

Introduction to System Architect Student Edition

(A Product of Popkin Software)

**Dr. DOROTHY G. DOLOGITE
Dr. RICHARD D. HOLOWCZAK**
Professors of Computer Information Systems
Zicklin School of Business, Baruch College, City University of New York

1/26/2005

Contents:

Part I: Getting Started with System Architect

- 1. Introduction2**
- 2. Starting Up System Architect3**

Part II: Structured Modeling with System Architect

- 3. Data Modeling using Entity Relationship Diagrams7**
- 4. Process Modeling using Data Flow Diagram17**
- 5. Process Modeling using Functional Hierarchy Diagram20**
- 6. Linking Structured Diagrams.....23**
- 7. Completeness Analysis and Rule-Based Error Checking26**
- 8. Transforming Entity Relationship Models to Physical Database Models28**

Part III: Object Oriented Modeling with System Architect

- 9. Object Oriented Modeling with Use Cases33**
- 10. Object Oriented Modeling with Class Diagrams44**
- 11. Object Oriented modeling with Sequence diagrams59**
- 12. Object Oriented modeling with Collaboration diagrams65**
- 13. Object Oriented modeling with Activity diagrams69**
- 14. Object Oriented modeling with State diagrams74**
- 15. Object Oriented modeling with Component diagrams80**
- 16. Object Oriented modeling with Deployment diagrams82**

Part IV: Reporting, Encyclopedia Management and Additional Features

- 17. Generating Reports.....84**
- 18. Project and Diagram Management87**
- 19. Additional Features of System Architect89**
- 20. Conclusions for this Tutorial93**

Part I: Getting Started with System Architect

1. Introduction

The purpose of this tutorial is to support learning the basic skills and features of the Popkin Software System Architect - Computer Aided Software Engineering (CASE) tool (called simply “SA” here). While the complete CASE tool has hundreds of features capable of addressing the complete enterprise systems development life cycle, this tutorial will focus only on a small, but important subset.

Students who work with this tutorial should have a basic understanding of the systems development life cycle. In particular, the role of different modeling techniques including Entity Relationship, Data Flow and Functional Decomposition should be studied first at the conceptual level (for example, from a Systems Analysis textbook such as those given in the references section or appropriate lectures) prior to undertaking this tutorial. Much of terminology used in this tutorial is derived from common Systems Analysis literature.

Students who work with this tutorial, and who are interested in so-called “object-oriented” methods, should have a basic understanding of the object-oriented development process. Again, a Systems Analysis textbook will provide a foundation prior to undertaking this tutorial.

In addition, students should have a working knowledge of Microsoft Windows (98, ME, NT, 2000 or XP) including working with multiple applications and windows, dialog boxes, left and right-clicking, dragging objects on the screen, and so on. This tutorial was developed based on the Student Edition version 8.8 of System Architect. Most recent versions (such as the prior System Architect 2001 version) have very similar basic features and most if not all of the examples in this tutorial should apply.

How to Use This Tutorial

In general, each section of the tutorial builds on previous materials although some background is given at the start of each section as a review and to set the scene for the lessons to be taught. Students need not complete the entire tutorial all in one sitting. Each of the sections can be completed one at a time. Once a student becomes familiar with navigating around SA and with the basics of drawing diagrams with various tools, rapid progress can be made through the latter portions of the tutorial. Two broad modeling approaches are introduced in this tutorial: Structured Modeling and Object Oriented Modeling. Structured modeling is covered in the following three primary topics:

- Data Modeling using Entity Relationship Diagrams
- Process Modeling using Data Flow Diagrams and Completeness Analysis and Rule-Based Error Checking
- Process Modeling using Functional Hierarchy Diagrams
- Linking Structured Diagrams

It is also recommended that students review Section 17 on Generating Reports, Section 18 on Project and Diagram Management and Section 19 that covers Additional Features of System Architect.

Object Oriented modeling is covered in the following topics:

- Object Oriented Modeling with Use Cases
- Object Oriented Modeling with Class Diagrams
- Object Oriented modeling with Sequence diagrams
- Object Oriented modeling with Collaboration diagrams
- Object Oriented modeling with Activity diagrams
- Object Oriented modeling with State diagrams
- Object Oriented modeling with Component diagrams
- Object Oriented modeling with Deployment diagrams

2. Starting Up System Architect

In this section, the basic techniques for starting up the Popkin Systems Architect Student Edition will be described. SA is a Windows software application that runs within the Microsoft Windows 98, ME, NT, 2000 or XP operating system. As with most Windows applications, starting SA is simply a matter of navigating the Windows Start menu as shown here:



Start the program by going to "Start" menu, selecting All Programs and then Popkin Software. Then click on the System Architect as shown in Figure 1:

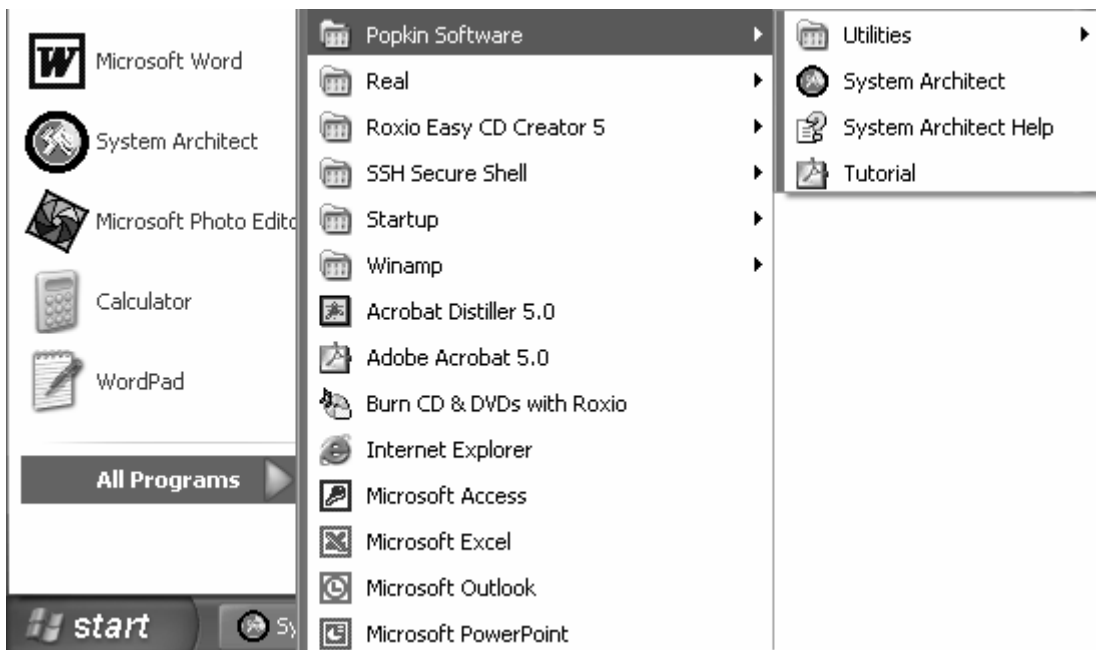


Figure 1 Windows Start menu showing Popkin Software and System Architect menus and icon

Note that on your own PC or in your School's computer lab, this menu may appear slightly differently.

When the software is started for the first time you may also see the Audit ID dialog box as shown in Figure 2. Type in any integer for the Audit ID field and then click on the OK button to close the dialog box.

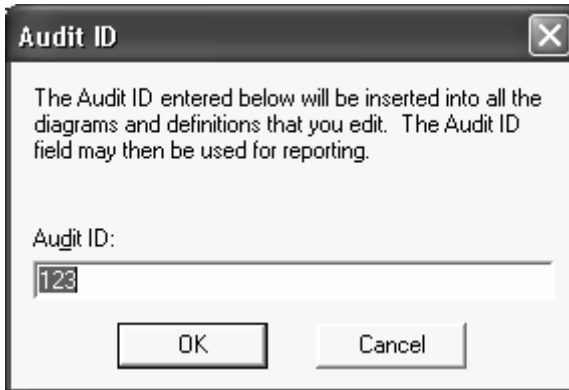


Figure 2 The Audit ID Dialog box

As with the previous dialog box, you may be prompted with a Configuration dialog box that will appear as shown in Figure 3.

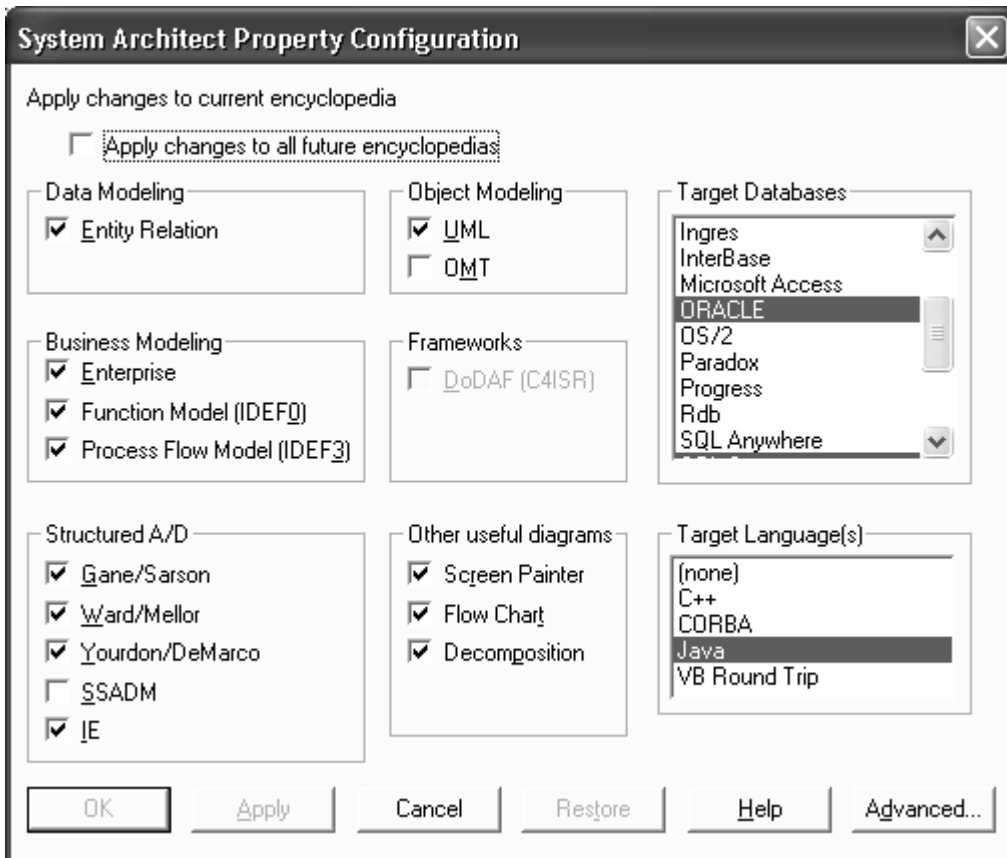


Figure 3 Configuration Dialog box

For the Student Edition of SA, simply click on the **Cancel** button. Once this is done, this dialog box may not appear in the future.

Next, you may see a SA “Tip of the Day”. Click on **Close** button to close this dialog box.

Once these few startup dialog boxes have been dismissed, the main SA screen should appear. The main screen is shown in Figure 4.

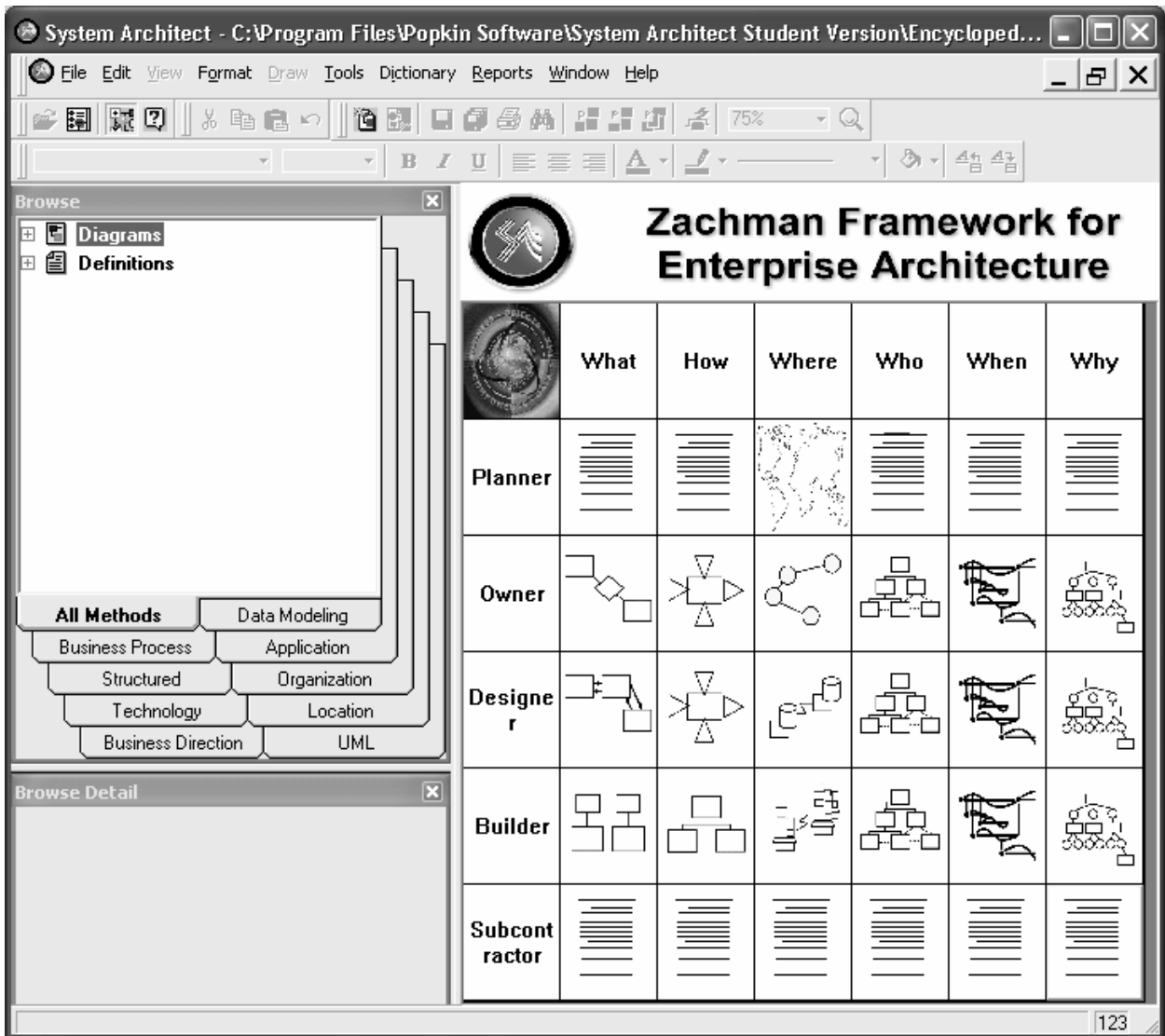


Figure 4 Main SA Screen

The SA main screen is divided into four main sections. Along the top are a series of menus and a button bar. The *Browser* appears on the left hand side and *Browse Detail* appears in a small window below it. The right hand side of the screen is reserved to display diagrams and models.

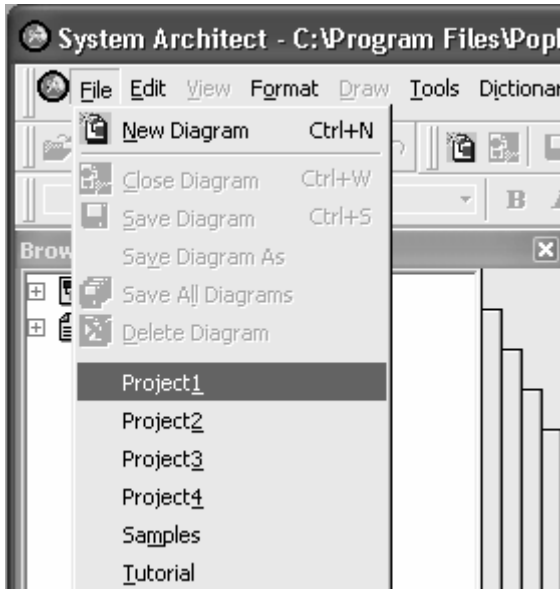
The Browser is organized into a series of tabbed sheets as can be seen in Figure 4. By default, the *All Methods* sheet is displayed. Depending on the size and resolution of the PC's screen, the application window may appear smaller and this may cause the Browser to display only abbreviations (e.g., "AM" for All Methods) on the tabs. If this is the case, try maximizing the main application window or drag the separator between the Browser window and the diagram display to the right to expand the Browser window.

All of the related models for a project are stored in a *SA Encyclopedia*. This is also called a Data Dictionary or Repository in most CASE tools and in systems analysis textbooks. In the full version of SA, the user may save a related set of models for a project into an encyclopedia that

they name. However, the Student Edition of SA is limited to only four such encyclopedias named: Project1, Project2, Project3 and Project4.

For the purposes of this tutorial, Project1 is assumed to be empty and different related models will be created in that encyclopedia. Later in the tutorial, Project2 will be used to demonstrate some more advanced features of SA.

To make certain the proper encyclopedia has been chosen, at this time, pull down the File menu and select the Project1 menu item as shown in Figure 5.



Technical Note: Encyclopedias are actually directories on the hard disk where SA is installed. Use Windows Explorer to look in the directory where SA is installed (C:\Program Files\Popkin Software\System Architect Student Edition by default) and locate the Project1 directory there.

Figure 5 Selecting the Project1 Encyclopedia

Should the Configuration dialog box appear (as in Figure 3), simply click Cancel to close it.

In the next section, a basic Entity Relationship diagram will be defined using the **Data Modeling** tab.

Part II: Structured Modeling with System Architect

3. Data Modeling using Entity Relationship Diagrams

The following section is based on materials about data modeling and ER diagrams found in most academic textbooks on systems analysis and design. In this section, a basic Entity Relationship data model will be created with two entities and a simple relationship between them. Make certain that the Project1 encyclopedia is currently selected.

The example business for this portion of the tutorial is an on-line music store. In the music store, customers are able to browse a catalog of compact discs (CDs) and purchase them over the

Internet using a credit card. In this section the Customer, their Customer Order and the Order Items will be modeled and the resulting diagram should appear as in Figure 6.

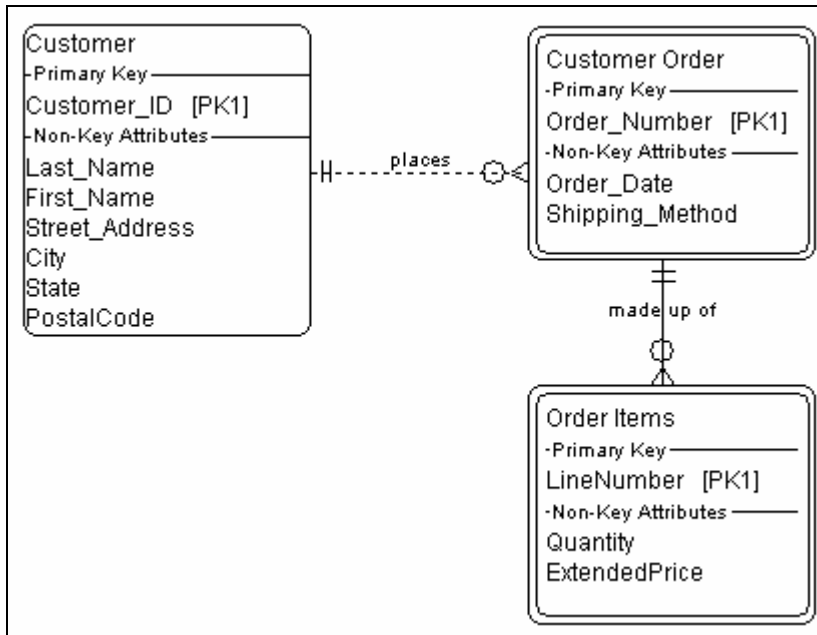


Figure 6 The Completed Entity Relationship Diagram for this exercise

To create a new diagram, click on the **Data Modeling** tab. The entries **Models** and **Shared** should appear. Pull down the File menu and choose the New Diagram menu item.

Select a new type of diagram by double clicking on the *Entity Relation* item in the list as shown in Figure 7.

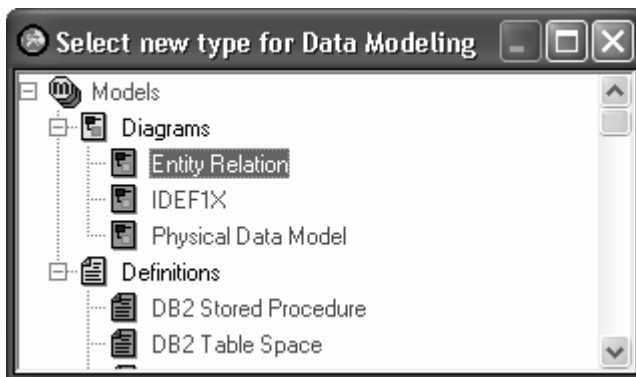


Figure 7 Selecting the Entity Relation diagram

A dialog box should appear asking for a name of the new diagram. Name this diagram `Music Store` and click on the OK button.

The next dialog box will prompt for the name of the data model. For this example, type “Project Data Model 1” and click on the OK button. No other options need to be selected at this time.

At this point, several new objects are created in the Project1 encyclopedia. To see the new entries, go to the Browse window and click on the + signs next to **Models** and open the sub entries of **Diagrams** and **Entity Relation** under Project Data Model 1. They should appear as in Figure 8.

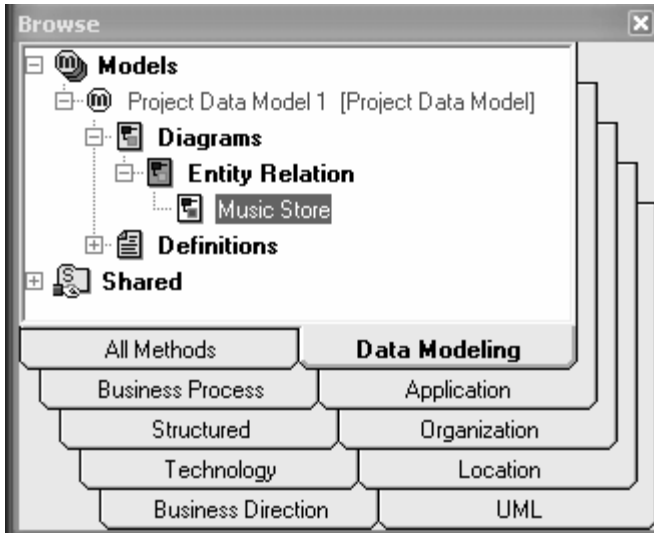


Figure 8 New Data Model and ER diagram

To start drawing the actual ER diagram, double click on the Music Store entry (as shown above). Note that the Browse Detail window shows a blank white rectangle and the right hand window is also blank and ready for drawing.

In this example three entities called “Customer”, “Customer Order and “Order Items” will be created as seen above in Figure 6. To begin creating an entity, locate the Entity icon on the

button bar as shown in Figure 9 and click on it. The mouse cursor should change to a pen once moved into the white drawing space.



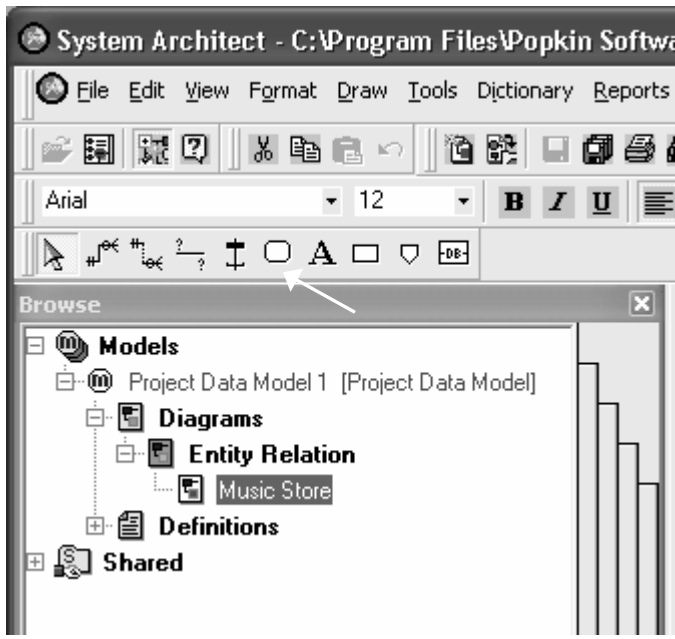


Figure 9 The button bar for Entity Relation diagrams with the Entity tool highlighted

Using the Entity tool, click on the empty space in the right hand window. When the new entity appears, it is given a default name. Type over this name with the new name of the entity: *Customer* as shown in Figure 10.

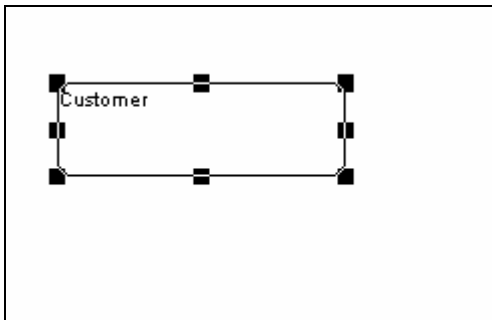


Figure 10 Naming the new entity Customer

Click the entity tool once more in the blank space to the right of the Customer entity and name this new entity *Customer Order*. Finally, click the entity tool in a space below the Customer Order entry and name this third entity *Order Items*. Should you forget to rename the entities or mistype their new names, edit the names as described in the following paragraph. The arrangement of the three entities should appear similar to those in Figure 11.

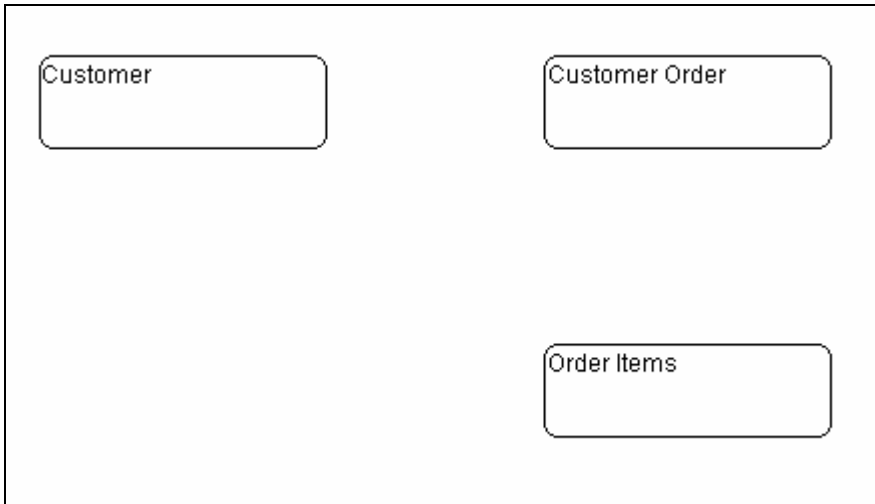



Figure 11 Three new entities arranged on diagram

Editing Names and Attributes for the Entities

Go back to the drawing toolbar and get rid of the pen-shaped cursor by clicking on the “Select-Mode” icon (the arrow ) on the very left. Now the selection cursor should be back in effect. This selection cursor can be used to drag and move the entities around, for example, if an item was created too close to another item. The selection arrow can also be used to edit the name of an entity as well as to add attributes to the entity.

In this example, attributes will be added to the Customer entity. Position the cursor over the Customer entity and click with the right mouse button. Then choose the **Edi t** item from the pop-up menu. In the dialog box that appears, the entity name can be changed or corrected if necessary and the space below the entity can be used to add attributes. For this example, the following attributes will be added:

Attribute Name	Data Type	Qualifiers	Primary Key?
Customer_ID	Character	10	Yes
First_Name	Character	30	No
Last_Name	Character	30	No
Street_Address	Character	40	No
City	Character	20	No
State	Character	2	No
PostalCode	Character	10	No

Fill in the dialog box as it appears in Figure 12. Be certain to check off the PK column for the Customer_ID as this will indicate Customer_ID is the primary key (or identifier) of the Customer entity.

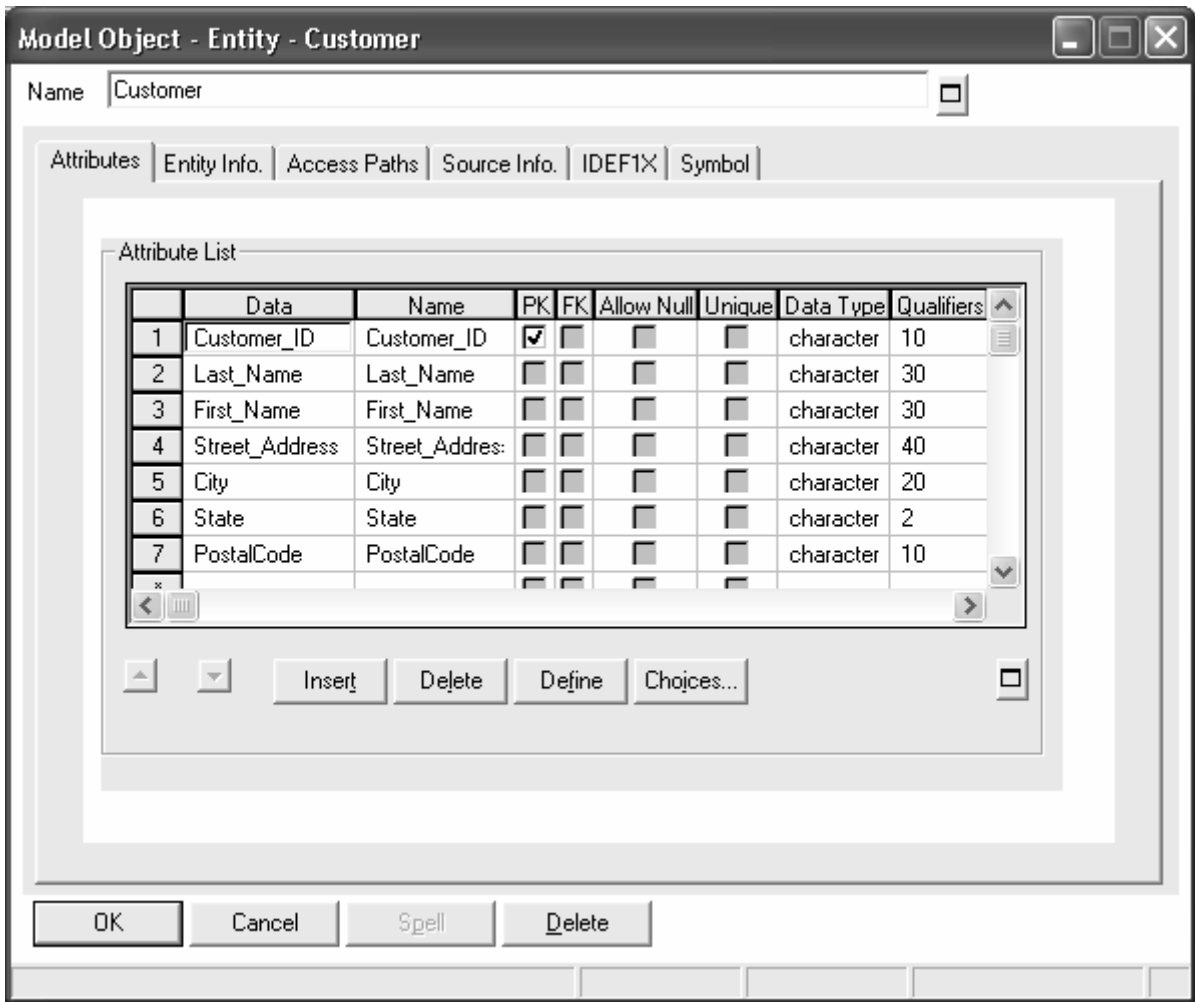


Figure 12 Adding attributes to the Travel Agent Entity

Note that the names provided in the Data and Name columns should be the same. Data Types can be chosen from the drop down list that appears when the cursor is in the Data Types field. Qualifiers give the size of the attribute and optionally the precision for numeric data types. Once the attributes have been added, click on the OK button to close the dialog box. The display should now change to show the attributes for the Customer entity. It may be necessary to resize the entity (expand it vertically) for all of the attributes to display.

Repeat the above exercise by adding the following attributes to the Customer Order entity:

Attribute Name	Data Type	Qualifiers	Primary Key?
Order_Number	character	10	Yes
Order_Date	date	leave blank	No
Shipping_Method	character	20	No

Finally, repeat the above exercise by adding the following attributes to the Order Items entity:

Attribute Name	Data Type	Qualifiers	Primary Key?
----------------	-----------	------------	--------------

LineNumber	Integer	3	Yes
Quantity	Integer	7	No
Extended_Price	Number	10, 2	No

The diagram at this point should look similar to Figure 13. It may be necessary to expand the size of each entity slightly so that all of the attributes are displayed,

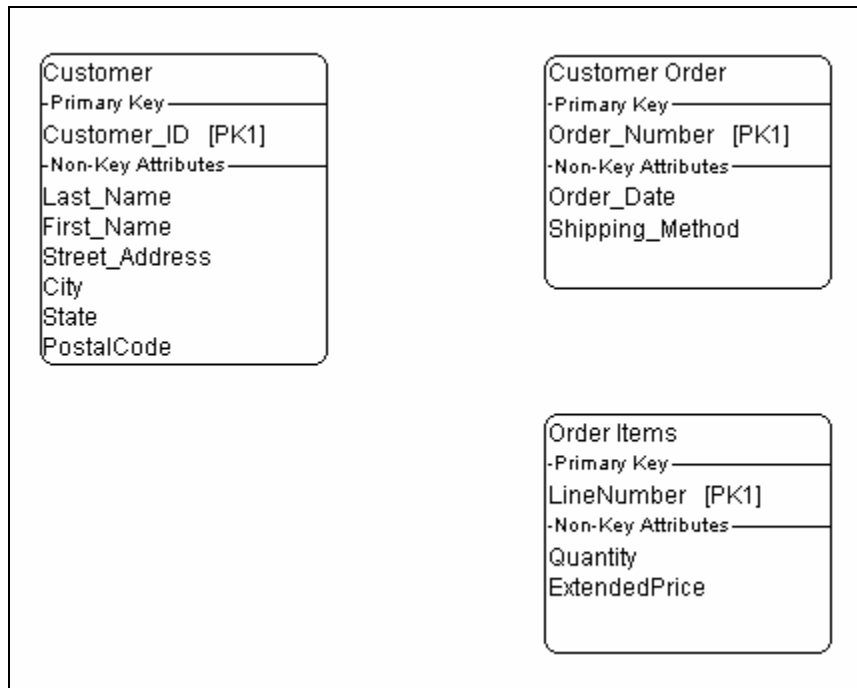



Figure 13 Three entities with attributes added

Defining a Relationship among the Entities

At this point, three entities have been defined. Now it is time to link these entities with relationships. In this example, we will link the entities with one-to-many relationships. A one to many relationship may be either:

- **Identifying:** In this case the key from the parent entity (one side) is combined with the key from the child (many side) entity to form a composite key.
- **Non-Identifying:** In this case the key from the parent entity appears as a foreign key only in the child entity.

Return to the drawing toolbar and select the *non-identifying* relationship tool:  The mouse cursor will take on the shape of a pen with a jagged line next to it.

Position the cursor over the Customer entity and click the left mouse button once. A bold plus sign **+** will appear on the edge of the entity. Then position the cursor over the Customer Order entity and click the left mouse button again. Do not hold the mouse button down while moving to the second entity.

A line representing the relationship will automatically be drawn between the two entities and a space will appear on which the name of the relationship can be entered. Type the word *places* as the title of this relationship. We would like to imply that the Customer “places” a Customer Order (or in this case, many customer orders). The relationship name can be changed or corrected by editing the relationship as described next.

To edit the relationship, return to the tool bar and select the selection tool (the arrow). Right click on the relationship line and choose the **E**dit item from the pop-up menu. The Relationship dialog box will appear as shown in Figure 14. Make any necessary changes to the names of the relationship and also add a relationship name for the reverse direction (called the Reverse Phrase). Use *placed by* as the reverse phrase.

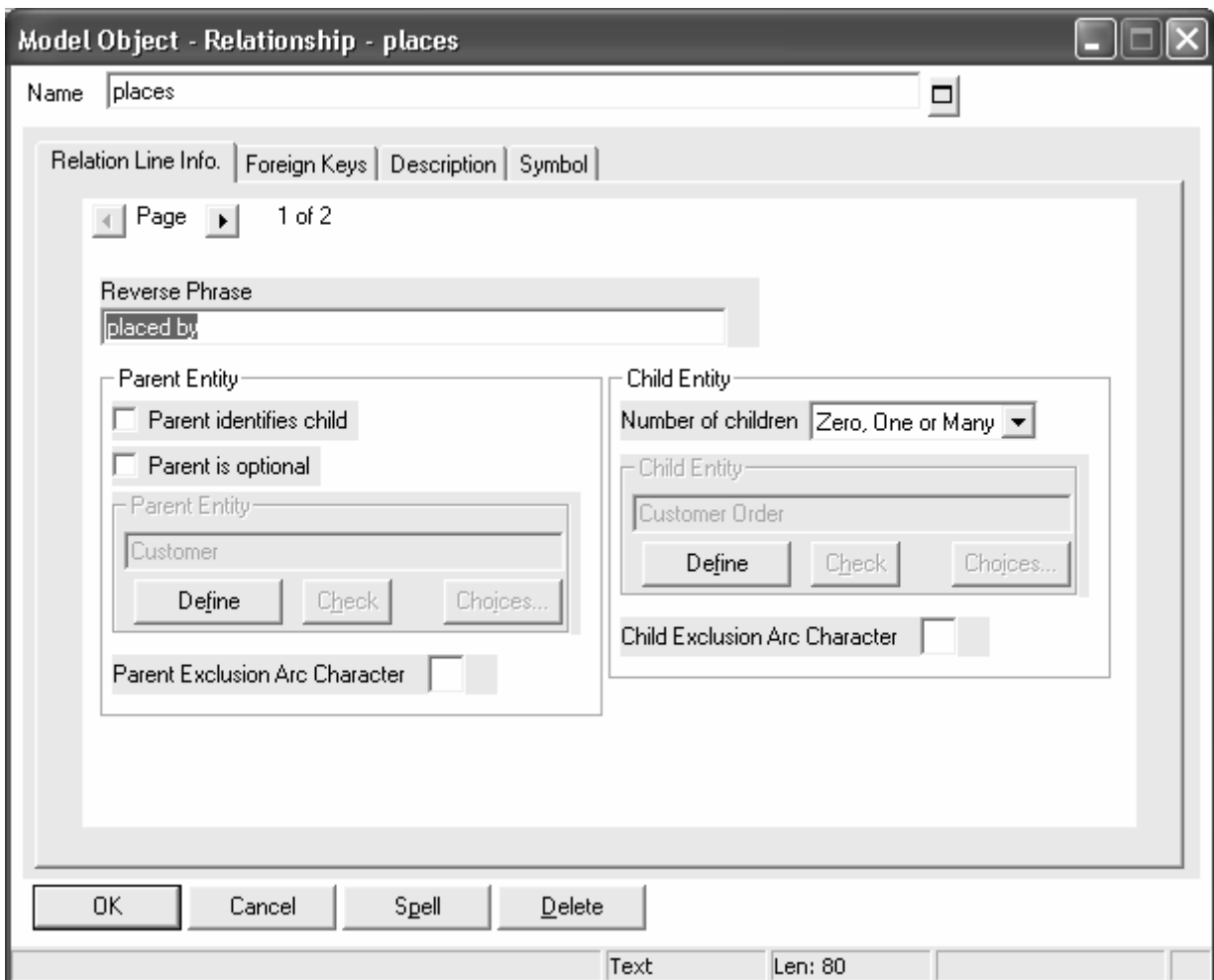



Figure 14 Relationship dialog box

After defining a relationship, the child entity (Customer Order) will be considered as a dependent of the parent entity (Customer) and therefore will be presented by double border.

If a relationship is specified incorrectly (for example if it is drawn in the wrong direction or between the wrong entities), it can be deleted by highlighting the relationship with the select tool

and by then pressing the Del (Delete) key on the keyboard (or by pulling down the Edit menu and choosing the Delete menu item).

Next, draw an Identifying relationship between the Customer Order entity and the Order Items entity. Select the identifying relationship tool:  Note that this appears next to the selection tool. The mouse cursor will take on the shape of a pen with a jagged line next to it.

Place the cursor over the Customer Order entity and click the left mouse button once. Then position the cursor over the Order Items entity and click the left mouse button again. Type the phrase *made up of* as the title of this relationship. We would like to imply that the Customer Order is “made up of” a one or more Order Items. Edit this new relationship and use *appears on* as the reverse phrase. The diagram at this point should appear as in Figure 15.

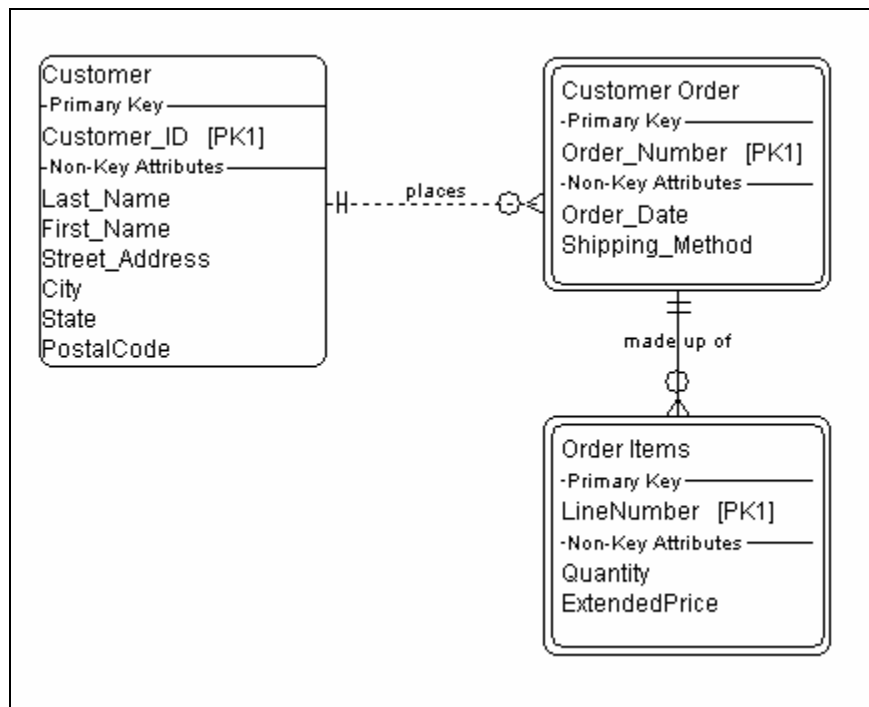


Figure 15 Three entities with attributes and relationships

To save the work completed thus far, press CTRL + S or pull down the File menu and choose the Save Diagram menu item. Click “Yes” if asked to confirm.

Additional details on changing the way ER models are displayed, including the display of foreign keys are given in Section 19.

Exercise: Editing an existing Entity and Creating an additional Entity and Relationship

In this exercise, an additional entity called **Product** will be added to the above diagram and then related to the existing Order Items entity. An outline of the steps follows:

- a) Use the Entity tool to add a new entity named “Product”. Place this to the left of the Order Items entity

b) Add the following attributes to the Product entity:

Attribute Name	Data Type	Qualifiers	Primary Key?
Product_ID	Character	10	Yes
Catalog_Number	Character	20	No
Album_Title	Character	40	No
Artist	Character	40	No
Release_Date	Date	Leave blank	No
Record_Label	Character	30	No
Standard_Price	Number	10, 2	No

c) Create a non-identifying relationship from Product to Order Items. Label this relationship “sold on” (reverse phrase is “selling a quantity of”)

d) Finally, edit the Customer entity and add three additional attributes as follows:

Attribute Name	Data Type	Qualifiers	Primary Key?
C_Username	Character	20	No
C_Password	Character	30	No
Email_Address	Character	40	No

The final result should appear as in Figure 16:

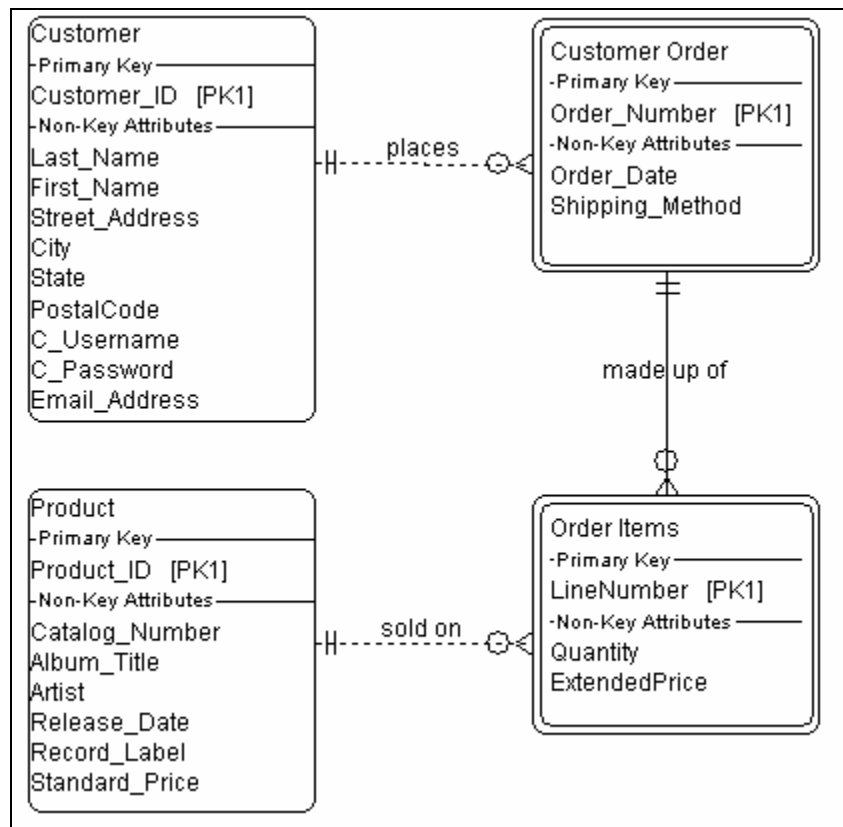


Figure 16 Completed Entity Relationship modeling exercise

When this exercise is completed, be certain to save the diagram (press CTRL + S or pull down the File menu and choose the Save Diagram menu item) and then close the diagram by pulling down the File menu and choosing the Close Diagram menu item.

Additional details on changing how ER models are displayed can be found in Section 19.

4. Process Modeling using Data Flow Diagrams

The following section is based on materials about data modeling and data flow diagrams found in student textbooks. In this section, the steps to create data flow diagrams will be introduced. Data Flow Diagrams (DFDs) represent the flow of data through an organization's business processes. DFDs deal with three main types of objects: Processes, Data Stores (internal to the organization) and external data sources. Figure 17 shows the DFD representing the customer registration and product browsing business process. Note that data enters the organization via the external entity Customer and then flows through the process to register the Customer in the database and then to browse the product catalog process. The browse process also requires information from the Products data store.

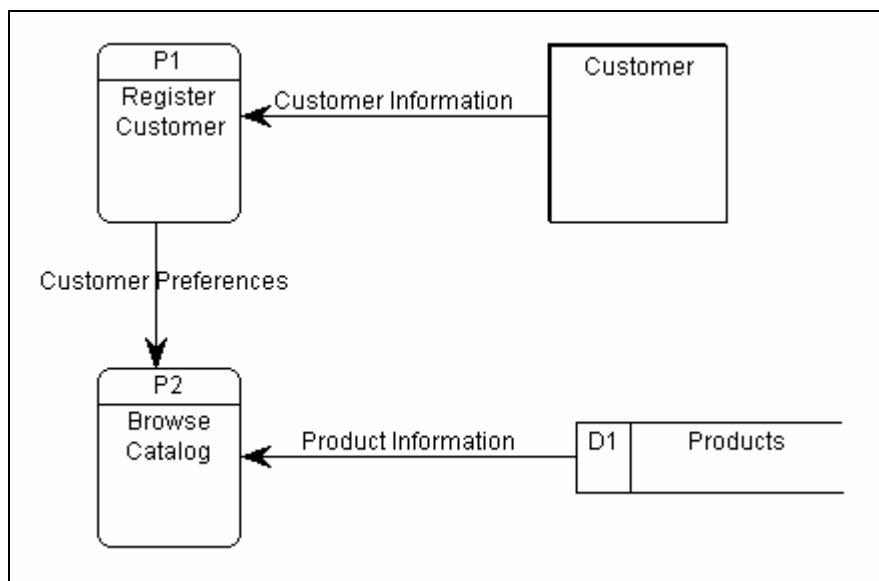


Figure 17 Completed Data Flow Diagram for the Browse Catalog business process

Before starting this section, make certain SA is running and that Project1 is the current Encyclopedia selected (see Section 2 for details). To begin drawing the DFD represented in Figure 17, select the **Structured** tab in the Browse window. Two entries: Diagrams and Definitions should appear. Right click on the Diagrams entry and choose the *New* item from the pop up menu. Double click on the entry "Data Flow Gane & Sarson" from the dialog box that appears as shown in Figure 18.

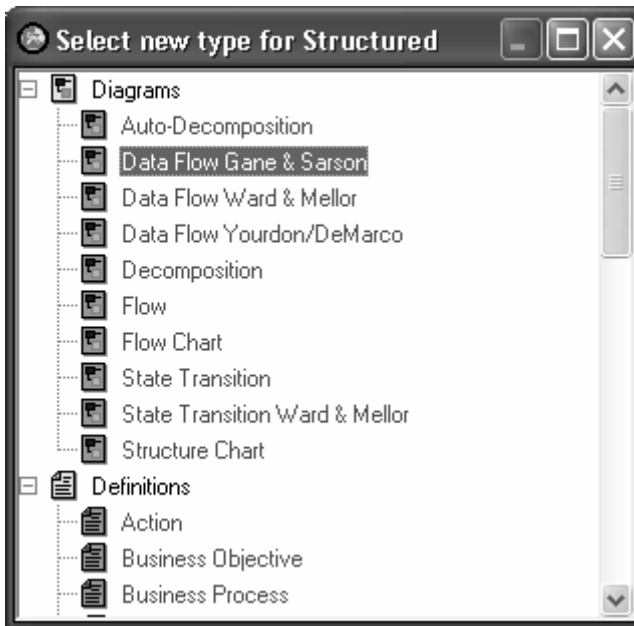
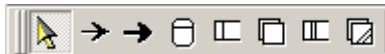



Figure 18 Selecting the Data Flow Gane & Sarson diagram (model) model type


Name this new diagram “Browse Catalog” when prompted and click on the OK button to continue. At this point, the new blank diagram has been created. Navigate the Browse menu by opening up the Diagrams entry and then Data Flow Gane & Sarson entry. Click on the new diagram called Browse Catalog that was just created to make that diagram appear.


For this diagram, two processes, one data store and one external entity will be created. Each of these objects has a corresponding icon in the button bar along the top of the main window. They are shown here:



Starting on the left, the tools include the general selection tool (arrow), the Data Flow tool, the Materials flow tool, the Process tool, the data store tool and the external data tool. The remaining two are the multiple data store tool and multiple external data tool (these will not be used in this tutorial).

Select the Process tool  and click in the right hand window to create a new process. As soon as a process has been created, type over the default name and label the process as “Register Customer”. Press Enter to confirm the new name. Next, click with the Process tool again below the Register Customer process and name this second process “Browse Catalog”. Refer to Figure 17 for placement and naming. Use the selection tool to reposition the processes as needed.

Next, select the External data tool  and click in the right hand window to create an external data source. Name this external data source “Customer”.

Third, select the Data Store tool  and click in the right hand window to create the new data store. Label this data store “Products”.

If any of the labels are missing or typed incorrectly, use the selection tool to right click on an object and choose the `Edit` item from the pop up menu. This will provide a dialog box where the names of the processes, external data and data stores can be changed. To add numbers to the process and data store symbols (e.g., “P1”, “P2” and so on), edit the process or data store and click on the “Symbol” tab. Then specify a new number at the “Number” prompt as shown in Figure 19.

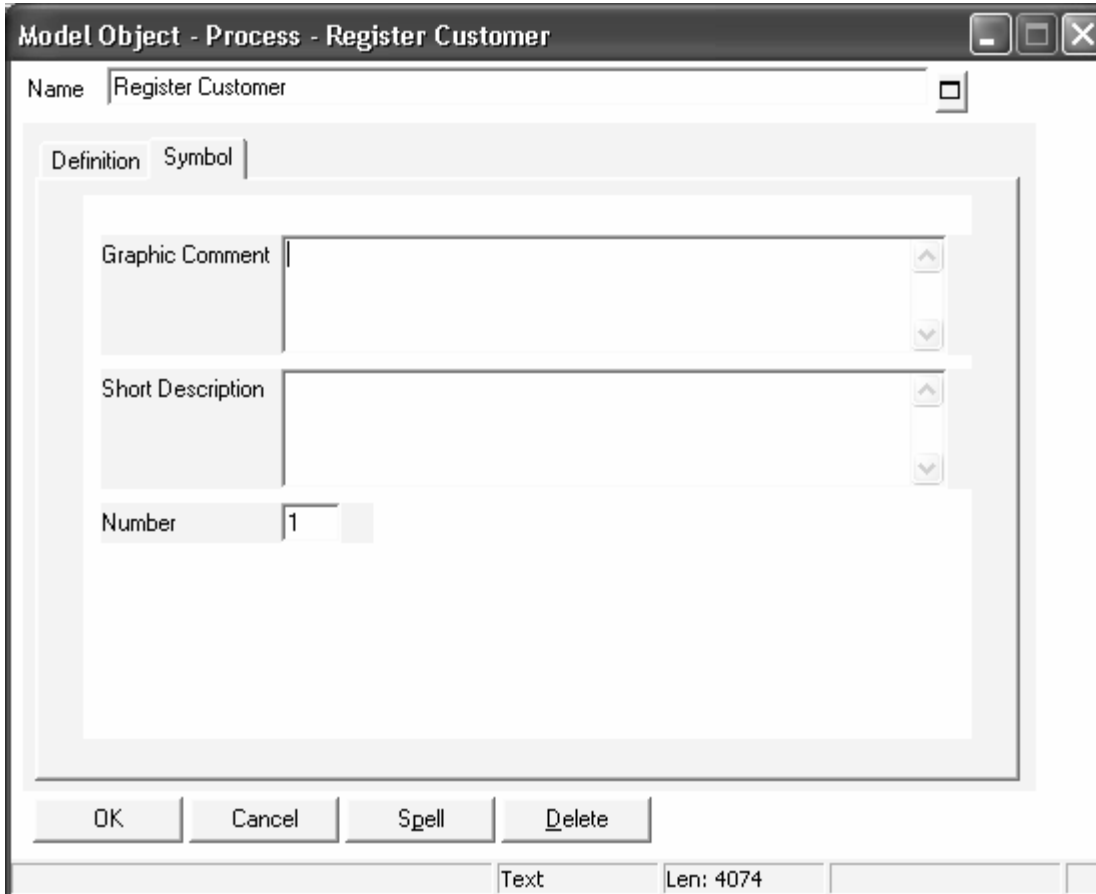



Figure 19 Specifying a Process number

Again, refer to Figure 17 for naming and placement of the objects in the diagram.

At this point all of the individual processes, external entities and data stores have been defined.

The next step is to connect them with Data Flows. To accomplish this, the Data Flow tool  will be used. To create the first data flow, select the Data Flow tool and click on the External Entity “Customer”. A bold plus sign **+** will appear on the Customer external entity. Move the cursor to the left and then click on the Process “Register Customer”. Name this data flow “Customer Information” by typing in the field and pressing Enter. When the Associative Properties dialog box appears, click on the OK button.

In a similar fashion, use the Data Flow tool to specify a flow from the Process “Register Customer” to the Process “Browse Catalog”. Label this flow “Customer Preferences”.

Finally, use the Data Flow tool to make a connection from the Data Store “Products” to the Process “Browse Catalog”. Label this flow “Product Information”.

If a data flow is incorrectly specified (for example, creating a flow between the wrong processes), it can be removed by highlighting the data flow with the Select tool and then pressing the Del (Delete) key on the keyboard (or by pulling down the Edit menu and choosing Delete).

This completes the construction of the Data Flow diagram. To save the work completed thus far, press CTRL + S or pull down the File menu and choose the Save Diagram menu item. Click “Yes” if asked to confirm the save. Before proceeding to the next section, close the diagram by pulling down the File menu and choose the Close Diagram menu item

An exercise on extending this DFD is given in Section 7.

5. Process Modeling using Functional Hierarchy Diagrams

Section 3 introduced modeling the data that an organization’s information system would be required to store and Section 4 discussed modeling the flow of data through an organization’s business processes. In this section, process modeling is viewed from the perspective of the functions a business is required to undertake. The example functional hierarchy for this business is shown in Figure 20. The on-line music store carries out two main organizational functions: Purchasing Wholesale Products and Selling Retail Products.

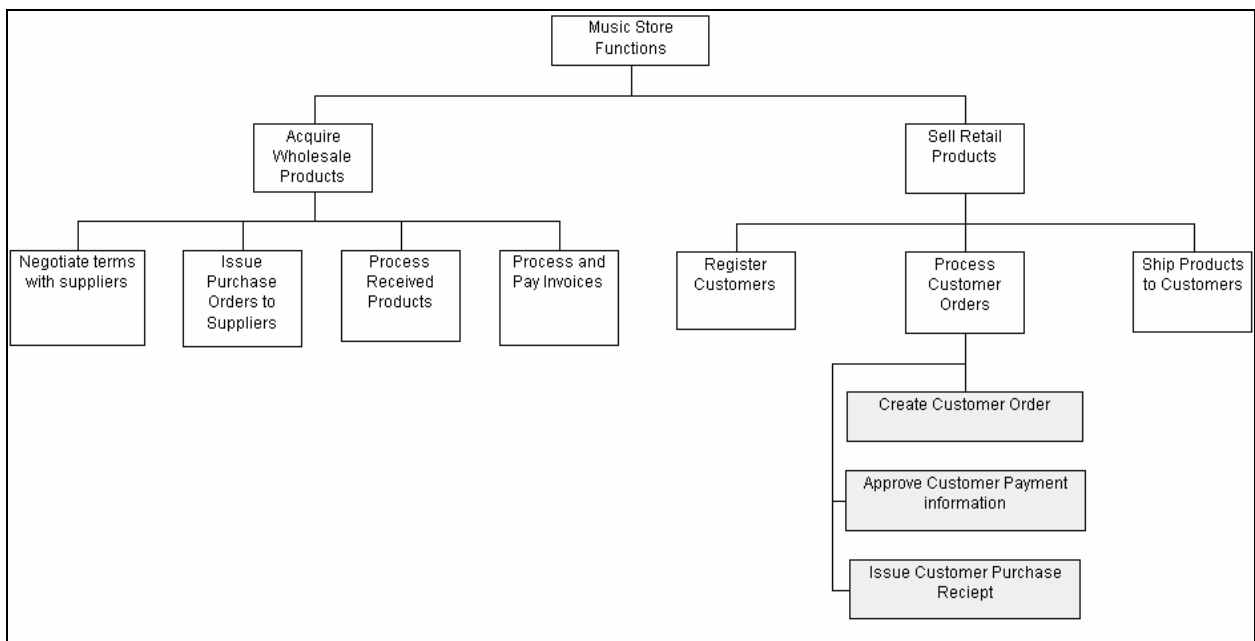


Figure 20 Completed Functional Hierarchy example

To begin drawing the function decomposition diagram, click on the **Business Process** tab in the Browser. Right click on the Diagram entry and select the *New* item from the pop-up menu. Double click on the Functional Hierarchy item in the dialog box as shown in Figure 21. Name the new diagram “Music Store Functions” and click on the OK button to create a blank diagram.

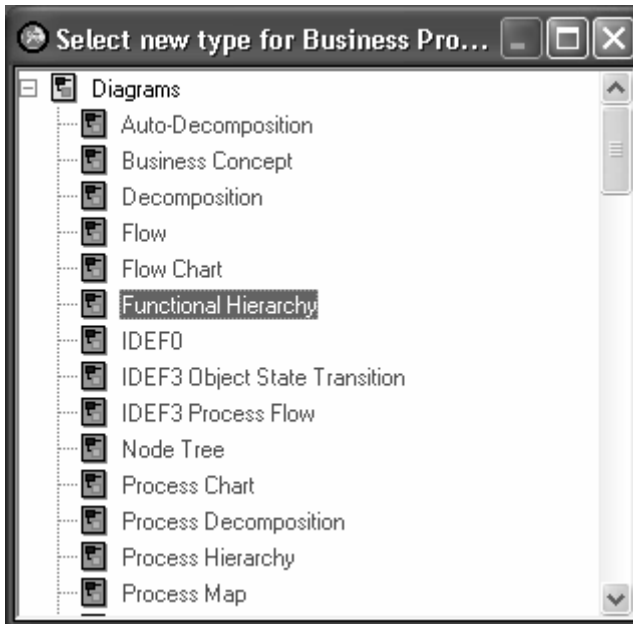





Figure 21 Selecting a Functional Hierarchy diagram

To begin drawing, open the Diagrams and Functional Hierarchy entries in the Browser window and then double click on the “Music Store Functions” diagram name. The new blank diagram should appear in the right hand window.

Each of the upper level nodes in the functional hierarchy will be created using the Organizational Function tool  while the lowest level nodes will be created using the Elementary Business Process tool .

Start by selecting the Organizational Function tool . Click on the upper part of the diagram to create a new function. Name this first function “Music Store Functions”. Create two additional functions just below the Music Store function named “Acquire Wholesale Products” and “Sell Retail Products”. Note that lines may be automatically drawn between the parent function and child functions depending on where the additional functions are drawn on the diagram. If lines automatically join the function boxes, simply drag them apart with the Select tool. At this point, the diagram should look similar to Figure 22.

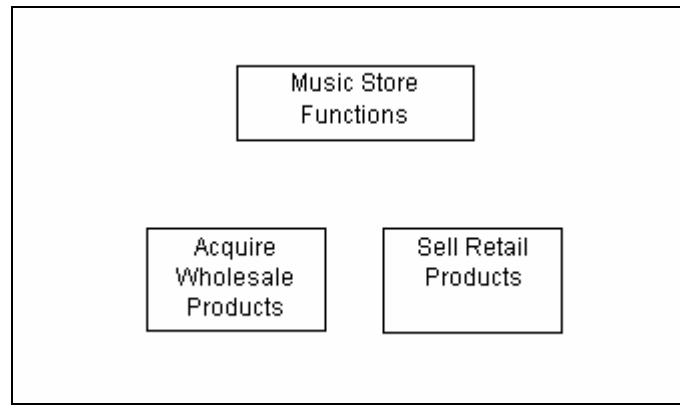


Figure 22 Adding three business functions to the diagram


Create four additional organizational functions on a third level under Acquire Wholesale Products according to Figure 20. Their names are:

- Negotiate terms with suppliers
- Issue Purchase Orders to Suppliers
- Process received products
- Process and pay invoices

Create three additional organizational functions on a third level under Sell Retail Products according to Figure 20. Their names are:

- Register Customers
- Process Customer Orders
- Ship Products to Customers

Note that at this time, no lines will be drawn connecting the organizational functions. By default, SA colors organizational functions yellow.

Next, switch to the Elementary Business Process tool  and create three new business processes below the Process Customer Orders. Note that the default color for elementary business processes is green. Name these three business processes:

- Create Customer Order
- Approve Customer Payment information
- Issue Customer Purchase Receipt

If any changes to process names are required, use the Selection tool to right-click on a process and choose the `Edit` item from the pop up menu. Make any necessary changes and then click OK to confirm.

Finally, connect the organizational functions and elementary business processes together as shown in Figure 20. Use the Selection tool to drag an object until it overlaps with its parent. For example, drag the “Acquire Wholesale Products” block until it overlaps with the “Music Store Functions” block. Once they overlap, a solid black plus sign **+** will appear indicating the connection will be made (see Figure 23). At that point, release the mouse button to complete the connection. The remaining blocks should re-organize themselves.

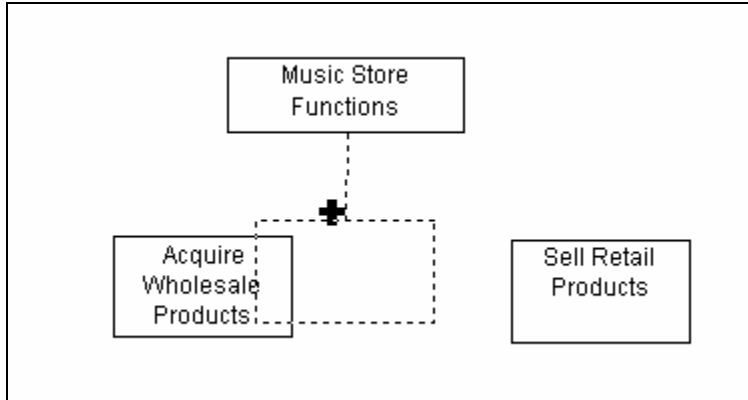


Figure 23 Connecting blocks in the Functional Hierarchy diagram

Perform the same operation to connect the elementary business processes to the respective organizational functions as shown in Figure 20. As a final step, the layout of the lowest level of the tree can be changed by right-clicking on a parent node (such as “Process Customer Orders”) and choosing the Arrange Children Vertically item from the pop-up menu.

As with all other diagram types, to save the work completed thus far, press CTRL + S or pull down the File menu and choose the Save Diagram menu item. Click “Yes” if asked to confirm.

Exercise: Creating additional Functions and Elementary Business Processes

For this exercise, continue the development of the On-line music store Functions diagram by adding the following:

- a) Add a new Organizational function called “Manage Inventory” under the “Music Store Functions” function.
- b) Add three new Organizational functions called “Intra-Warehouse Moves”, “Inter-Warehouse Moves” and “Physical Inventory”. Create all three of these under the “Manage Inventory” function.
- c) Add new elementary business processes under the “Physical Inventory” function. Think about how a physical inventory might be accomplished in a warehouse. (e.g., printing current inventory location records, checking observed quantities with inventory records, etc.) and fill in the Elementary Business Processes you believe an inventory manager would need to undertake to perform a physical inventory of a warehouse.

If proceeding to section 6 on Linking Structured Diagrams, leave the current Functional Decomposition diagram open.

6. Linking Structured Diagrams

In the previous sections, the steps to create and manipulate Entity Relationship diagrams, Data Flow Diagrams and Functional Decomposition diagrams were presented. At this point, each of these diagrams is a separate representation of some aspect of the business information system

being developed. However, there are relationships among these different diagrams. For example, Entities in an ER diagram may represent data stores from a DFD. Similarly, a DFD may provide the data flows and functions used to carry out a particular function in a Functional Decomposition diagram. Linking diagrams enables system analysis and designers to quickly navigate between relevant portions of each of the different types of diagrams as well as make the connections between the models explicit.

In this section, techniques used to link diagrams will be introduced. In order to proceed with this section, Section 3 on developing an ER model, Section 4 on creating a process model using a DFD and Section 5 on creating a functional decomposition must have been completed.

For a first example, the “Register Customers” function in the Music Store Functions diagram created in Section 5 will be linked to the “Browse Catalog” data flow diagram created in Section 4. To begin this process, open the Music Store Functions functional diagram. Right click on the “Register Customers” function and choose the `Child Attach` menu item from the pop-up menu as shown in Figure 24.

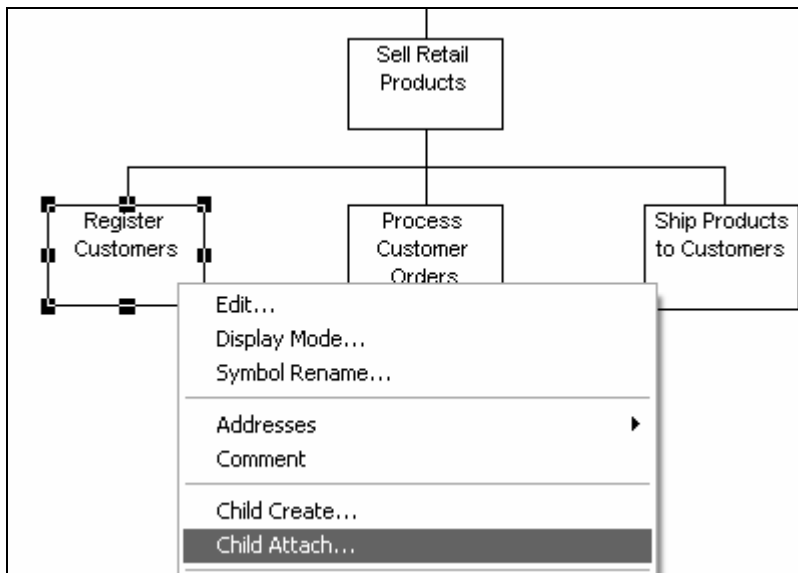


Figure 24 Attaching a child diagram to a business function

A list of existing diagrams in the current encyclopedia will appear as shown in Figure 25. For this example, highlight the “Browse Catalog” diagram and click the `Attach` button. At this point a prompt to save the current diagram will appear. Click the `Yes` button and the DFD diagram should appear.

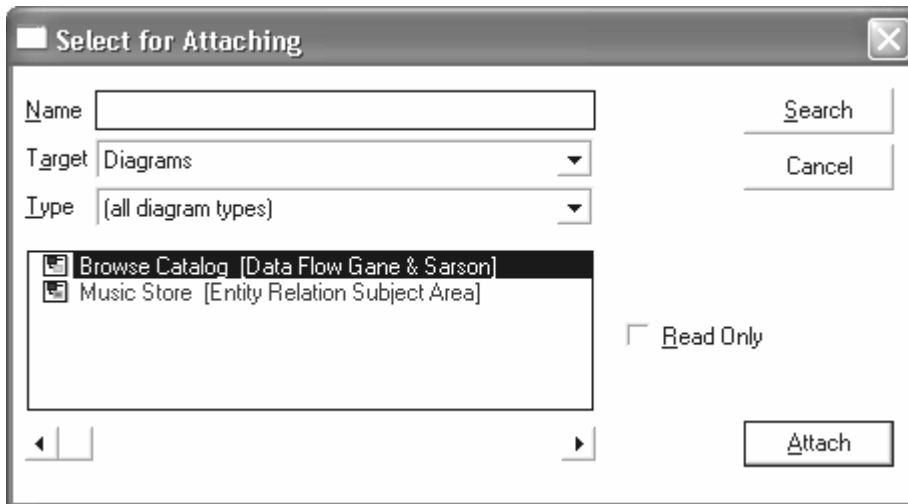


Figure 25 List of diagrams to attach

To return to the Functional Hierarchy diagram, right click on any blank area on the DFD and choose the `Parent` menu item from the pop-up menu. Once diagrams are linked in this fashion, three dots will appear over the symbol to indicate a child diagram exists. This is shown in Figure 26. In the future, to navigate to a child diagram, simply right click on the Function and select the `Child Open` menu item from the pop-up menu. To unlink a child diagram, right click on the Function and select the `Child Detach` menu item from the pop-up menu.

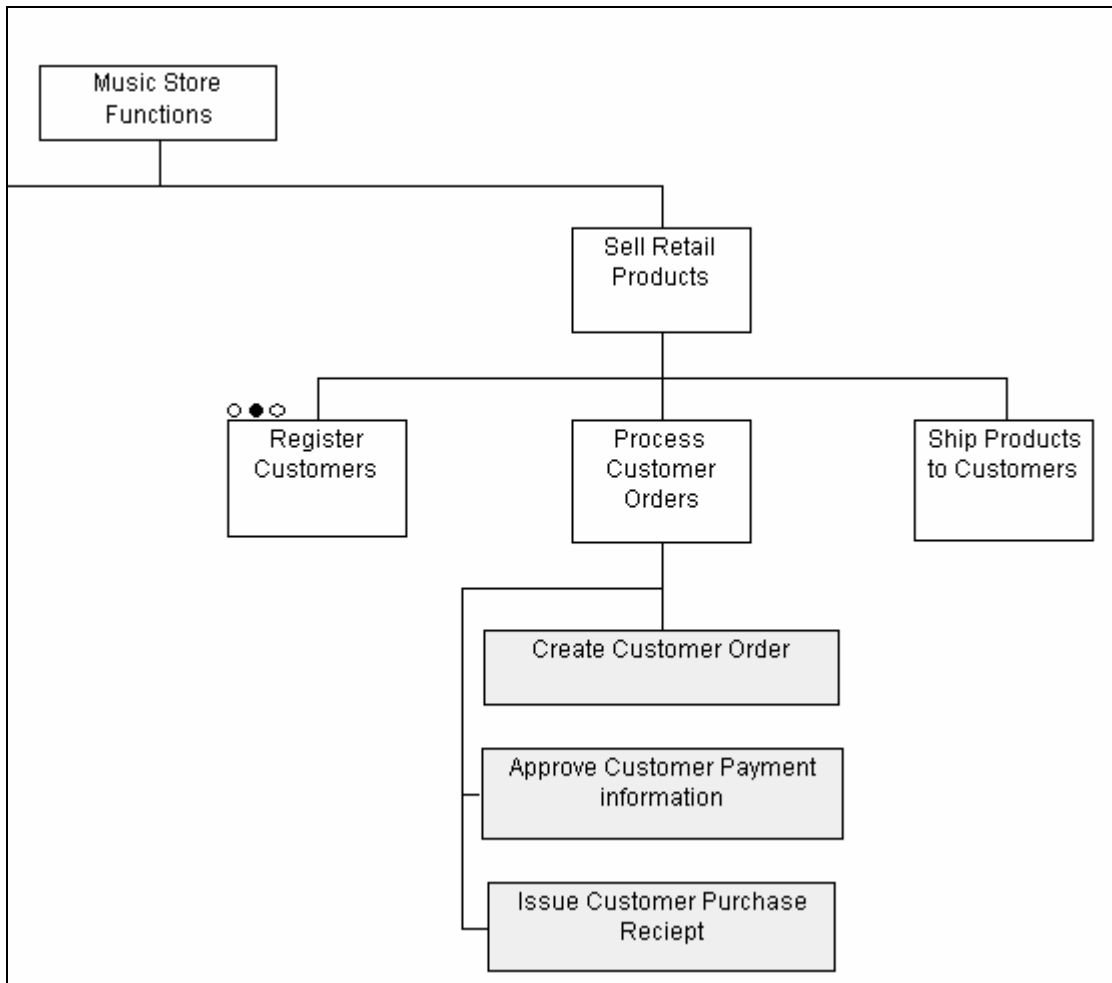


Figure 26 A portion of the Functional Hierarchy diagram with child diagram attached (three dots)

Exercise: Linking a Data Store to an Entity Relationship Diagram

In this exercise, the “Products” data store in the “Browse Catalog” DFD will be linked to the “Music Store” ER diagram. First, close the Functional Decomposition diagram used earlier in this section. Then open the “Browse Catalog” DFD as created in Section 4 and right click on the “Products” data store. Choose **Child Attach** and then select and Attach the “Music Store” Entity Relationship diagram. Practice navigating among the three diagrams.

7. Completeness Analysis and Rule-Based Error Checking

As with many CASE tools, SA provides a collection of rule based error-checking routines for each model. As projects grow in size, rule based error checking can automatically spot potential trouble areas such as entities not related to the rest of an ER model or processes that are lacking inputs or that have no outputs in a DFD model. To demonstrate this feature, rule base error checking will be done on the DFD created in Section 4.

To begin the error checking process, click on the **Structured** tab in the Browse window and open the Diagrams and then Data Flow Gane & Sarson entries until the “Browse Catalog” DFD appears. Double click on the Browse Catalog DFD to open it up.

Before proceeding, make certain the Windows computer you are working on has at least one printer installed. This is required to generate the reports on screen (even if you chose not to print them).

Rule Based Error checking can now be carried out on this diagram by pulling down the Reports menu and selecting the Rules Check menu item. A message may appear stating: “Since the report file did not contain a Rules Check report specification, the default specification will be used”. Simply click on the OK button to proceed. A new window should appear in a short time with the resulting Rules Check report. An example rule check report is shown in Figure 27. Note that the text may wrap around due to the long lines of output.

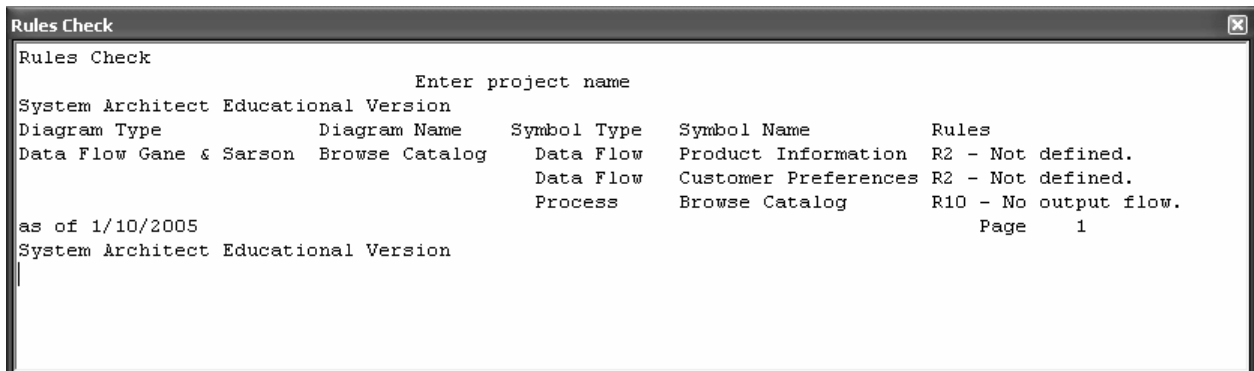



Figure 27 Output from the Rules Check Report

The text can be highlighted and copied to the Windows clipboard (drag the mouse over the text to highlight it and then right click and select Copy from the pop-up menu). This text can then be pasted into MS Word or other software for storage or inclusion into a master report.

Figure 27 documents three instances where rules were violated. For example, Rule 10 warns that a process Browse Catalog has no output flow thus any data generated by that process would have no way to be stored by the information system being designed. This will be fixed in the exercise below. Rule 2 violations appear in cases where formal definitions have not been given for objects such as data flows and data stores.

To close the report window, click on the small Window Close  in the upper right corner of the report window. Whenever SA finds issues that might be errors, it makes marks on the diagram where they occur. To clear these marks, pull down the Reports menu and select the Clear Errors menu item.

To fix the definition problems, right click on the object in question and choose Edit from the pop-up menu. Supply appropriate definitions in the resulting dialog box and click OK to save the definitions. The following exercise will correct the missing output flow problem.

Exercise: Adding a new Data Store to a DFD

In this exercise, the missing output flow problem reported in the Rules Check report will be resolved by creating a new data store called “Customer Orders”. An outline of the steps follows:

- Click on the **Structured** tab and open the Diagrams and Data Flow Gane & Sarson entries.
- Double click on the Browse Catalog diagram to display it.
- Use the Data Store tool to create a new Data Store called “Customer Orders”. Position this new Data Store below the Process “Browse Catalog”.
- Right click on the Customer Order data store and choose **Edit** from the pop-up menu. Click on the **Symbol** tab and then change the number of this item to 2.
- Use the Data Flow tool to create a data flow from the Process “Browse Catalog” to the new Data Store “Customer Orders”. Label this data flow “Products Ordered information”.

Be sure to save the changes to the diagram. The results should appear as in Figure 28.

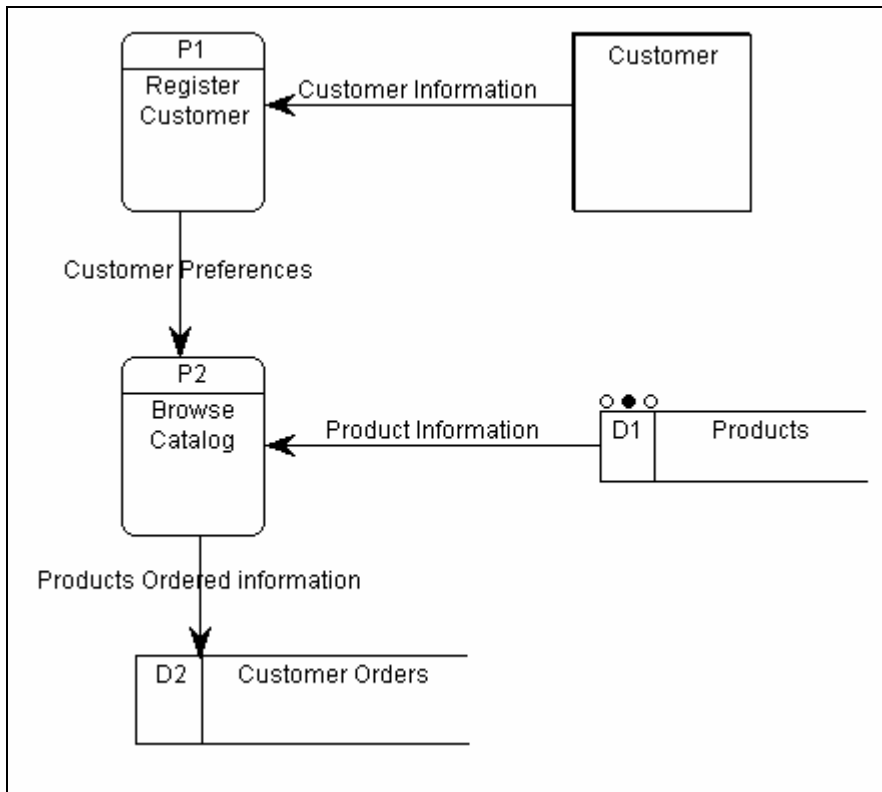


Figure 28 Revised Data Flow Diagram with additional Data Store

At this point, with the revised DFD open, re-run the Rules Check report and confirm that the R10 Output Flow problem has indeed been resolved.

8. Transforming Entity Relationship Models to Physical Database Models

Section 3 introduced data modeling with Entity Relationship diagrams. In general, ER models cannot be directly implemented in relational databases. A step towards implementation is to transform the ER model into a Physical Database Model (PDM). To demonstrate this, open the Music Store ER model created in Section 3 by navigating to the Diagrams and Entity Relation branches in the **All Methods** tab of the Browse window. Right click on the Music Store diagram and choose Open from the pop-up menu. The diagram should appear as shown in Figure 16 (after the Product entity has been added during the Exercise for Section 3).

As a first step, foreign keys should be propagated through the ER diagram. The steps for this are given in detail in Section 19.2. With the current ER model displayed, pull down the Dictionary menu and select the Update FKs menu item. A dialog box will appear showing a log of the changes made to the diagram. After dismissing this dialog box, the ER Diagram will redraw with the foreign keys in place. Be sure to save the diagram after this change.

With the foreign keys propagated, the physical database model can be created by pulling down the Dictionary menu. Highlight the Create Data Model menu item and choose Physical Data Model... from the fly-out menu as shown in Figure 29.

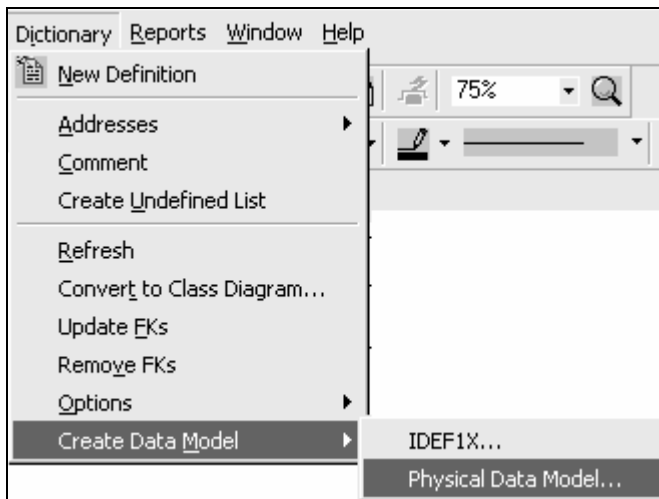


Figure 29 The Create Data Model menu item

A dialog box will appear as shown in Figure 30. Give the new diagram the same name as the ER model and leave all of the other options as they are set as defaults. Click the OK button to continue. A window will appear as shown in Figure 31 with log entries as the transformation process is started. Do not close this window (this will be done in a later step).

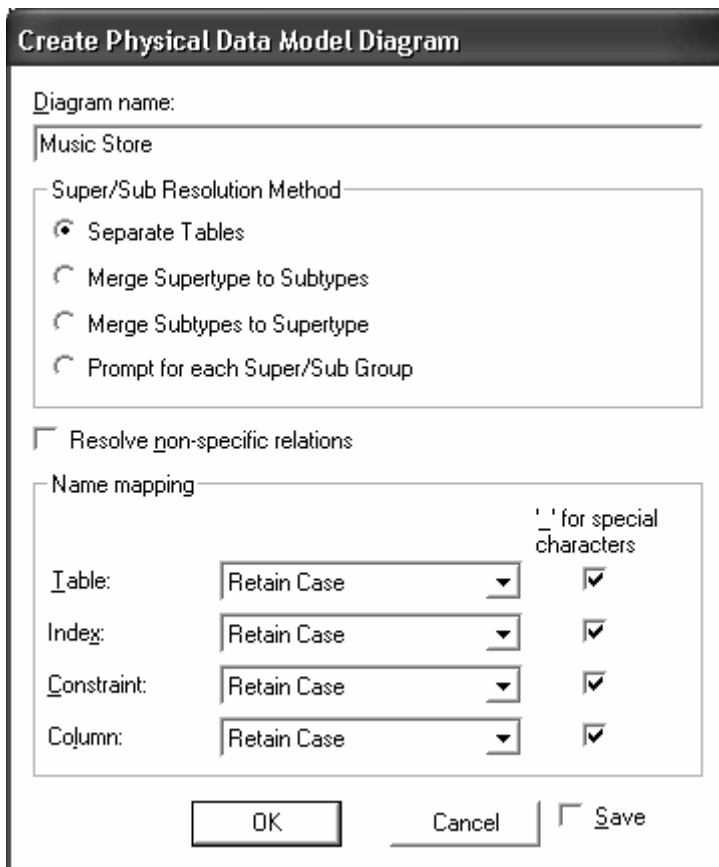


Figure 30 Create Physical Data Model diagram options dialog

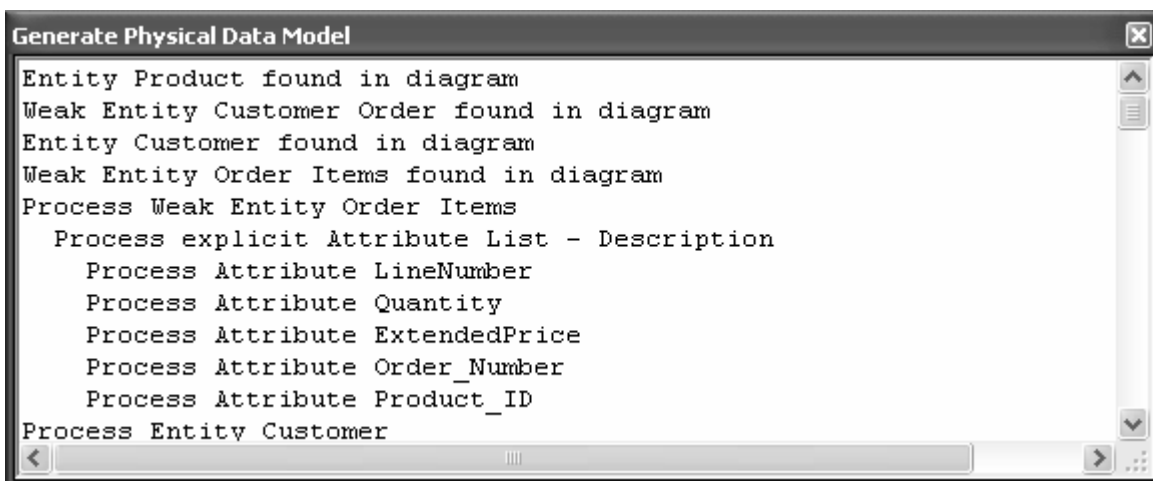


Figure 31 Log output for the generate physical data model process

The next dialog box to appear as shown in Figure 32 and will prompt for the name of the new database. For this example, type “MusicStoreDB” in the Database Name field and click the OK button.

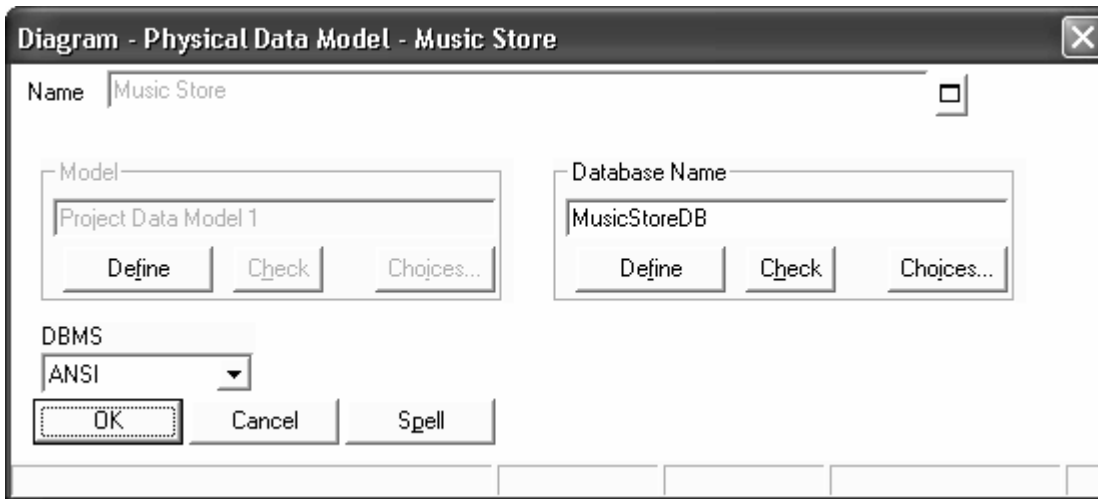


Figure 32 Naming the target database

After more log entries are added to the pop-up window (Figure 31), the new diagram will draw in the background. Once it is completed, the pop-up window can be closed. After some re-arranging and resizing of the entities, the diagram should appear as shown in Figure 33.

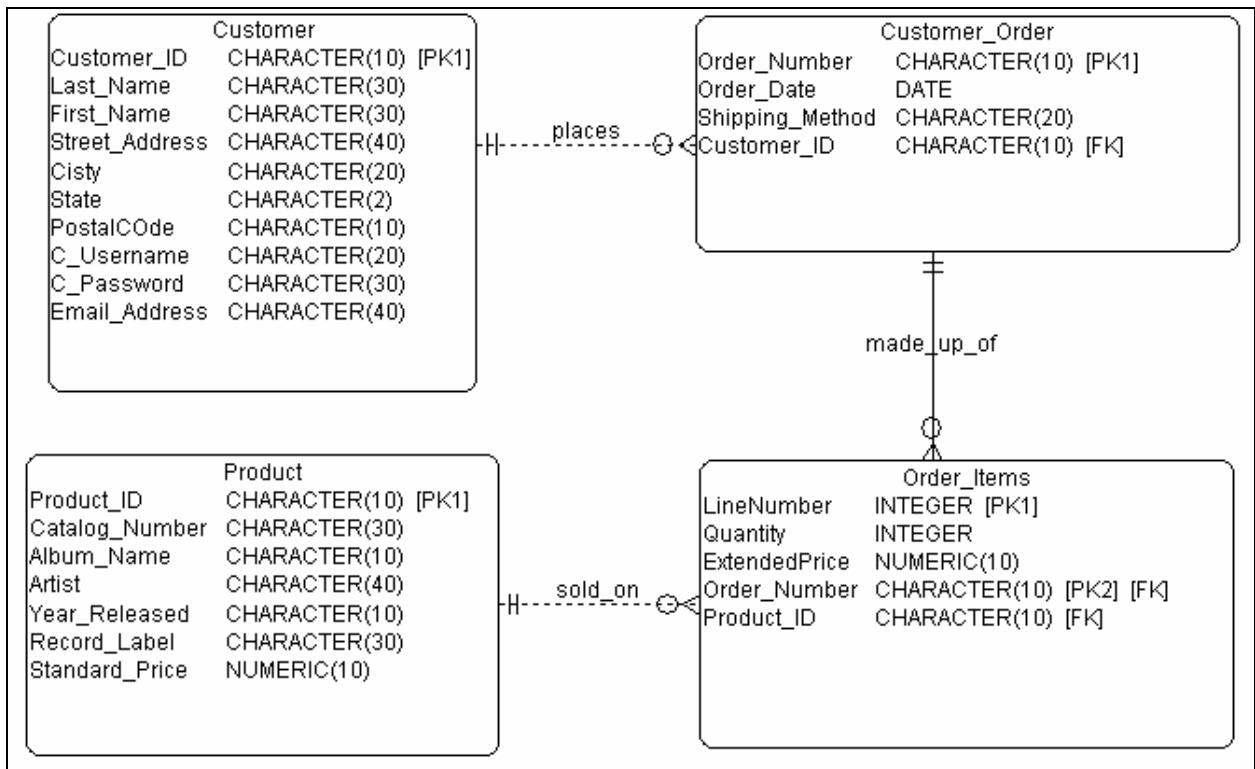


Figure 33 Physical Data Model

To save the new physical data model, press CTRL + S or pull down the File menu and choose the Save Diagram menu item. Click “Yes” if asked to confirm the save.

The commercial version of Popkin System Architect provides additional functionality to automatically generate the Data Definition Language (DDL) statements that would be used to create the schema in a relational database. The Student Version does not have this capability, however.

Part III: Object Oriented Modeling with System Architect

In this part of the tutorial, a number of object oriented modeling techniques will be introduced using the Unified Modeling Language (UML) notation.

Before continuing on to this next set of sections, choose a new Encyclopedia that is different from the one used for the prior sections. For example if the PROJECT1 encyclopedia was used for Sections 3 through 8, choose the PROJECT2 encyclopedia for the sections on Object Oriented modeling. To select a different encyclopedia, pull down the `File` menu and choose the appropriate encyclopedia from the list. As described in Section 2, the Configuration dialog box may appear (as shown in Figure 3). Click the cancel button to proceed.

9. Object Oriented Modeling with Use Cases

In this section, the basics of object oriented modeling with UML use cases will be introduced. A use case is a behaviorally related sequence of steps (a scenario), both automated and manual, for the purpose of completing a single business task. Use cases describe the system functions from the perspective of external users and in the manner and terminology in which the users understand. Use cases are the results of decomposing the scope of system functionality into many smaller statements of system functionality. A use case itself is not considered a functional requirement, but the story the use case tells captures the essence of one or more requirements. Use cases are initiated or triggered by external users or systems called actors.

The following example will develop the use case for a customer registering a new account at our on-line music store, browsing the catalog and requesting assistance from a customer service representative. The resulting Use Case diagram is shown in Figure 34.

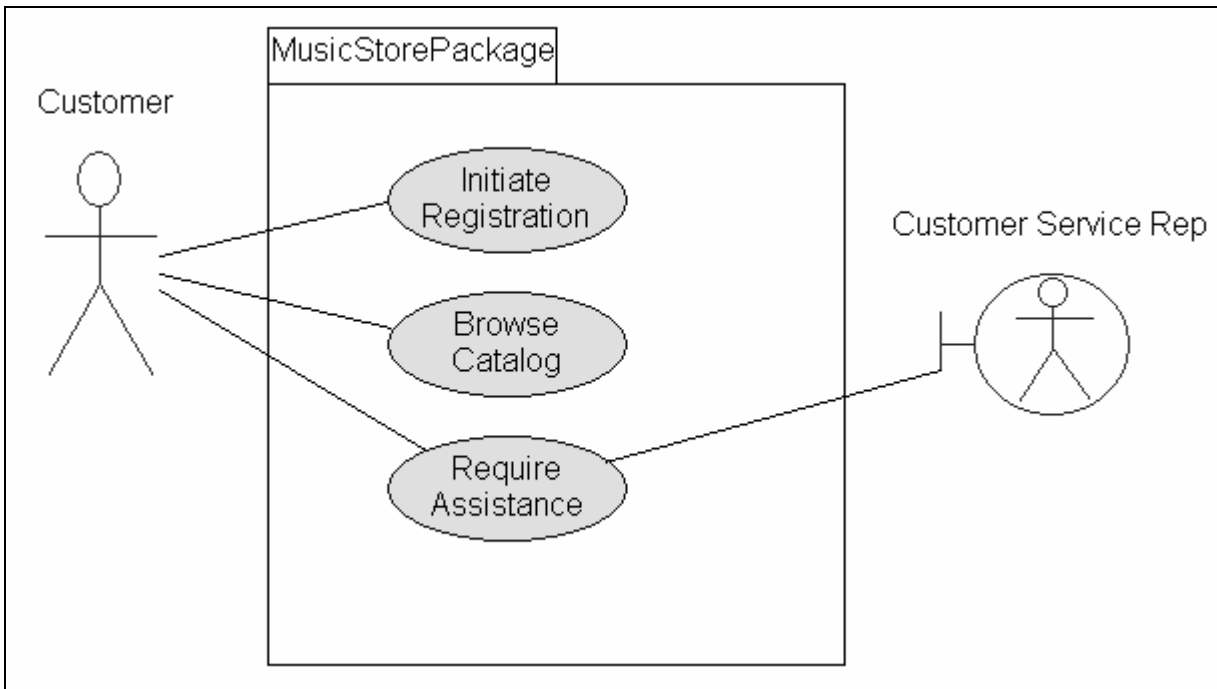


Figure 34 Example use cases

There are four main steps to creating a Use Case. First, a Package is created to hold the use case objects. Next use cases are added to the diagram. Third, actors and their associations with use cases are specified. Finally, the details of the use case can be filled in.

To begin, select the **UML** (Unified Modeling Language) tab in the Browse window. Pull down the File menu and choose the New Diagram menu item. Double click on the UML Use Case item to select it as shown in Figure 35.

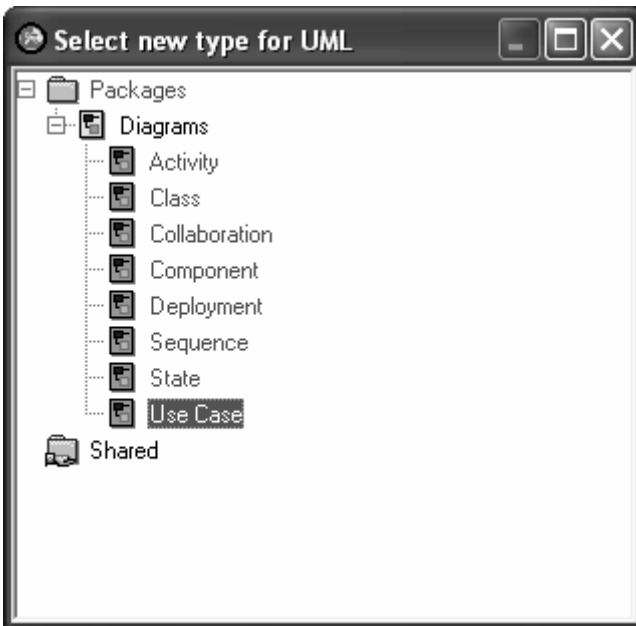


Figure 35 Selecting a new UML Use Case diagram

Name the new diagram “Customer Registration Browsing and Assistance” and click on the OK button as shown in Figure 36.

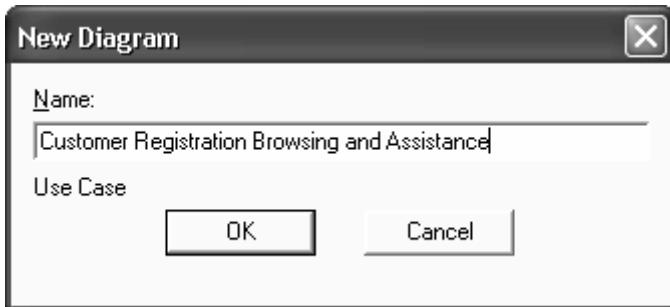


Figure 36 Naming a new UML Use Case diagram

The next prompt will be to create a new Package. Name the package “MusicStorePackage” as shown in Figure 37 and click on the OK button.

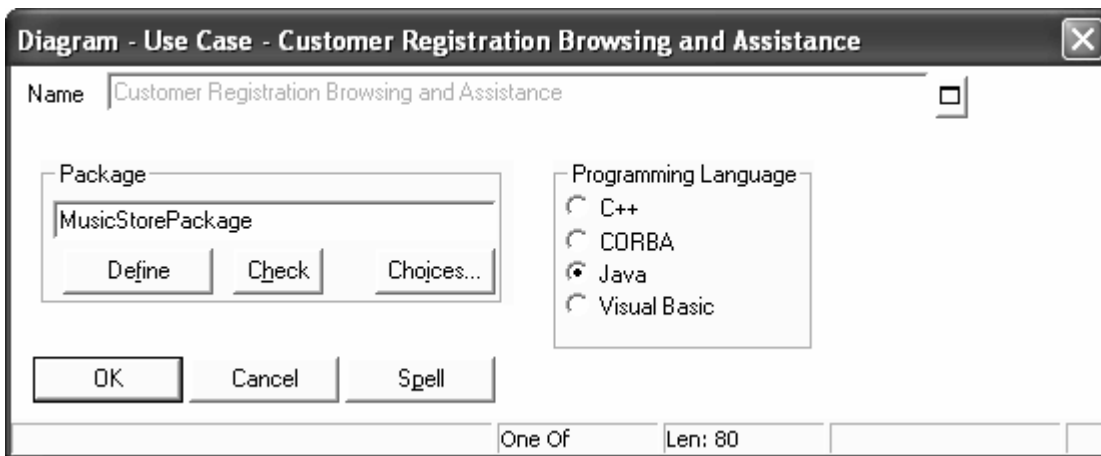



Figure 37 Creating a new package for the use case

With the “MusicStorePackage” Package created, it can then be shown on the UML diagram by selecting the Package tool . With this tool, click on the open space on the blank diagram to the right to display this package. Name the package “MusicStorePackage” and then stretch out the box a bit as shown in Figure 38.

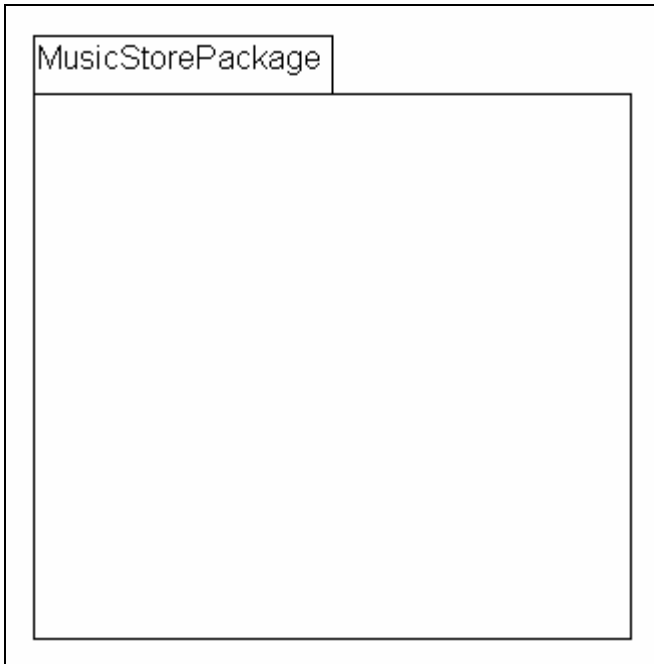


Figure 38 Creating a package in the Use Case diagram

Now that the package is displayed, individual use cases can be added to the diagram. To add a new use case, click on the Use Case tool icon on the button bar. This appears as a hollow oval:



Click on the diagram space inside of MusicStorePacakge to create the new Use Case. A green circle will appear briefly followed by a dialog box prompting for the name of the new Use Case. For this example, provide “Initiate Registration” as the name of the use case as shown in Figure 39 and click the OK button to continue. Note also that the use case is automatically created within “MusicStorePackage”.

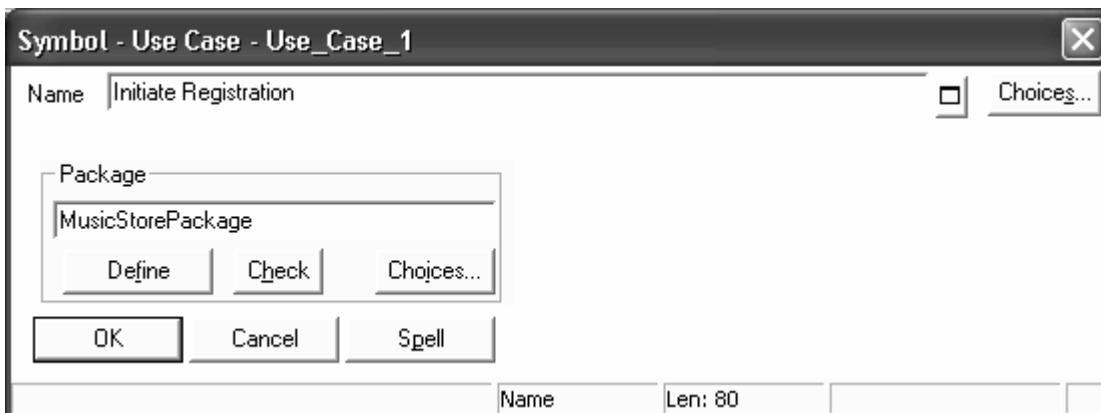


Figure 39 Naming a new Use Case

Use the Use Case tool to create two additional use cases. Name these:

- Browse Catalog
- Require Assistance

The resulting diagram should appear as shown in Figure 40.

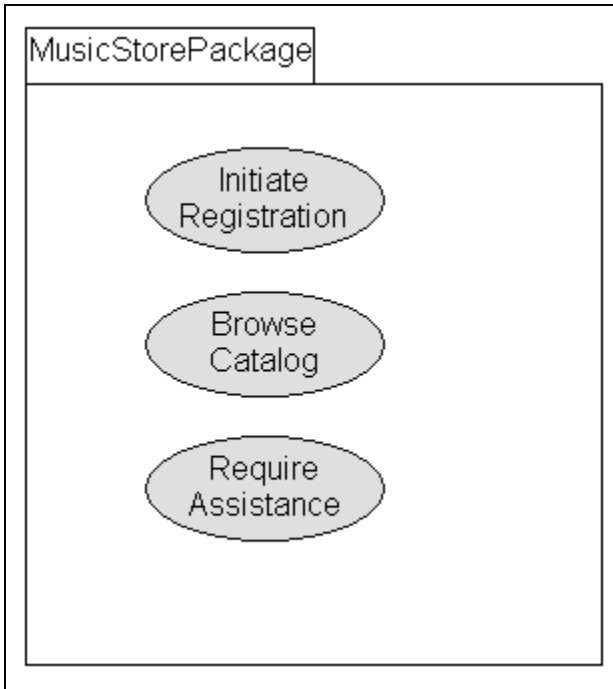



Figure 40 UML diagram with three Use Cases

The next step in the Use Case diagram is to put in “Actors”. The Actor tool icon appears as a stick figure: . In some cases, it may not appear automatically along the button bar. If this icon is missing, pull down the extended list of icons by clicking on the small down arrow as shown in Figure 41.

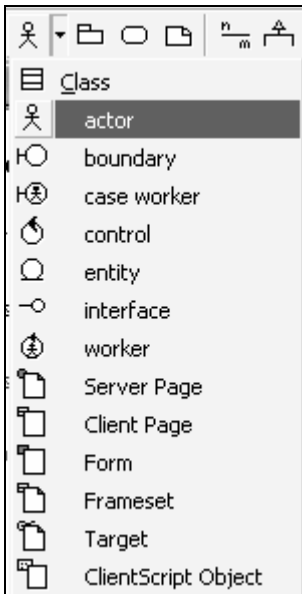


Figure 41 Selecting the Actor tool from the drop down list

With the Actor tool selected, click on the diagram in the open space to the left of MusicStorePackage. Name the actor “Customer” as shown in Figure 42.

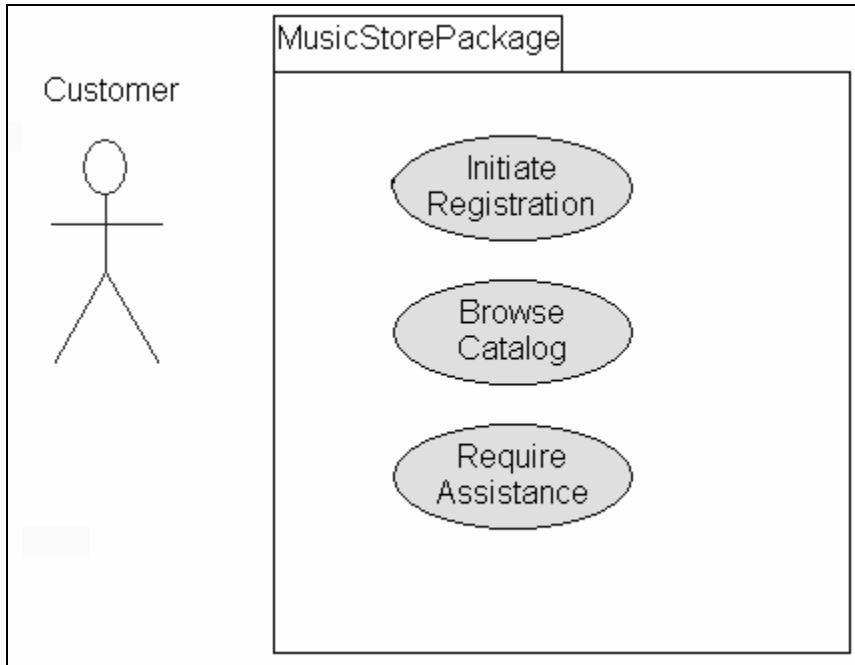



Figure 42 Adding an actor to the Use Case diagram

Next, select the “Case Worker” tool  case worker and create a Case Worker named “Customer Service Rep” on the right hand side of MusicStorePackage.

The next step is to draw associations between the actors and the use cases. Start by selecting the

Use Case Associations tool by clicking on its icon: . To draw an association between the Customer and the Registration Use Case, first click on the Customer. A bold plus sign **+** will appear on the Customer actor. Then click on the Registration Use Case as shown in Figure 43.

With the Use Case Association tool still selected, draw three more associations as follows (refer to Figure 34):

- Between Customer and Catalog
- Between Customer and Assistance
- Between Customer Service Rep and Assistance

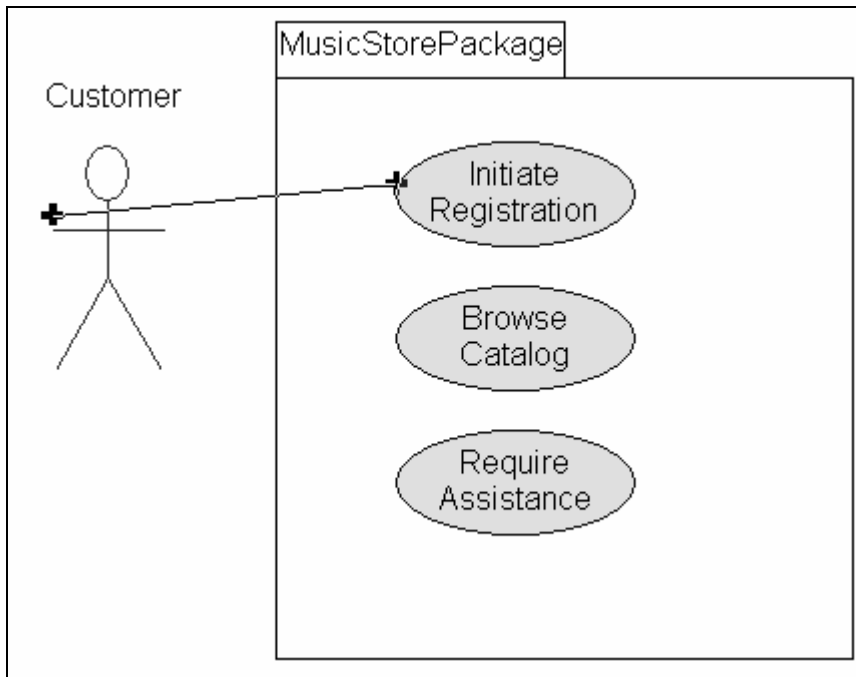


Figure 43 Associating Customer and the Registration Use Case

In a similar fashion, add associations between Customer and the Browse Catalog use case and between Customer and the Require Assistance use case. Finally, draw an association between the Customer Service Rep case worker and the Require Assistance use case. Refer to Figure 34 as needed.

At this point, three use cases, an external actor (Customer) and an internal worker (Customer Service Rep) have been created and associations between them have been specified. The next layer of detail requires defining the details of each use case. For this example, the “Browse Catalog” Use Case will be specified at a greater level of detail.

To begin, right click on the Catalog Use Case and select the `Edit` item from the pop-up menu. Starting with the Definition tab (shown in Figure 44), provide a description for the use case as follows:

After registration, Customers may browse the product catalog and select products they are interested in. Selected products will be saved in the Customer's shopping cart that can later be used to create a Customer Order.

Next provide a list of Exceptions as follows:

Visitors to the web site who have not registered will not see any product pricing, may not place items in a shopping cart and may not place orders.

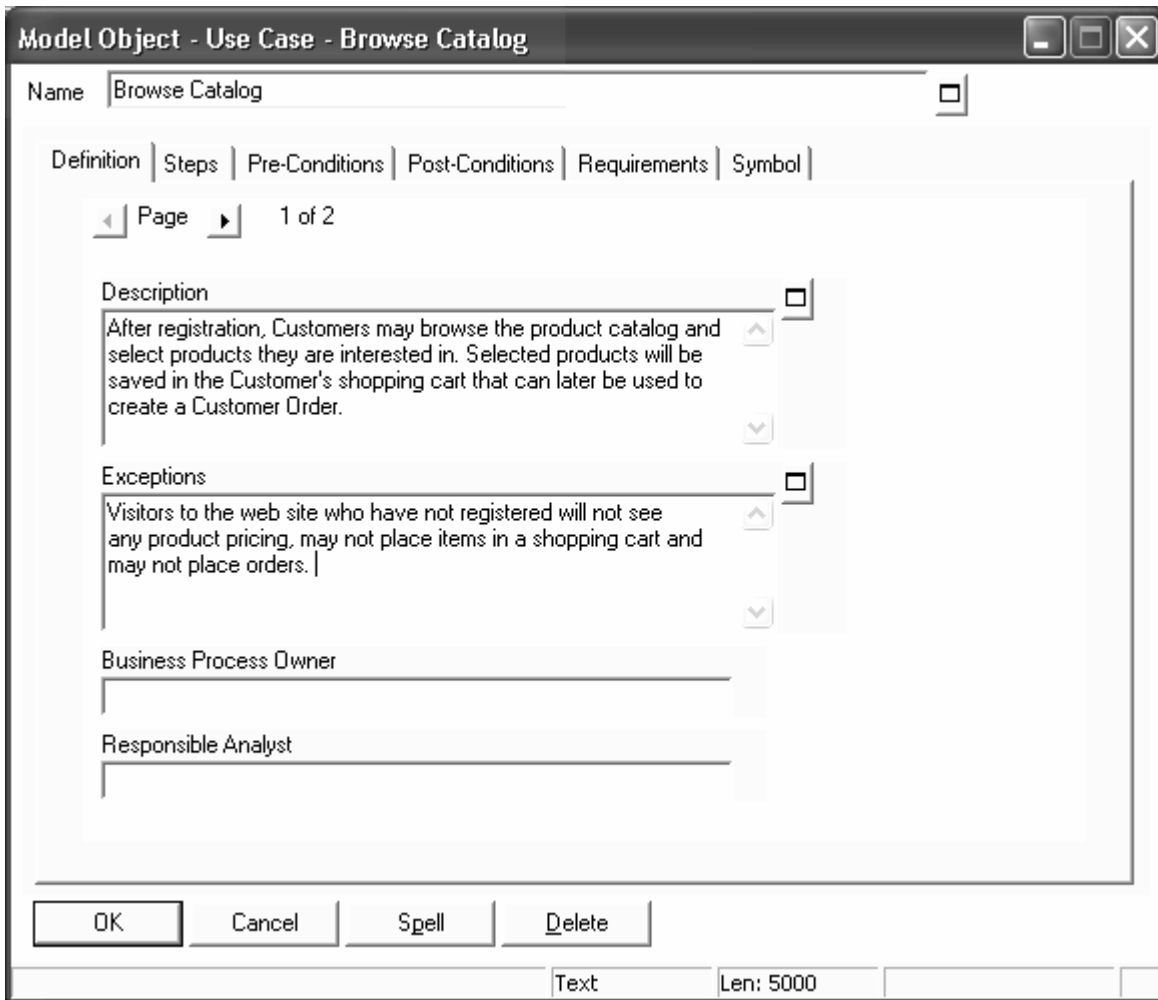


Figure 44 The Definition tab of the use case properties

Next, click on the Steps tab to enumerate the different steps to be carried out for this use case. The Steps tab is shown in Figure 45. Fill in the name of each step and the Step Text as shown below:

Name	Step Text
SelectCategory	Customers begin by selecting the category (genre) of music
BrowseArtists	Customers browse lists of artists/groups
BrowseAlbums	Customers browse list of albums by Artist
SelectAlbum	Customers select albums and add them to their shopping cart

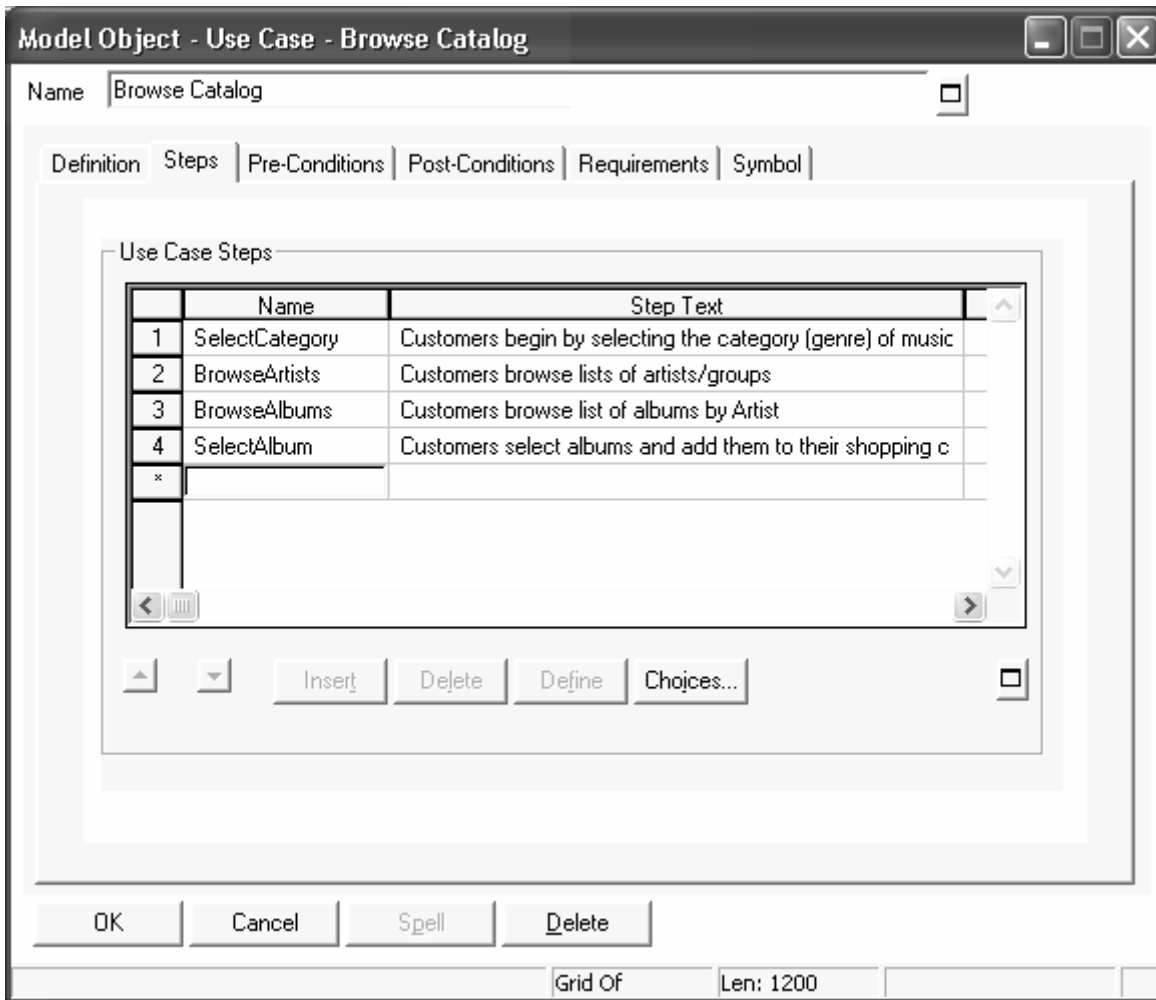


Figure 45 The Steps tab of the use case properties

Next, click on the Pre-Conditions tab as shown in Figure 46. Pre-conditions are like pre-requisites for the process to take place. To browse categories and albums, Customers must be registered. Preconditions can be entered like “steps” except that instead of the step text box, a business rule box appears if the “Define” button is clicked. Any details about the business rule can be provided in the separate Description box. For this example, specify a pre-condition as follows:

Name: CustomerRegistration Business Rule: Customer must register before browsing

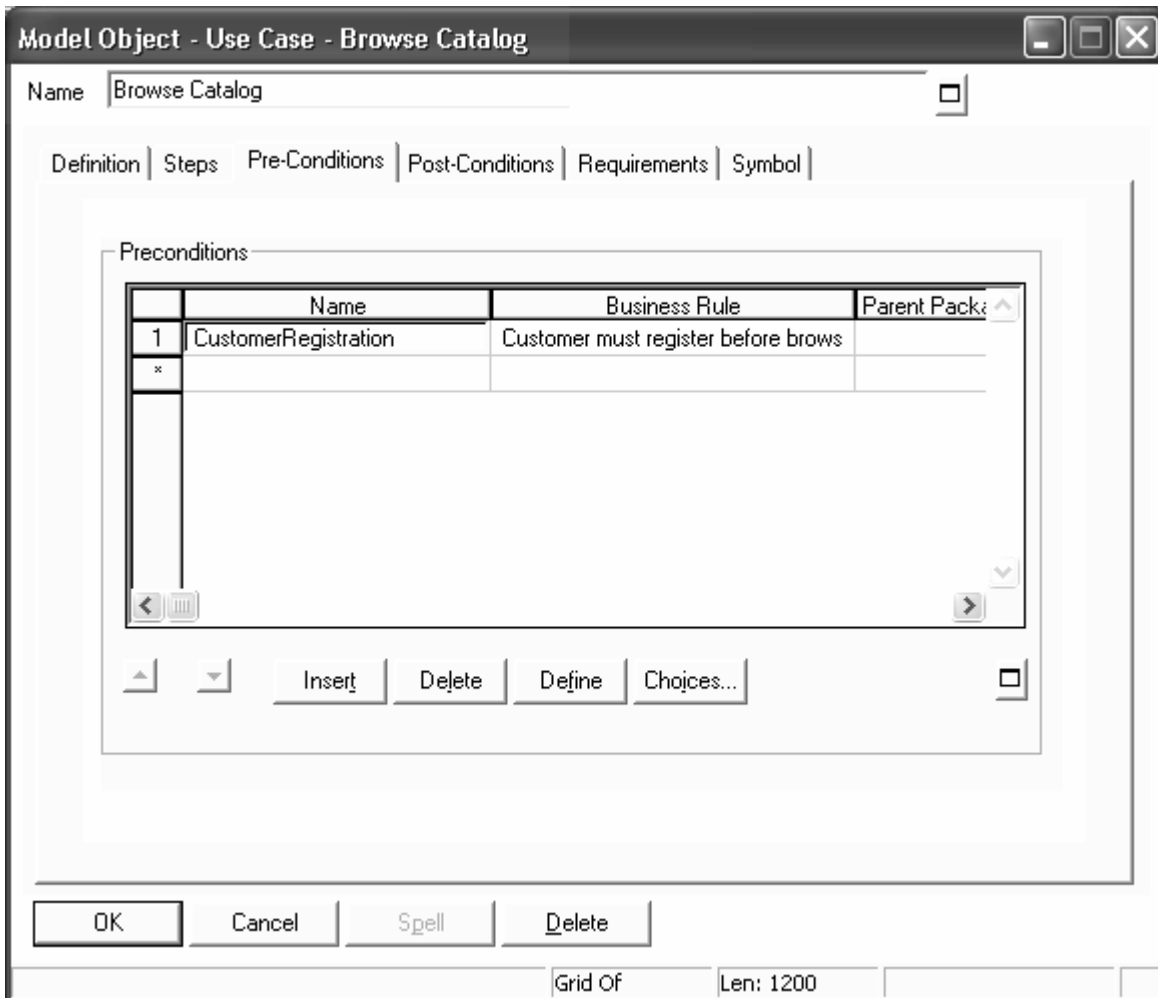


Figure 46 Pre-Conditions tab of the use case properties

Finally, click on the Post-Conditions tab as shown in Figure 47. Post conditions are conditions that are required to ensure that the process finishes as desired. They are added in the same way as pre conditions. Specify this post condition as follows:

Name: SaveShoppingCart Business Rule: Shopping cart must be saved

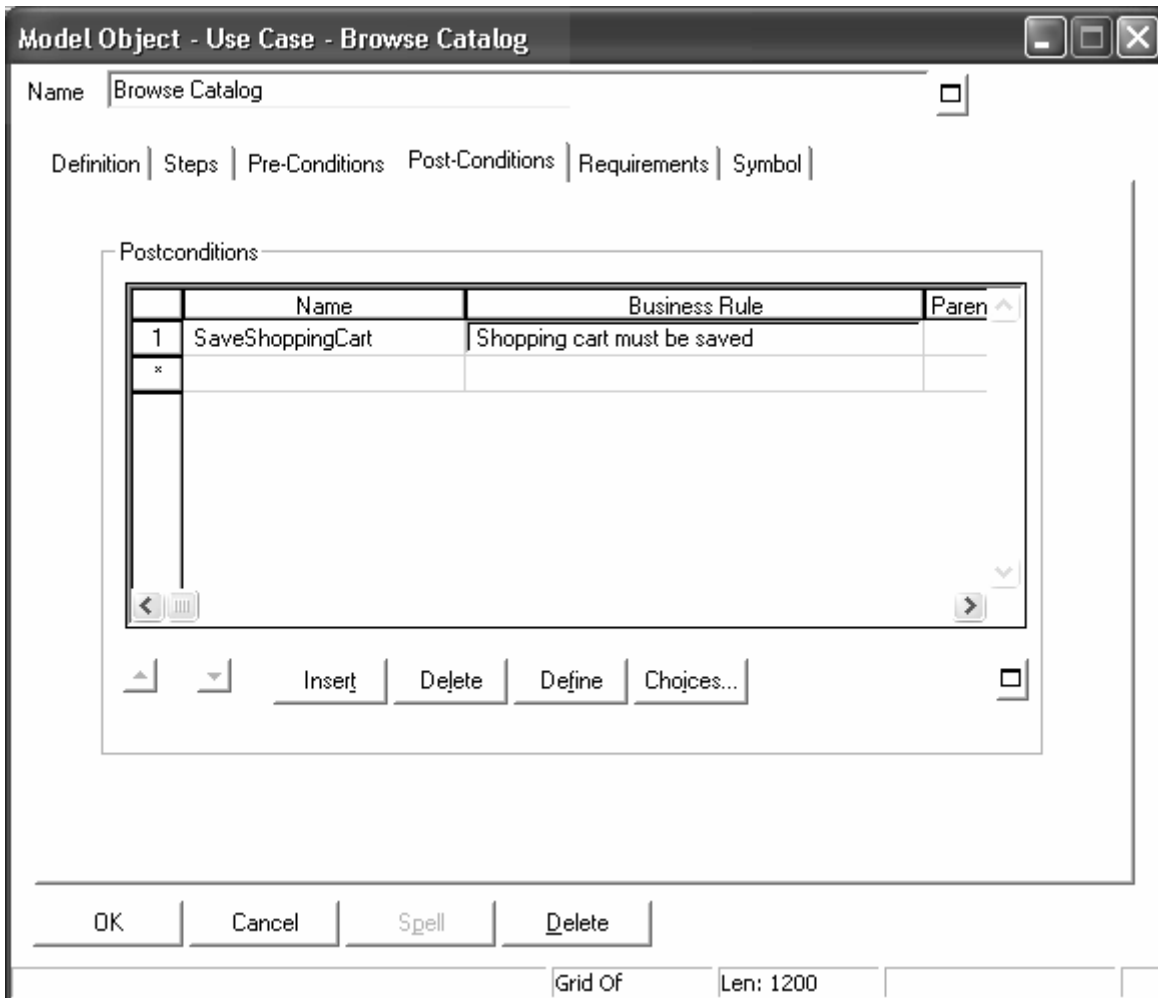


Figure 47 The Post-Conditions tab of the use case properties

Once the pre- and post-conditions have been entered, click on the OK button to complete the creation of the new Use Case.

To save the work completed thus far, press CTRL + S or pull down the File menu and choose the Save Diagram menu item. Click “Yes” if asked to confirm.

Exercise: Create a new Use Case, Associations and Use Case details

For this exercise, create a new Use Case with the following information:

Use Case Name	Place Customer Order	
Description	A customer places an order by viewing his/her shopping cart and proceeding to “check out”.	
Exceptions	If the shopping cart is empty, an order may not be placed.	
Step 1	CopyCartItems	Copy all shopping cart items to customer order
Step 2	ConfirmOrder	Prompt user to confirm order
Pre-Condition	ShoppingCartItems	At least one item must be in the shopping cart

Post-Condition	EmptyShoppingCart	Empty the shopping cart after a successful order
----------------	-------------------	--

To complete this exercise, draw a Use Case Association between Customer and Customer Order, and name this association “places”. The completed exercise is shown in Figure 48.

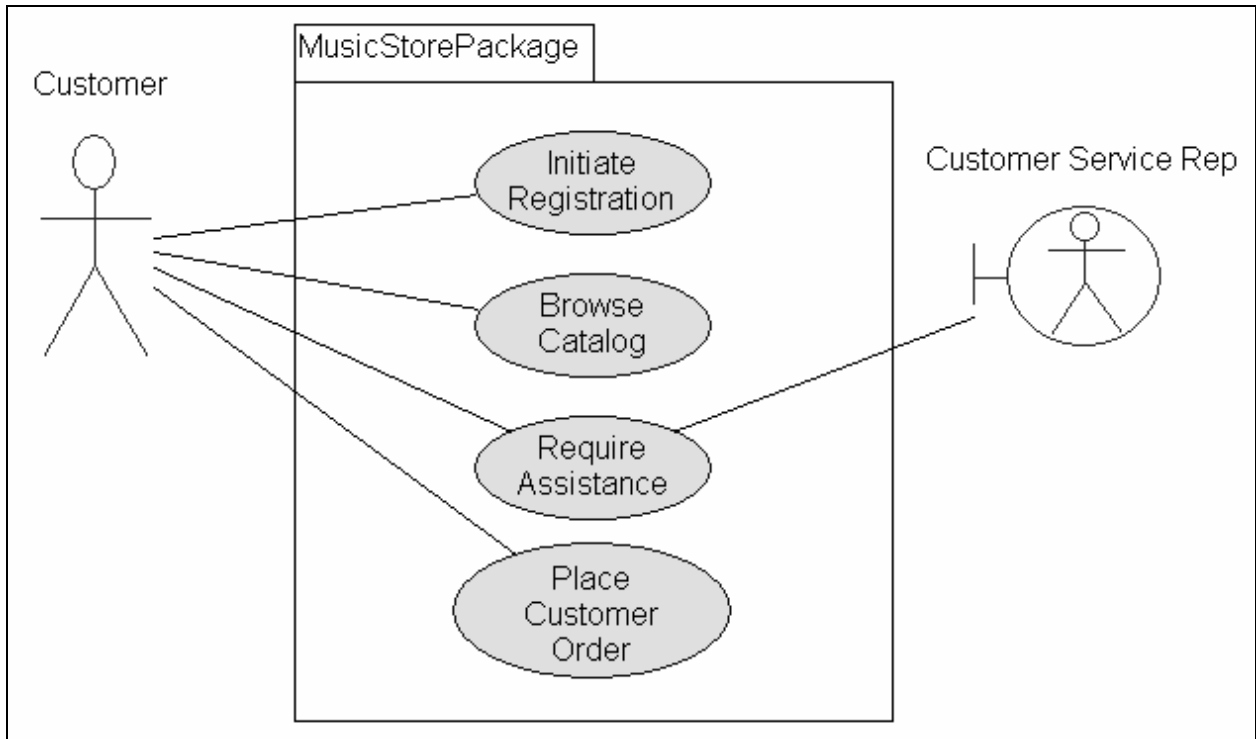


Figure 48 Completed Use Case exercise

10. Object Oriented Modeling with Class Diagrams

This section introduces structural modeling with class diagrams. A class diagram is used to graphically depict objects (with attributes and methods) and their associations and relationships. OO concepts such as multiplicity, associations, generalizations/specialization relationships, and aggregation relationships can all be shown on a class diagram. For this example, a class diagram for the Music Store will be created as shown in Figure 49. This example builds upon and makes use of some of the items created in the Use Cases developed Section 9.

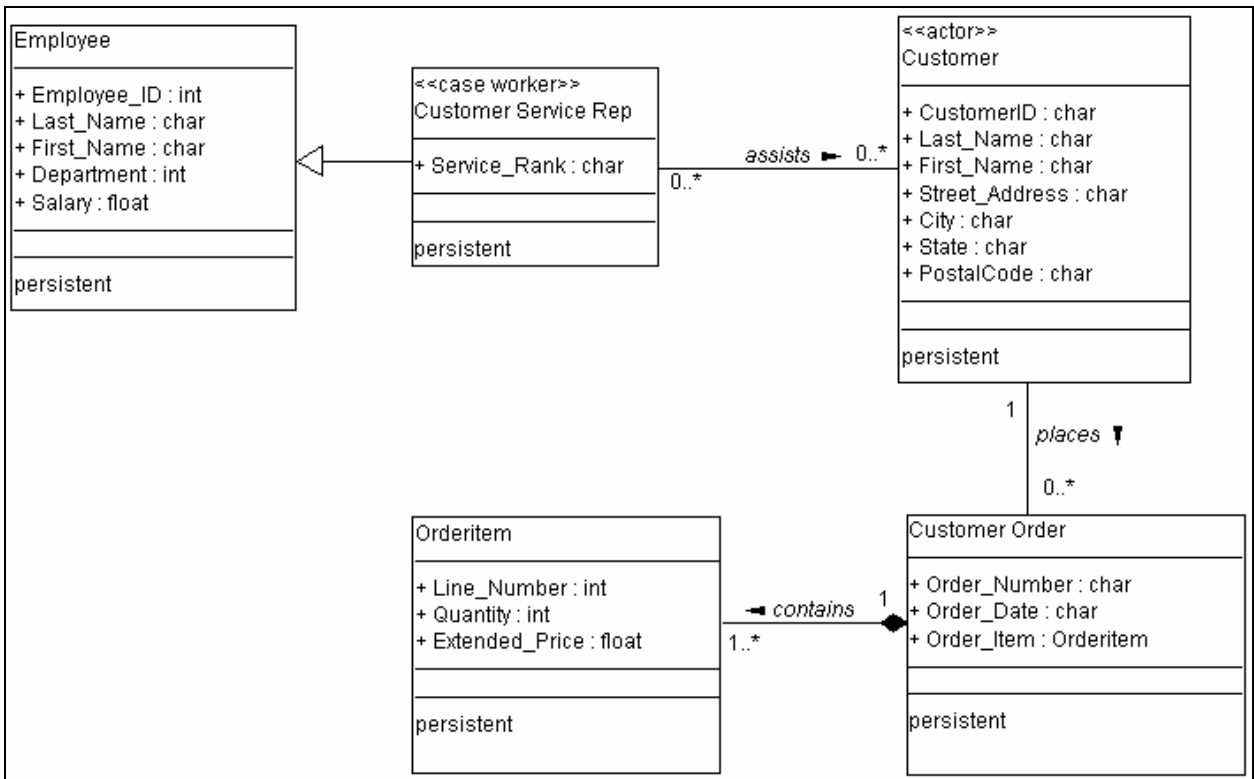


Figure 49 Example UML Class diagram

Start by choosing the **UML** (Unified Modeling Language) tab. Pull down the **File** menu and select the **New Diagram** menu item. Choose **UML Class** by double clicking on it as shown in Figure 50.

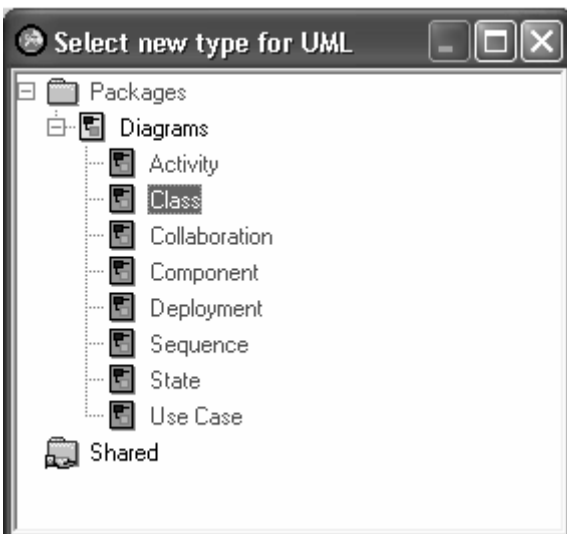



Figure 50 Selecting the new Class diagram

Name the new diagram “Music Store Customer Classes” and click the OK button. On the next dialog box, click on the Choices button and drag the MusicStorePackage over to the package name field as shown in Figure 51. Close the Package list by clicking on the close button  and then click the OK button on the Music Store Customer Classes dialog box to complete creating the new diagram.

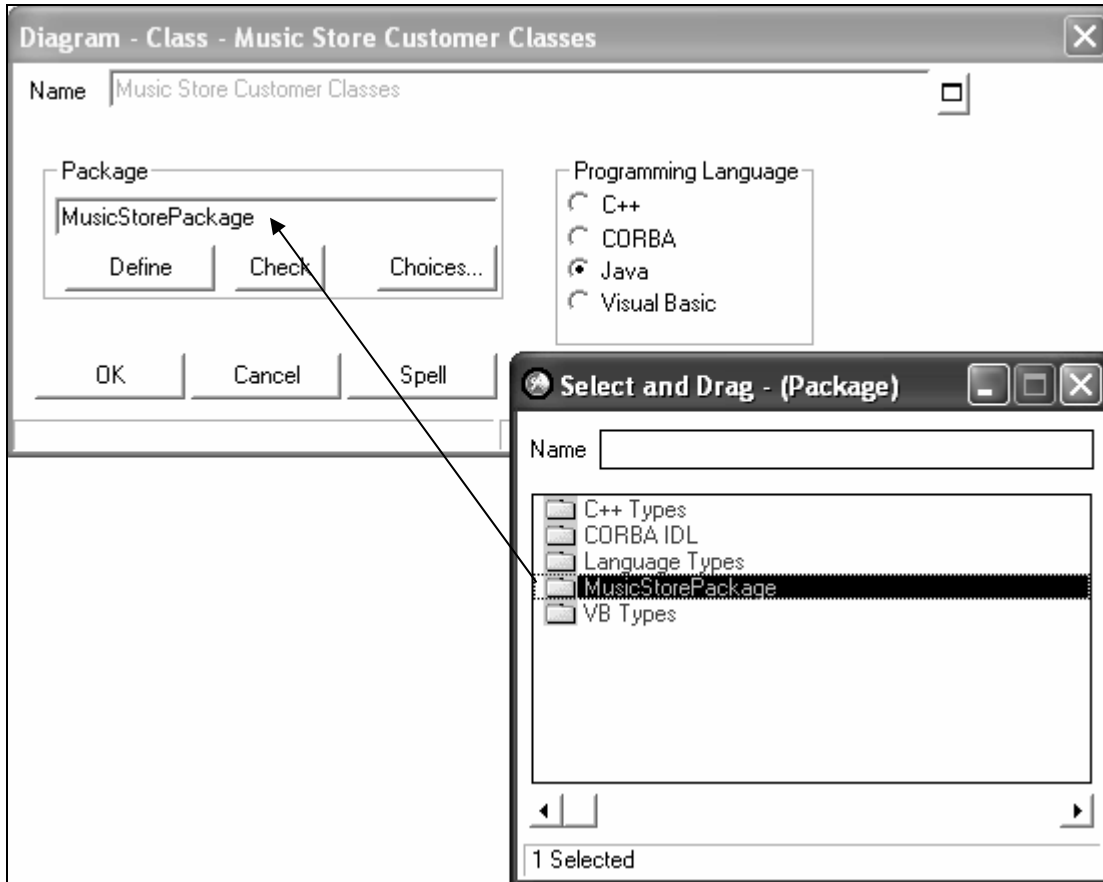




Figure 51 Naming the new diagram and associating it with the MusicStoreCustomer package

At this point a new blank diagram should be opened in the right hand window and new classes can be drawn in. To create a new class, select the Class icon  and then click in the open area of the diagram. A dialog box will open offering the opportunity to name the new class. For this example, the Customer class should be added. However we have a previous definition of a Customer that was done in Section 9 on Use Cases. To bring in this definition, click on the Choices button to the right of the Name field to display a list (the dialog box has “Select and Drag – (Class)” in the title) and then drag the Customer actor over to the Name field as shown in Figure 52. Close the list by clicking on the close button . Finally click on the OK button to create the new class.

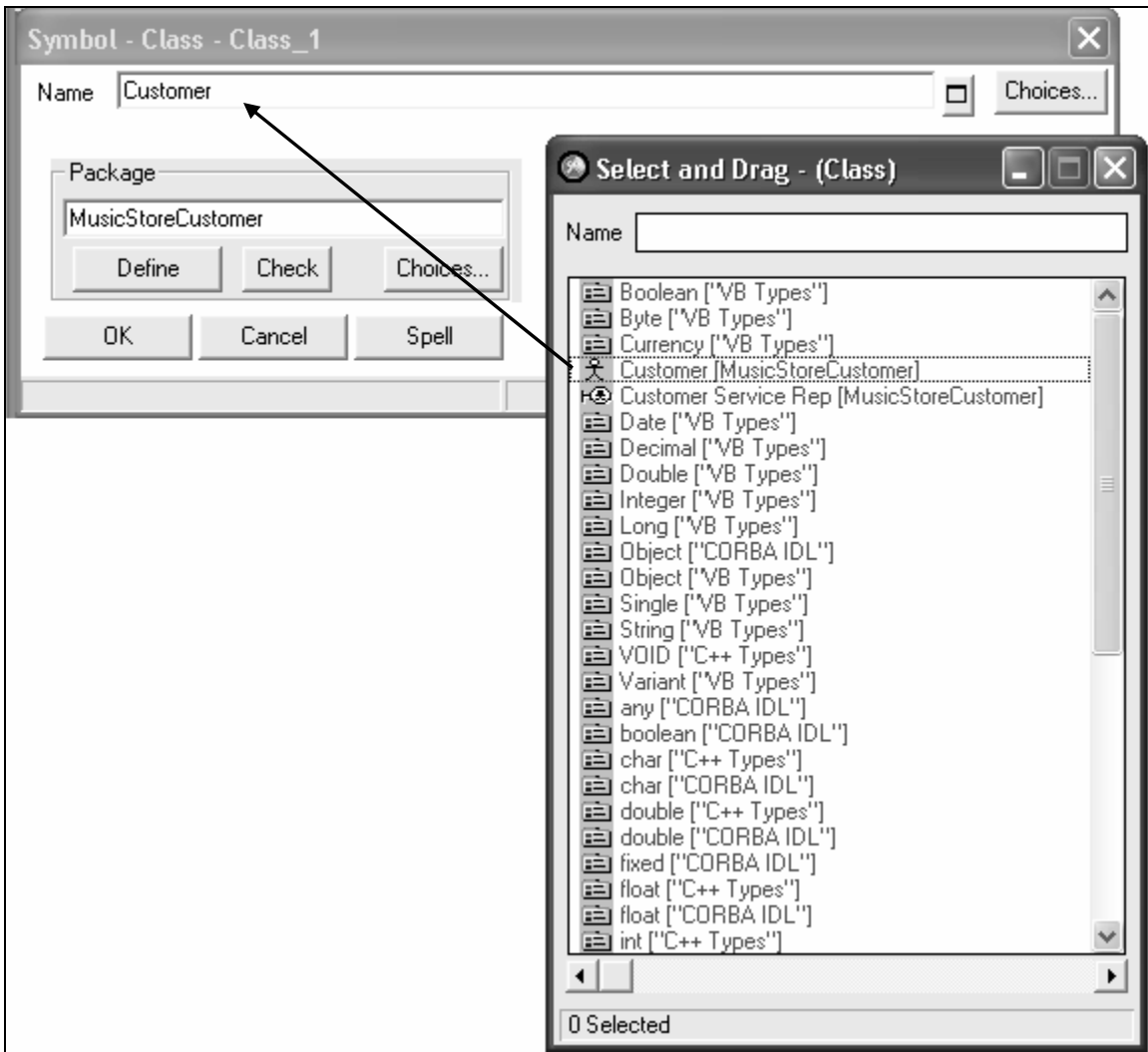


Figure 52 Bringing in the Customer class

With the Class tool still selected, repeat this set of steps and bring in the Customer Service Rep. class. The results at this point should appear as in Figure 53

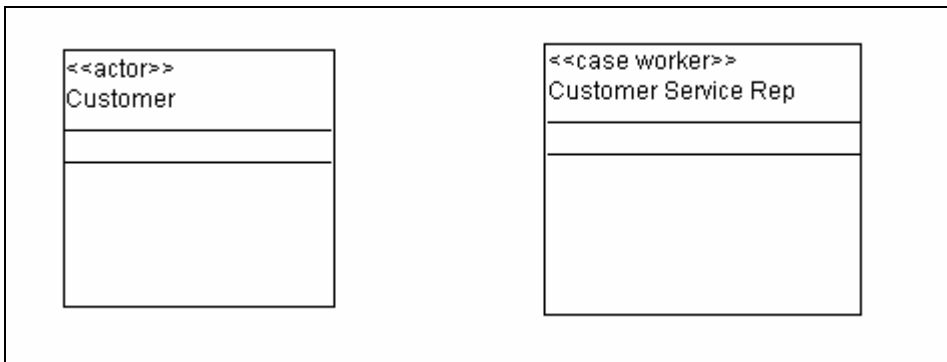


Figure 53 Class diagram after Customer and Customer Service Rep have been added

Next, several more classes will be added. Still using the Class tool, click in the open diagram and create two new classes named Employee and Customer Order. Use the example Class diagram shown in Figure 49 as a guide to placing the classes on the diagram. If a class is named incorrectly, use the Select tool to right click on the class and choose Edit from the pop-up menu.

Specifying Attributes for each class

At this point five different classes have been added to the diagram. The next step will be to further define each class with attributes. To add attributes to a class, use the select tool to right-click on the Customer class and then choose the Edit from the pop-up menu. The first page of the dialog box prompts for a description of the class. Provide text similar to the following: “A customer is a person who registers on our music store web site, browses the product catalog and purchases products from us.”

Next, click on the Attributes tab and put in attributes for the Customer as follows:

Attribute Name	Type	Access?	Primary Key (PK)?
Customer_ID	char	Public	Yes
Last_Name	char	Public	No
First_Name	char	Public	No
Street_Address	char	Public	No
City	char	Public	No
State	char	Public	No
Postal_Code	char	Public	No

Click on the OK button to close this dialog box. Next, edit the Employee class and add the following attributes:

Attribute Name	Type	Access	Primary Key (PK)?
Employee_ID	int	Public	Yes
Last_Name	char	Public	No
First_Name	char	Public	No
Department	int	Public	No
Salary	float	Public	No

Click on the OK button and edit the Customer Service Rep class. Add the following single attribute (in a later step, other attributes will be inherited from the Employee class defined above):

Attribute Name	Type	Access	Primary Key (PK)?
Service_Rank	char	Public	No

Click on the OK button and then edit the Customer Order Class using attributes:

Attribute Name	Type	Access	Primary Key (PK)?
Order_Number	char	Public	Yes
Order_Date	char	Public	No
Order_Item	Orderitem	Public	No

When specifying the data type “Orderitem”, a dialog box will appear prompting to create this new class in a specific Package. Click the Choices button and drag over the MusicStorePackage as shown in Figure 54. Then close up the list and click on the OK button. A new class called Orderitem will be created and the dialog box will open automatically allowing the specification of this new class. Add the following attributes to the new Orderitem class:

Attribute Name	Type	Access	Primary Key (PK)?
Line_Number	int	Public	Yes
Quantity	int	Public	No
Extended_Price	float	Public	No

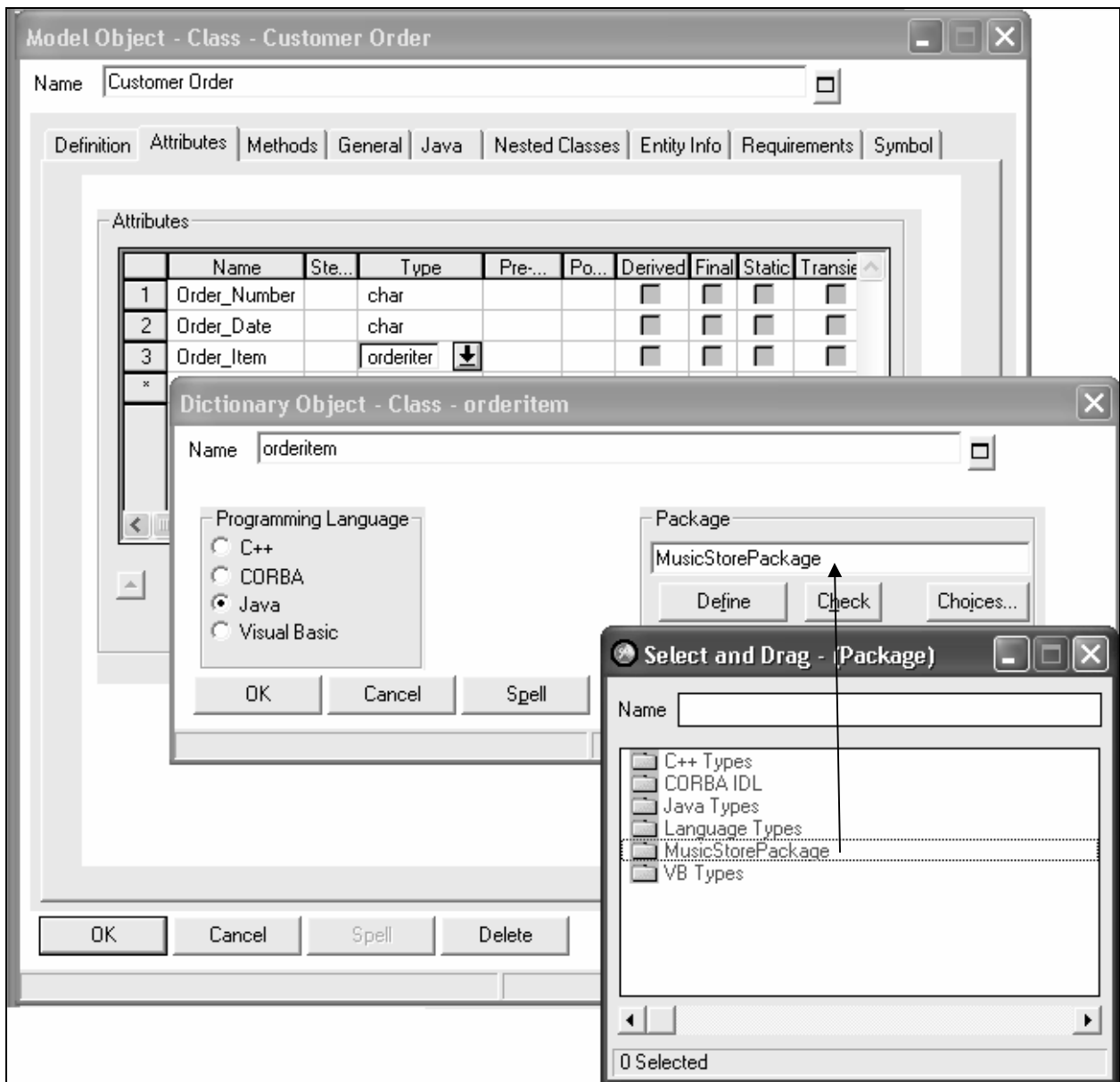


Figure 54 Specifying a new orderitem class

To display this new Orderitem class on the diagram, open the Definitions section of the Browser. Then open the Packages section and inside of that open the MusicStorePackage as shown in Figure 55. Drag the “Orderitem [Class]” entry over to the UML diagram.

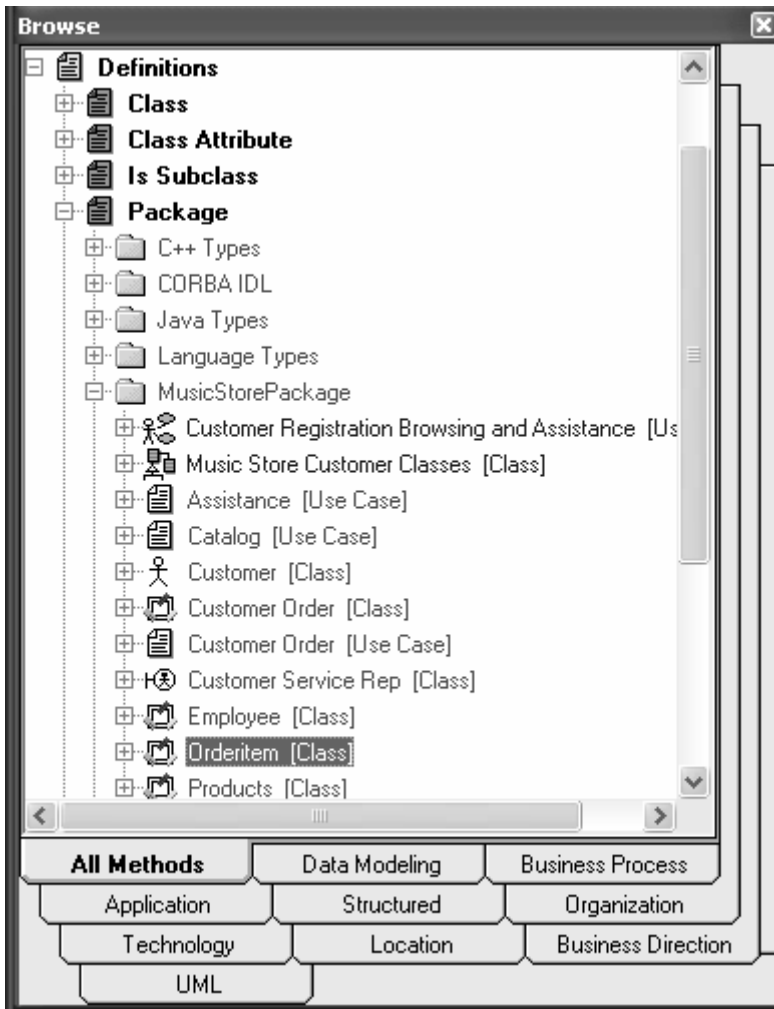


Figure 55 Browsing the contents of the MusicStorePackage package

At this stage, the UML Class diagram should look similar to Figure 56. If the “Persistence” keyword is not displayed in the classes (as shown in the figure), right-click on one of the classes on the diagram and select the `Display Mode...` option from the pop-up menu. Make certain the “Persistence” item is checked off as shown in Figure 57. Note that in this example, only entity classes (that are persistent) are being modeled. Boundary and control classes are typically not persistent and would be marked as “transitory” as shown on the second page of the General tab of the Class edit dialog box as shown in Figure 58. This figure is shown only as an illustration and the class will not be added to the current model.

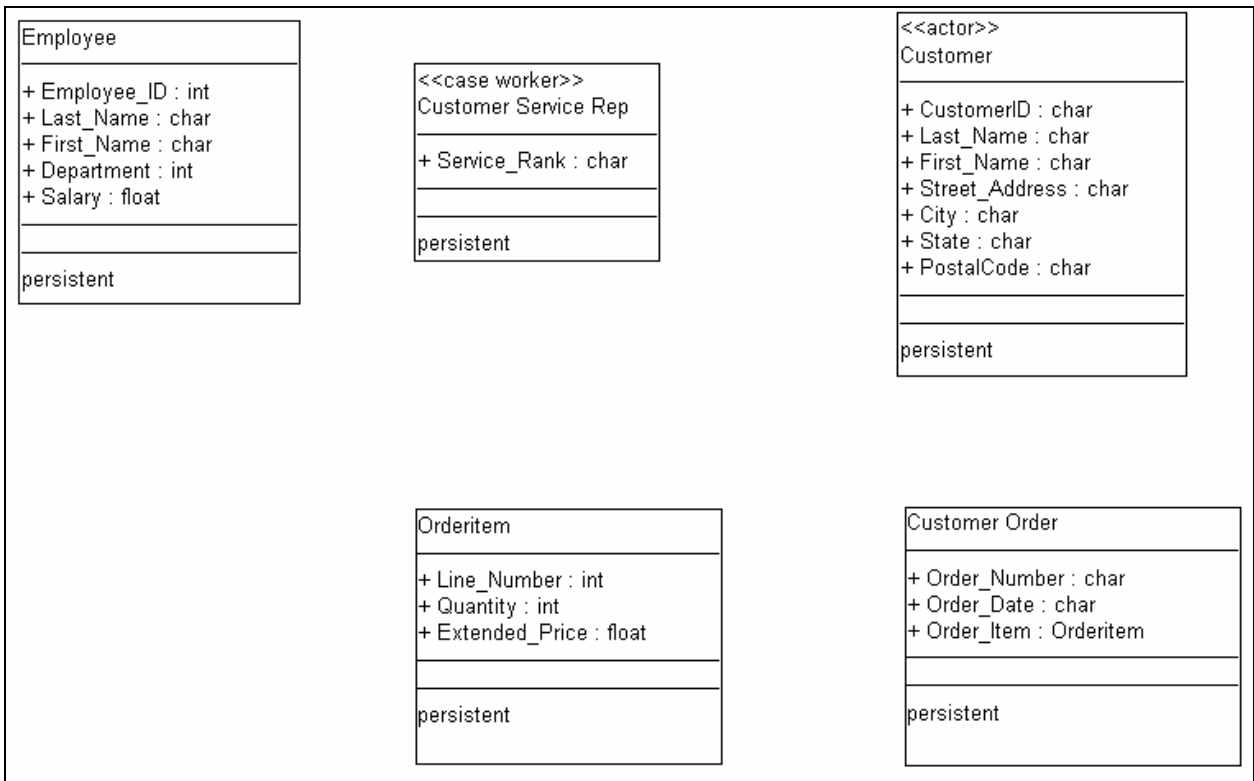


Figure 56 UML Class diagram after specifying 5 classes

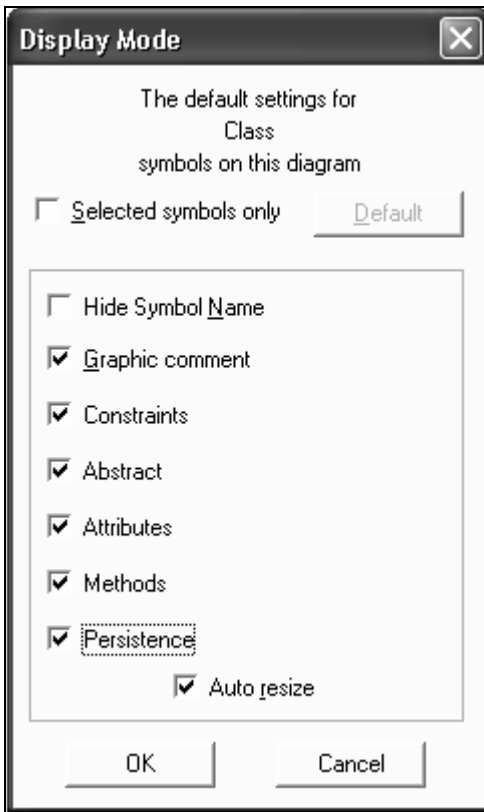


Figure 57 Displaying the Persistence indication on the class diagram

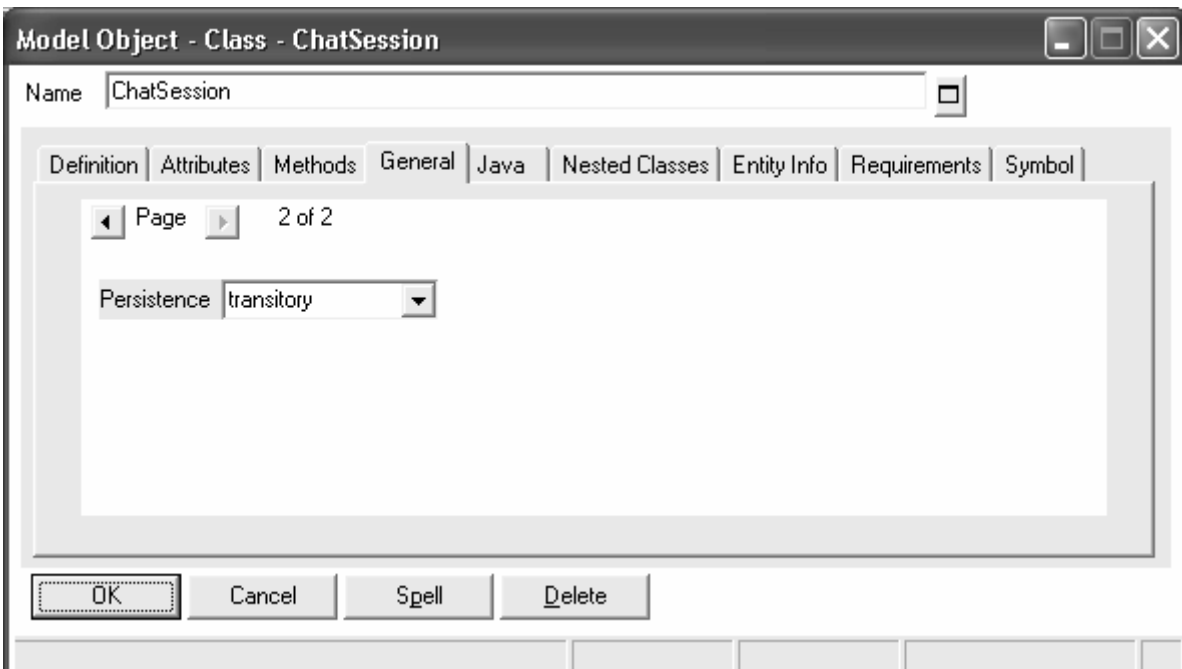



Figure 58 Example of defining a transitory class

Drawing associations and other relationships between classes

Now that the classes have been added to the diagram, associations and other types of relationships can be drawn between them. To begin drawing associations, select the Association tool from the button bar: . The first association will be drawn between the Customer Service Rep and the Customer classes. Begin by clicking on the Customer Service Rep class. A bold plus sign **+** will appear on the edge of this class. Move the mouse over to the Customer class and click on that class. A line will connect the two classes as shown in Figure 59. Once the mouse button is released, a prompt will appear over or next to the association line. Type in “assists” as the name of the association for this example as shown in Figure 60.

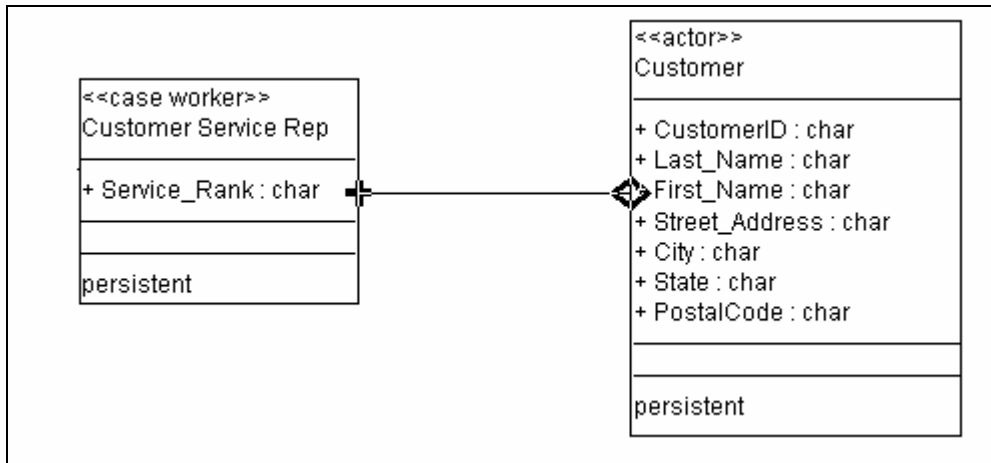


Figure 59 Drawing an association between two classes

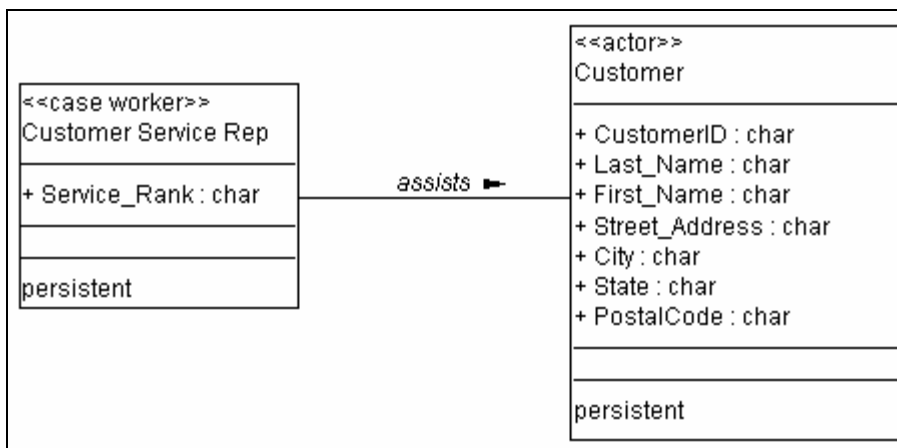


Figure 60 Specifying a name for the association

Right-click on the association line and choose **Edit** from the pop-up menu. In this example, a customer could be assisted by Zero, One or More Customer service representatives. A Customer Service Rep could assist Zero, One or More Customers. Set the options for the Multiplicity of each end of the association accordingly as demonstrated in Figure 61.

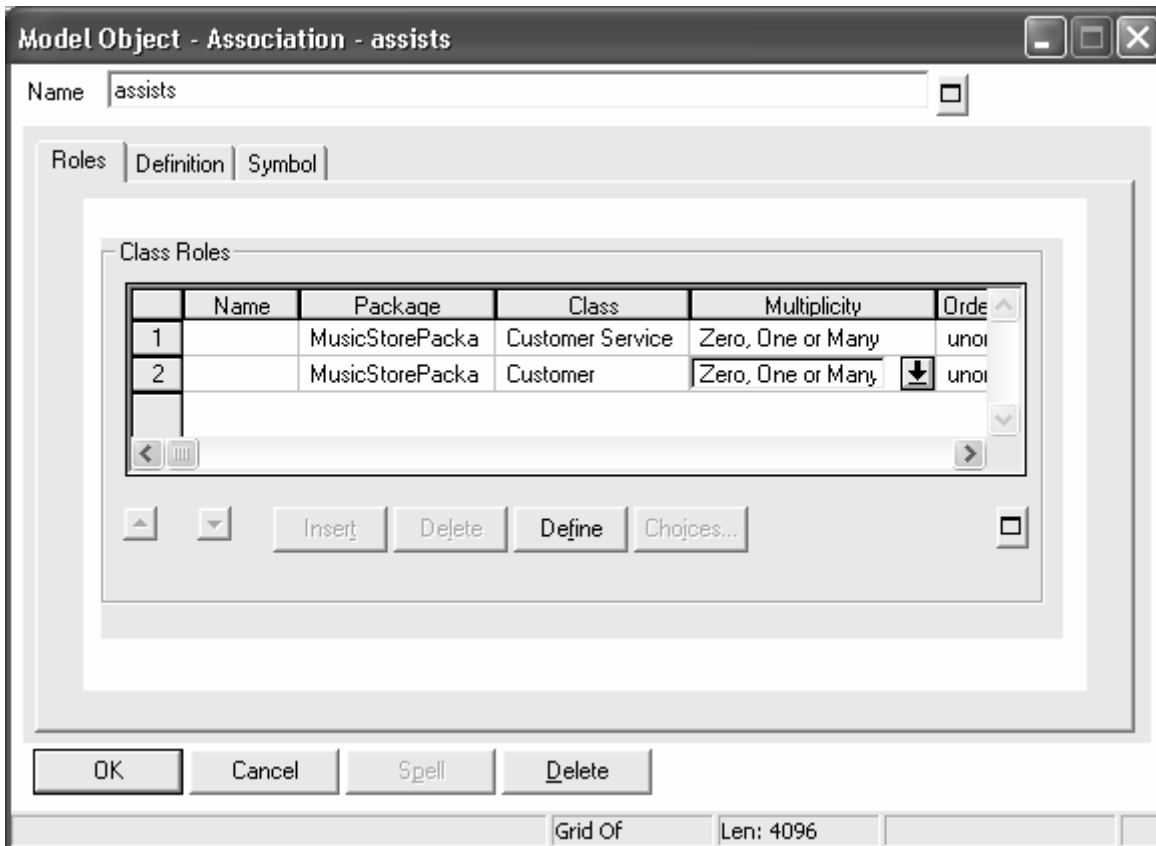


Figure 61 Specifying the multiplicity of the two ends of the association

Draw an additional association between Customer and Customer Order named “places” with the multiplicities of “Only One” on the Customer side and “Zero, One or Many” on the Customer Order side. This will indicate that a Customer places zero, one or more orders, while an Order is placed by one and only one Customer.

Draw an additional relationship between the Customer Order and Orderitem classes. Name this association “contains”. Edit this association and specify the multiplicity of “Only One” on the Customer Order side and “One or Many” on the Orderitem side. Before clicking on the OK button, scroll to the right and change the Aggregation property from “none” to “composite” for the Customer Order side of the association as shown in Figure 62. This will indicate that the Orderitem is a part of the Customer Order.

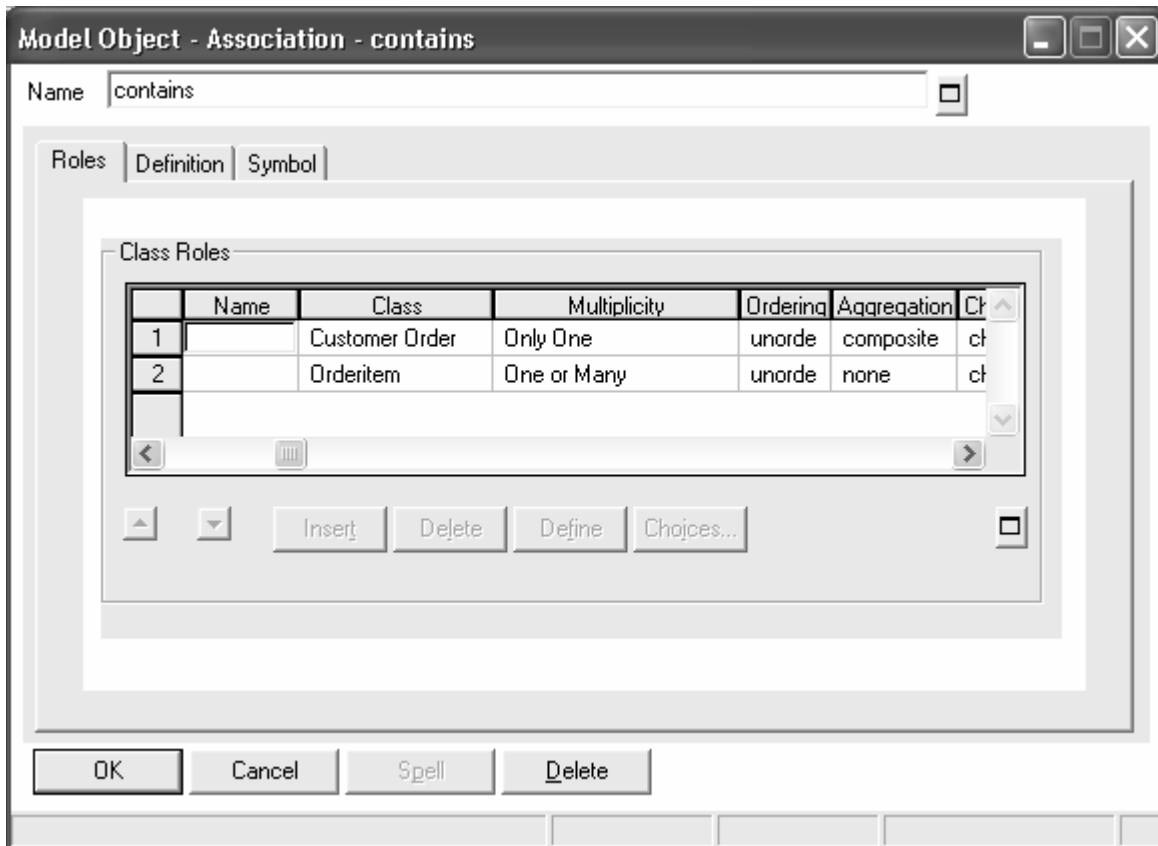



Figure 62 Specifying multiplicity and aggregation properties of an association

A third type of relationship is inheritance. In this example, the Customer Service Rep is a type of Employee. To show this relationship, select the “Inherits From” tool by clicking on its icon: . Using this tool, click on the Customer Service Rep class and then click on the Employee class. The arrow should point from Customer Service Rep to the Employee as shown in Figure 49 at the start of this section.

To save the work completed thus far, press CTRL + S or pull down the File menu and choose the Save Diagram menu item. Click “Yes” if asked to confirm.

Exercise: Extending the UML Class Diagram

For this exercise, extend the existing UML Class Diagram with the a Product class using the following attributes:

Attribute Name	Data Type	Access	Primary Key (PK)?
Product_ID	char	Public	Yes
Catalog_Number	char	Public	No
Album_Title	char	Public	No
Artist	char	Public	No
Genre	char	Public	No
Release_Date	int	Public	No
Record_Label	char	Public	No

Standard_Price	float	Public	No
----------------	-------	--------	----

Associate this class with the Order Items class and name the association “purchased”. Edit the relationship such that one Product may be purchased on zero, one or many order items and an order item is an item for one and only one product. A portion of the completed diagram is shown in Figure 63.

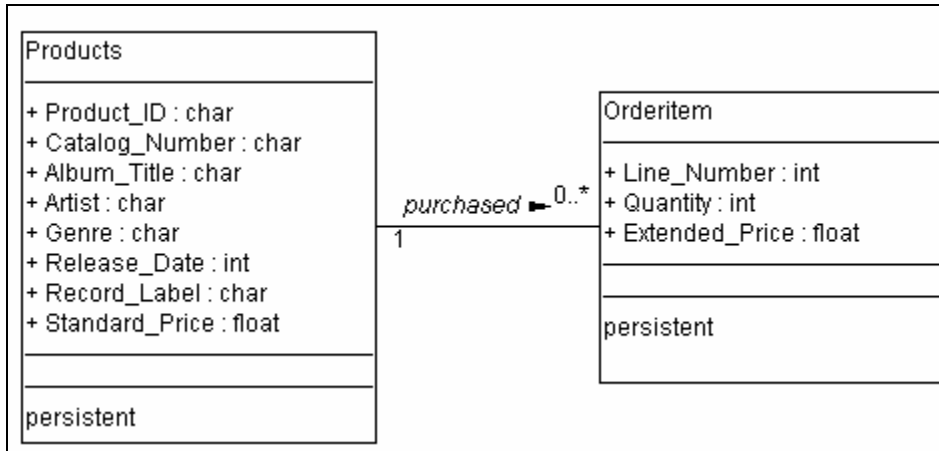


Figure 63 A portion of the completed class diagram after adding the Products class

Optional Exercise: Mapping Persistent Classes to Entities

It is often the case that information systems designed using Object Oriented modeling techniques are ultimately implemented using relational databases for data storage. In such cases, persistent classes such as Products, Customers, Customer Orders and Orderitem modeled in this section will ultimately be implemented as tables in a relational database management system. To assist in this process, a common step is to convert the UML Class diagram into an Entity Relation (ER) diagram. This process is called “mapping” as each persistent class is “mapped” onto one entity.

To demonstrate this facility, the UML Class diagram drawn in this section will be converted to an ER diagram. To get started with this exercise, make certain the UML class diagram is still displayed and has been saved. Pull down the **Dictionary** menu and select the **Convert to ER Diagram...** menu item as shown in Figure 64. A dialog box will appear as shown in Figure 65. Use “Project Data Model 1” as the Destination model (this may have to be typed in) and provide “Music Store Class ER” as the Destination diagram as shown in the figure. Click on the **OK** button to continue.

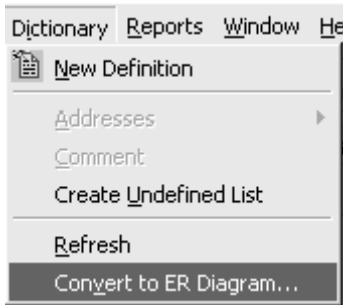


Figure 64 The Convert to ER Diagram menu item

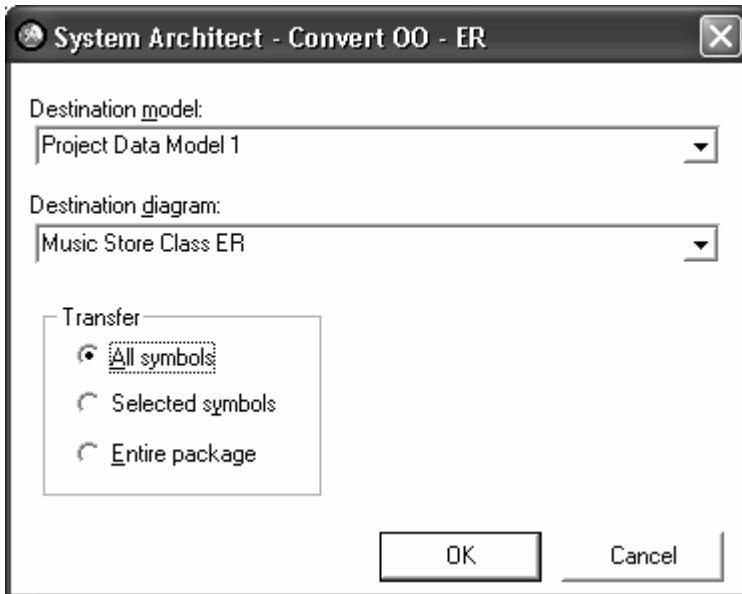


Figure 65 Convert OO Class diagram to ER diagram dialog box

After the conversion is complete, the entities can be re-arranged as shown in Figure 66. The dashed box surrounds the Employee and Customer Service Rep Supertype/Subtype entities. From this point additional work may need to be done on the diagram. For example, the relations should be replaced with proper relations that specify identifying or non-identifying attributes. Once the ER model has been carefully reviewed and revised, it can be transformed into a physical data model as described in Section 8.

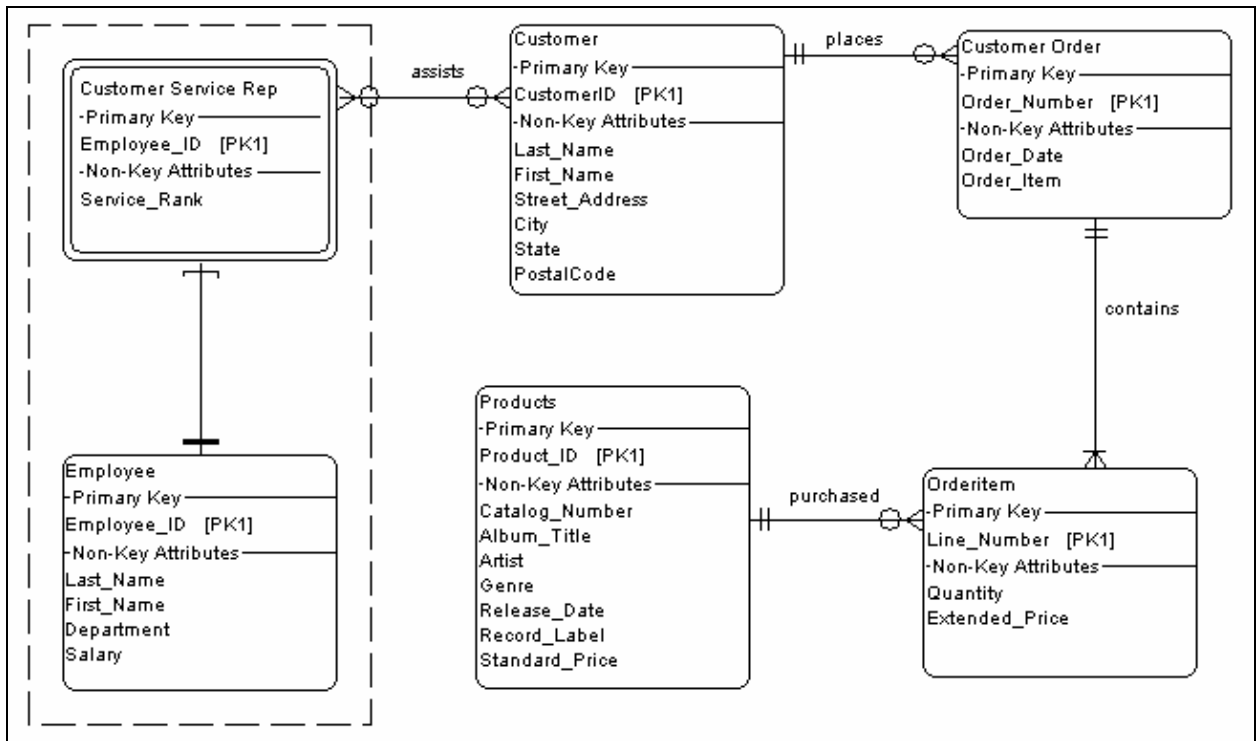


Figure 66 ER diagram resulting from converting the UML Class diagram

11. Object Oriented modeling with Sequence diagrams

The purpose of Sequence diagrams is to document how objects interact with one another over time at fine level of detail. The examples in the section assume that the classes created in Section 10 are available in the currently selected encyclopedia. The classes should include: Customer, Customer Service Rep., Customer Order and Products.

To begin creating a new Sequence diagram, choose the **UML** (Unified Modeling Language) tab. Pull down the **File** menu and select the **New Diagram** menu item. Choose **Sequence** by double clicking on its entry as shown in Figure 67. Name the new Sequence diagram “Music Store Customer Sequences” as shown in Figure 68. When prompted for the Package in which to create this diagram, choose the “MusicStorePackage” as shown in Figure 69.

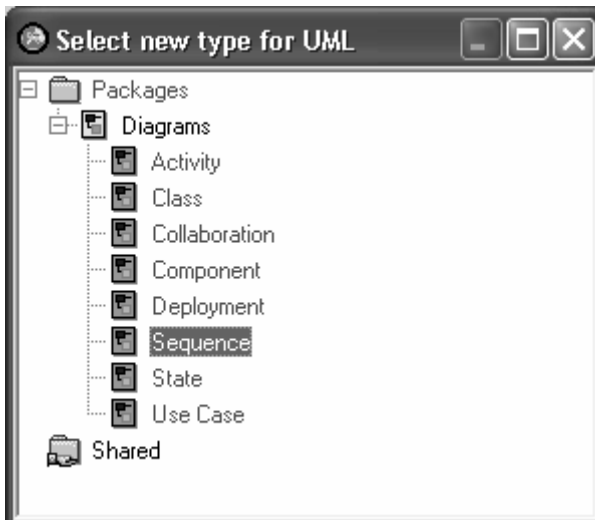


Figure 67 Creating a new Sequence diagram

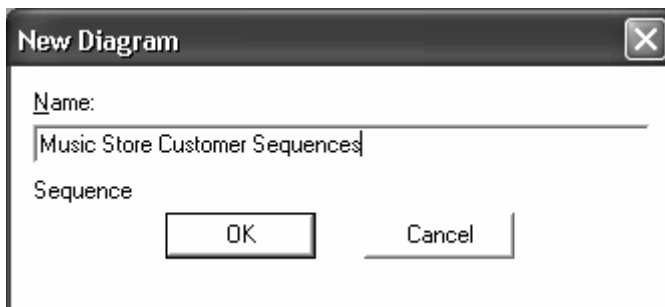


Figure 68 Naming the new Sequence diagram

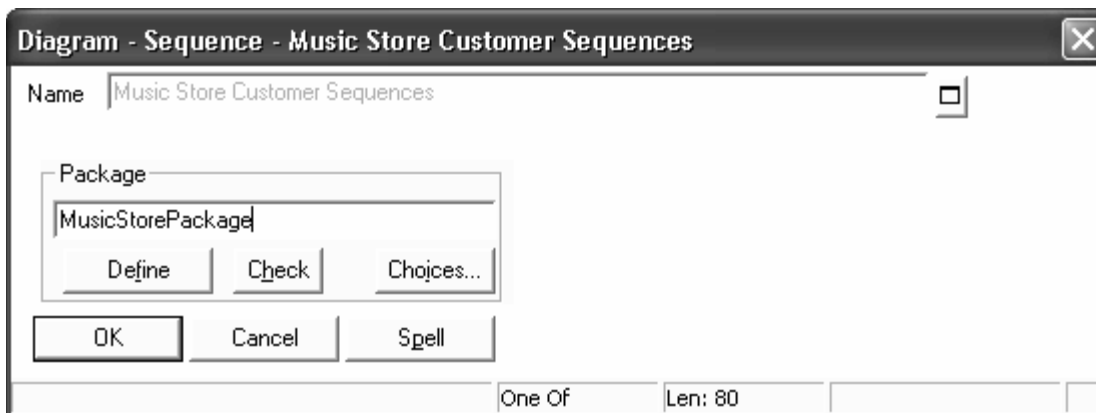


Figure 69 Creating the new Sequence diagram in the MusicStorePackage

The main focus of this example sequence diagram will be on the interaction between the Customer and the Products classes. The Customer will likely spend the majority of their time interacting with the different products the music store carries by browsing music genres (Rock, Country, Hip Hop, Classical, etc.), searching for albums by artists that they enjoy and so on. Ultimately, the Customer will elect to purchase some products. A completed example of a sequence diagram depicting this interaction is shown in Figure 70.

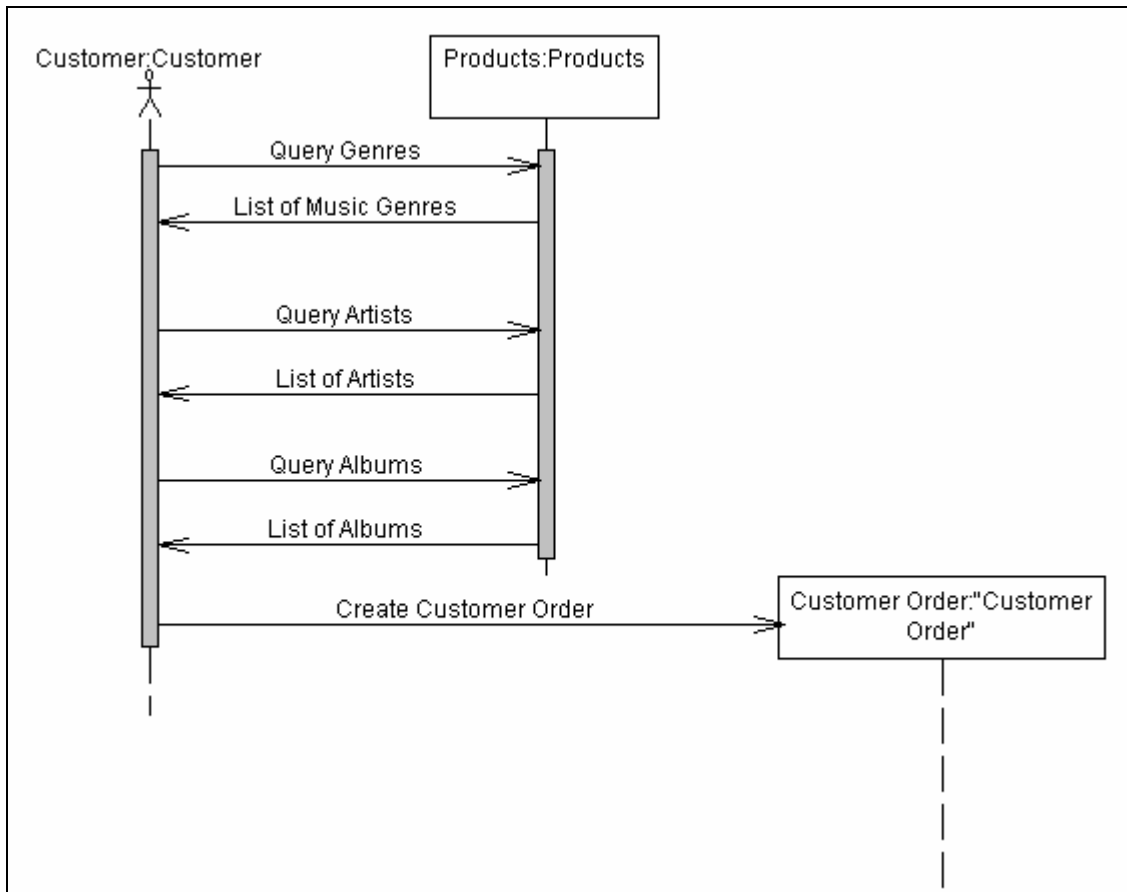


Figure 70 Example Sequence diagram

The necessary classes to be used for this diagram have already been created as part of Section 10. To populate the new Sequence diagram, all that is necessary is to drag and drop existing classes on to the diagram. To accomplish this, keep the new Sequence diagram open and use the Browse window and click on the **All Methods** tab. Navigate to the Definitions tree and then open the Class branch of the Definitions tree. The list of classes will appear as in Figure 71. Locate the Customer class and drag it onto the diagram as shown in the figure. Repeat this for the Products class and the Customer Order class. Arrange the three objects on the diagram as shown in Figure 70. Note that on Sequence diagrams, classes or objects are displayed with a box (or other icon as per object stereotype) at the top with a dashed “life line” leading down from them. Lifelines may be extended by dragging the resize handles of the objects downwards.

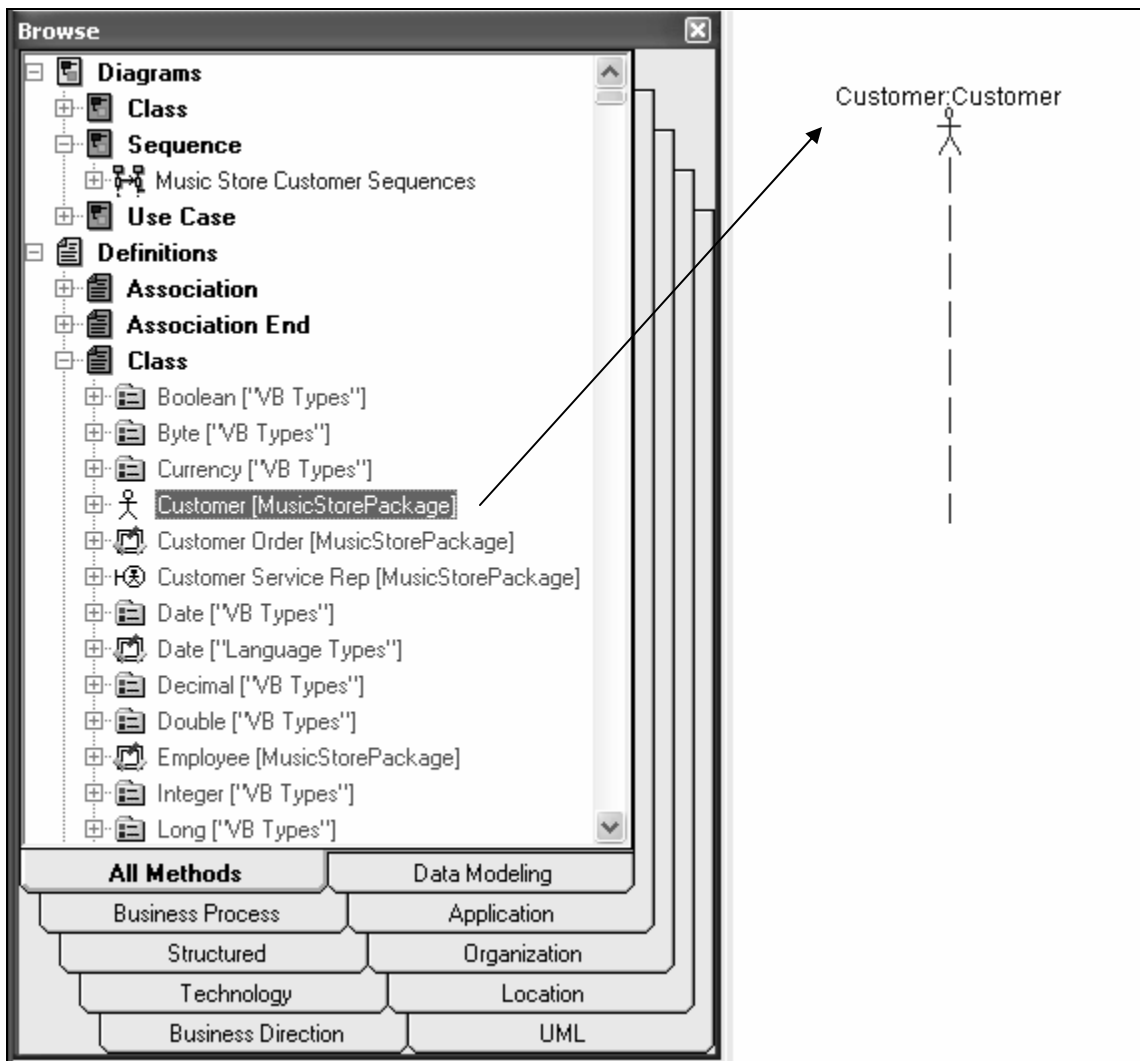



Figure 71 Dragging and dropping the Customer class onto the sequence diagram

Once the three classes are on the diagram, the message/stimulus lines can be added. Choose the “Message/Stimulus” tool from the button bar: . Click on the Customer lifeline just below the Customer class. A bold plus sign **+** will appear. Next, click on the Products lifeline to indicate the destination of the message/stimulus. This process is demonstrated in Figure 72. A prompt will appear for the name of the message. For this example, use “Query Genre” as shown in Figure 70 and press Enter. This message indicates that the Customer has requested the list of music genres available in the product catalog. The return message in response to this request should be a list of music genres. To show this response, use the same Message/Stimulus tool to click on the Product lifeline first and then on the Customer lifeline. Name this message “List of Music Genres” as shown in Figure 70. The results of these two steps should appear as shown in Figure 73.

Create the four additional messages to the Customer and Product lifelines as shown in Figure 70. If a message is created in the wrong direction by mistake, simply select it, delete it and draw it

again. To rename a mislabeled message, right-click on the message line and choose `Edit` from the pop-up menu.

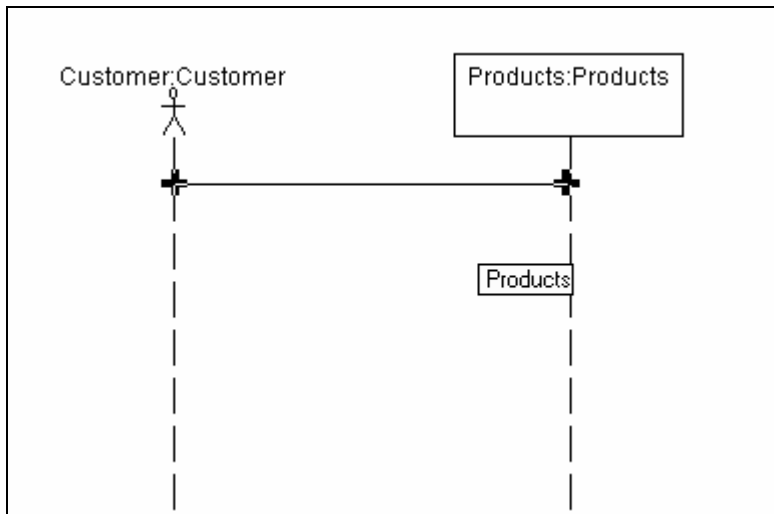


Figure 72 Adding a message between Customer and Products

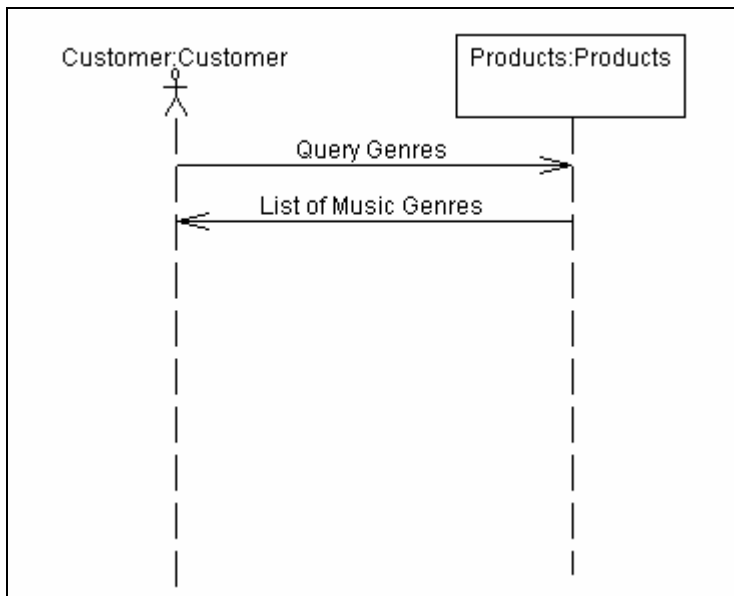



Figure 73 Sequence diagram after two messages have been added

At this point the sequence diagram shows the interaction between the Customer and the Products. The final element of this interaction will be the point where the Customer decides to purchase a product and in doing so will create a Customer Order. To show this on the Sequence diagram, first make certain the Customer Order object is positioned below the other classes as demonstrated in Figure 70. Using the Message/Stimulus tool  draw a message line between the Customer lifeline and the object box for the Customer Order (not the Customer Order lifeline) as shown in Figure 74. Name this message “Create Customer Order”.

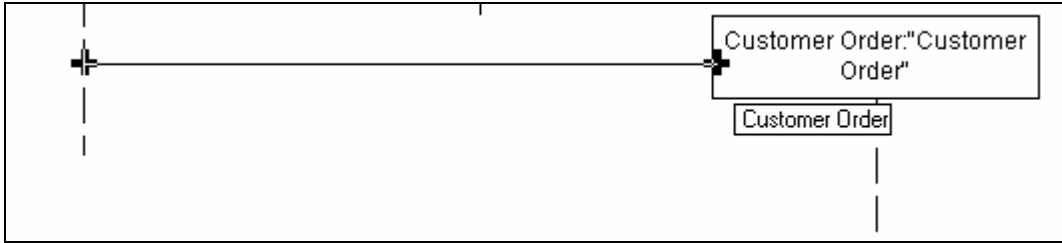



Figure 74 Drawing a message from Customer lifeline to the Customer Order object

Designating Focus of Control areas

As a final step, the lifelines can be marked with areas of focus of control. These indicate where the object is active. To designate focus of control on the Customer lifelines, choose the Focus of Control tool  and click on the Customer lifeline. A small gray shaded rectangle will appear on top of the lifeline. Choose the Select tool and then click on the Focus of Control rectangle. Use the grab boxes to stretch and/or position the rectangle until it covers all of the points of message interaction as shown in Figure 75. Repeat these steps to specify the focus of control on the Products class again, using Figure 70 as a guide. Note that the focus of control of the Products class does not cover the interaction between the Customer and Customer Order.

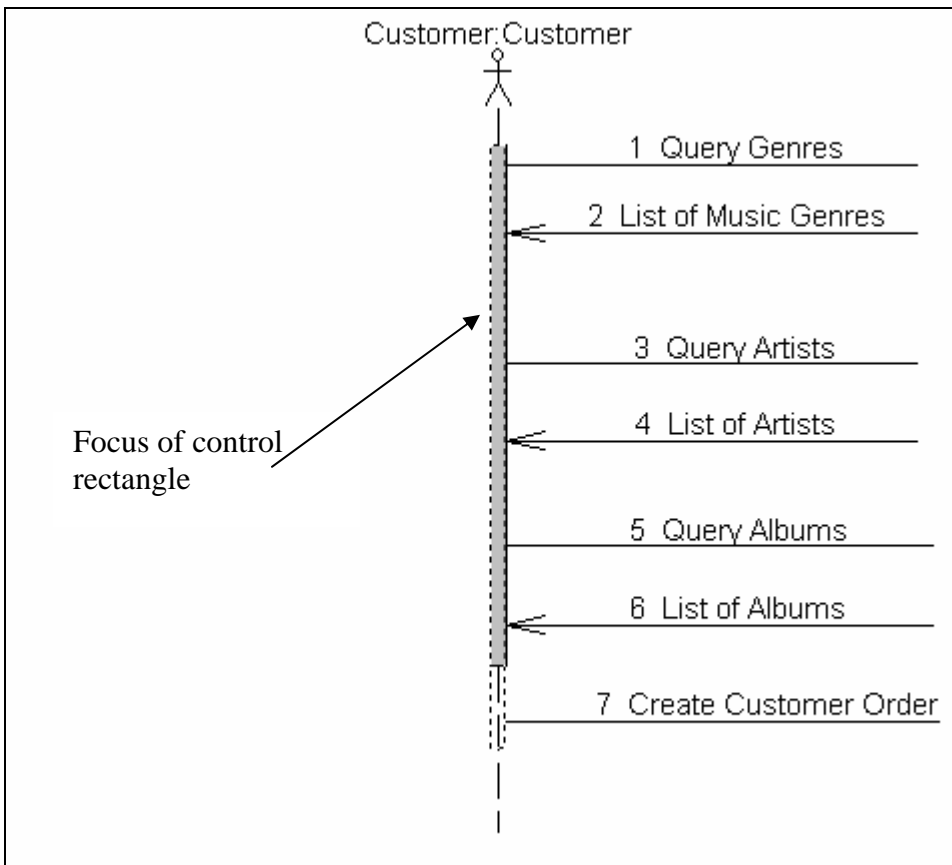


Figure 75 Adding a focus of control area to the Customer lifeline

To save the work completed thus far, press CTRL + S or pull down the File menu and choose the Save Diagram menu item. Click “Yes” if asked to confirm.

Exercise: Modeling Customer interaction with the Customer Service Rep.

In this exercise, the interaction between Customer and Customer Service Rep (CSR) will be modeled in a new diagram. To get started, create a new Sequence diagram named “Music Store CSR” in the MusicStorePackage and drag and drop in the Customer and Customer Service Rep classes. Create a simple message exchange between the two and add the focus of control rectangles as shown in Figure 76.

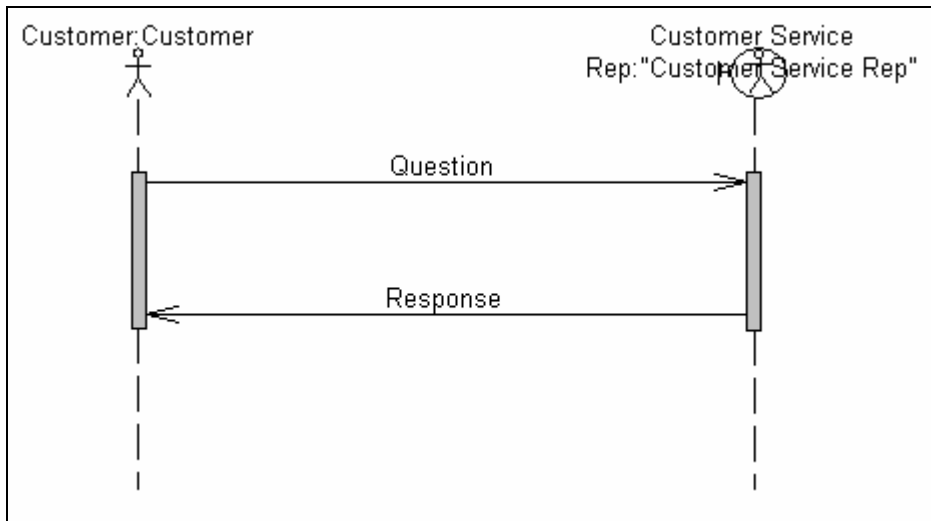


Figure 76 Customer and CSR interaction

12. Object Oriented modeling with Collaboration diagrams

This section introduces collaboration diagrams. Collaboration diagrams, like sequence diagrams, provide a view of the dynamic aspects of an object-oriented system. A collaboration diagram is essentially an object diagram that shows message passing relationships instead of aggregation or generalization associations. Collaboration diagrams are very useful to show process patterns, that is, patterns of activity that occur over a set of collaborating classes.

Collaboration diagrams are equivalent to sequence diagrams, but they emphasize the flow of messages through a set of objects, while the sequence diagram focuses on the time ordering of the messages being passed. Therefore, if the intent is to understand the interactions among a set of collaboration objects, a collaboration diagram should be used. If the intent is to understand the time ordering of the messages, then a sequence diagram should be used. Typically, both types of diagrams will be employed to more fully understand the dynamic activity of the system

Given that a sequence diagram has been created previously as was done in Section 11, the fastest way to create a Collaboration diagram is to generate it automatically from the Sequence diagram. To begin this process, open the Music Store Customer Sequence diagram (as shown in Figure

70) by clicking on the **All Methods** tab in the Browse window. Navigate to the Diagrams and then Sequence branches. Right click on the name of the Sequence diagram and choose Open from the pop-up menu..

To automatically create the corresponding Collaboration diagram, pull down the Draw menu and choose the Synchronize Diagram menu item as shown in Figure 77. A dialog box (shown in Figure 78) will appear confirming the Sequence and Collaboration diagrams are now synchronized. The resulting Collaboration diagram will appear as in Figure 79.

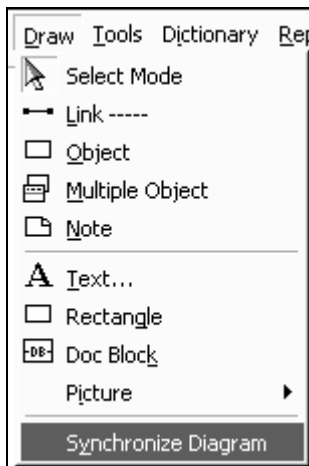


Figure 77 The Synchronize Diagram menu item

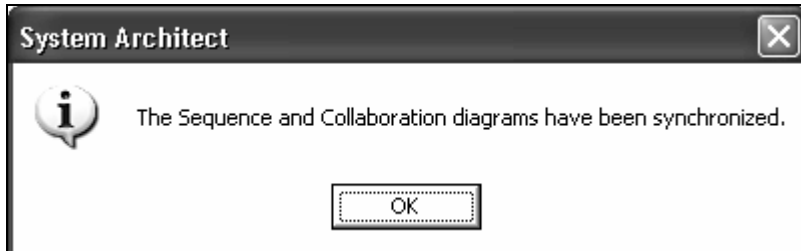


Figure 78 Confirmation after synchronizing the Sequence diagram

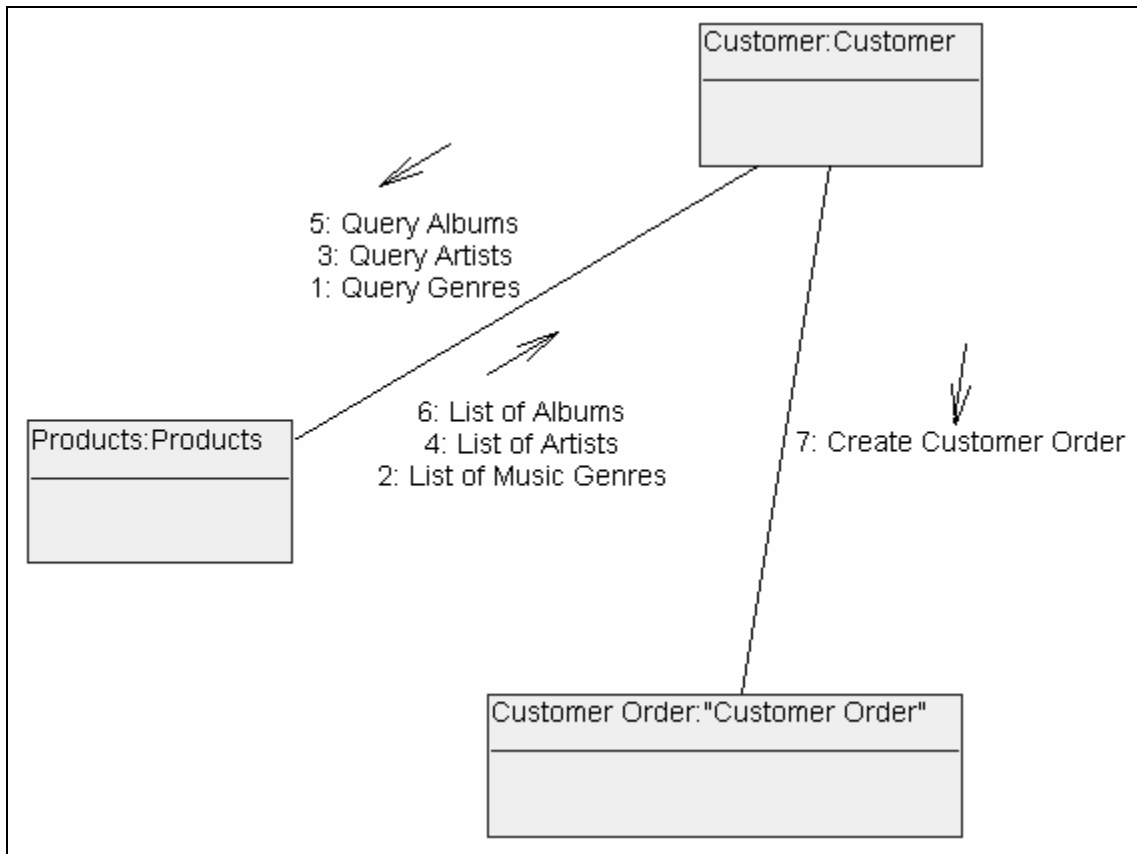



Figure 79 Collaboration diagram created from the Sequence diagram

Note that the default collaboration diagram may not appear exactly as shown in Figure 79. Some re-positioning of the classes may be required.

By default, all messages passed between objects are shown as “Simple” with the double arrowhead. However other message (or event) synchronization types are possible. For example, in this situation, the Customer might query for the list of genres or artists at any time, regardless of whether or not the Products object has indicated it is ready to accept such a request. For this situation, an asynchronous event model is appropriate. To change the synchronization, right-click on the messages numbered 1, 3 and 5 and choose the *Associative* item from the pop-up menu.

The Associative properties of the messages are shown in Figure 80. By default, all of the Message/Stimuli entries should have “simple” as the synchronization method. Three main changes should be made as follows: First highlight message number 1 “Query Genres”. Change the Synchronize entry from “simple” to “asynchronous” and then click on Replace button. Make the same alternation to messages 3 “Query Artists” and message 5 “Query Albums”. Next, change the visibility of the Products:Customer end of the link to “Local” and finally change the Products:Products end of the link to “Global”. The end results should match Figure 80. Note that after clicking OK, the diagram should change slightly with the arrow from Customer to Products changing to a half-arrow:  to indicate asynchronous message synchronization.

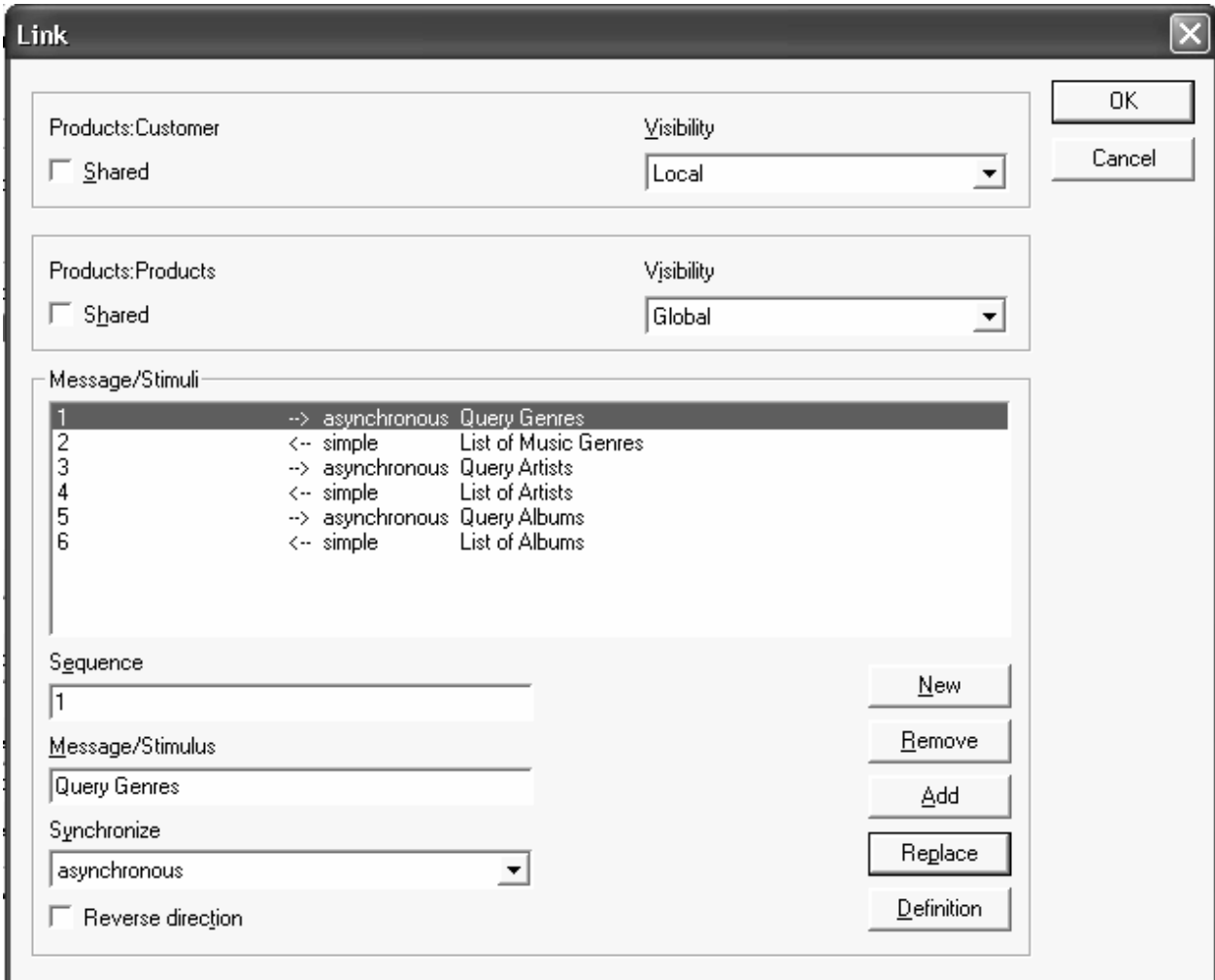


Figure 80 Editing the associative properties of messages

As a final step, synchronize the Collaboration and Sequence diagrams by pulling down the Draw menu and choosing the Synchronize Diagram menu item as shown in Figure 77. SA will prompt for confirmation that the Sequence diagram will be updated to reflect the changes made to the Collaboration diagram as shown in Figure 81. Click Yes to continue the update.

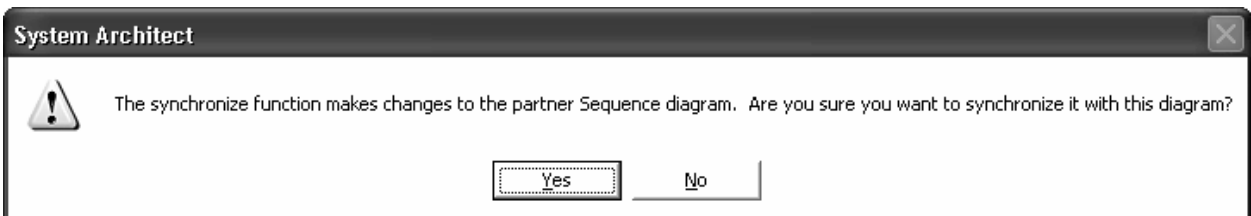


Figure 81 Synchronize diagrams confirmation dialog box

Be certain to save both diagrams (press CTRL + S or pull down the File menu and choose the Save Diagram menu item) before closing them.

Exercise: Creating a Collaboration diagram for the Music Store CSR Sequence diagram

For this exercise, open the Music Store CSR sequence diagram created during the exercise in Section 11. Use the `Synchronize Diagram` method to automatically create a Collaboration diagram. The results should appear similar to the diagram shown in Figure 82. Note that some rearrangement of the classes may be required to make the diagram match Figure 82.

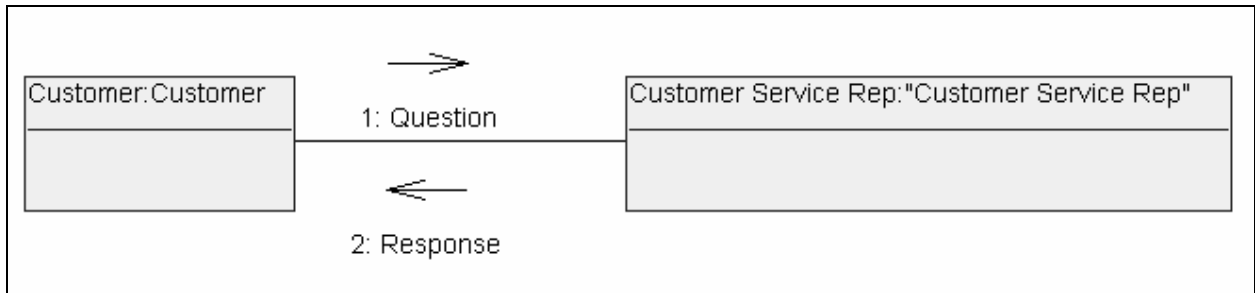


Figure 82 Collaboration diagram for the Music Store CSR example

13. Object Oriented modeling with Activity diagrams

Activity diagrams are similar to flowcharts and data flow diagrams in that they graphically depict the sequential flow of activities of either a business process or a use case. They are different from flowcharts in that they provide a mechanism to depict activities that occur in parallel.

To begin creating a new Activity diagram, choose the **UML** (Unified Modeling Language) tab in the Browser window. Pull down the `File` menu and select the `New Diagram` menu item. Choose `Activity` by double clicking on its entry as shown in Figure 83. Name the new Activity diagram “Music Store Customer Activity”. When prompted for the Package in which to create this diagram, choose the “MusicStorePackage” as done in previous sections. Finally, type in “MusicStoreActivity” as the name for the Activity Model as shown in Figure 84.

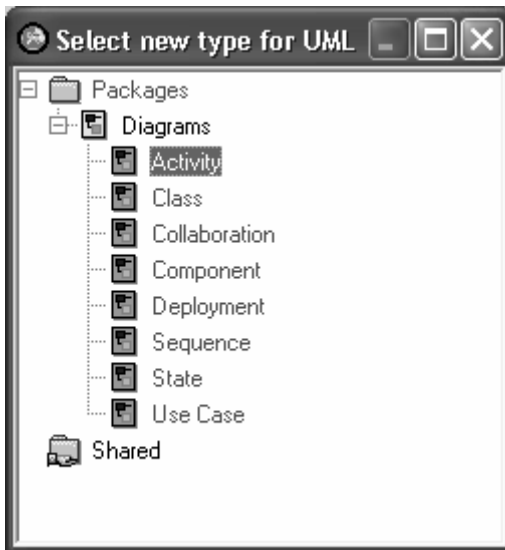


Figure 83 Creating a new Activity Diagram

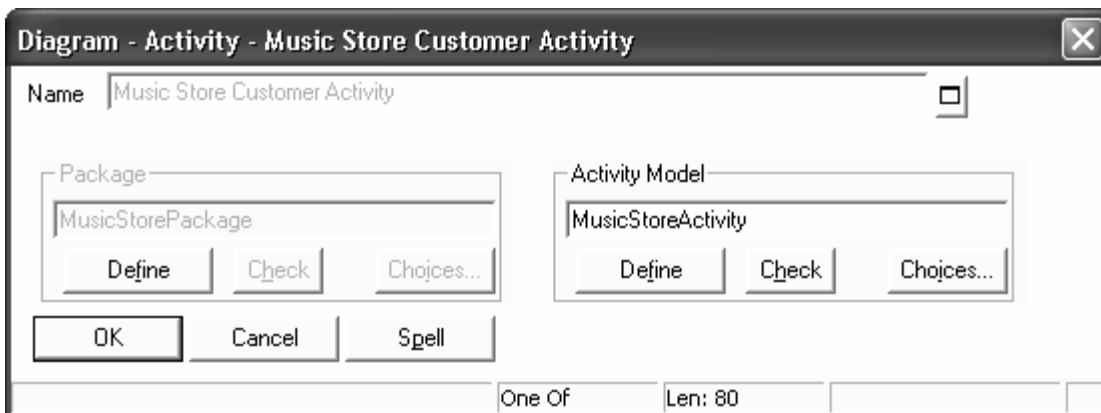


Figure 84 Specifying the MusicStoreActivity as the activity model

The example activity diagram that will be constructed is shown in Figure 85. The activities to be modeled begin when the customer visits the music store web site. The customer has a choice as to whether they would like to browse the product catalog by clicking through music genre, artist and album or if they would prefer to use a music search function that will accept the name of an album or artist and perform a search of the catalog. The system keeps track of the artists the customer browses so that the music store can take advantage of targeted marketing opportunities at a later time.

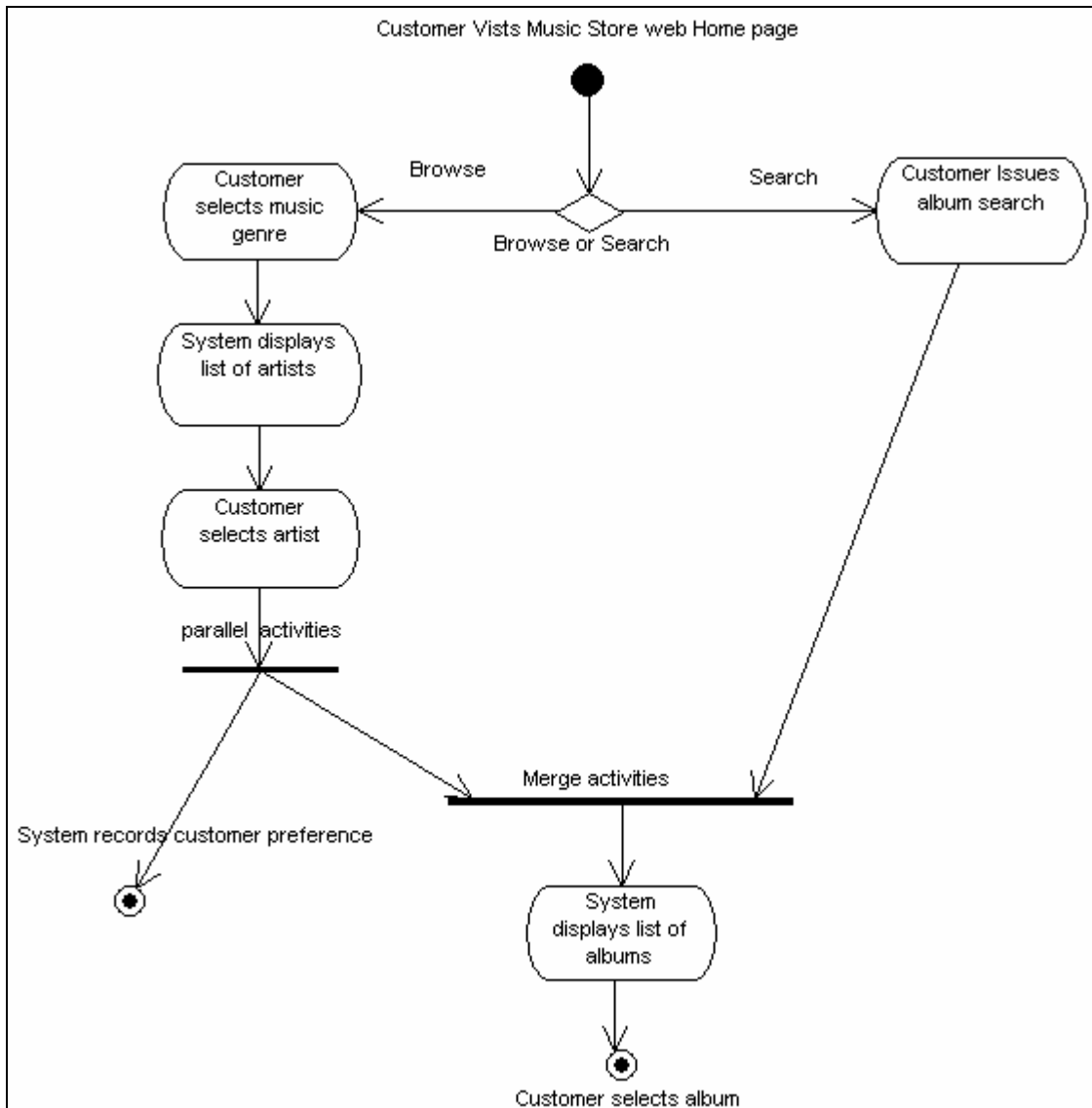




Figure 85 Example Activity diagram for Customer interaction with a Music store web site


The first step in drawing the activity diagram is to create an Initial Action State where the activity begins. Select the Initial Action State tool (solid circle)  from the button bar and click on the open space of the new diagram towards the top. Name this initial action “Customer Visits Music Store web site” and press enter.


The next step is to add regular Action States. Select the Action State tool (hollow oval)  from the button bar and click below and to the left of initial action state. Name this new action state “Customer selects music genre”. With the Action State tools still selected, add the following additional action states to the diagram using Figure 85 as a guide:


- System displays list of artists
- Customer selects artist

- Customer issues album search (positioned to the right)
- System displays list of albums (positioned towards the bottom of the diagram)


As with other diagram types, if an item on the diagram is mislabeled, use the Select tool to right-click on an item and choose `Edit` from the pop-up menu. The name of the item can be changed in the dialog box that will be displayed.

Two Final Action States will also need to be added. Select the Final Action State tool (button style circle)  and click on the diagram towards the bottom of the diagram. Label this final action state “Customer selects album”. With the Final Action State tool still selected, click once more on the bottom left of the diagram and label this “System records customer preference”.

Next a Decision activity will be added to the diagram. Select the Decision tool  from the button bar and click on the diagram just below the initial action state. Label this decision “Browse or Search”.

At this point all of the major action states have been added to the diagram. The final step will be to add simple and complex transitions to the diagram. The complex transitions will be added first. Complex transitions are used when transitions need to proceed in parallel or when parallel activities need to be joined or merged together. Select the complex transition tool  from the button bar and click on the diagram just above the “System displays list of albums” action state. Label this complex transition “Merge activities”. With the complex transition tool still selected, click once more below the “Customer selects artist” action state. Label this complex transition “parallel activities”.

Simple transitions will now be used to connect all of the action states. Select the Action

Transition tool (single arrow)  from the button bar. The first transition will be drawn from the initial action state to the decision diamond. Begin by clicking on the initial action state. A bold plus sign **+** will appear on the initial action state. Position the mouse pointer over the “Browse or Search” decision activity and a second plus sign will appear. Click on the decision activity and the transition will be created as shown in Figure 86. Leave this particular transition label blank. It is not necessary to label each transition. As a final step, the associative properties dialog box will appear offering the chance to specify the direction of the transition. For this example, ensure that the Flow (Sequence) radio button under the “To” entry is selected as shown in Figure 87 and click the OK button.

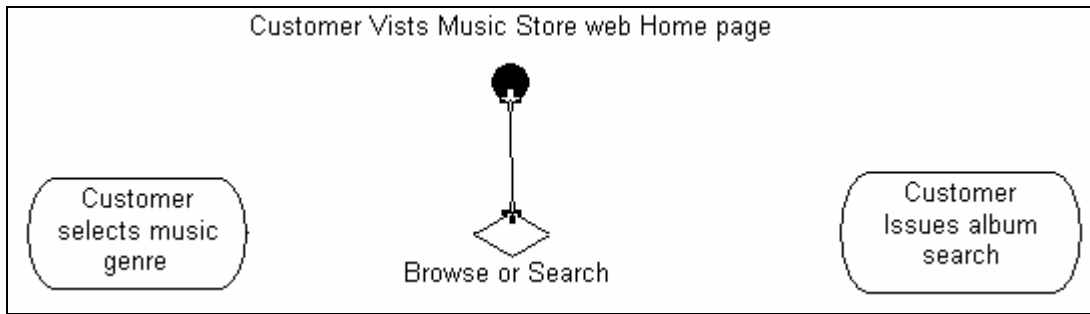


Figure 86 Connecting action states with the Action Transition tool

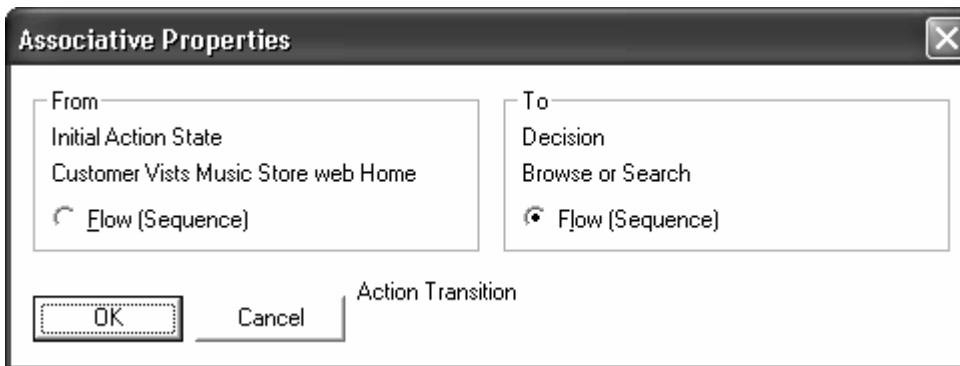


Figure 87 Confirming Associative properties of the action transition

With the Action Transition tool still selected, create the remaining transitions as shown in Figure 85. Only two transitions have been labeled. The transition from the decision to the left is labeled “Browse” and the transition going to the right is labeled “Search”.

Exercise: Creating an Activity diagram for Customer assistance

For this exercise, create a new Activity diagram named “Music Store Customer Assistance” that will document how a customer obtains assistance for questions they may have. Create this new diagram in the MusicStorePackage and use the same MusicStoreActivity activity model as shown in Figure 84. The diagram should have the following items:

- The initial action state should be that the Customer requires assistance
- A Decision state should be next where the Customer chooses between browsing an on line FAQ (Frequently Asked Questions) or contacting a customer service representative
- The Action states for “Browse On-line FAQ” and “Call Customer Service Rep.” should then follow according to which option the Customer chooses.
- Add Decision states such that if the Customer’s question has been answered we reach a Final Action State. If the Customer’s question is not answered then control returns to the first Decision state.

The resulting diagram should resemble the one shown in Figure 88

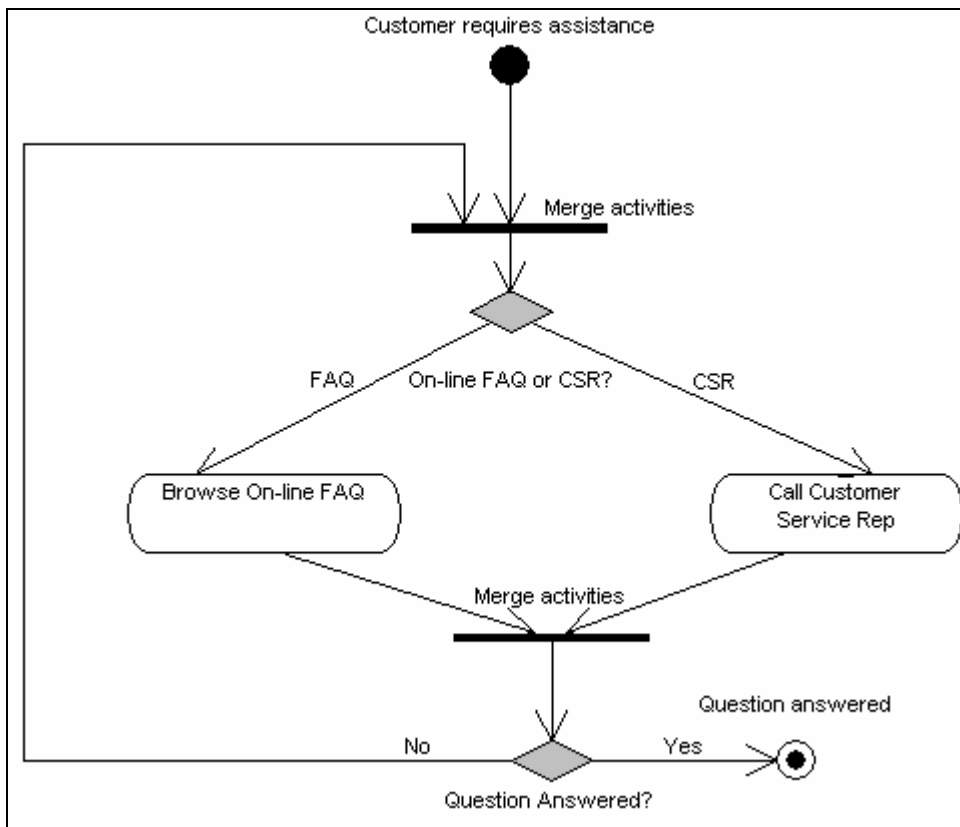


Figure 88 Exercise for Customer Assistance Activity diagram

14. Object Oriented modeling with State diagrams

A state diagram is a dynamic model that shows the different states that a single object passes through during its life in response to events, along with its responses and actions. Typically, state diagrams are not used for all objects, but just to further define complex objects to help simplify the design of algorithms for their methods. In the On-line Music Store example, the Customer object is central to the activities in a large portion of the total system. Therefore, the various states of the Customer object will be modeled for this section of the tutorial.

To begin creating a new State diagram, choose the **UML** (Unified Modeling Language) tab in the Browser window. Pull down the **File** menu and select the **New Diagram** menu item. Choose **State** by double clicking on its entry as shown in Figure 89. Name the new State diagram "Music Store Customer States". When prompted for the Package in which to create this diagram, choose the "MusicStorePackage" as done in previous sections. For the Class name, click on the **Choices** button and drag and drop the Customer Order class into the field as shown in Figure 90. Close the Class list by clicking on the close button and then click **OK** to create the new diagram.

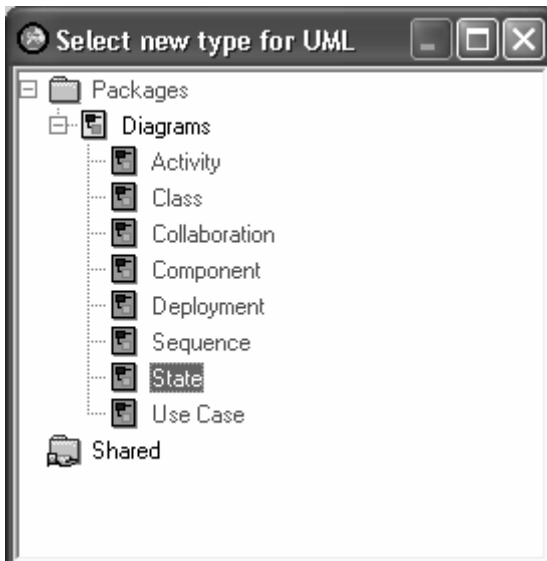


Figure 89 Creating a new State diagram

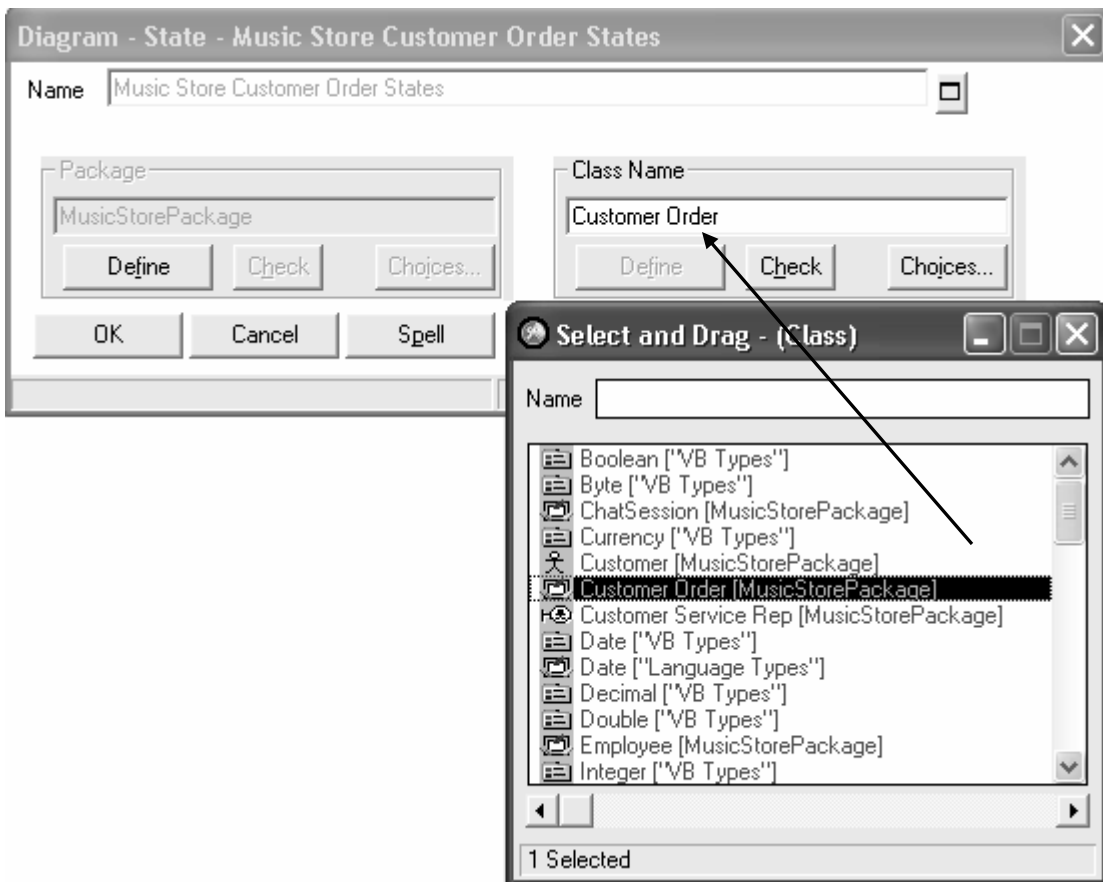


Figure 90 Specifying the target class name for the State diagram

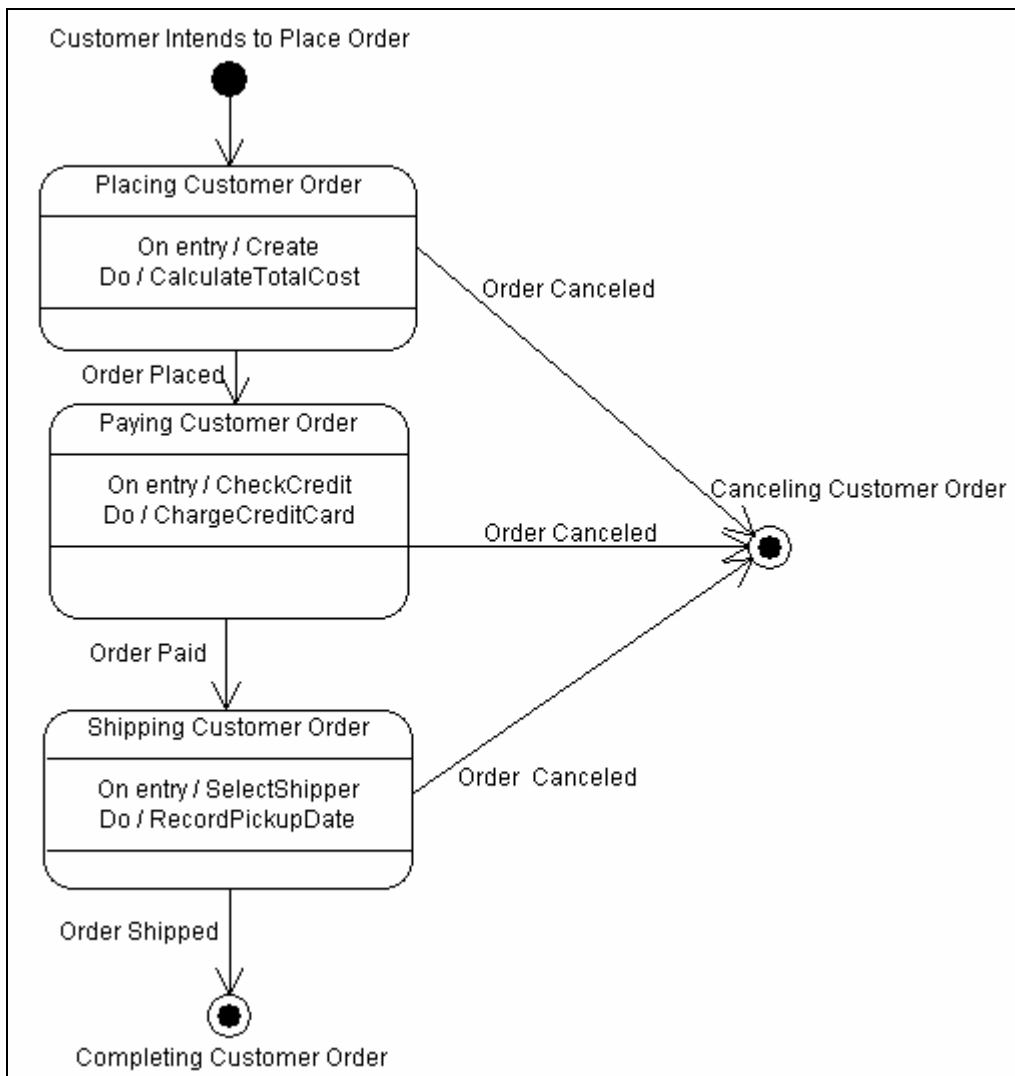




Figure 91 Example State diagram for Customer registration

The example State diagram shown in Figure 91 represents the states the Customer Order object encounters as an order is placed, paid, shipped and/or cancelled. To begin drawing this State diagram choose the Initial State tool  which is represented as a large solid circle (not to be confused with the “Junction” tool which is a slightly smaller solid circle). Click on the diagram in the upper right corner and label this initial state “Customer intends to place order”.

Next, select the State tool  and add the following three states:

- Placing Customer Order
- Paying Customer Order
- Shipping Customer Order

Two final states will be added next. Select the Final State tool  and create a final state labeled “Completing Customer Order”. With the Final State tools still selected create an additional final state labeled “Canceling Customer Order”. At this point the diagram should appear as shown in Figure 92. As with other diagram types, if an item on the diagram is mislabeled, use the Select

tool to right-click on an item and choose `Edit` from the pop-up menu. The name of the item can be changed in the dialog box that will be displayed.

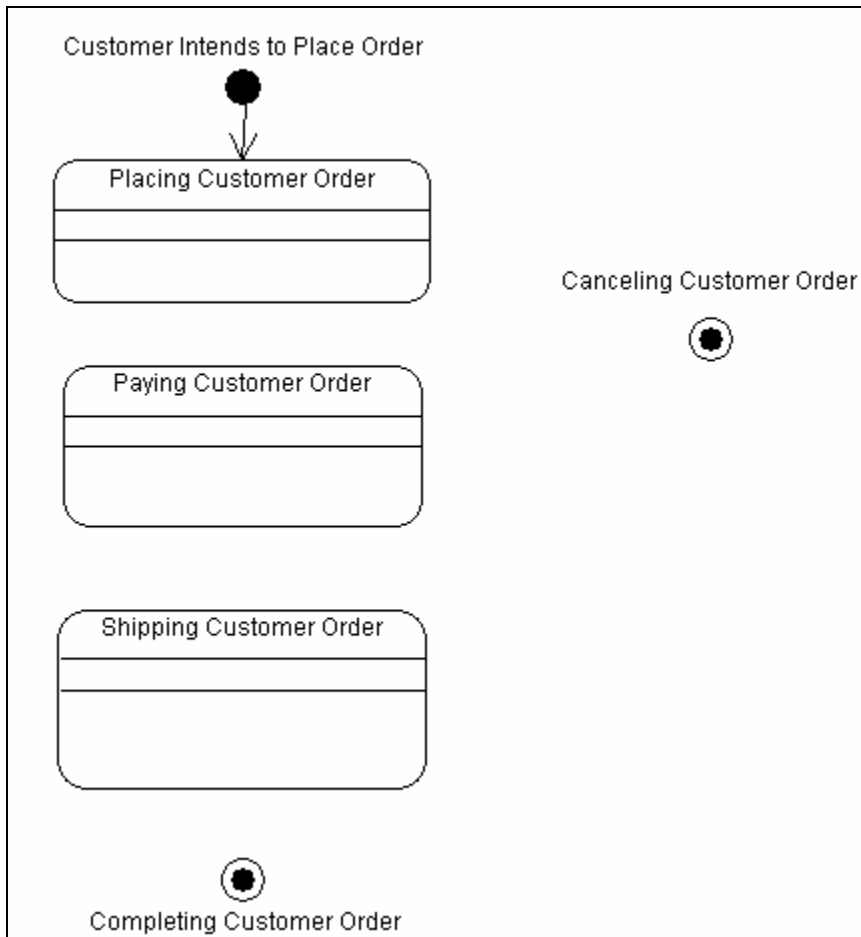



Figure 92 State diagram after adding initial, final and regular states

With the states in place, transitions can now be added to the diagram. Select the Transition tool (single arrow)  from the button bar. The first transition will be drawn from the Initial State to the "Placing Customer Order" state. Begin by clicking on the initial action state. A bold plus sign **+** will appear on the initial action state. Position the mouse pointer over the "Placing Customer Order" state and a second plus sign will appear. Click on the state and the transition will be created as shown in Figure 93. With the Transition tool still selected add four additional transitions as shown in Figure 91. The transition labels are as follows:

- Order Placed
- Order Paid
- Order Shipped

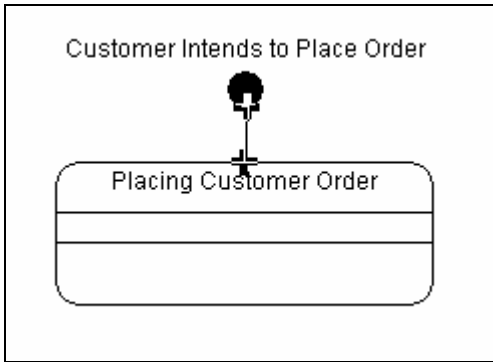


Figure 93 Drawing a transition between two states

A Customer Order can be cancelled during any of the three main states. Therefore, it is necessary to create Transitions from each of the three main states to the “Canceling Customer Order” final state. Name each of these transitions “Cancel Order”.

At this point all of the states have been added to the diagram and transitions between those states have been drawn. A final step for the State diagram is to add actions to the states that indicate the kinds of activities that are carried out at each state. Right-click on the “Placing Customer Order” state with the Select tool and choose `Edit` from the pop-up menu. Navigate to the Actions tab and provide a new action named “Create” that will be triggered “On entry” to the state. Provide a second new action named “CalculateTotalCost” that will run as the order is placed (choose “do” under the When column) as shown in Figure 94. Click the OK button when finished.

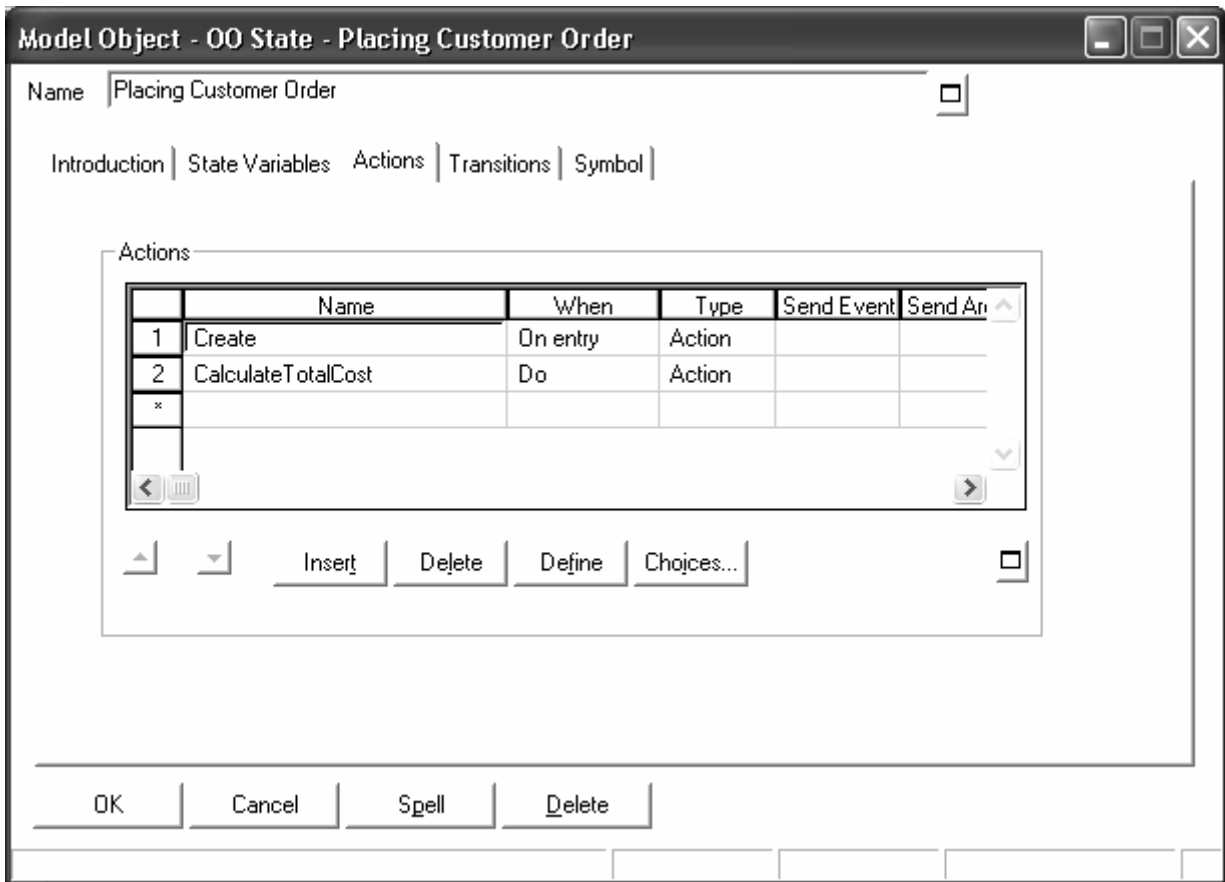


Figure 94 Specifying an Action for a state

In a similar fashion, edit the additional states as per Figure 91. The Actions to be specified include:

State	Action	When
Paying Customer Order	CheckCredit	On entry
Paying Customer Order	ChargeCreditCard	Do
Shipping Customer Order	SelectShipper	On Entry
Shipping Customer Order	RecordPickupDate	Do

To save the work completed thus far, press CTRL + S or pull down the File menu and choose the Save Diagram menu item.

Exercise: States of the Product object

For this exercise, the states of the Products object will be documented. Create a new State diagram named “Music Store Product States” based on the “Products” class and complete the following steps:

- Draw an Initial State named “New Album Released”
- From this initial state, the product may be queried by Customers or new products may be delivered requiring an inventory update.

- After responding to a Customer query, a Customer purchase may lead to depleting inventory or more product could be delivered leading to another inventory update.

15. Object Oriented modeling with Component diagrams

A component diagram is used to model the structure of the software, including dependencies among software components, binary code components, and executable components. Such a diagram can be used to show how programming code is divided into modules and depict the dependencies between those components.

To begin creating a new Component diagram, choose the **UML** (Unified Modeling Language) tab in the Browser window. Pull down the **File** menu and select the **New Diagram** menu item. Choose **Component** by double clicking on its entry as shown in Figure 95. Name the new Component diagram “Music Store Components”. When prompted for the Package in which to create this diagram, choose the “MusicStorePackage” as done in previous sections.

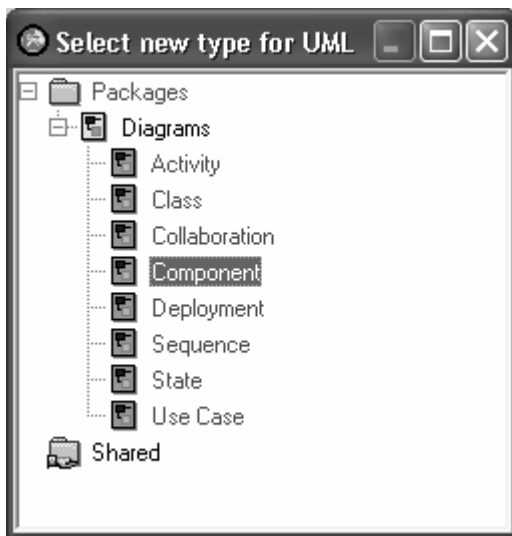





Figure 95 Creating a new Component diagram


The Component diagram to be drawn in this example is shown in Figure 96. In this example, stereotypes are displayed for the User Interface (UI) components which are shown as Java Server Pages (JSP) and the middle-tier logic which are shown as Java servlets.


To begin drawing this Component diagram, select the Package tool  from the tool bar and draw a large package named MusicStorePackage on the diagram. Use the JSP stereotype component tool  (click on the down arrow next to the generic component tool icon) to draw three components named:

- Web Site Home page UI
- Customer Registration UI
- Product Search UI

With the Servlet stereotype component tool  draw three middle-tier servlet components named:

- Web site generation logic
- Registration business logic
- Product search business logic

With the generic component tool  draw a final component named “Database Access Component”. Arrange the components within the MusicStorePackage as shown in Figure 96.

Finally, use the dependency tool  to draw dependencies between each UI component and its middle-tier business logic components and from each middle-tier component to the Database Access Component as shown in Figure 96.

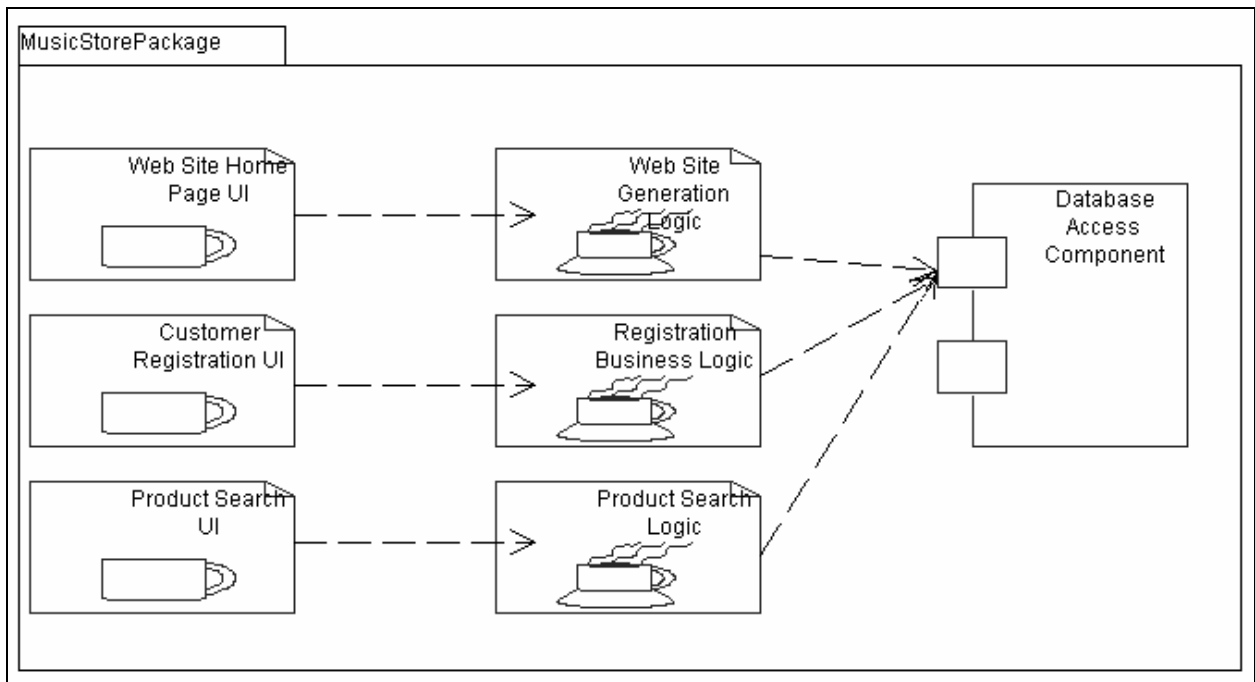


Figure 96 Example Component diagram

Exercise: Extending the Component diagram

For this exercise, extend the component diagram by adding a user interface (UI) component named “Customer Order Entry” with associated middle-tier business logic that in turn is dependent upon the Database Access Component.

16. Object Oriented modeling with Deployment diagrams

Deployment diagrams are implementation-oriented diagrams that describe the physical architecture of the hardware and software in the system. In the Component diagram example from Section 15, a three-tier system was modeled with a front end user interface (UI), a middle tier Servlet engine with the business logic and a back-end database server accessed through a common database component. An example deployment diagram is shown in Figure 97.

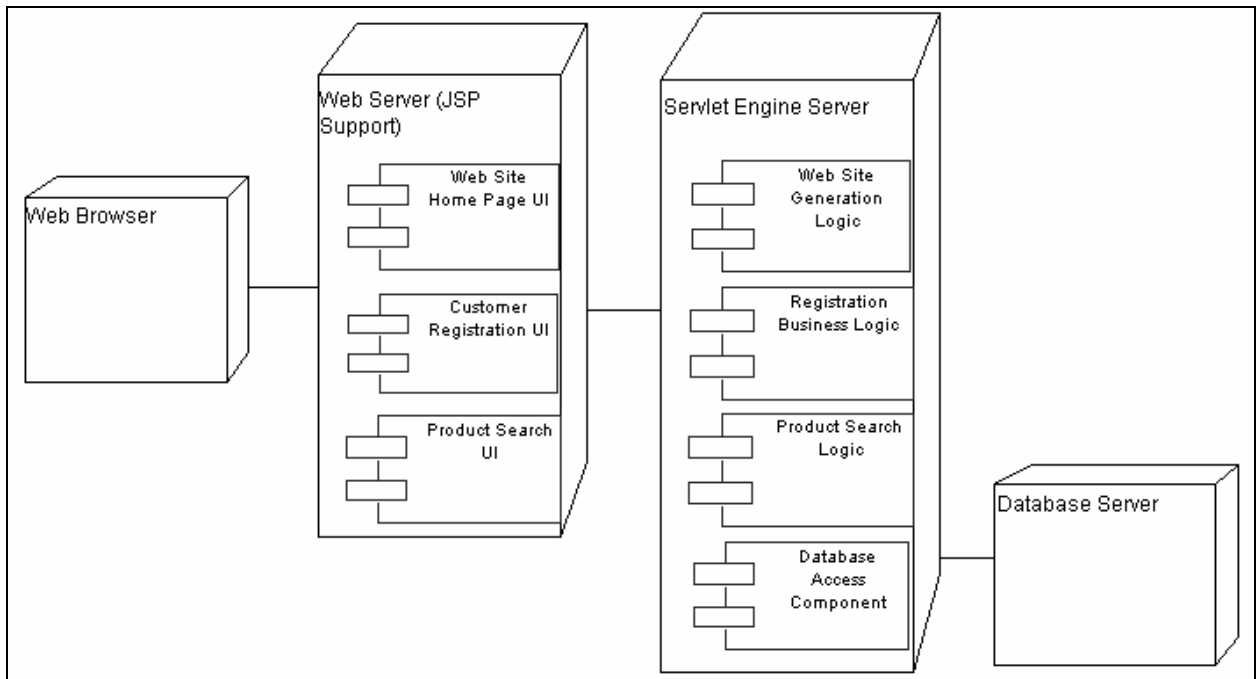


Figure 97 Example Deployment diagram

To create a new Deployment diagram, choose the **UML** (Unified Modeling Language) tab in the Browser window. Pull down the **File** menu and select the **New Diagram** menu item. Choose **Deployment** by double clicking on its entry as shown in Figure 98. Name the new Component diagram “Music Store Deployment”. When prompted for the Package in which to create this diagram, choose the “MusicStorePackage” as done in previous sections.

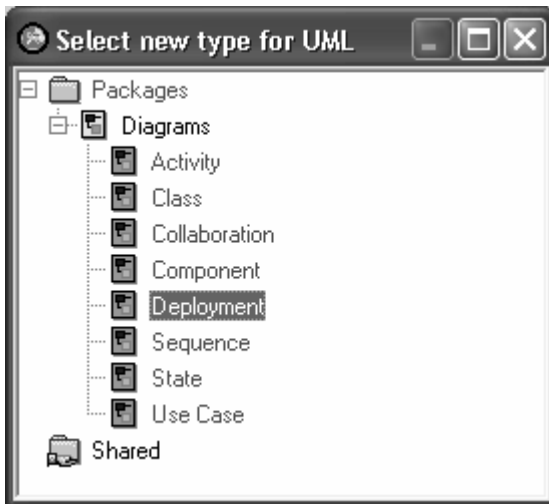



Figure 98 Creating a new Deployment diagram

To begin drawing this Deployment diagram, select the Node tool  from the tool bar and draw four nodes named:

- Web Browser
- Web Server (JSP Support)
- Servlet Engine Server
- Database server

Expand the sizes of the Web Server and the Servlet Engine Server. The components created previously in Section 15 can be easily added to the Deployment diagram by dragging them from the Browse window as shown in Figure 99. To accomplish this, open the Shared and Definitions tree and then navigate to the Component branch. Upon opening this branch up, all of the components created in Section 15 will be displayed. Drag each of the components to its appropriate place as shown in the example Figure 97.

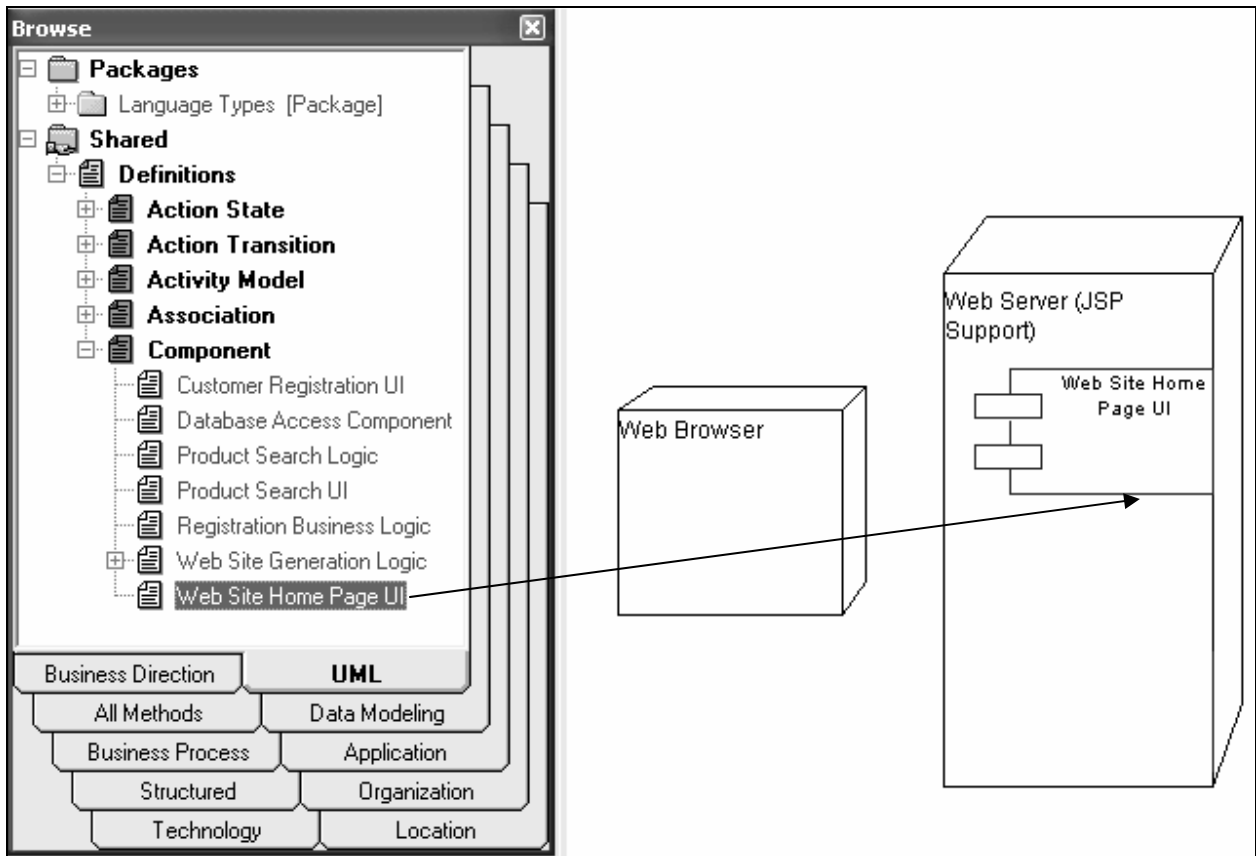



Figure 99 Placing existing components on the Deployment diagram

The final step for finishing up the Deployment diagram is to create connections between each of the nodes. Select the Connection tool  from the tool bar. Click on the Web Browser node (A bold plus sign **+** will appear) and then click on the Web Server node. Follow the same steps to connect the Web Server node to the Servlet Engine node. Finally, follow the same steps to connect the Servlet engine node to the Database Server node.

Exercise: Extending the Deployment Diagram

For this exercise, extend the existing Deployment diagram by creating a new node named “Inventory Management System” and connecting this node to the Database Server.

Part IV: Reporting, Encyclopedia Management and Additional Features

17. Generating Reports

System Architect has a number of different report generation tools. Some of these tools such as the “Rules Check” were demonstrated earlier in this tutorial (see Section 7 on Page 26). In this

section, the more comprehensive HTML Reports generator will be demonstrated. For this example, the Project containing the UML diagrams created in Part III of this tutorial will be used as the basis for the reports.

To begin generating an HTML report, pull down the Reports menu and choose the HTML Reports menu item. A dialog box will appear as in shown in Figure 100. While there are dozens of options that appear on the five different tabs, the main options that require attention are:

Tab	Field	Description	Example value(s)
General	Publish Home Page	The directory and start-up HTML page for the report.	c:\rich\tutorial\myReport.htm
General	Options:	The style of the report.	Encyclopedia Report
Diagrams	Type	Limit the display of diagrams to specific types	(any)
List Management	Options:	Publish All Diagrams	Checked

Once the options have been selected, click on the Publish button to create the comprehensive HTML reports. If this is the first time reports have been generated in the specified directory (General tab, Publish Home Page field), a warning will be displayed as shown in Figure 101.

Once the report generation has completed, use a web browser to open the HTML file given in the General tab, Publish Home Page field. The results of one trial report run are displayed in Figure 102.

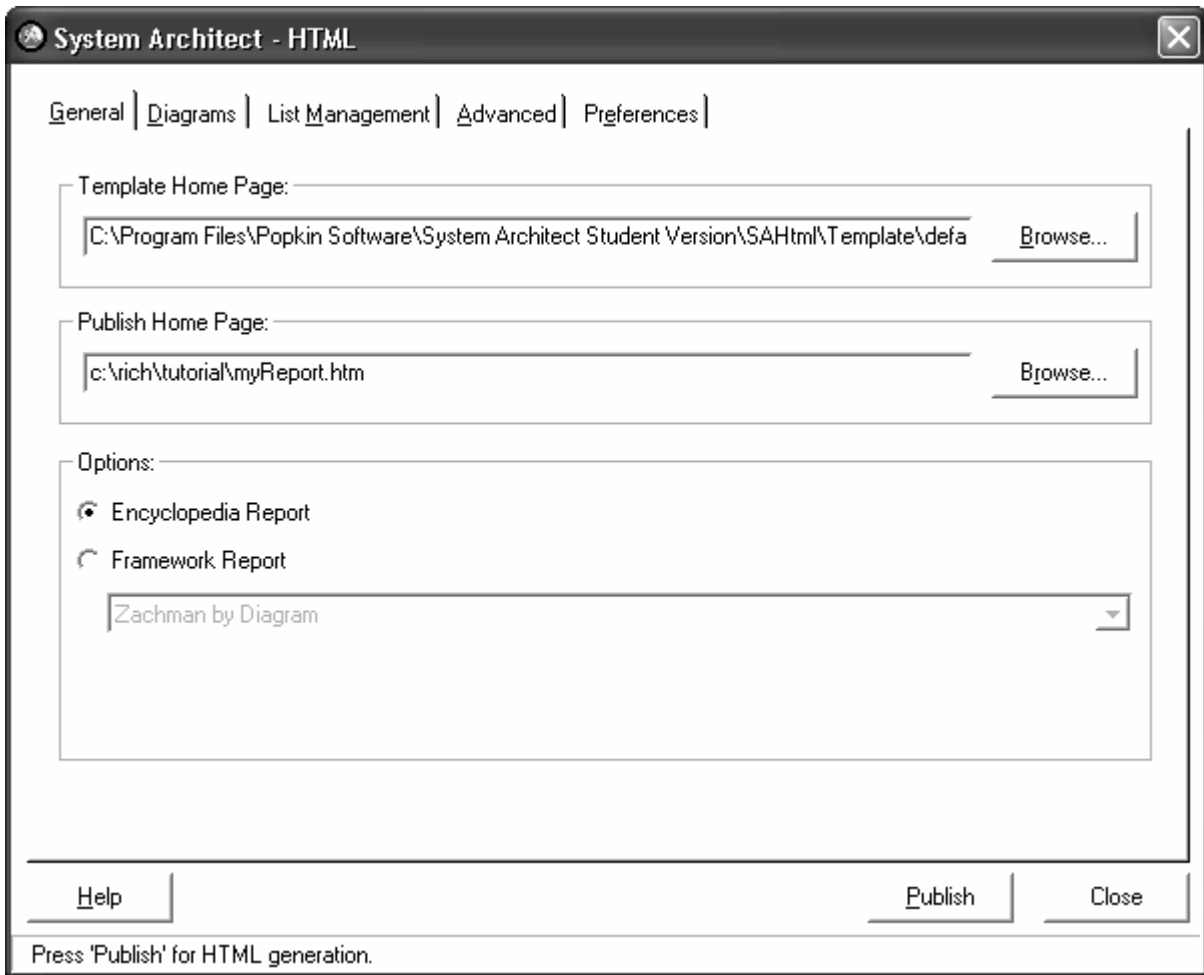


Figure 100 HTML Reports Options

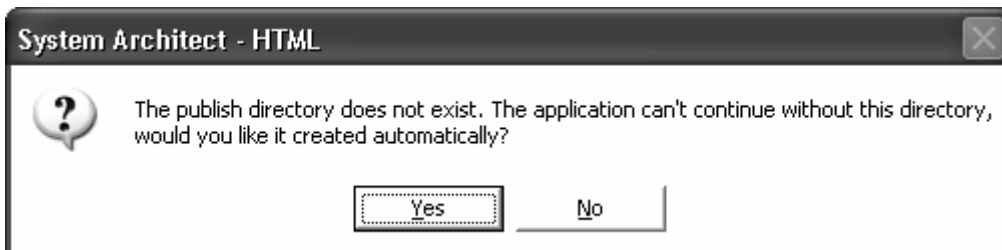


Figure 101 Confirming creating a new directory for the report output

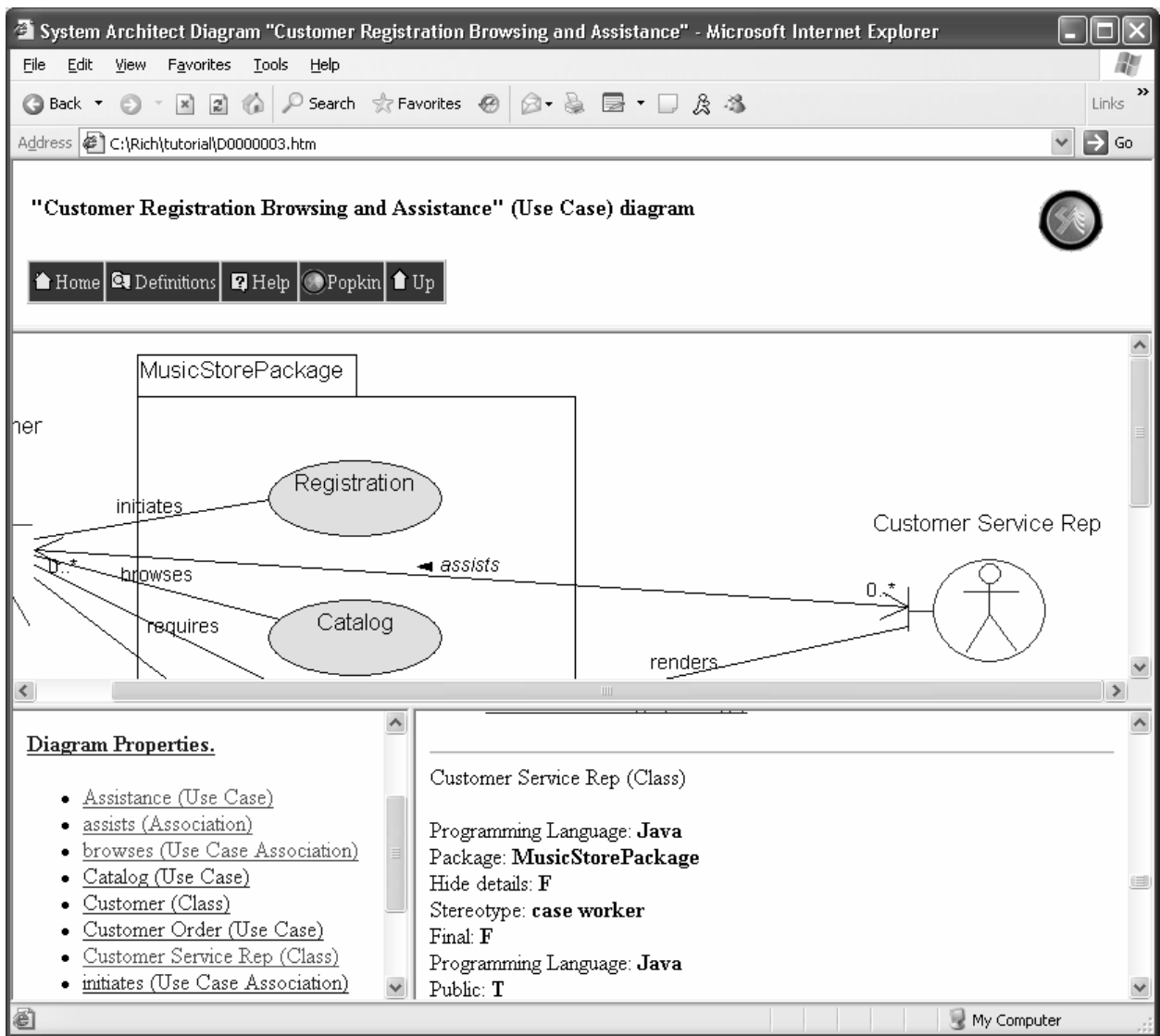


Figure 102 Example HTML Report

18. Project and Diagram Management

In this section, some techniques for managing SA projects will be introduced.

18.1 SA Encyclopedias

The Student Edition of SA provides four encyclopedias named Proecjt1, Project2, Project3 and Project4. It is possible to “install” additional projects such as those passed along by a team member or provided from the class instructor. In the following exercise, assume a .ZIP file containing all of the encyclopedia files for a project have been made available to the student and have been downloaded to the local user’s hard disk.

First, before attempting any of the following steps, make certain that SA is NOT running. Make a backup of existing projects by using the Windows Explorer to navigate to the C:\Program Files\Popkin Software\System Architect Student Edition\Encyclopedias\ directory (this is the default directory used during the installation) as shown in Figure 103. **Error! Reference source not found.** Copy the Project1 and Project2 directories to another disk or another directory on the same hard disk for safekeeping.

Use an unzip utility (such as WinZIP) to unzip the new project and its files. Again using Windows Explorer copy the set of files into the appropriate project directory such as C:\Program Files\Popkin Software\System Architect Student Edition\Encyclopedias\Project2

The next time SA runs, it will find these new project encyclopedia files in this directory and will automatically make use of them.

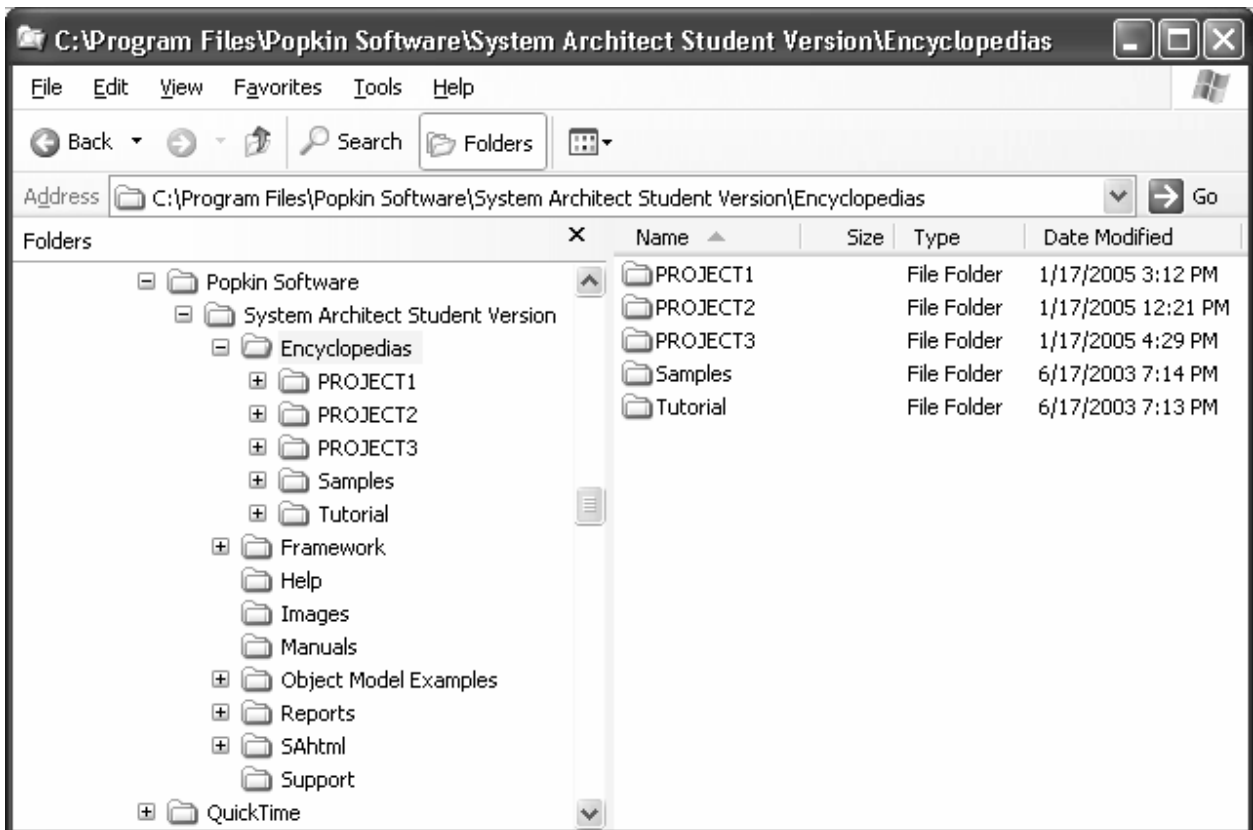


Figure 103 Popkin SA Encyclopedia directories in Windows Explorer

18.2 Managing Diagrams within a Project

While working on a project, it is often desirable to save several different versions of the same diagram. For example, before embarking on a series of major changes to a diagram, one might

wish to save the current version for safekeeping or reference as the subsequent changes are made. This can be accomplished by using the Save Diagram As menu item on the File menu.

For example, suppose the designer wished to continue expanding the “Music Store” ER diagram created earlier in this tutorial while retaining the original. With this diagram open, pull down the File menu and choose the Save Diagram As menu item. Provide a new name for this diagram (such as “Music Store 2”) and select the appropriate type of model as shown in Figure 104.

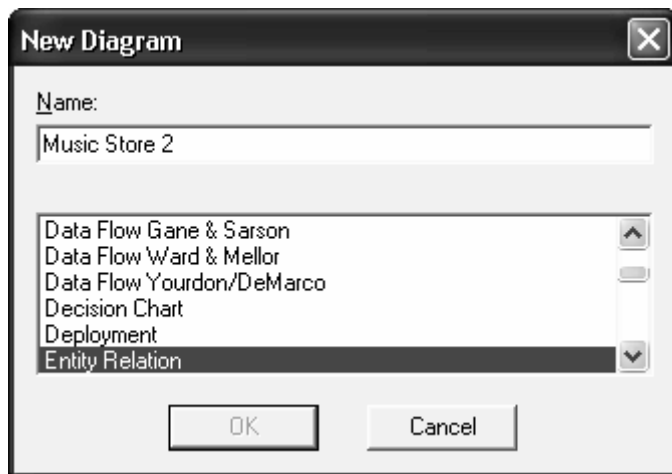


Figure 104 Saving a diagram under a new name

Once the name and type are chosen, click on the OK button. The Project Data Model dialog box *may* appear next. In this example, this new diagram will be saved in the same data model that was first created at the start of this tutorial (“Project Data Model 1”). Click on the OK button and the newly named diagram will appear in the right hand window.

19. Additional Features of System Architect

The SA product has a vast number of features. In this section a collection of tips for using some of the more helpful features are briefly introduced.

19.1 ER Model Display Mode Options

In Section 3 an ER model was created that included the display of attributes with primary key descriptions. It is also possible to change how ER diagrams are displayed. Make certain the Project with the structured models is selected (Project1 was used earlier in the tutorial) by pulling down the File menu and selecting the Project from the list. Open the ER Diagram (Go to the **All Methods** tab, Diagrams, Entity Relation and then double click on the “Music Store” diagram). Highlight/select one of the entities on the diagram, pull down the View menu and choose the Display Mode menu item. The resulting dialog box appears in Figure 105. The display mode choices are as follows:

- Entity Relation – Only entities with titles and relationships are shown

- Key Based – Entities with titles and their identifiers (keys) with relationships are shown
- Fully Attributed – Entities with all attributes including keys and relationships are shown
- Physical – Same as Fully Attributed with the addition that data types for each attribute are shown

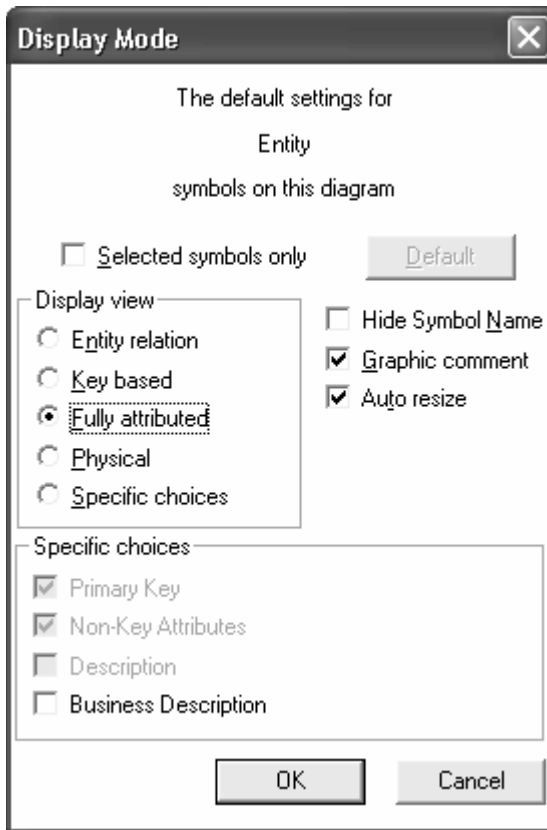


Figure 105 Display Mode Dialog Box for ER Diagrams

19.2 Including Foreign Keys in ER Diagrams

Foreign keys (FK) are generally not shown in traditional ER modeling. However, prior to creating a physical data model based on the conceptual ER model, the designer may wish to view how foreign keys will be propagated through the model. To perform this change in SA, open an ER Diagram and make certain the Fully Attributed display mode is set. Then pull down the Dictionary menu and select the Update FKs menu item. A dialog box, demonstrated in Figure 106, will appear showing a log of the changes made to the diagram. After dismissing this dialog box, the ER Diagram will redraw with the foreign keys in place. An example is shown in Figure 107. note that it may be necessary to resize some entities so that all of the attributes are displayed.

To remove the foreign keys, pull down the Dictionary menu and select the Remove FKs menu item.

```

Foreign Key Propagation
Propagating PK Order_Number into Order Items as FK; added
Propagating PK Customer_ID into Customer Order as FK; added
Propagating PK Product_ID into Order Items as FK; added
2 entities changed

```

Figure 106 Status report after propagating foreign keys

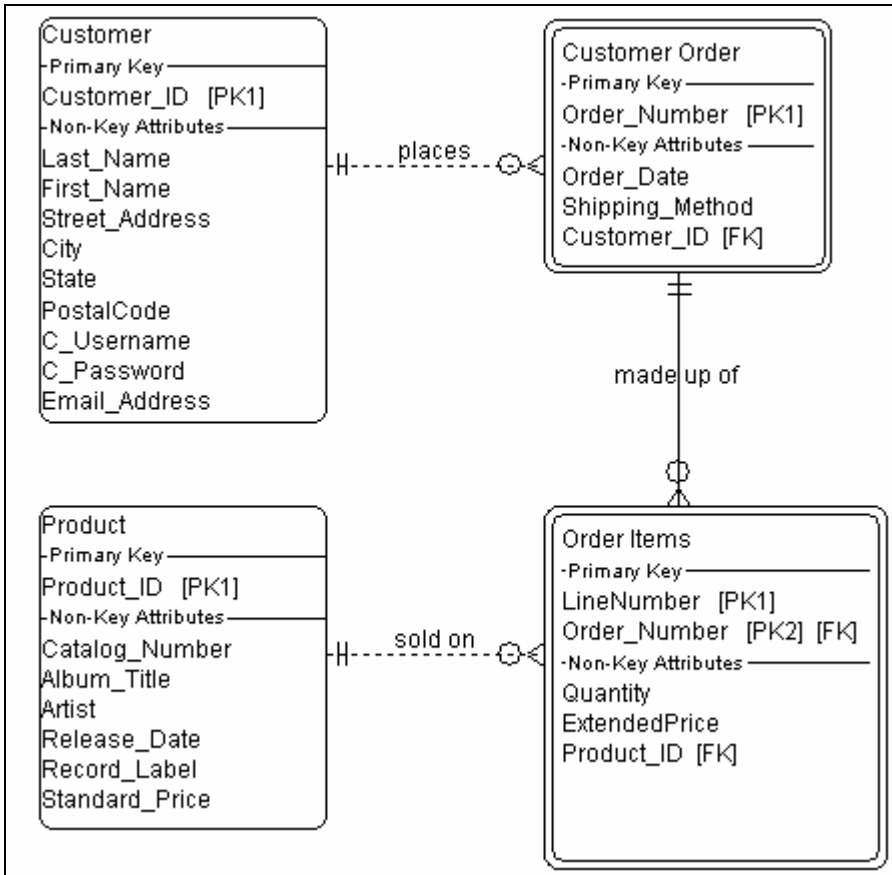


Figure 107 ER Model after foreign keys have been propagated

19.3 Changing the Symbol Style for ER Diagrams

The default symbol style for the ER diagrams is called the “Crow’s feet” notation. In this notation a single relationship line indicates a maximum relationship cardinality of “1” while the crows foot (three small lines at the end of the relationship line) indicates a maximum relationship cardinality of “Many”. There are other notations that explicitly give the minimal and maximal cardinalities in terms of number and in terms of other types of symbols. To change the notation used for the ER Diagram, open the diagram, pull down the **Format** menu, choose **Diagram** format and finally **Notation** from the fly-out menu. The Notation options dialog box is

shown in Figure 108. To see the differences in notation, try selecting C-Y (short for Coad-Yourdon) or Unified for example.

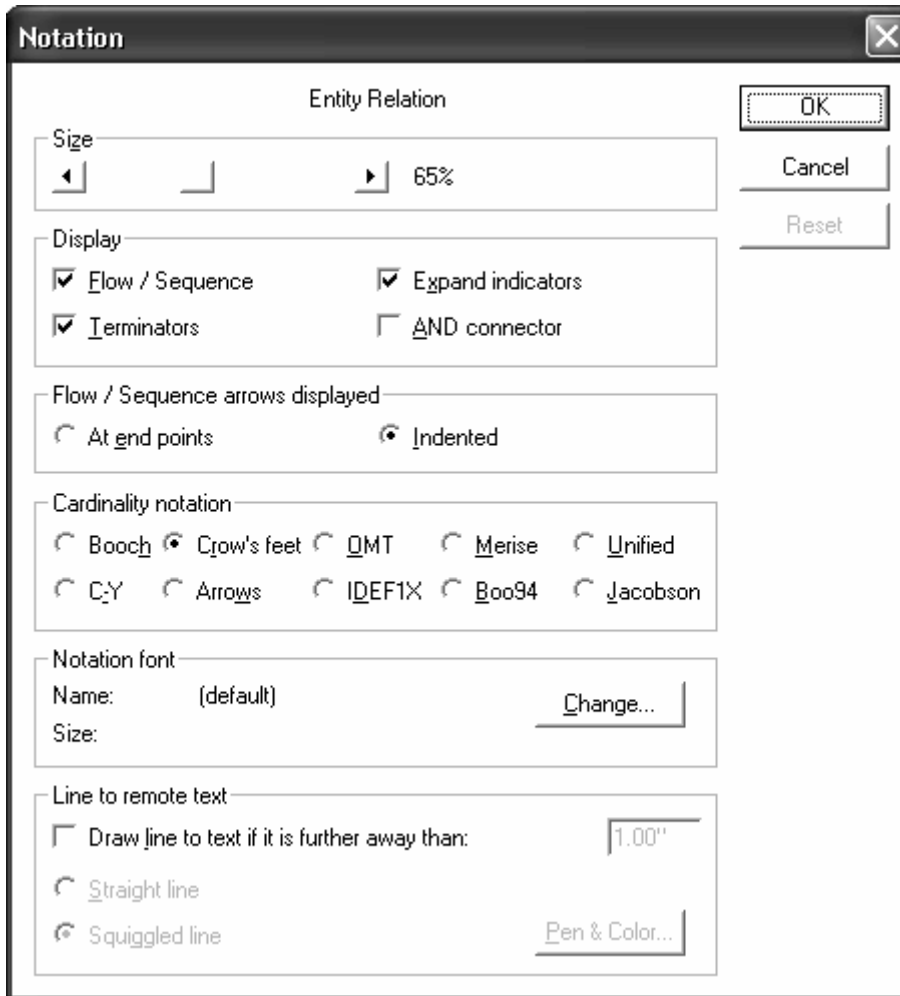



Figure 108 ER Diagram Notation Options dialog box

19.4 Adding Documentation to a Diagram

It is important to include documentation on each diagram so that it can be easily identified. Documentation can be easily and automatically added to any diagram using the “Doc Block” drawing tool. Open any diagram and select the Doc Block icon  or, pull down the Drawing menu and choose the Doc Block menu item. The cursor should change to a pen. Find some open space on the diagram and click once. The documentation block is automatically added. Once created, the documentation block can be re-sized and re-positioned using the selection tool. Additional comments can be added by right-clicking on the block and choosing the Edit menu item from the pop-up menu. An example is shown in Figure 109.

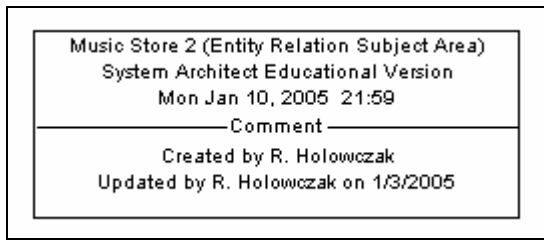


Figure 109 Example Documentation Block with additional comments

19.5 Copying Diagrams to a Word Processor

Producing documentation for a project can be streamlined by copying and pasting diagrams from SA into MS Word or into another Windows based word processing application. For example, to copy an ER Diagram, open the diagram, pull down the Edit menu and choose the Select All menu item. This will highlight all of the entities and relationships in the diagram. Pull down the Edit menu again and select the Copy menu item. This will place a copy of the diagram on the Windows clipboard in Enhanced Metafile format. At this point, switch over to MS Word or other word processor, pull down the Edit menu and select the Paste menu item.

19.6 Changing the colors and fonts of diagrams

It is possible to change the background color, line color and fonts (type style and size) for virtually any object in any of the diagrams SA provides. One reason to change some of the colors is that they may not print clearly on black and white printers or may not photocopy well. To change the colors, open any diagram and select an object (e.g., an Entity or a Function). Pull down the Format menu, highlight the Symbol Format item and then choose Color from the fly-out menu. Choose the colors for the outline Pen, background Fill and font colors by clicking on the appropriate Colors button. Click the OK button when finished. In a similar fashion, the font can be changed by pulling down the Format menu, highlighting the Symbol Format item and then by choosing Font from the fly-out menu.

To change the colors, fonts or other display attributes for a group of objects on a diagram, select multiple objects first (hold down the SHIFT key while clicking on additional objects) and choose the appropriate menu item as described above.

20. Conclusions for this Tutorial

In this tutorial, the basic steps were outlined for creating and manipulating a number of different models used for systems analysis and design. The SA product has an extensive number of additional model types as well as reporting tools. Readers are directed to Popkin's web site for additional documentation and tutorials.