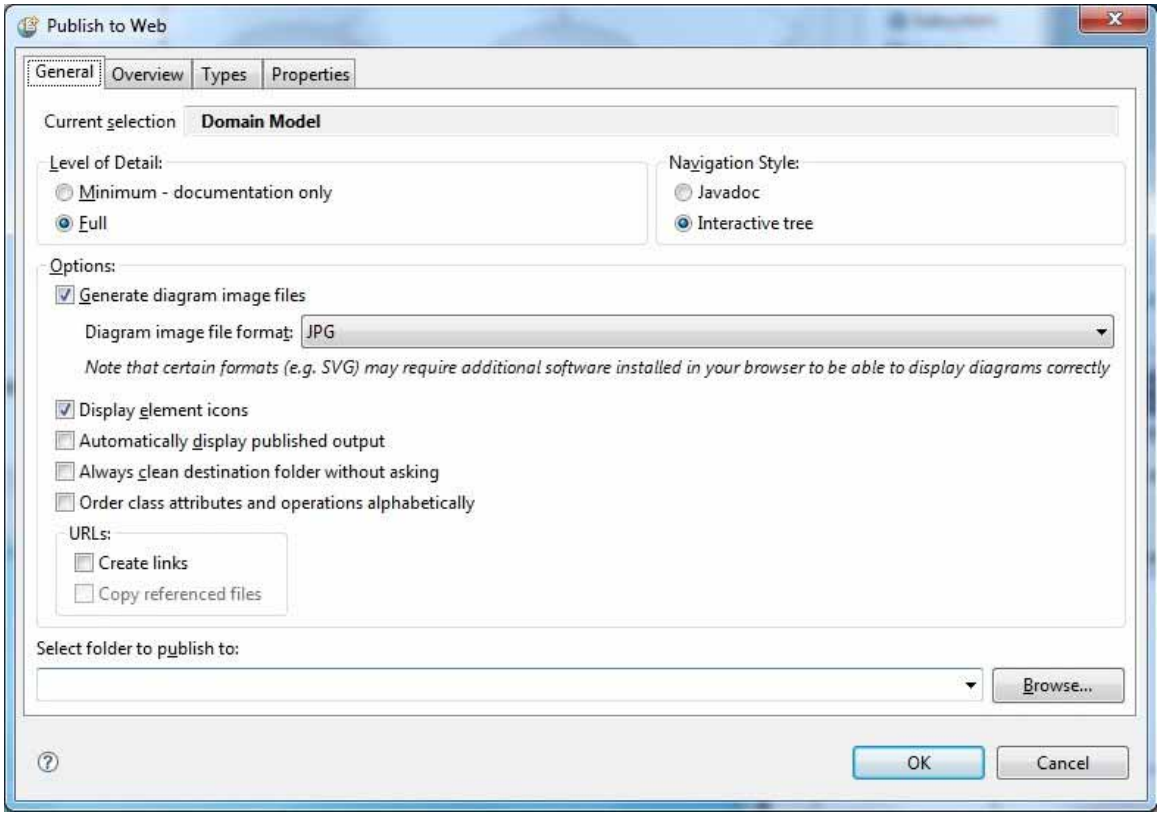
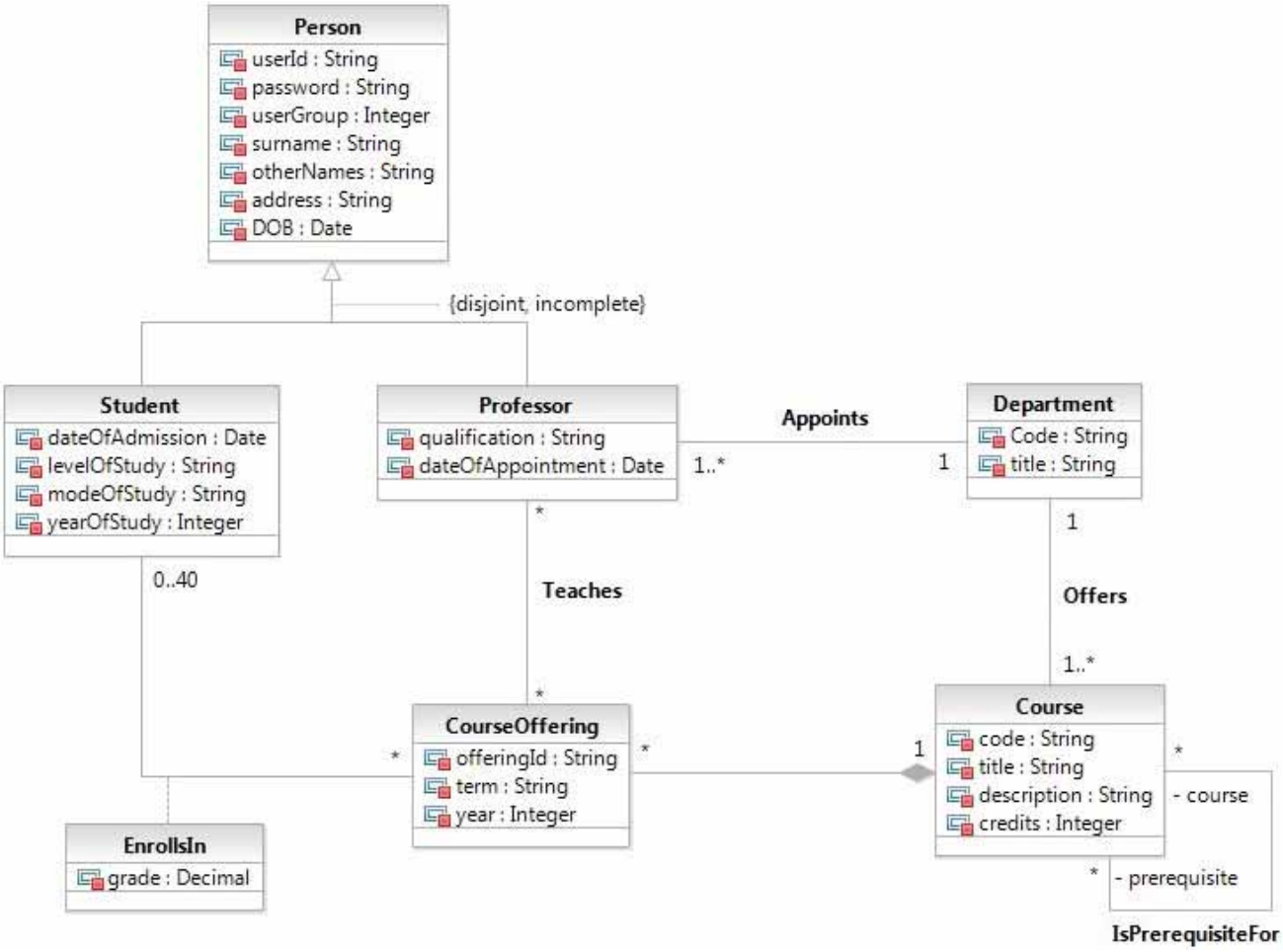
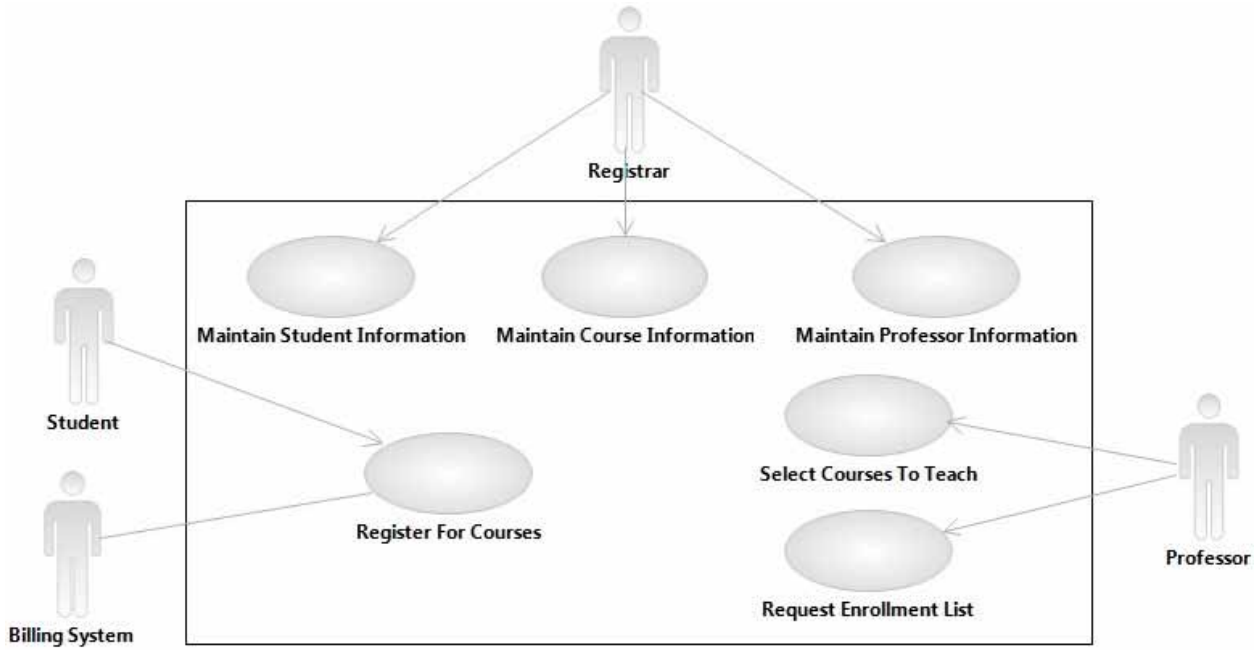


Figure 32: Generating a report.

APPENDIX A: DOMAIN MODEL CLASS DIAGRAM FOR ASU APPLICATION



APPENDIX B: USE-CASE MODEL USE-CASE DIAGRAM FOR ASU APPLICATION



APPENDIX C: GENERAL GUIDELINES AND TECHNIQUES FOR ORGANIZING INTERNAL STRUCTURES OF MODELS IN RATIONAL SOFTWARE MODELER

The primary tool for organizing the content of UML models is the Package, which serves two purposes:

3. partitioning, organizing, and labeling model information by
    - grouping model elements that correspond to a specific subject matter in the problem or solution domain.
    - separating different types of model information such as interfaces, implementations, diagrams, etc.
    - grouping model elements in order to define and control their dependencies on other elements.
    - grouping diagrams that provide alternative views on the same model.
4. establishing namespaces
    - for model elements.
    - for implementation artifacts generated from model elements (this might involve mappings between model and implementation language namespaces).
    - for a unit of reuse.

Traditionally, UP has a proposed specific packaging strategy for various model types. Those strategies are reflected in the model type-specific sections of these lab notes. RSM also introduces some additional organizational tools, which are described in the rest of this section.

Represent Viewpoints Using «perspective» Packages

In cases where it is desirable to see model elements organized in more than one way, you can create additional packages with diagrams that depict the alternate organizational schemes. This same technique can be used anywhere there is a need to represent a particular view on model content that cuts across the model's packaging scheme. RSM supports this technique by providing a «perspective» package stereotype as part of its UML 'base profile'. You can think of a «perspective» package as generally the equivalent of a UP for Systems Engineering or IEEE 1417 "Viewpoint".

Do not place semantic model elements (classes, packages, associations, and so on) within «perspective» packages. Just place diagrams within them that depict views based upon the alternate organizational concern or application viewpoint. Applying the «perspective» stereotype to a package does several things. It visually identifies that package as representing a particular viewpoint. It also supports a model validation rule that warns you when semantic model elements are placed in a «perspective» package. It also serves as a designator of packages that should be bypassed by RSM transformations.

Create Self-Updating Depictions of Specific Concerns Using Topic Diagrams

In contrast to "normal" diagrams wherein you manually place the model elements you wish to depict, the contents of a Topic Diagram are determined by a query that is run against existing model contents. To create a Topic Diagram, you select a "topical" model element, then define what other model elements you wish to appear in the diagram based upon the types of relationship(s) they have to the topical model element. As the semantic content of the model changes, the Topic Diagrams adjust accordingly.

Examine Models via Browse Diagrams

Browse Diagrams are not specifically a tool for model organization. Their purpose is to facilitate discovery and understanding of model content without having to manually compose diagrams. However, in the context of model organization it is good to be aware of them since they might reduce your need to compose persisted diagrams. That, in turn, could reduce the size and complexity of your models, leaving them easier to organize.

Browse diagrams are a bit like Topic diagrams, but with the key difference that Browse Diagrams are never persisted; they are always generated on the fly. To produce a Browse Diagram, you select a model element (from a diagram or the `Project Explorer`), and use the context menu to "Explore in Browse Diagram". This will produce a diagram depicting the selected model element as the 'focal point' with related model elements presented in a radial layout around the focal point. Of course, you can then select one of the related model elements in that Browse Diagram, and make it the focal point of another Browse Diagram, and continue in this manner as long as you like.

Inter-Diagram Navigation

In RSM there are two mechanisms for inter-diagram navigation:

5. It is possible to drag a diagram node from the `Project Explorer` to some other 'host' diagram. Then you can double-click the resultant icon on the host diagram to open the referenced diagram.
6. Whenever you create a new UML package in a model, a "Main" diagram (freeform diagram) is automatically created. By default, this "Main" diagram is created as the 'default' diagram of the package. You can rename the diagram to something other than "Main", and it will still be treated as the 'default'. You can also select a different diagram in the package and make it the 'default' diagram for that package. The purpose of the 'default' diagram is this: if you place the package itself onto some other 'host' diagram, you can then double-click the package, which will open its default diagram.

These mechanisms support the following organizational guidelines, which can apply to models of any type:

7. Compose the Main diagram (or other default diagram) of each modeling file to depict:
   a. each top-level package in the modeling file.
   b. the diagram icons for any other diagrams that reside in the root package of the modeling file (in other words, do not depict the icon for the default diagram itself).
8. Compose the Main diagram (or other default diagram) of each top-level package to depict:
   a. the packages that it directly contains
   b. the diagram icons for any other diagrams that it directly contains
9. Repeat this pattern for each successively lower level of packages.

Acknowledgement

This appendix is extracted from the document *Model Structure Guidelines for Rational Software Modeler and Rational Software Architect (2004 Release)* V1.0 by Bill Smith, Model Driven Development, IBM Rational Software, September 8, 2004.
(Available at http://www-128.ibm.com/developerworks/rational/library/04/r-3155/)