**Practical: Requirements Engineering**

# Introduction

The practical exercises makes use of RE-Tools modules of StarUML – a software modelling environment. RE-Tools is an open source requirements modelling toolkit that supports the following notations: NFR Framework, *i*\* Framework, KAOS, Problem Frames, UML and BPMN. For this practical, however we will focus on the UML technique which stands for Unified Modelling Language, and is used for object-oriented modelling.

In the following sections we will demonstrate how UML Use Cases are applied in requirements engineering following real-life application scenarios/examples.

# Tools Download and Installation

* Since the latest version of StarUmlV3 needs to purchase related services to execute RE-Tools. So here, we installed the earlier version.
* Download StarUML via the link: <https://staruml.informer.com/5.0/>
* Mac user can try StarUML first if it cannot load the RE-Tools please use draw.io to draw the use case diagram + soft goals.
* Run the downloaded StarUML setup
* Follow the prompts to complete installation.

(Close the StarUML window to install RE-Tools next)

* Download RE-Tools via the link: <http://sourceforge.net/projects/re-tools/>
* Run the downloaded RE-Tools setup

Since the previous exercises have already installed the .NET Framework on the computer, there is no need to reinstall the relevant framework

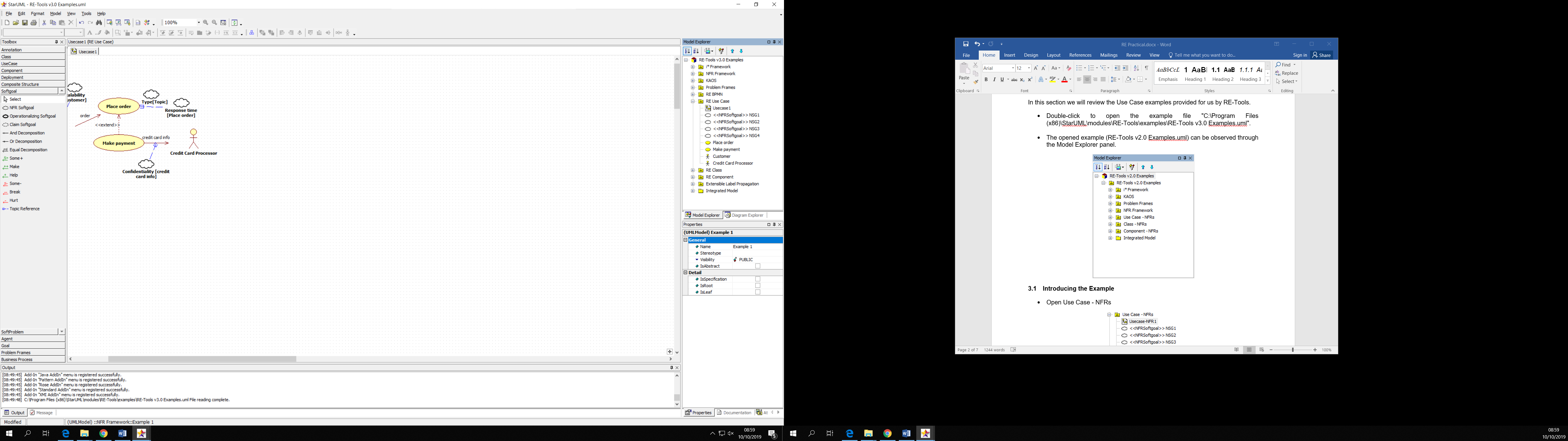
* Rename or remove "Contributors.xml" in "C:\Program Files (x86)\StarUML" folder.

(This file contains the names of the StarUML's contributors that are used as the default names for newly created model elements. However, without renaming or removing this file, these names could cause naming conflicts when RE-Tools is used. To acknowledge their contributions, URL the content of the original "Contributors.xml" file is available at the end of the document.)

# Examples

In this section we will review the Use Case examples provided for us by RE-Tools.

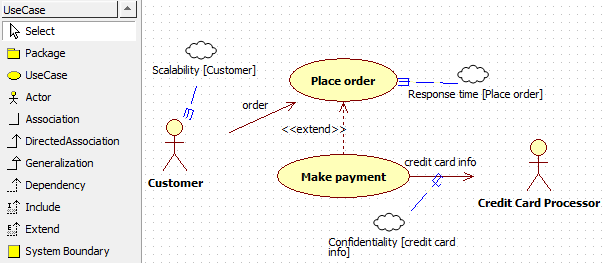
* Double-click to open the example file "C:\Program Files (x86)\StarUML\modules\RE-Tools\examples\RE-Tools v3.0 Examples.uml".
* The opened example (RE-Tools v3.0 Examples.uml) can be observed through the Model Explorer panel.



## Introducing the Example

* Open RE Use Case
* Double-click Usecase 1

(The work area loads the Use Case diagram and also activates the UseCase panel in the Toolbox area on the left side of the work area, as seen below)



* The UseCase panel, provides the components which can be used to create a Use Case diagram for requirements. The components provided includes; Select, Package, UseCase, Actor, Association, DirectedAssociation, Generalization, Dependency, Include, Extend and System Boundary.
* From the example above we have;

(For the functional requirements)

* + Two Use Cases: Place order and Make payment
  + Two Actors: Customer and Credit Card Processor
  + Two Directed Associations: order and credit card info
  + One Extend: <<extend>>

(For the non-functional requirements or NFR)

* + Three Dependencies
  + Three NFR Soft Goals: Scalability, Response time and Confidentiality

(The NFR Soft Goal can be obtained from the Agent panel of the Toolbox)

## Stepping through the Example

The explanation below describes what the Use Case diagram represents (You can view it as deriving a requirement statement from the Use Case Model).

* A customer can place an order
* A customer can make a payment

Here customer is an actor (a person or something performing an action). *Place order* and *make payment* are actions (known as use cases).

* Navigate to StarUML in your installed programs to launch it.
* Go to File > New Project By Approach…

To create an actor:

* Under UseCase panel of Toolbox, click Actor , then click on the work area.
* Double-click the actor and rename to **Customer**.

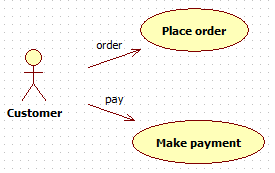
To create use case:

* Under UseCase panel of Toolbox, click UseCase , then click on the work area.
* Double-click the use case and rename to **Place order**.
* Repeat the step above to create **Make payment** use case.

To create a directed association:

* Under UseCase panel of Toolbox, click DirectedAssociation .
* Click down pointing from *Customer* to *Place order*, then release the mouse.
* Double-click the directed association arrow and rename to **order**.
* Following the step above, create **pay** directed association which links from *Customer* to *Make payment*.

(Your final image will look as follows)

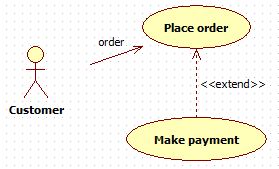


Note: *order* and *pay* are directed associations (associations link actions to actors).

* But a payment is derived from an order, and not independent from an order – as depicted by the image above.
* Now delete the **pay** directed association (as it is not suitable for our requirement statement).
* Under UseCase panel of Toolbox, click Extend .
* Click down pointing from *Make payment* to *Place order*, then release the mouse.

(The updated diagram will look as below)

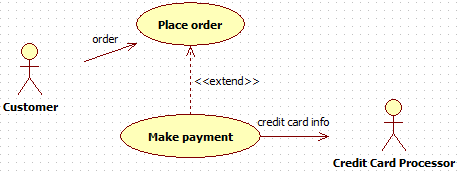
Therefore the right figure to show that payment is derived from order is shown below;



When a payment is made by a customer, the payment details (e.g. credit card info) is sent to another actor (Credit Card Processor) for processing.

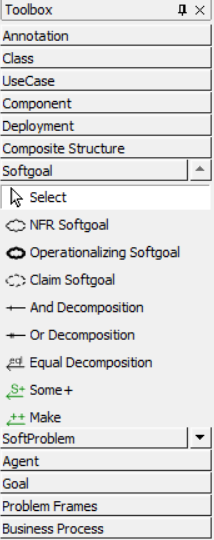
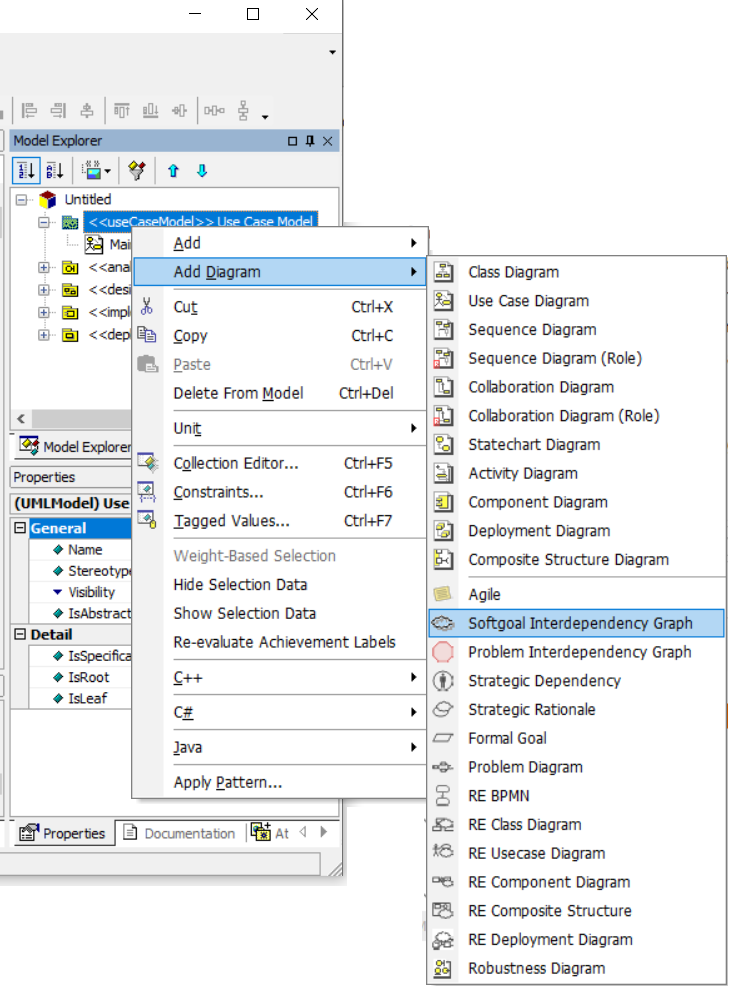
* Create another actor called **Credit Card Processor**, and place it at the right side of *Make payment*.
* Create another directed association called **credit card info** linking from *Make Payment* to *Credit Card Processor*.

(The updated diagram will look as below)



Now that is the end of our functional requirement.

Next we need add NFR Softgoal into Toolbox.



Next we want our system to support any number of customers not just one. This means our system needs to be scalable.

* Under Softgoal panel of Toolbox, click NFR Softgoal , then click on the work area.

(Note: do not change the text in the highlighted edit field)

* To rename, single-click on the new NFR Softgoal to select it.
* From the Standard Toolbar, select Tagged Values icon as shown below.



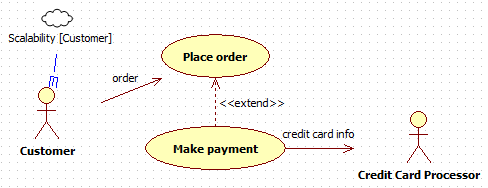
* Under RE tab, give the Softgoal a Type: **Scalability**
* Click Close

(Your new Scalability Softgoal looks like below)



* To link the Scalability Softgoal; Under Softgoal panel of Toolbox, click Topic Reference 
* Click down pointing from *Scalability* to *Customer*, then release the mouse.

(The updated diagram will look as below)



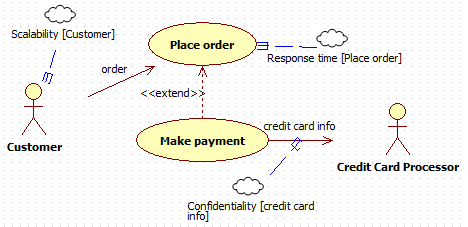
Also we want to ensure that when a customer places an order there is no delay in the response to the order request. We called this the *Response Time* requirement.

* Create a Softgoal as previous with Type as **Response time**
* Now link it to *Place order* use case, using a *Topic Reference* component

Finally we want to ensure that the network for our system is safe for any private data that the customer sends during payment – such as credit card info. We called this requirement; *Confidentiality*.

* Create a Softgoal as previous with Type as **Confidentiality**
* Now link it to *credit card info* directed association, using a *Topic Reference* component

(Our final model will look as follows)



# Exercise

**Exercise 1**

Following the example above, attempt the exercise below;

Create a UML Use Case model for the following requirements:

* Design a system that allows a user to;
  + register their details
  + sign-in
  + partially view some resources for free when signed in (e.g. an article)
  + when signed in; make a payment to download or gain full access to any resource
* NFR;
  + the system should support multi-users (i.e. any workload)
  + the registration process should validate input data
  + the system should ensure strict authorization (to ensure full access to resource is paid for)
  + the system should support a secure payment process, and
  + the system should support optimum performance when loading/viewing resources as a registered user.

**Execise 2**

Not all components in UseCase panel of the Toolbox were utilised in our examples. Investigate what they mean and how you may utilise them in your model.

# Directed Study

* Information on other Requirements Engineering techniques: <http://www.utdallas.edu/~supakkul/tools/RE-Tools/index.html>
* Contributors to StarUML is found here: <http://www.utdallas.edu/~supakkul/tools/RE-Tools/StarUML-Contributors.html>