Practical Session 2: The Use Case and Requirements Model

The following report uses a retail banking scenario to illustrate how Enterprise Architect can be used to develop Use Cases and a Requirements Model. The report also gives you hands on experience with Use Case Diagrams, Business Rules, Features and the Relationship Matrix.

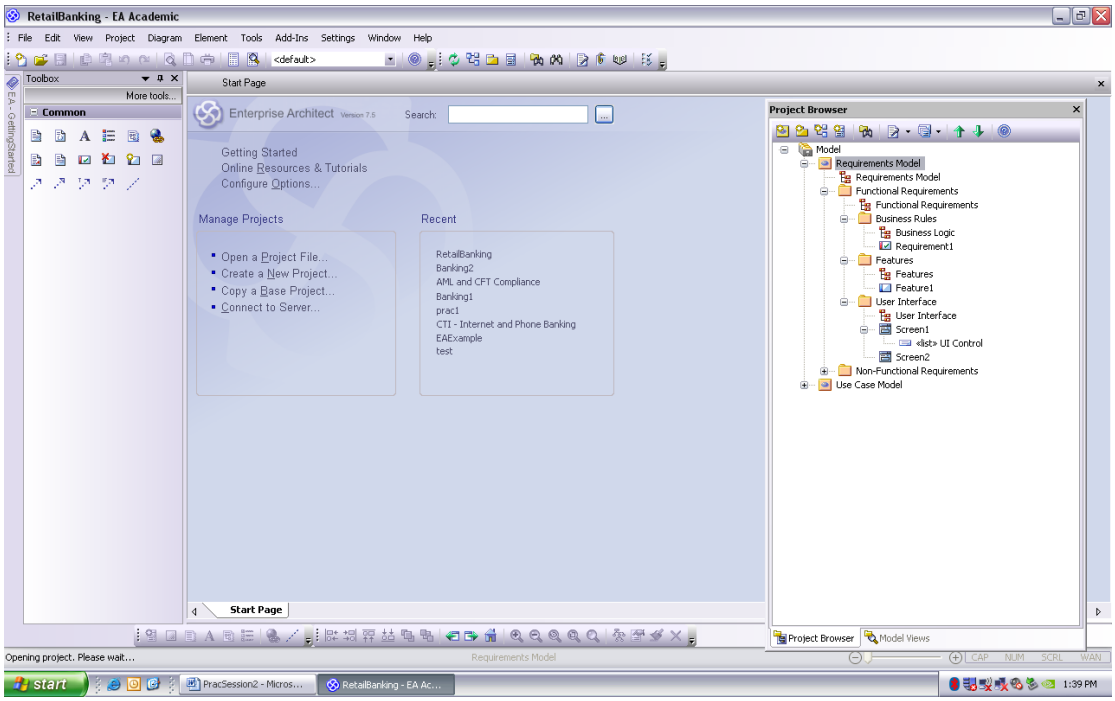
Using Enterprise Architect (EA): Session 2

The Use Case and Requirements Model

You need to create a new project for this and the subsequent sessions. Call it **RetailBanking.**

Select a *Use Model* and a *Requirements Model*.

Ensure that the RetailBanking and *project browser* are both open as shown below.



For the purposes of these exercises delete the default templates for the user interface package (right click on the user interface folder in the project browser and select delete from the menu).

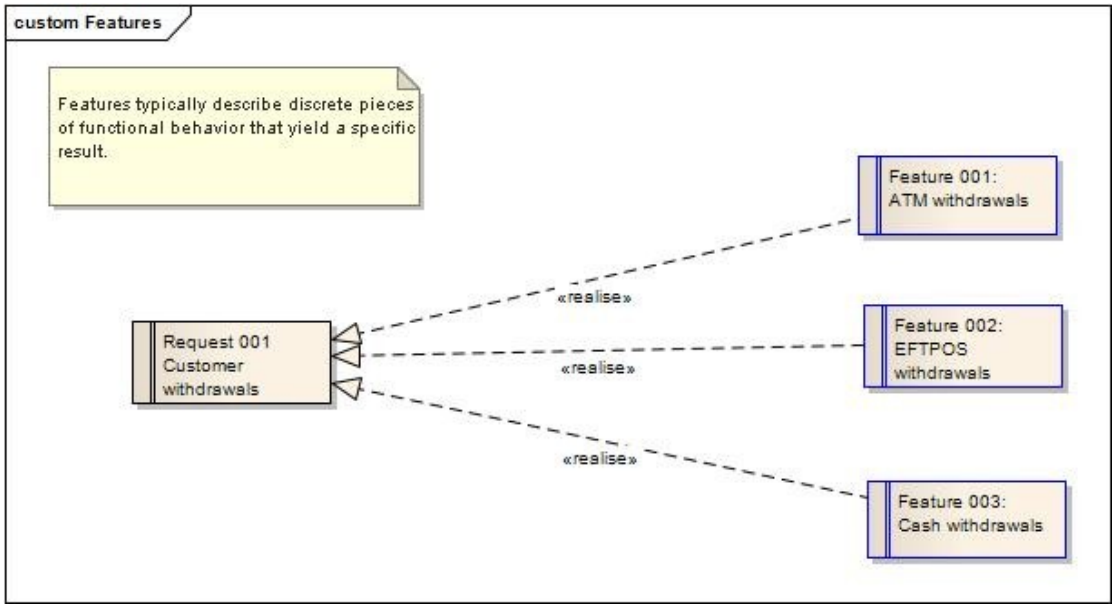
We will now add *three features* and a *business rule* for the retail banking example from the theory sessions. Ensure you have the *requirements toolbox* selected.

Feature 001: The system will provide ATM withdrawals
Feature 002: The system will provide EFTPOS withdrawals
Feature 003: The system will provide cash withdrawals from a branch

Business Rule 001: The maximum daily amount that can be withdrawn from a savings account is $900.

In the *features custom diagram* add the following stakeholder(customer) request as a requirement.

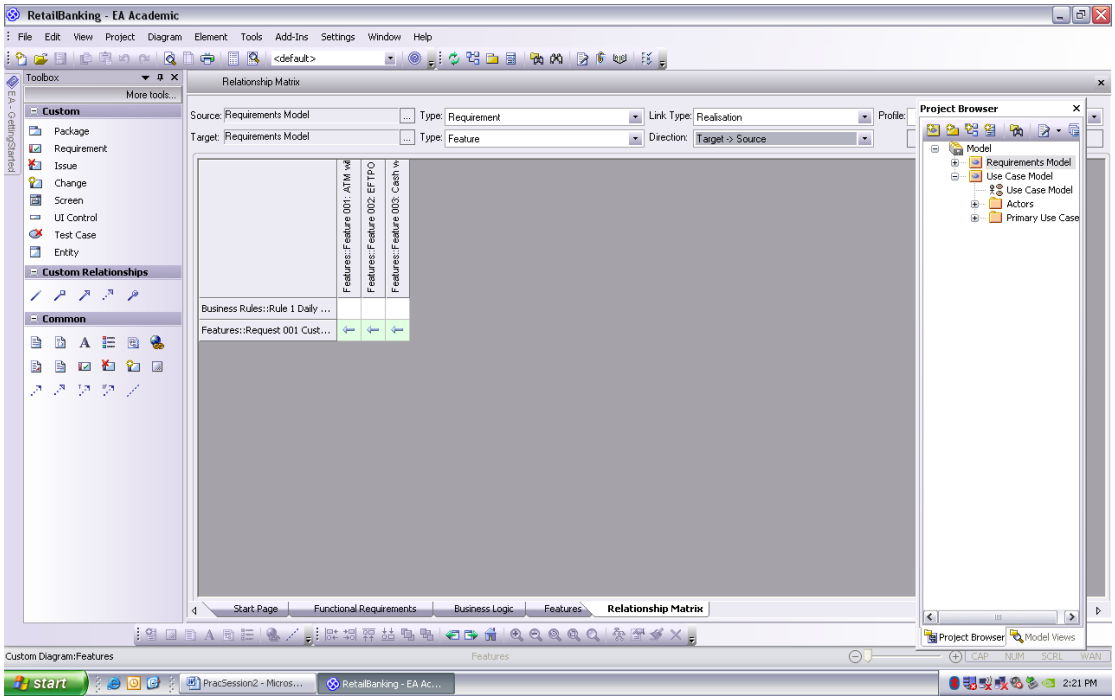
Request 001: A customer is able to withdraw money from their accounts
Now establish a *realisation association* between each feature and the request. Access the properties of each of the three associations and in the stereotype field enter *realise*. The stereotype should appear in the diagram as shown below.



These realisation associations could have been set through a *relationship matrix*, rather than through the diagram. However as they have been set through a diagram, they will appear in the relationship matrix.

Viewing the matrix

From **View** Menu, select **Relationship Matrix.** You will need to ensure that you select the correct models, types, link types and direction to see the following.



Now to the Use Case Model

We will use the default *actors* and *primary use case* packages. However delete the sample actor and primary use provided. These will be replaced with your own. You may delete the hyperlinks and notes on the Use Case Model diagram, leaving just the empty folders.

To add an actor to the actor folder …

In the project browser:

Highlight the folder, right mouse click and select **add.**

Select **element**, then **actor** and **create.**

Close the properties window for the *customer actor*, and it now will be visible in the actor package in both the browser and the diagram.

Create the following actors: *retail customer*, *business customer*, *ATMsystem*, *shop assistant*, *banking system*, *credit card system*.

To add a use case to the primary use case folder.

In the project browser:

Highlight the folder, right mouse click and select **add.**

Select **element,** then **use case** and **create**. Name the use case, *withdrawCashATM.*

Close the properties box (later on you can add some detail for *withdrawCashATM*).

Create the following use cases: *validatePIN*, *withdrawCashEFTPOS*, *withdrawCashBranch*, *validateAmount*, *makePayment*, *directDebitPayment* and *creditCardPayment*.

Note

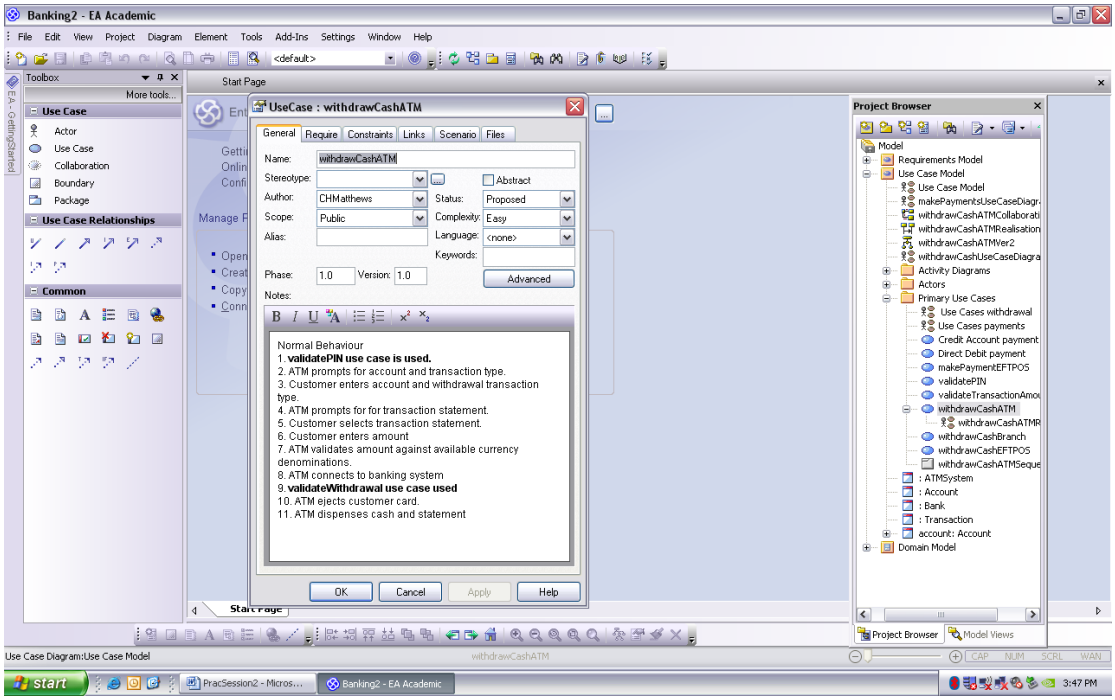
These use cases and actors could have been created by drawing a diagram using the use case tool box. The approach taken in this session is to create the model elements (use cases and actors) first and then later to assemble them into diagrams to view the model as required.

One disadvantage of creating them using a diagram is that the elements will appear at the same hierarchy level as the diagram in the project browser, rather than in a dedicated package.

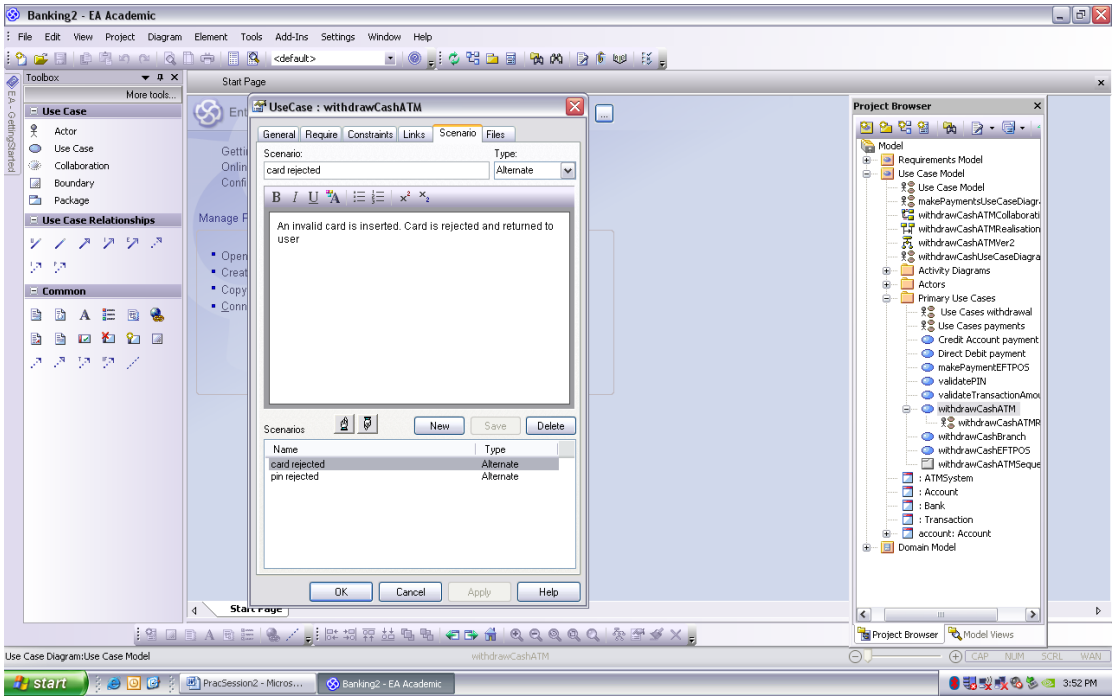
Documenting the Use Cases

Use cases are *narratives* describing normal and alternate behaviour. These narratives can be included with the use case properties. To access the property box, highlight and click on the required use case in the project browser.

The following figure shows the normal behaviour for the use case *withdrawCashATM* being recorded under the *General Tab* in the properties window.

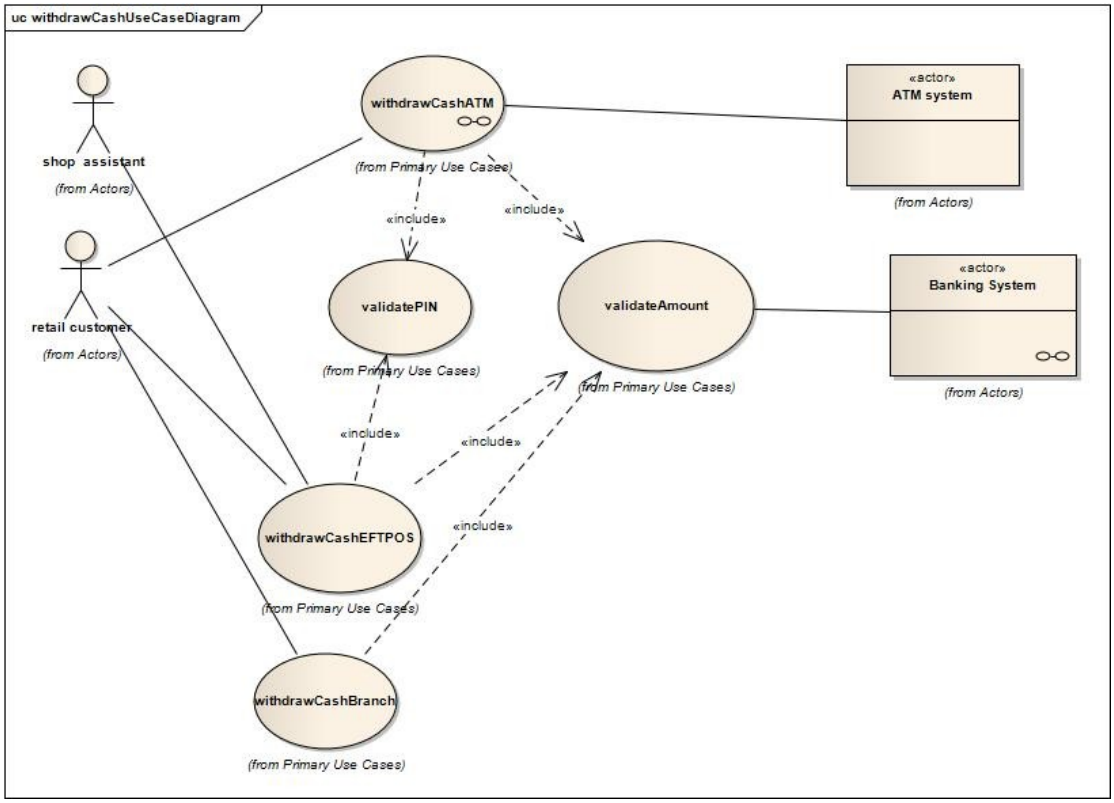


… and the scenarios, documenting alternate behaviour, recorded under the Scenario Tab.


Banking2 - EA Academic
File Edit View Project Diagram Element Tools Add-Ins Settings Window Help
Toolbox
Use Case
Actor
Use Case
Collaboration
Boundary
Package
Use Case Relationships
Common
Start Page
UseCase : withdrawCashATM
General Require Constraints Links Scenario Files
Scenario:
card rejected
Type:
Alternate
An invalid card is inserted. Card is rejected and returned to user
Scenarios
New Save Delete
Name Type
card rejected Alternate
pin rejected Alternate
OK Cancel Apply Help
Project Browser
Model
Requirements Model
Use Case Model
Use Case Model
Activity Diagrams
Actors
Primary Use Cases
Use Cases withdrawal
Use Cases payments
Credit Account payment
Direct Debit payment
makePaymentEFTPOS
validatePIN
withdrawCashATM
withdrawCashBranch
withdrawCashEFTPOS
: ATMSystem
: Account
: Bank
: Transaction
account: Account
Domain Model
Project Browser
Model Views
Use Case Diagram:Use Case Model
withdrawCashATM
start
3:52 PM

Viewing the use case model, using use case diagrams

The following is a use case diagram illustrating the three ways of withdrawing money from an account.



Your task is to draw this using Enterprise Architect

Hints:

You need to add a further use case to the Primary Use Case package.

It is to be drawn as part of the Use Case Model (not the Use Case Model diagram) i.e. highlight *Use Case Model* package, right mouse click, select **add** and then select **diagram**. Select **use case diagram** (as part of the UML behavioural group).

Now right click on the blank drawing canvas and open the properties window for the diagram. Change the default name to *withdrawCashUseCaseDiagram*.

Ensure that the use case tool box is open. However **do not** use it to create actors or

use cases. Instead highlight and drag the required use case elements from the project browser to the drawing canvas.

Enter each as a *simple link*.

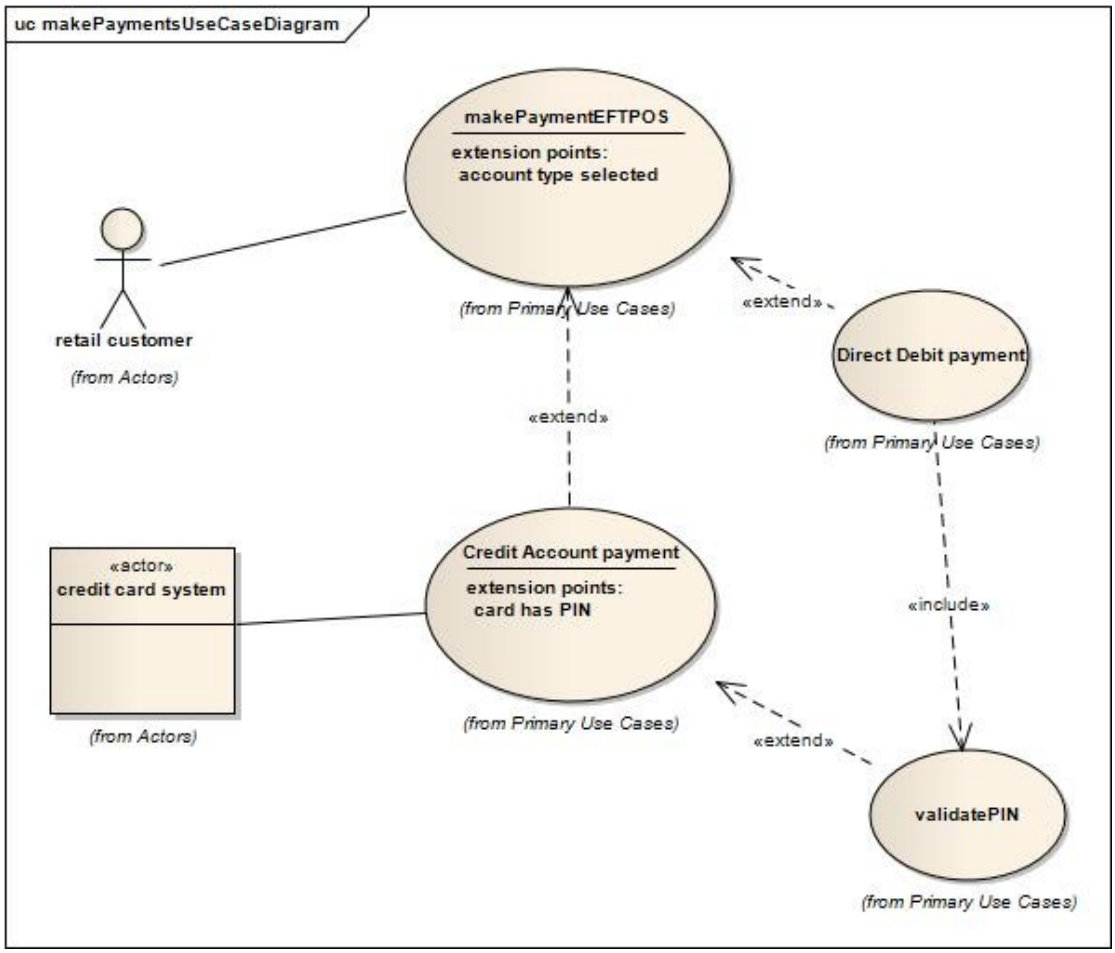
The relationships/associations can be created using the use case toolbox.

At this stage ignore the composite diagram icon (∞). It will be introduced later.

The display for an actor can be changed from the default stick figure to a rectangular form.

Select the icon and right click to activate a menu. Select **Advanced** and then **Use Rectangle Notation.**

Now here is a further use case diagram for viewing part of the use case model.

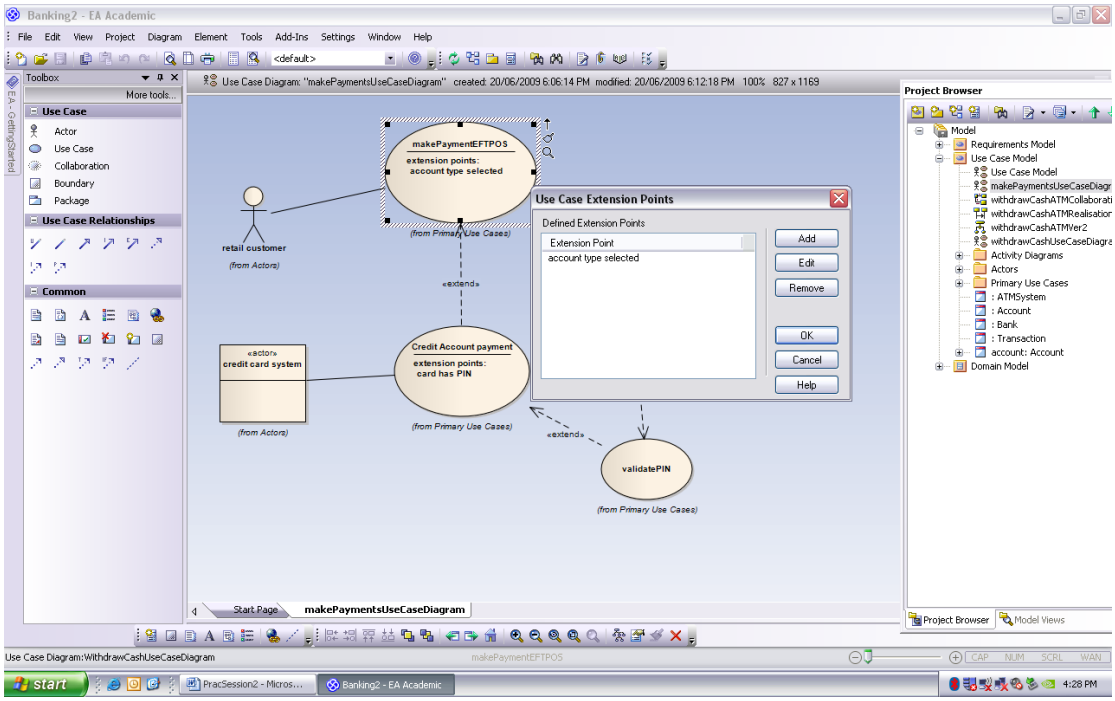


You are to draw this one.

Hints (continued)

It is to appear at the same level as the previous one i.e as part of the Use Case Model.

To enter extension points for a use case, access the properties window for the required use case and right click to activate the menu. From **Advanced** select **Edit Extension Points …** and the following window will open. Multiple extension points can be added if required.



One further hint:

If you wish to delete elements from a project it has to done through the project browser. *Right click* on the element and select **delete** (the last on the list). Deleting elements in a diagram just removes them from the diagram, not the model.

The connectors (in this case associations and relationships) in a diagram can either be removed from the diagram or hidden (still there but not seen). Highlight the connector, *right click* and select **delete**. You will be given the option of removal or hidden.

Creating a realisation relationship matrix between the features and the use cases

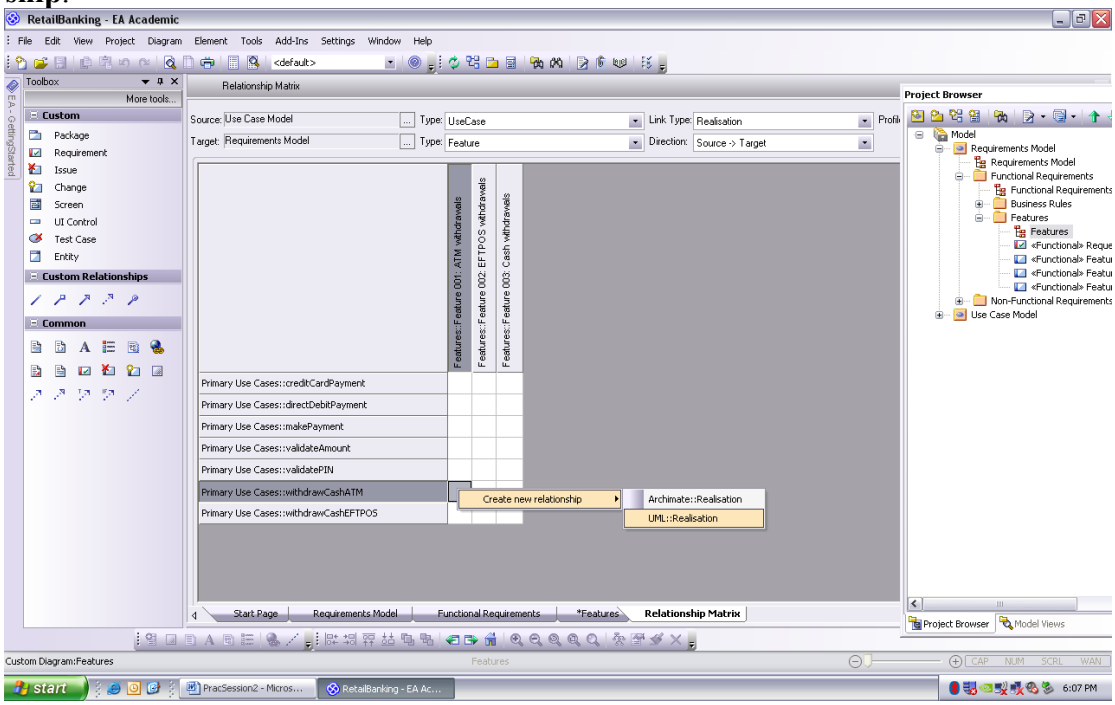
From the **view** menu, select **relationship matrix.**

In this case the *target* is the *requirements model* and the *source* is the *use case model* (because the use cases will realise the feature).

The *type* will be *UseCase* for the *use case model* and *Feature* for the *requirements model.*
Link type is *Realisation* and *direction* is *Source* → *Target* i.e use case to requirements.

To establish the link right click on the appropriate cell in the matrix, select **create new relationship** and then **UML realisation.** See screen snapshot below.

If you wish to delete a relationship, right click on the cell and select **delete relation ship**.



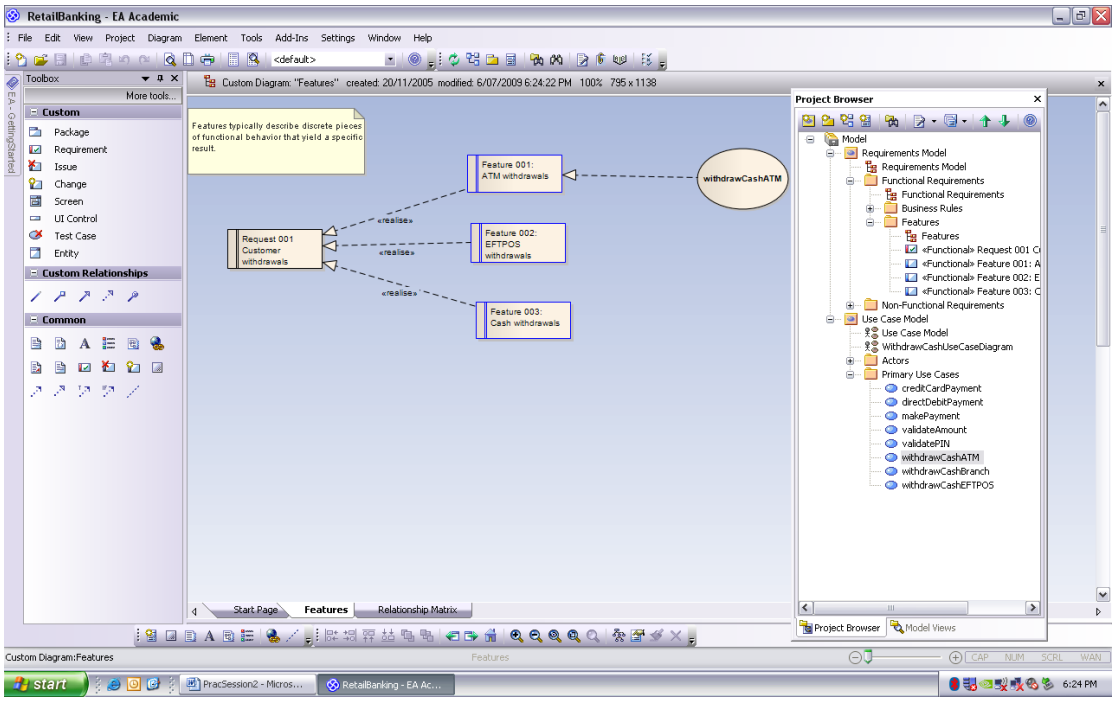
Incorporating this in the features custom diagram

Close the relationship matrix diagram view.

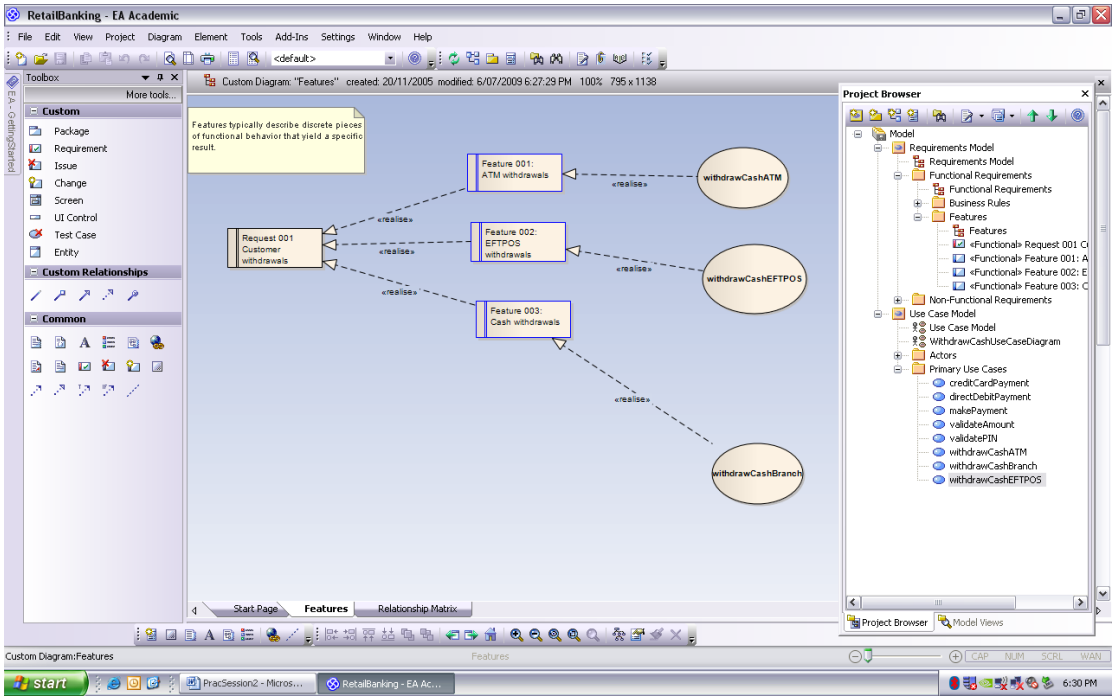
Open the features diagram, showing the three features and the stakeholder request.

Ensure the primary use case package is open in the project browser.

Now highlight and drag the *withdrawCashATM* use case across to the diagram as a *simple link*. Notice that a realisation relationship is established (see below).



Do the same for the other two use cases. If you wish to attach the realisation stereotype select the relationship, right click to access the realisation properties. In the stereotype box enter the required stereotype i.e *realise*.


RetailBanking - EA Academic
File Edit View Project Diagram Element Tools Add-Ins Settings Window Help
Toolbox
More tools...
Custom
Package
Requirement
Issue
Change
Screen
UI Control
Test Case
Entity
Custom Relationships
Common
Custom Diagram: "Features" created: 20/11/2005 modified: 6/07/2009 6:27:29 PM 100% 795 x 1138
Features typically describe discrete pieces of functional behavior that yield a specific result.
Feature 001: ATM withdrawals
Request 001 Customer withdrawals
Feature 002: EFTPOS withdrawals
Feature 003: Cash withdrawals
«realise»
withdrawCashATM
withdrawCashEFTPOS
withdrawCashBranch
Start Page
Features
Relationship Matrix
Project Browser
Model
Requirements Model
Functional Requirements
Business Rules
Features
Non-Functional Requirements
Use Case Model
Actors
Primary Use Cases
creditCardPayment
directDebitPayment
makePayment
validateAmount
validatePIN
withdrawCashATM
withdrawCashBranch
withdrawCashEFTPOS
Model Views
Custom Diagram:Features
start
6:30 PM