UML Class Diagrams with Magicdraw

Robin Beaumont robin@organplayers.co.uk

Tuesday, 04 October 2011

Contents

Video of this practical chapter

You can see this tutorial as the videos 6 to 9 in the following playlist at:

http://www.youtube.com/playlist?list=PL5D590974B7967000

1. Introduction

1.1. Where to obtain the software

This depends on who you are:

- MSc students at Edinburgh University or The Royal College of Surgeons (Edin.) will be provided with the software along with the academic licence.
- All others can obtain the software by visiting and registering at: http://www.magicdraw.com/ for the free community edition. An alternative with a similar interface is Visual Paradigm (VP-UML).

1.2. Before you begin this tutorial?

Before you work through the appropriate chapters see section 11 at: http://www.robin-beaumont.co.uk/virtualclassroom/contents.html

1.3. What are the aims of this practical chapter?

This chapter is the second in a series of practical tutorials to introduce you to using a specific CASE tool, MagicDraw Personal Edition (MD/PA). This practical chapter assumes that you have worked through the first tutorial so in several places briefly describes what to do. By the end of this tutorial you will feel confident about using MD/PA to draw UML compliant Class diagrams.

This tutorial will make use of a specific narrative given on the next page.

2. The Narrative

A Primary care centre (PCC) consists of Employees, Clients (patients), Voluntary workers and Students. In terms of how the employees are paid they are classified as being either casual, part-time, full time or honorary staff. General Practitioners, a particular group of employees, can be either qualified or in training which because both are salaried are different from the various other students that are at the PCC. Nurses (of which there are several varieties) carry out consultations, home or institutional visits, client teaching sessions (one to one and group) and various clinics such as toddler, ulcer and diabetic management.

Patients can either be registered or visitors, either can see a variety of the above people. They may see a person for either an individual or a planned series of visits. A visit may be a group session/ consultation or one or more treatments (blood and/or urine test or just Blood pressure check, etc).

Each client's record has several aspects. One aspect consists of one or more Problems which may be open, referred, being managed or resolved. For example a patient presenting with a leg ulcer may be referred by the GP to a practice nurse who will dress the wound until it is healed, The GP may request a follow up appointment which may be either a one off event or several (such as every two weeks for 10 weeks). Associated with each problem may be specific treatments (each which may relate to more than one problem) However the treatment will always only relate to a single patient. Alternatively the GP may just refer the client to a consultant, where the Problem would have the status 'referred'.

Another aspect of the client's record are the diagnoses. Clients may have zero or more diagnoses which may be linked to a particular problem and /or specific treatment, sometimes a diagnosis may be a stand alone detail such as Klippel-Feil syndrome.

The client record also contains appointment details which may be either (missed, attended, patient abandoned or practice abandoned), once the client actually sees the person (usually the person they have the appointment with) a visit is recorded. The visit can be with anyone discussed above, a visit may be with more than on e person, such as a GP and a trainee GP.

The PCC makes use of both the BFI for advice about various treatments as well as an in house formulary both of which are available electronically.

Another group of people are the administrators (who can be employees, voluntary workers or students) such people operate various phone and reception services offered at the PCC. They vary from operating the front desk (logging and possibly editing appointments), to arranging repeat prescriptions and organizing telephone consultations with GP's or nurses for Blood pressure monitoring and Blood taking amongst other things.

The voluntary workers are managed by a voluntary worker co-coordinator who is also herself a voluntary worker.

We will not model this entire scenario but take some aspects of it to demonstrate various modelling procedures in Magicdraw.

Exercise 1

List below where in the above scenario you might want to model:

Aggregation or composition

Recursion

Inheritance

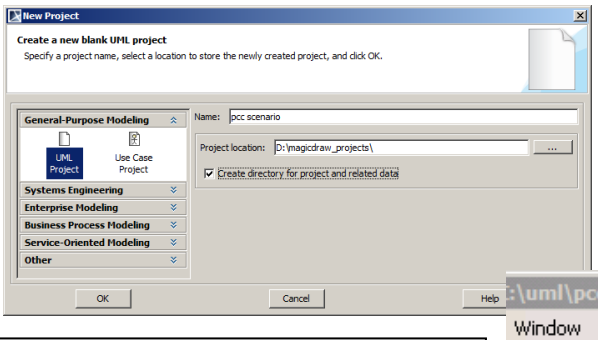
Ternary associations

We will now look at examples of each of these in turn and see how we can implement them in MagicDraw.

3. Aggregation and Composition

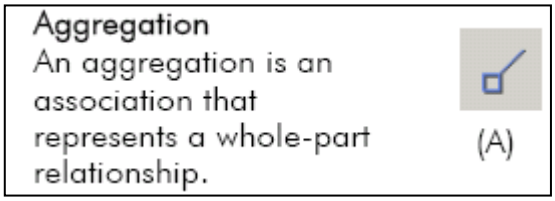
A good example in the above narrative of where aggregation could be modelled in UML is the sentence:

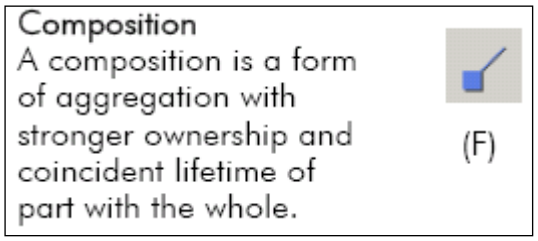
"A Primary care centre (PCC) consists of Employees, Clients (patients), Voluntary workers and Students"

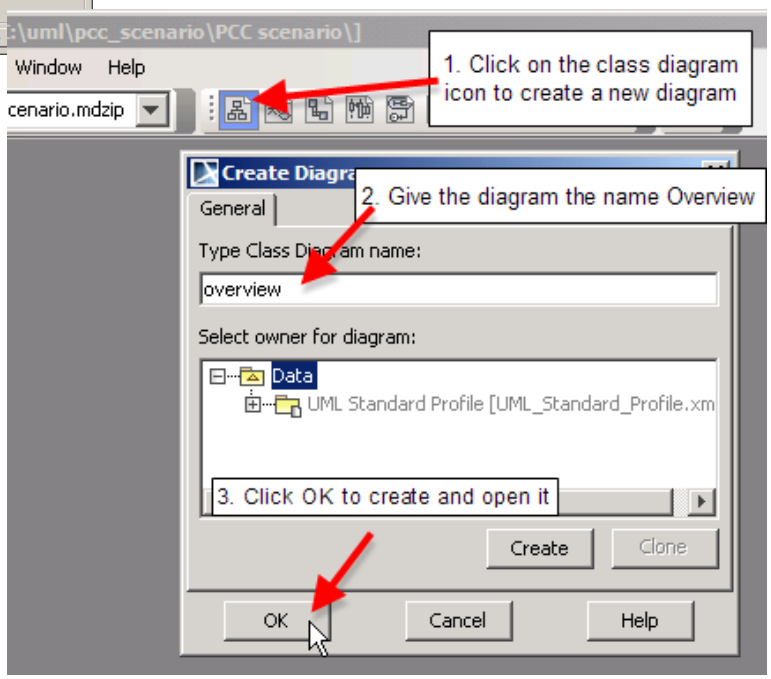


First you need to create a new project. I have given it the name **PCC scenario** and specified a new folder for it.

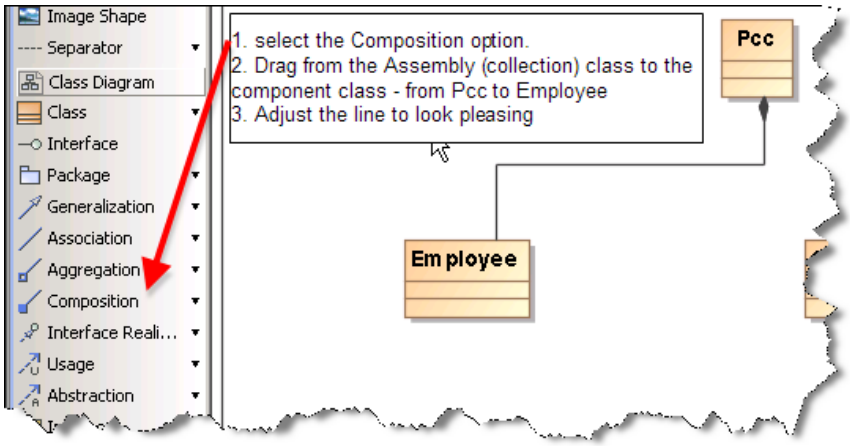
You then need to create a new Class diagram; follow the steps below to do that.







You then add the required classes to the diagram and then draw the required Composition association line as show below.

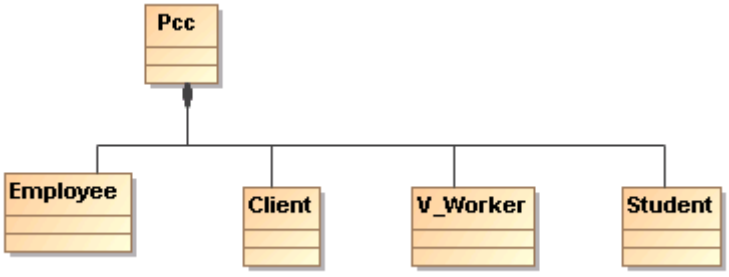


Exercise 2

Create the diagram opposite; you will need to move the various lines carefully to make them look as neat as this.

Also check the Youtube video "Drawing UML aggregation and composition associations" at:

http://www.youtube.com/playlist?list=PL5D590974B7967000

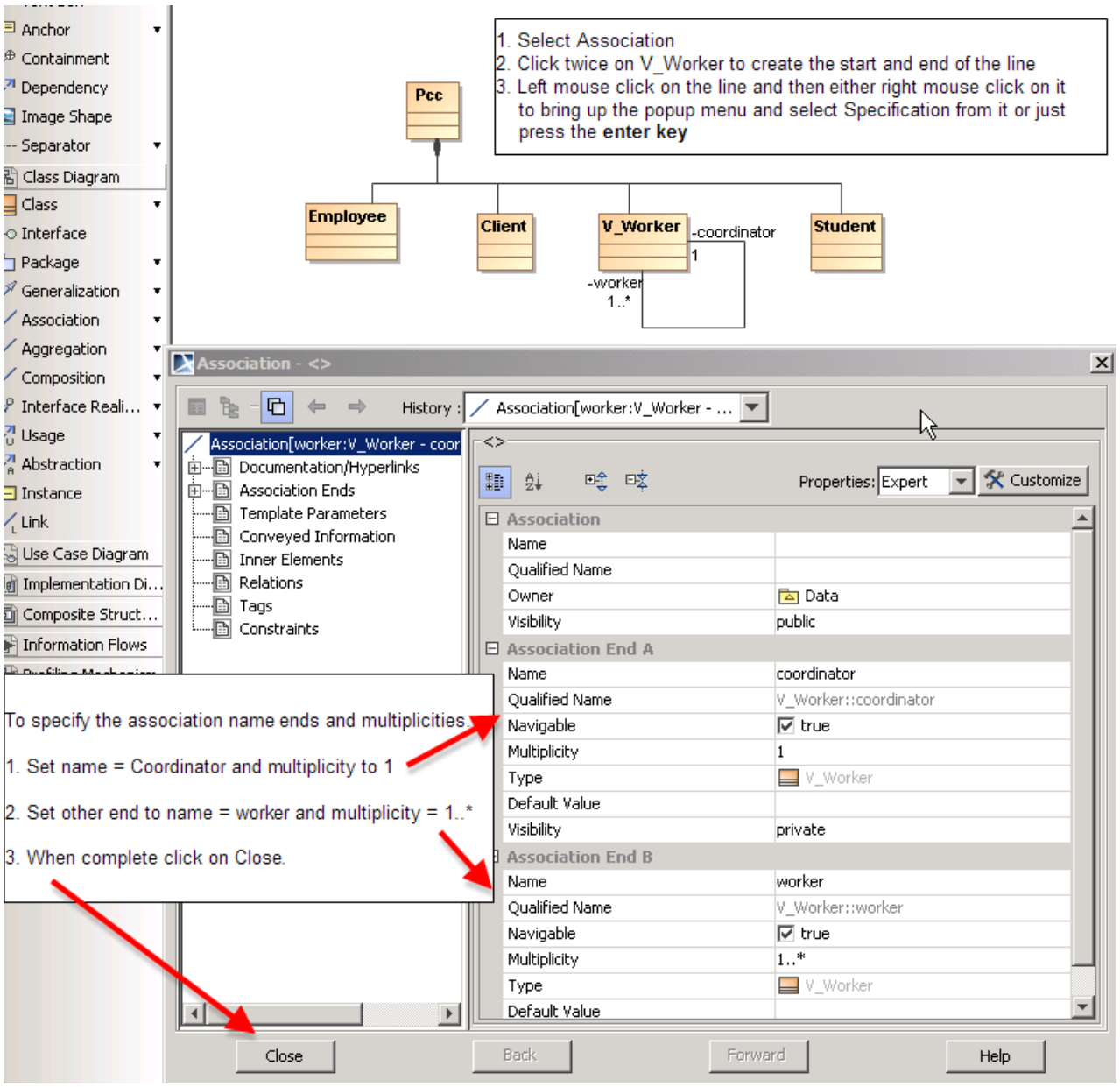


4. Recursion

A good example in the above scenario of where recursion could be modelled in UML is the sentence:

"The voluntary workers are managed by a voluntary worker co-coordinator who is also herself a voluntary worker."

The edited screen shot below provides details of how to achieve this in Magicdraw.



Exercise 3

Create the recursive association described above, you will need to move the various lines and classes carefully to make them look neat.

Also check the Youtube video "creating recursive associations in UML using Magicdraw" at:

http://www.youtube.com/playlist?list=PL5D590974B7967000

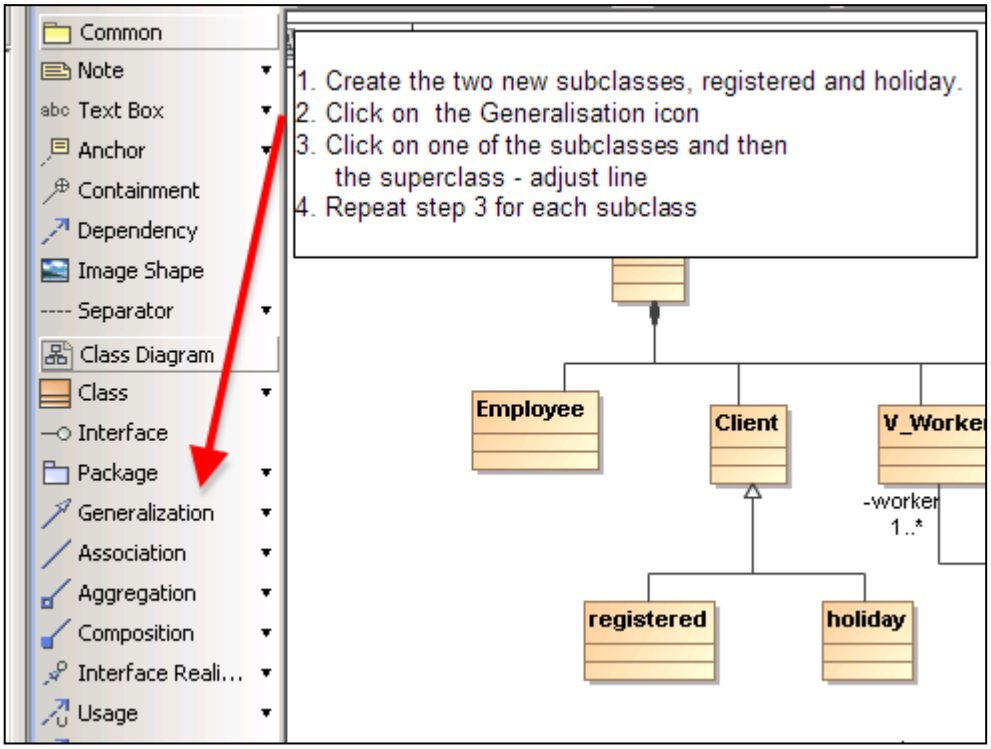
5. Inheritance/Generalisation

A good example in the above scenario of where inheritance could be modelled in UML is the sentence.

“Patients can either be registered or visitors”

You might be thinking that this would best be modelled as simply having an attribute in the PATIENT class indicating that they are either registered or visitors, and two of my YouTube videos discuss this point entitled 'misuses of inheritance'. On balance the better solution would probably be the attribute rather than inheritance approach here, however for now we will assume that we would collect different data about visitors compared to registered patients - we might be particularly lazy GPs and not collect a medical history or next of kin information for the visitors just giving then a prescription!

We have considered patients to be clients in the model because in most of the narrative description this term is used.

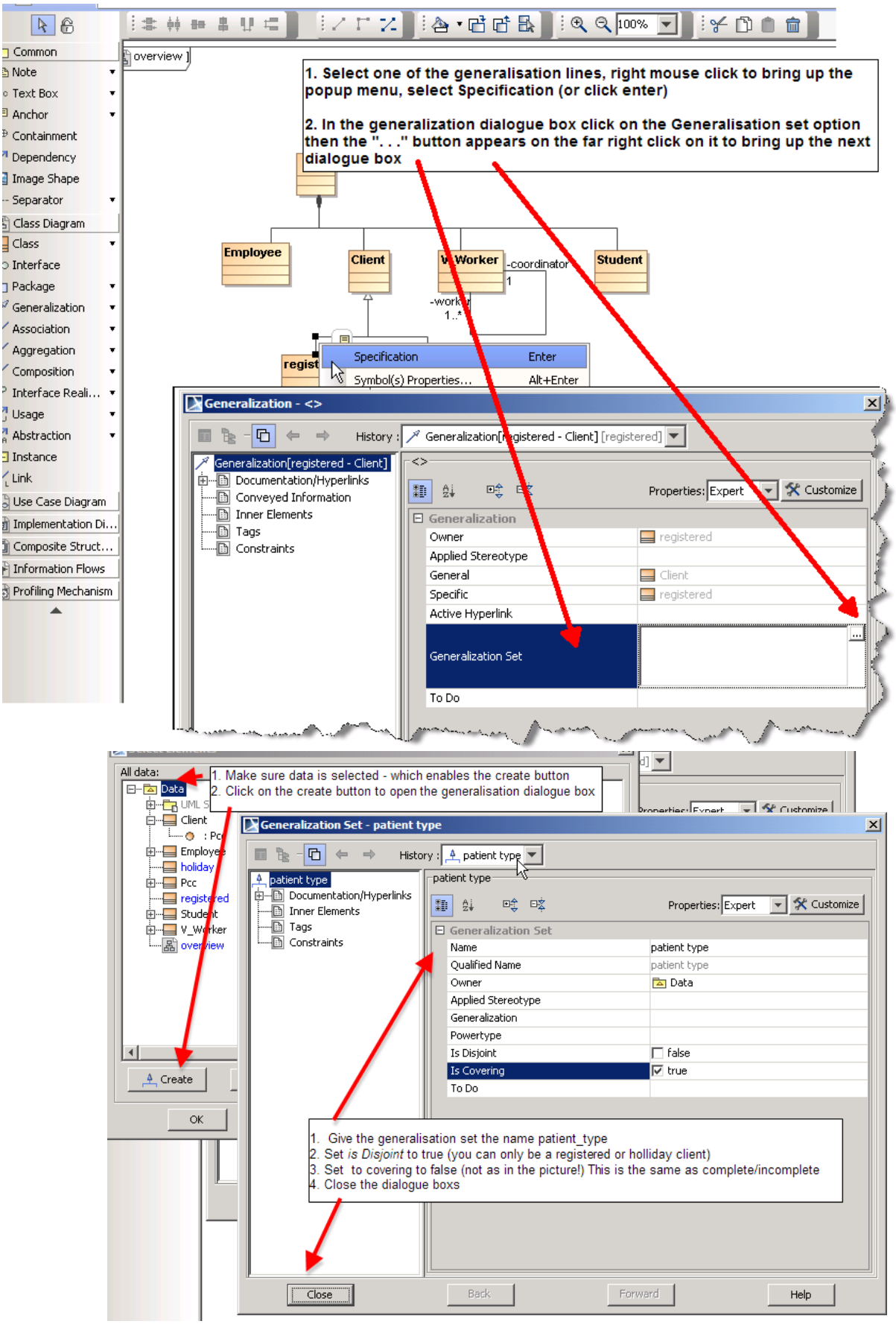


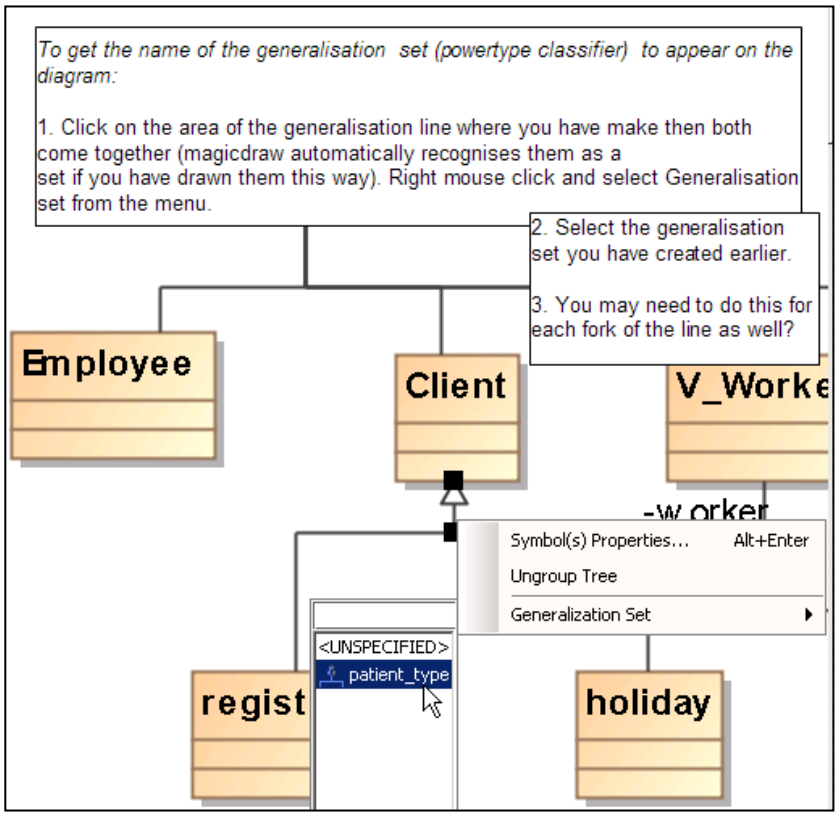
Exercise 4

Create the two subtypes described above. Remember you will need to move the various lines and classes carefully to make them look neat.

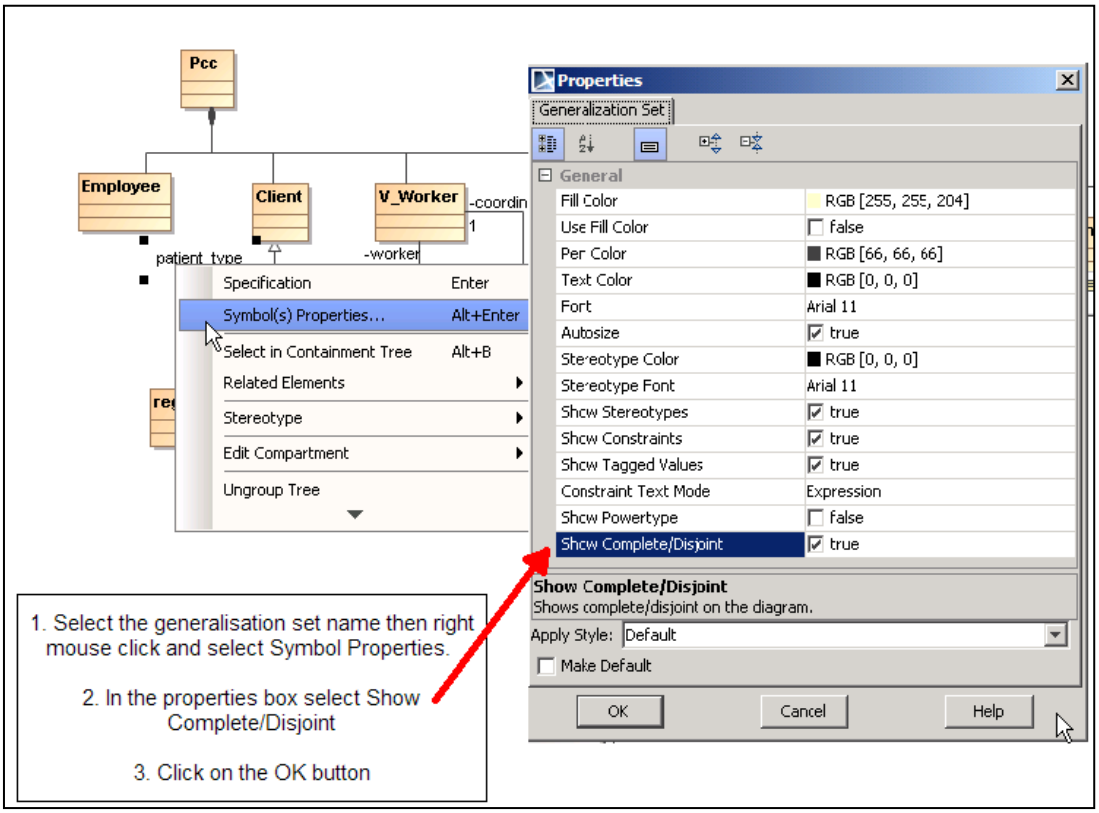
6. Generalisation sets

The Chapter "An introduction to UML - part 2 Associations" explains what generalisation sets are. We will create a generalisation set for the two subtypes we have just formed and give it the name patient_type. This involves rather a complex set of dialogue boxes – good luck.



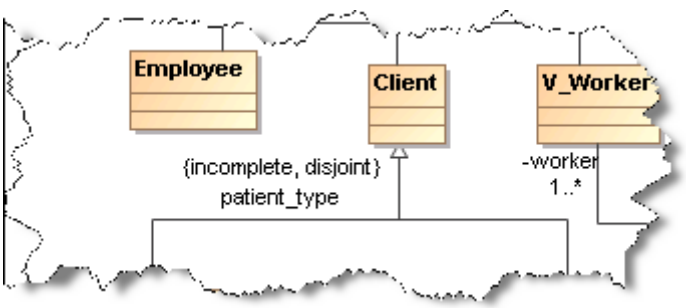


And finally to get the constraint details to be displayed:

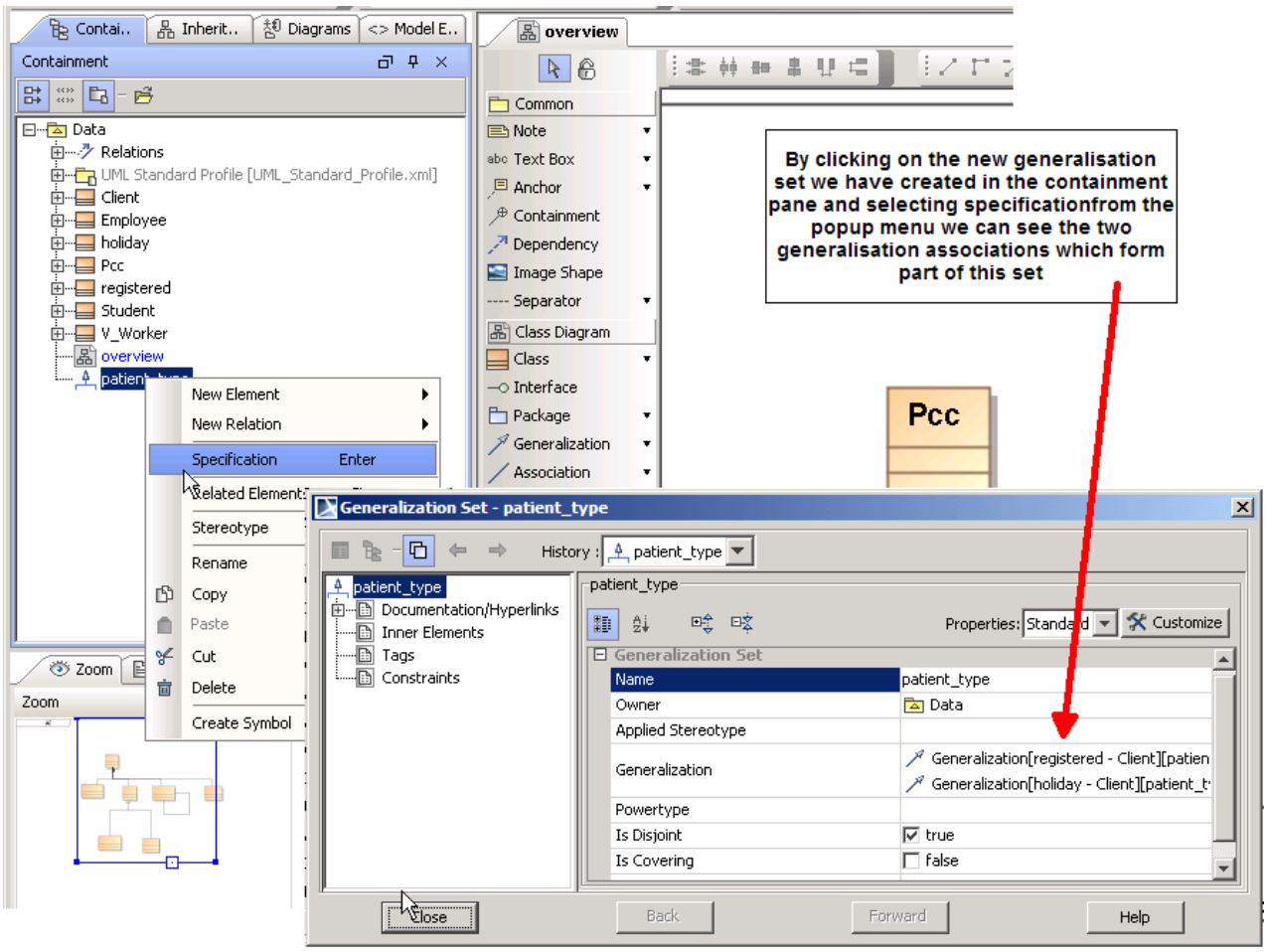


Exercise 5

Create the generalisation set described above and also make sure the name and constraint details are displayed as shown opposite.



You can see what you have done by looking at the containment panel on the left hand side of the screen:

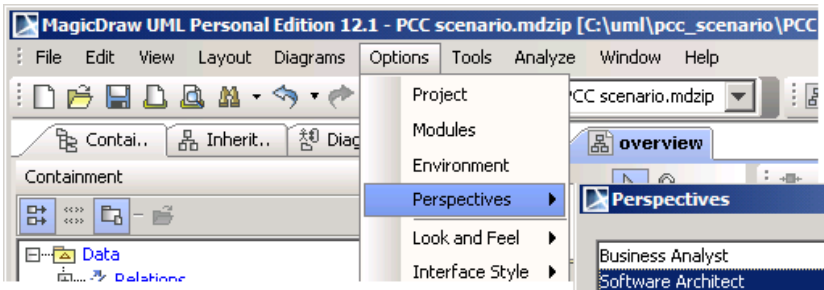


Exercise 6

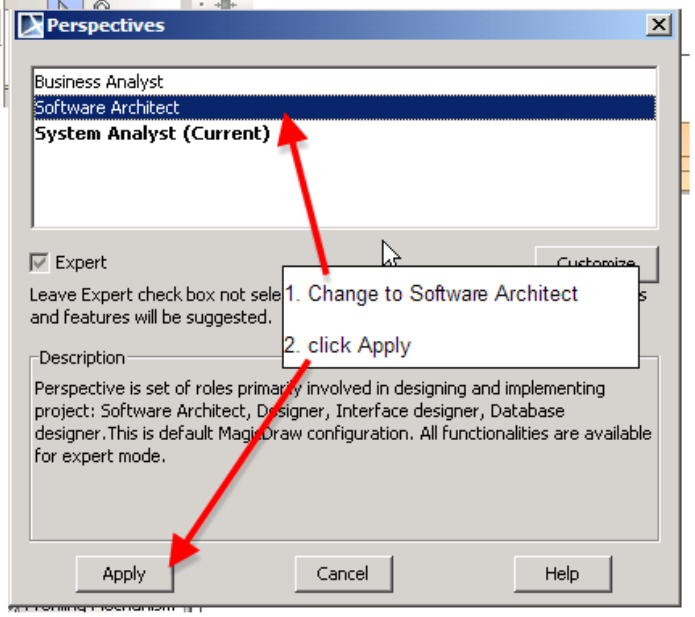
Have a look at the containment panel.

7. Ternary associations

The Chapter "An introduction to UML - part 2 Associations" explains Ternary associations and also information can be found in MagicDraw in help under, "N-ary Association". Unfortunately the diamond symbol you require to draw this type of association only appears on the menu after you have changed the ***perspective*** to software architect, You may remember that when you initially installed Magicdraw you set it to the System Analyst perspective.



To set the MagicDraw perspective to Software Architect select the menu option Options -> perspectives -> perspectives then follow the instructions:



Setting the scene

The scenario states

*"… once the **client** actually sees the person (usually the person they have the appointment with) a **visit** is recorded. The visit can be with **anyone discussed above**, a visit may be with more than one person, such as a GP and a trainee GP"*

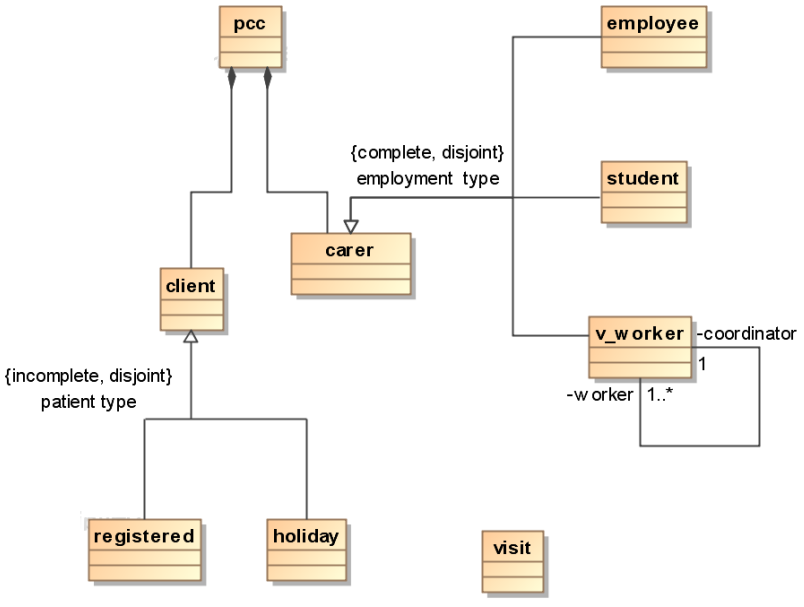
For the time being we will ignore the multiplicity details for each of the associations described above. More importantly we have a situation of a client seeing **anyone discussed** above which is rather problematic. To model this I have created a new superclass called carer (you may argue that some employees may not be Carers – but will ignore this for the moment). The only other new class that needs to be added is Visit.

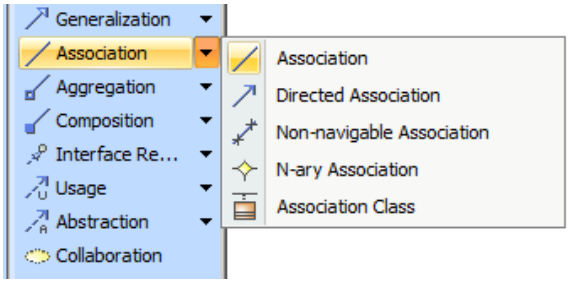
Exercise 7

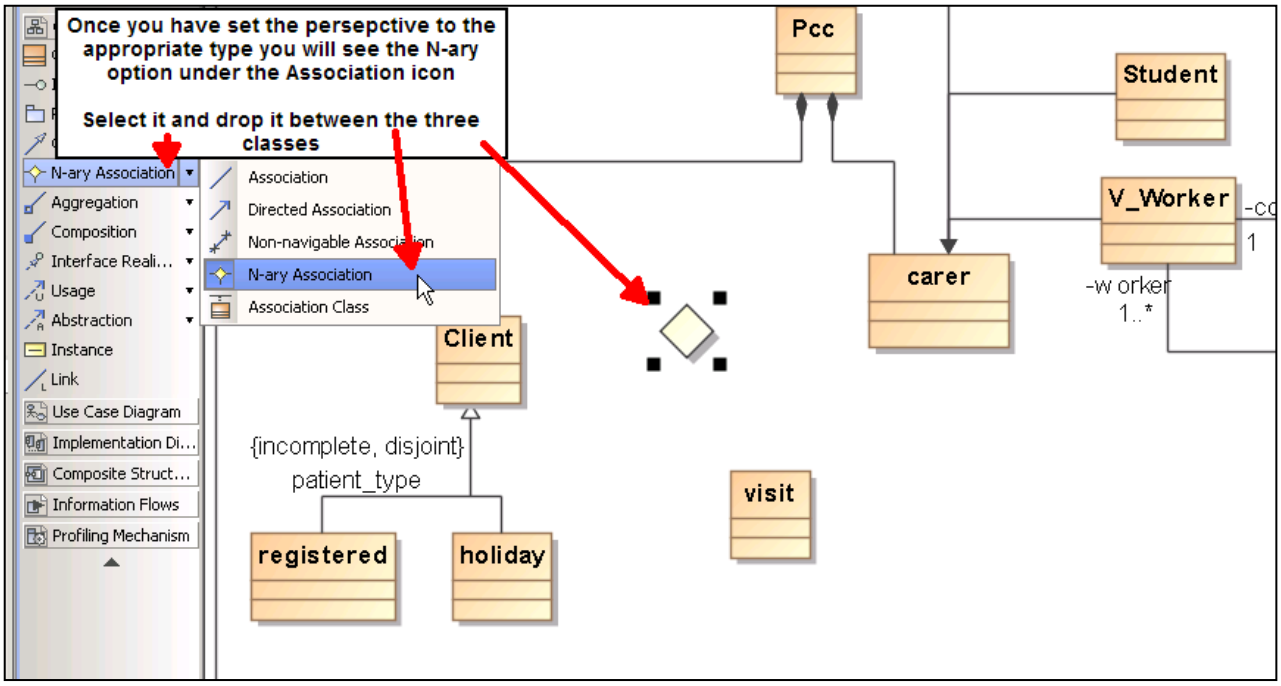
Edit your diagram to look like the one below where I have:

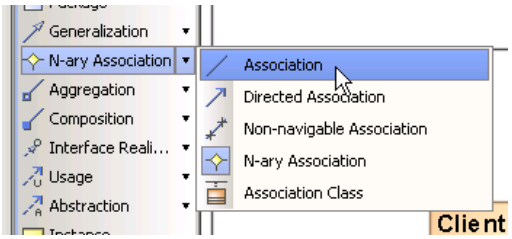
1. Added a CARER class and created another generalisation set called employee type, making it complete/disjoint

2. Added a VISIT class



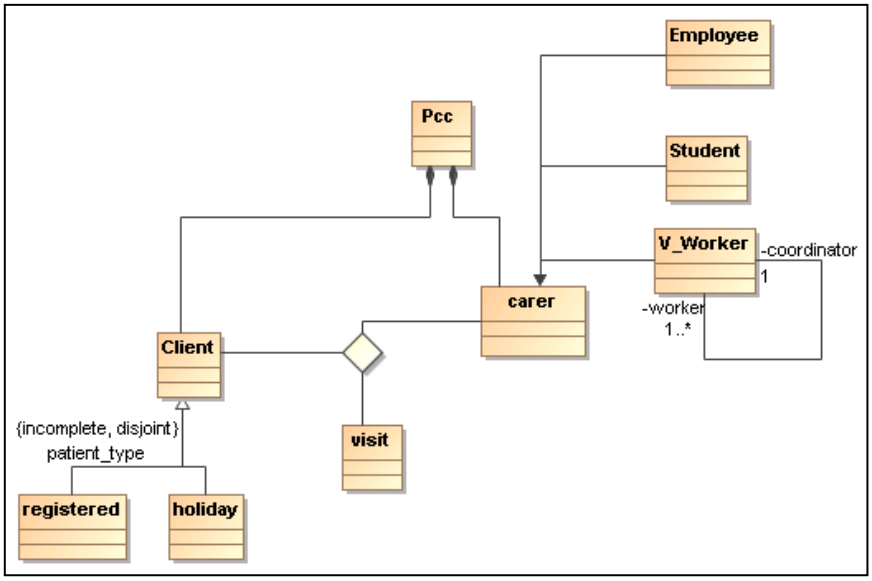






Now to draw the Associations, unfortunately the options on the drawing palette show the last chosen option so you need to select now N-ary Association -> Association to create the three simple associations between the diamond and each of the classes.

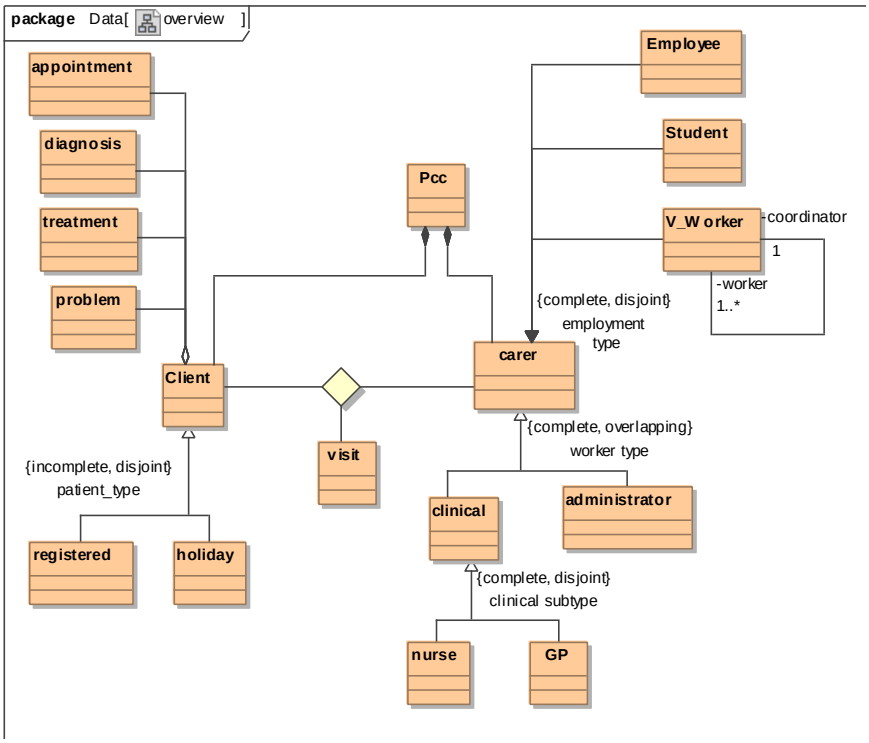
The finished result:



Exercise 8

Edit your diagram to look like the one above.

8. Extended Exercise



As you will have realised from the above exercises, and know from the material you have read, the development of any model is an iterative process. After re-visiting the narrative and comparing it with the model on the previous page I have again improved it, my new model is given opposite. How do you think it improves upon the previous model, and also do you think it is still lacking in certain areas (I'm sure it is!).

Exercise 9

Update and edit your model to mimic the one opposite. You will need to do this before moving onto the next tutorial concerned with UML sequence diagrams.

Important note: You may think that it is annoying to have to edit/change a model after you have developed one but being able to change a model quickly and with the minimum amount of effort is a very useful skill - You will soon realise this when you develop your own models where they will probably get changed hundreds of times!

9. Summary

We have considered in this practical chapter several more advanced UML Class diagram semantics, however we have still only really scratched the surface. There are a wide range of association line types, along with various options that can go with them that we have not considered. These will become known to you are you look at the drawing menu palette in detail and search Magicdraw help and the pdf user manual (also in the help option).

Two aspects that people often get bogged down with are, stereotypes and Packages, both of these I have deliberately avoided, but you can find out more details about them in my “Introduction to Class modelling using UML” chapter.

Also I have deliberately not introduced the complexity of adding methods to each class here, but will do so in latter tutorials. In the previous tutorial you added attributes to various classes so I have included a revision exercise below.

Exercise 10

1. Please add a number of attributes to each of the classes you have on the diagram. Remember this is a proprer UML class diagram so do NOT include foreign keys.

2. When you have done that found out a way to:

- Display the Class diagram showing the attributes for each class
- Display the Class diagram hiding the attributes and therefore only displaying the class name.

End of document Tuesday, 04 October 2011