ICT284 Systems Analysis and Design: Tutorial 4

**Domain modelling**

**ABOUT THIS TUTORIAL**

In the previous tutorial you commenced documenting the functional requirements of the Conference Coordinator information system using use case modelling. The next key concept for defining requirements is **domain modelling**, used to identify and define the ‘things’ in the system that we need to keep track of and store information about. In this tutorial, you will gain practice with techniques for identifying domain classes, and in representing them in a **domain model class diagram**. Use case modelling and domain modelling are complementary techniques, and together provide an overview of the entire system at a conceptual level.

**LEARNING OUTCOMES FOR THIS TUTORIAL**

**After completing this tutorial you should be able to:**

* Identify ‘things’ in the problem domain using the brainstorming technique
* Identify ‘things’ in the problem domain using the noun technique
* Draw domain model class diagrams using Microsoft Visio

**This tutorial addresses the following learning outcomes of the unit:**

LO 4. Model system requirements using UML, including use case diagrams and descriptions, activity diagrams and domain model class diagrams

LO 10. Present systems analysis and design documentation in an appropriate, consistent and professional manner

**REFERENCE MATERIAL**

* Topic 4 lecture notes and videos
* Satzinger, Jackson & Burd, Chapter 4
* Conference Coordinator Information System Case study (on LMS)

**QUIZ**

This week’s quiz includes some questions relating to material covered in this tutorial.

**Domain modelling for CCIS**

Last week you undertook use case modelling for the CCIS. This week, you proceed to data modelling in order to gain a more complete picture of the information required to support the operations needed in the CCIS. You are going to create a domain model class diagram to represent the information needs of the system.

1. Use a combination of the **brainstorming technique** and the **noun technique** to identify potential domain classes and their attributes. Consider the relationships among the classes you find.
2. Create the model on paper first, showing all domain classes and attributes. Show the multiplicity of all associations. If you find a many-to-many association, include an association class.
3. Where you have questions or there is insufficient information, discuss with your colleague and lecturer, and note any assumptions you make.

**Domain model class diagram**

1. When you are happy with your diagram, draw it in Visio. From New, choose the UML Class Diagram and select the Basic UML Class Diagram type. (From within Visio, follow the Shapes menu Software and Database; UML Model Diagram; UML Class Diagram).

Note that by default, your classes will probably have three boxes like this:



However, this is more like what the eventual design class diagramwill look like. The domain class diagram should not include the third box for methods. To remove it, just remove the dotted ‘separator’ line:



To show the associations between classes, use the Association connector. To show multiplicity, right-click on the association and choose Multiplicity. You can change the placeholders to show 1..1, 1..n, etc.

* Remember to include a title and legend to your diagram. Complete your domain model class diagram before the next tutorial.

**REVIEW: WHAT SHOULD I NOW BE ABLE TO DO?**

* Identify ‘things’ in the problem domain using the brainstorming technique
* Identify ‘things’ in the problem domain using the noun technique
* Draw domain model class diagrams using Microsoft Visio

**WHAT’S NEXT?**

You’ve now completed the two major conceptual modelling tasks for the CCIS: use case modelling, to identify and document the functional requirements of the system, and a domain model class diagram, to represent its information requirements. The next step is to review these models for consistency and to document parts of them in more depth, as we begin to move from analysis towards design. We’ll do this in the next tutorial.