

Domain Modelling

Outline:

The first two workshops allow you to apply some of the theories and ideas you've learnt in lectures. Working in groups of 4 you are to carry out some Domain Modelling on the systems identified below. Domain Modelling allows us to conceptualise a described system and is used in the first stages of discussions with a client to enable the development of a system.

In the first workshop (week 2), you can focus on the tasks in the Precursor; in the second workshop (week 3), you can focus on the domain modelling tasks. However, you are encouraged to look at those domain modelling tasks; we will keep looking at domain modelling in the following weeks.

Precursor:

Most likely in the following workshops you need to work in a group, which will be assigned. The first task is that within your group you need to select your favourite collaborative environment so that your group can work together. Any collaborative environments are acceptable, for example, Padlet and Trello. The second task is to learn to use a version control system and here we choose to use Git. Find the learning slides for Git which is in the blackboard and get to know it and then create an account in GitHub which is an open source software development platform. In the following workshops, you will use them to collaborate within your group.

Domain Modelling

- Go through the available data sources; such as the original description; Use Cases; Glossary etc.
 - With this, you must then identify possible Domain Classes (noun phrases)
- Identify basic relationships between the domain classes.
- Add additional key attributes
- Pay attention to the relationships, like names and multiplicities – it is very important that these are correct!

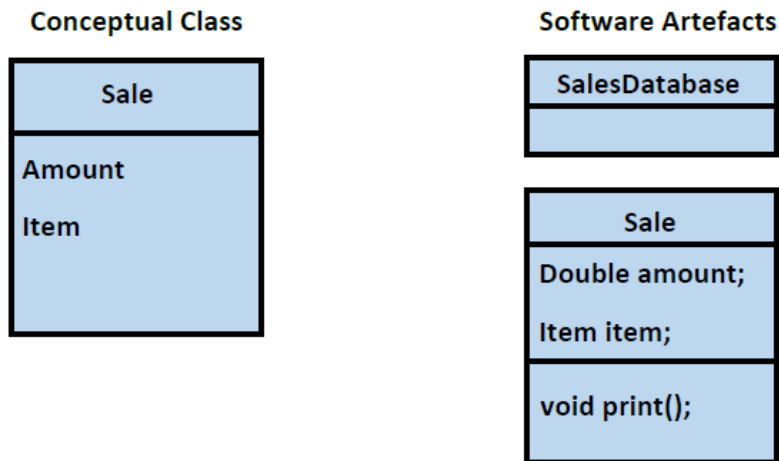
Performing Domain Analysis:

After carrying out Verb / Noun analysis on the outlines below, you need to analyse each of the selected words to see how they fit into the system, the following rules allow for this –

- Does the word represent a person carrying out an action?
 - If so, then this is an 'actor' (think UML) – 'R'
- Is the word also a verb (for example 'deposit' from the ATM example)?
 - Then this could be a 'method' (or function) – 'M'
- Or is it just a value, for example a 'colour' (string) or 'money' (integer)?
 - This would then most likely be an 'attribute' – 'A'
- Which noun phrases are not marked?
 - Then these could be a 'class' – 'C'
- Verbs can also sometimes be classes. This is where domain experts come in handy.

Conceptual Classes:

A Domain Model is a 'Conceptual' artefact, **not** a Software Artefact, however it does help to develop and lead to the Software Artefact. The diagram below shows the difference.



Task 0: (Individual)

Let's consider the following basic 'Student Library System':

***Courses** may have recommended items of **Reading Material**, which may be either complete **Books** or individual **Chapters**. Items of Reading Material may also have **Reviews** associated with them.*

The first step here is to identify the conceptual classes that would be associated with the above system (the description has hints!).

Once these have been identified, create a domain model for the system, paying attention to entity relationships. (Show this to a demonstrator before proceeding)

Task 1:

In your groups you are to create a Domain Model for the popular board game of Monopoly. The first part of creating the domain model is to perform a Verb / Noun analysis, in order to do this you will need to write the description of Monopoly, you may be able to find a good description of this on-line.

Remember to follow the rules:

- Identify candidate conceptual classes
- Draw them in a UML domain model
- Use existing names for things, the vocabulary of the domain.

Show the demonstrator your Verb / Noun analysis, and your final Domain Model UML diagram showing the game of Monopoly.

Task 2:

This time, create the Domain Model diagram for the system described below.

"We need a point-of-sale system for our company Acme Traders Co. to allow us to record the sales made at our shop and provide customers with a receipt for purchases. The system needs to be operated by any of our sales staff."

Point of Sale system:

Flow of system: -

1. Customer arrives at POS checkout with goods
2. Cashier starts a new sale
3. Cashier enters item identifier
4. System records sale line item and presents item description, price, and total

Cashier repeats steps 2-3 until complete

5. System presents total with tax (VAT) calculated
6. Cashier tells Customer the total, and asks for payment
7. Customer pays and System handles payment
8. System logs completed sale and updates stock
9. System presents receipt
10. Customer leaves with receipt and goods.

Task 3: (In your own time)

This is to be done in workshops or in your own time before the next workshop.

Create the following Domain models:

- 1) Hospital wards, teams of doctors and patients.

Ward is a division of a hospital or a suite of rooms shared by patients who need a similar kind of care. In a hospital, there are a number of wards, each of which may be empty or have on it one or more patients. Each ward has a unique name.

Wards are differentiated by gender of its patients, i.e. male wards and female wards. A ward can only have patients of the gender admitted to it. Ward and patient have constraint on Gender.

Every ward has a fixed capacity, which is the maximum number of patients that can be on it at one time (i.e. the capacity is the number of beds in the ward). Different wards may have different capacities.

The doctors in the hospital are organised into teams. Each team has a unique name or code and is headed by a consultant doctor. Consultant doctor is the senior doctor. They can supervise fellows, residents, and medical students. The rest of the team are all junior doctors. Each doctor could be a member of no more than one team.

2) Domain model – Treatments, Patients and doctors

From the diagram below, write the description of the structure of the system.

