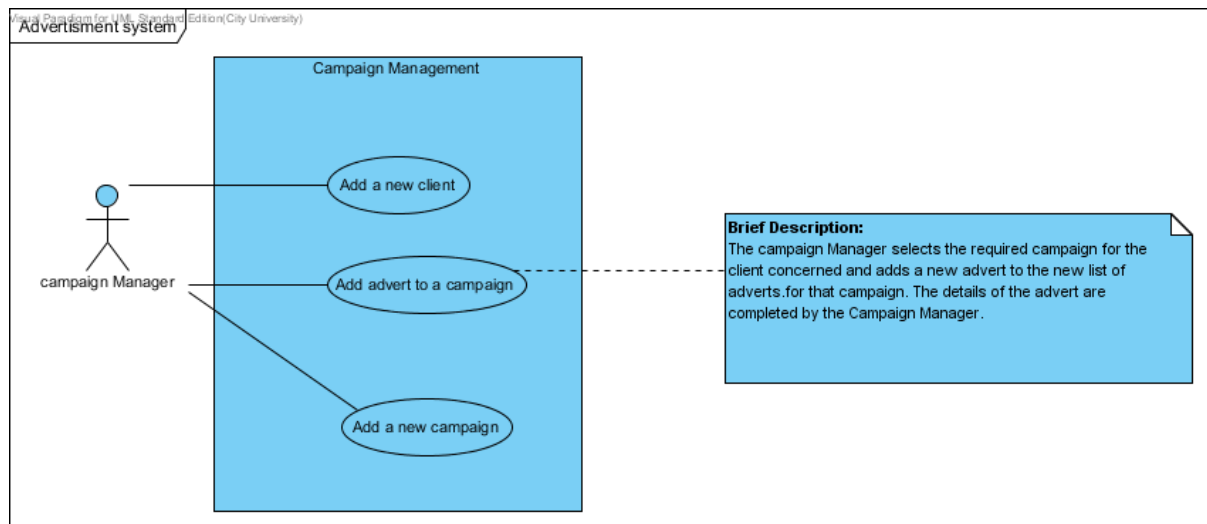


IN2013, CRC session in class, 15 October, 2013

Consider the following simple use case model (a simplified version of a more extensive example developed in Bennett, S., McRobb, S. and Farmer, R. “Object – Oriented Systems Analysis and Design using UML”, 4th Ed., McGraw – Hill, ISBN - 10 0 – 07 – 712536 - 3).



The specification of one of the use cases, ‘Add advert to a campaign’ is provided in the comment.

We will organise a brain storming session with CRC cards.

Scanning the description we could find 3 classes: Client, Campaign and Advert (noun-verb analysis will discover these). We need three volunteers to play the *roles of instances* of these classes. The Campaign Manager is an actor, i.e. is outside the system boundaries.

Let’s discuss the responsibilities of the classes:

- What they do
- What they know.

The non-obvious answer is whose responsibility it is to maintain the list of campaigns.

Provide a discussion of the options. Each of the role playing student must express an opinion.

Consider also how you would have implemented this example in Java to solve the problem with the list of campaigns.

After some deliberation, the team eventually reach an agreement on responsibilities.

Now consider whether an instance of a particular class will have to rely on other instances (how?) to deliver its own responsibilities.

Again, after a discussion, an agreement was reached that the CRC cards will have the following content (I used Visual Paradigm to prepare this model answer).

Visual Paradigm for UML, Standard Edition (City University)

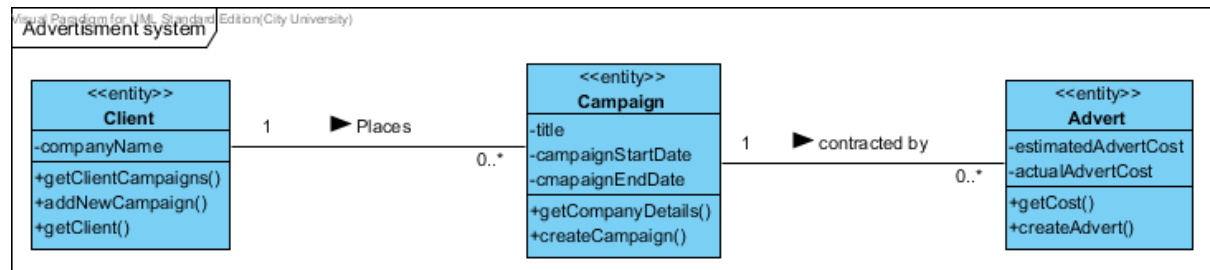
Client	
Attributes:	
Name	Description
name	
address	
email	
Responsibilities:	
Name	Collaborator
Provide client information	
Provide list of campaigns	Campaign - provides campaign

Campaign	
Attributes:	
Name	Description
campaignStartDate	
campaignFinishDate	
title	
Responsibilities:	
Name	Collaborator
Provide campaign information	
Provide list of adverts	Advert - provide advert details
Add a new advert	Advert - construct new advert object

Advert	
Attributes:	
Name	Description
estimatedAdvertCost	
actualAdvertCost	
Responsibilities:	
Name	Collaborator
Proved advert details	
Construct advert objects	

Now let us translate the responsibilities into attributed/operations and collaborations into associations between the classes. There seem to exist two simple 1 to many associations which are shown in the diagram below.

Visual Paradigm for UML, Standard Edition (City University)



Peter Popov