Modern Method Software Engineering

Home work 2

Sike Huang	**850414-A394**	**sike.huang@gmail.com**	**0762320173**
Shanbo Li	**840810-A478**	**shanboli@Gmail.com**	**0704646157**

2007-9-23

Question 1: What is difference between <<extend>>, <<include>> and use case generalization relationships in use case diagrams. Emphasize your answer by giving an example. (Bonus Point 01)

Answer:

The difference between include and extend relationships is the location of the dependency. Suppose we are going to add a few use cases for the actor Dispatcher, say, OpenIncident, AllocateResources, and ConnectionDown.

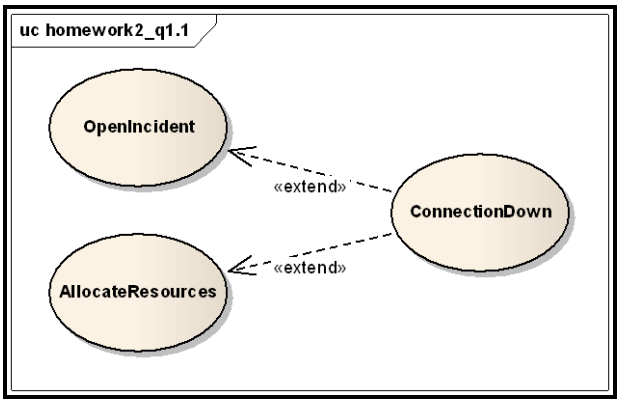

Figure 1-1 Simple diagram of extend relationship

If we model the ConnectionDown use case with include relationships, the authors of OpenIncident and AllocateResources use case has to know about and include the ConnectionDown use case. Conversely, if we use extend relationships instead, only the ConnectonDown use case needs to be changed to extend the additional use cases. In short, exceptional cases are modeled with extend relationships, whereas common behavior shared by use cases are expressed by include relationships.

The extend relationships and generalization relationships are also different. In a extend relationship, each use case shows a different flow of events to achieve a different task. In *Figure 1-1*, the OpenIncident use cases depicts the actions which take place when the Dispatcher creates a new Incident, in contrary the ConnectionDown use case describes the actions that occur during network crisis.

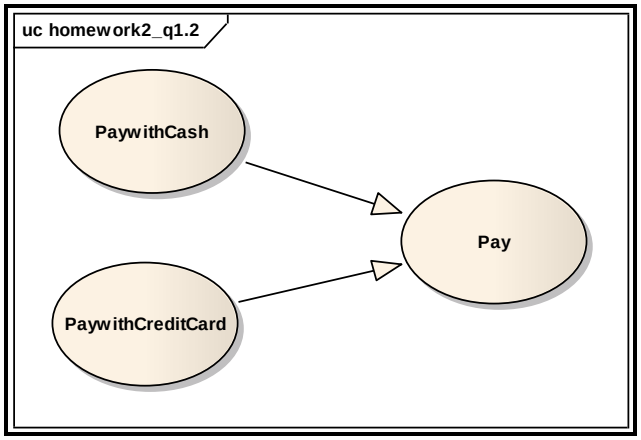

Figure 1-2 Simple diagram of generation relationship

In *Figure 1-2*, PaywithCash and PaywithCreditCard both explain actions that occur during Payment, however at different abstraction levels.

Question 2: Identify functional and nonfunctional requirements from the following description

Answer:

We have refined follow requirements from the description.

Functional Requirements:
● **The allocation of staff to production lines should be mostly automated.** ■ It describe a requirement of allocation of staff, if it cannot mostly automated allocate the staff to the production line, the staff in production planning(system operator) will have to do it himself. It is an interaction between user and system. So it is a functional requirement.
● **Process based on the skills and experience of operatives.** ■ It is what customer asked the allocation system developer to realize. And it is the rule to carry out the allocation.
● **Details of holidays and sick leave will also be taken into account.** ■ It is what customer asked the allocation system developer to realize. And it is also the rule to carry out the allocation.
● **Only staff in production planning will be able to amend the automatic allocation to find-tune the list.** ■ It reflects only certificate user could operate the allocation system. It is something about interaction between user and system.

Table 2-1 Functional Requirements

Nonfunctional Requirements:
● **A process will be run once a week to carry out the allocation.** ■ Performance
● **A first draft allocation list will be printed off by 12.00 noon on Friday for the following week.** ■ Performance, time associate.
● **Final Allocation list is printed out by 5.00 pm.** ■ Performance, time associate.
● **The system must be able to handle allocation of 100 operatives at present.** ■ Performance
● **The system should be capable of expansion to handle double that number.** ■ Performance

Table 2-2 Nonfunctional Requirements

Question 3:

Answer

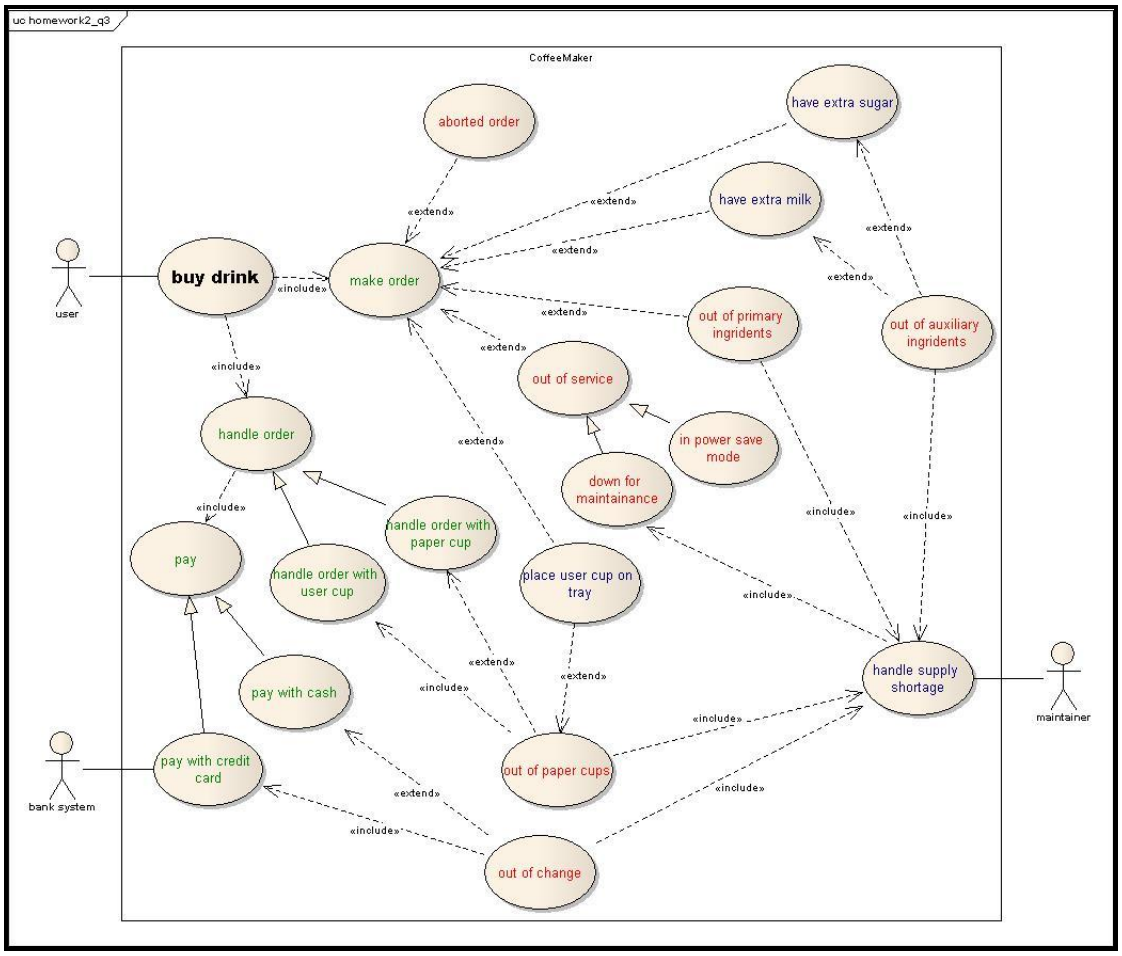

Figure 3-1 Refined use case diagram of CoffeeMaker

We arrange the use cases into four categories:

Categories	Describe	Member
Main use case	High-level use case.	Buy drink
Primary use case	Describe the core events that coffeemaker system is interested in.	make order, handle order(handle order with user cup, handle order with paper cup), pay(pay with credit card, pay with cash).
Secondary use case	These use cases stands for the events that coffeemaker system takes care, but they are not mainly focused on.	have extra milk, have extra sugar, place user cup on tray, handle supply shortage
Exceptional use case	These use cases happens only under exceptional conditions.	aborted order, out of primary ingredients, out of auxiliary ingredients, out of paper cup, out of service(down for maintains, in power save mode), out of change

Table 3-1 Use case categories table

Question 4: Consider an Academic Payroll System for a University:

1. Identify few actors (at least 2) and their scenarios (at least 1 for each actor) and identify use cases for the problem (at least 02 use cases). These use cases should be presented as textual description (see p. 163)

Answer:

Actors:

● **Employees in the Personnel Department** ■ Manager of AP system
● **Full-time academics** ■ View payment details, leave balances, and personal information; update personal details and payment method.
● **Casual academics** ■ Submit timecard to AP system, receive notification from AP system(if required)
● **The University** ■ Deducts standard tax rates from payments made to academics.

Table 4-1-1 Actors of AP system

Scenario:

Scenario name	**AddNewCasualAcademic**
Participating actor instances	Tom: EmployeeInPersonnelDepartment Jerry, Danna: CasualAcademic
Flow of events	1. The University just contracts with two new Casual Academics. As an Employee In Personnel Department, Tom is asked to add Jerry and Danna to AP system. 2. Tom login to AP system via a windows-based interface on his laptop. 3. Tom adds Jerry to AP system as a Casual Academic with Jerry's information. Then he gets AP system's echo with "New Casual Academic Adding Successful". And Tom also adds Danna with the same way. 4. AP system sends e-mails to Jerry and Danna, and notifies them that they have been enrolled to AP system, and ask them to confirm their information. 5. Jerry confirms the information. 6. Danna finds that her age is not correct and sends an e-mail back to Employee In Personnel Department with right information. 7. Tom receives Danna's e-mail and edits the age of Danna in AP system. 8. AP system sends Danna a email with new information, and ask for her confirmation. 9. Danna confirms the information update.

Table 4-1-2 AddNewCasualAcademic scenario

Scenario name	**UpdateAddressAndCheckPayment**
Participating actor instances	John: FullTimeAcademic
Flow of events	1. One Wednesday, John moves to a new house. He realizes that it is necessary to update his address in AP system. So he login to Academic Kiosk via safari, and wants to update his address. 2. John inputs his new address and press update button. 3. He finds that system echoes “personal information update successful”. 4. John also checks if they pay him the recently fortnight salary. 5. He finds that he got the money.

Table 4-1-3 UpdateAddressAndCheckPayment scenario

Use Case:

Name	**ViewPaymentDetails**
Participating Actors	Initiated by FullTimeAcademics(FA for short)
Flow of events	1. FA click view payment details button on Academic Kiosk(AK for short) 2. AK redirect to view payment details window. 3. AK asks FA if he wants to see all history or just wants to check the latest details. 4. FA chooses to see all history payment details. 5. AK shows all payment details history. Including payment date, payment summary and payment method. 6. FA reads all his payment details and requests to see the latest payment details. 7. System shows the FA’s latest payment details. Including payment date and time, payment summary, and payment details.
Entry Condition	● FA login to AK system via bowser.
Exit Conditions	● FA has checkout his payment state details
Quality Requirements	● FA can connect the EPD if he found his payment state is not the same as he wishes. ● Payment state has to be changed as soon as the University transfers money to FA’s bank account.

Table 4-1-4 ViewPaymentDetails use case

Name	UpdateFullTimeAcademicsInformation
Participating Actors	Initiated by EmployeesInThePersonnelDepartment(EPD for short) Update FullTimeAcademics(FA for short) information
Flow of events	1. EPD operators AP system and chooses to update FAs information. 2. System redirect to update FAs' Information window. 3. EPD enters the key word to search a FA. 4. System search FA with the key word. If it matches some of the FA in database, system shows the FA which match the key word (maybe more than one). Ask EPD to choose one of FA to display information. 5. Otherwise, if no matched FA, system go back to start of search step and asks EPD to enter a new key word. 6. EPD chooses one of FA to show his/her information. 7. System shows the FA's information. 8. EPD edits the FA's information and upload to system. 9. System saves the latest data and notifies FA that his/her information has already updated. 10. System shows information update successful and asks EPD if he want to search a new FA or exit UpdateFullTimeAcademicsInformation mode. 11. EPD chooses to search a new FA. 12. Otherwise EPD finds that all FA's information is up-to-date and exit UpdateFullTimeAcademicsInformation mode. 13. System goes back to start of search step. 14. Otherwise system exits UpdateFullTimeAcademicsInformation mode.
Entry Condition	● EPD login to AP system
Exit Conditions	● ALL FAs' information are up-to-date.
Quality Requirements	● FA can receive the information update notification and communicate with EPD if the FA found any information that have just been updated are wrong.

Table 4-1-5 UpdateFullTimeAcademicsInformation use case

2. Extend of the identified use cases with exceptions handling

Answer:

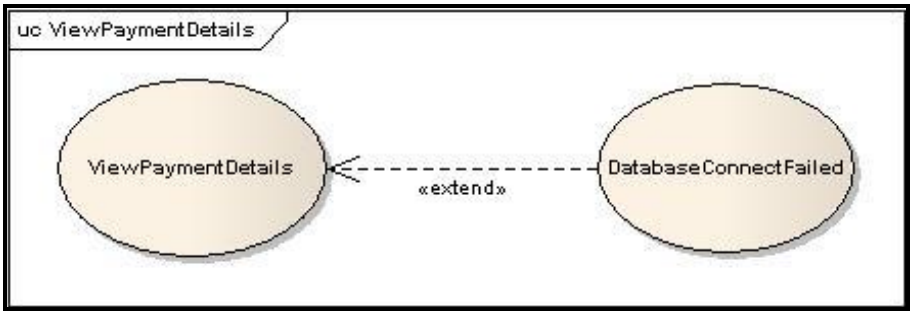

Figure 4-2-1 Exception handling of ViewPaymentDetails use case

DatabaseConnectFailed:	AK warns that Database is not available at the moment, and asks AF to try later.

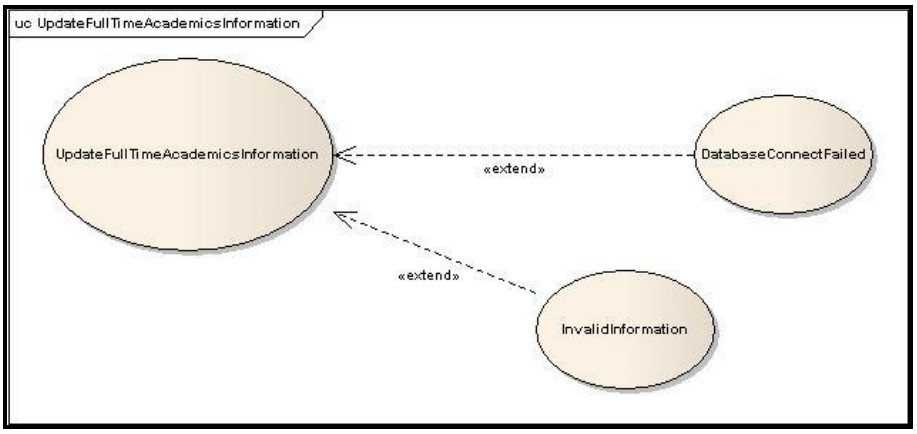

Figure 4-2-2 Exception handling of UpdateFullTimeAcademicsInformation use case

InvalidInformation:	AP system warns that the information which EPD just inputs is invalid, and asks EPD to re-input information
DatabaseConnectFailed:	AP system warns that Database is not available at the moment, and asks EPD to try later.

3. Describe of non-functional requirements

Answer:

Category	Nonfunctional Requirements
Usability	● In case a new AP has been added to AP system, She/he can access AK via a browser. ● The AP system should be shipped with user manual. ● The AP system should not throw technical error message to user.
Reliability	● If exception occurs during EPD's connect with AP system, it can restart. ● If exception occurs during FA's connect with AK, it can restart. ● Automatically do backup of AP system fortnightly.
Performance	● AK must support many parallel connects from CAs. (e.g. 100) ● AP system must support many parallel connects from EPD(e.g. 10)
Supportability	● EPD must be able to add new items to FA or CA's personal information on AP system. ● EPD must be able to change the standard of salary of FA or CA from AP system.
Implementation	● All FA should be able to access AK from a web browser supporting cookies, JavaScript. ● FA or CA cannot access from web browser. ● EPD should be able to access AP system via a windows-based desktop interface. ● AP system should run on any Unix operating system (e.g. MacOS X, Linux, Solaris).
Operation	● AK should be very easy to use so that FAs do not need any training in order to use it.
Legal	● All EPDs require secure authentication to use AP system. ● Any EPD cannot publish FA or CA's payment details. ● An FA or CA's payment details cannot be seen by any others except EPD. ● FA or CA can claim for a correct payment as required by local laws, if he cannot get agree with University on the salary they paid.

Table 4-3-1 nonfunctional requirements of AP system

Abbreviation about the nonfunctional requirements table above:

AP system	=	Academic Payroll system
EPD	=	Employees in the Personnel Department
FA	=	Full-Time Academics
CA	=	Casual academics
AK	=	Academic Kiosk

Question 5: Consider an information system of an advertising consultancy company:

1. Glossary for the problem

Answer：

Campaign：

A campaign is a competition among a number of companies. In AdConslt software system, the term refers to advertisement campaigns, where each campaign can have a multiple advertisements, which could be of several types.

Timesheet：

A timesheet keeps the track of the schedules and appointments assigned to a Staff, and it indicates the amount of in-office and out-of-office hours of consultancy a Staff has performed.

Template：

A template is a form used in negotiation between Director and Client, the template consists of client information, type of campaign needed, resources, and campaign budget information, start and end time.

2. Identified actors of the problem (at least 3 actors)

Answer:

Client：

Company that is assisted by Advert Consultancy Inc to compete in an advertising campaign,

Staff：

Staff provides consultancy to the Client, and present workload by the Timesheet.

Director (Department director)：

Direct performs negotiation with the Client, and assigns staff to the client, each department in Advert Consultancy Inc has its own director.

Board of governor：

Board of governor sees a summary of all active campaigns, utilization of staff, and current negotiations.

3. Identified scenarios of the problem (at least one for each actor)

Answer:

Scenario name	StartCreativeCampaign
Participating actor instances	David: Director Alice: Client James: Staff
Flow of events	1. Alice wants to take part into the advertisement campaign, and Alice goes to Advert Consultancy Inc to ask for consultancy for creating advertisements. 2. David from creation department meets Alice, David negotiates the campaign with Alice based on a negotiation template. 3. During the negotiation, David checks the current active campaigns with his department, and finds James is available, so David assigns James to Alice to consult with advertisement creation. 4. David and Alice agree with the negotiation and sign the contract, the campaign becomes active.

Table 5-3-1 StartCreativeCampaign scenario

Scenario name	WorkloadInChange
Participating actor instances	James: Staff
Flow of events	1. James finishes his work by the end of Friday. 2. James calculates the amount of hours of consultancy he has performed in this week. 3. James figures out 30 hours in-office work and 10 hours out-of-office work. 4. James updates his timesheet by writing down the amount of hours.

Table 5-3-2 WorkloadInChange scenario

Scenario name	StaffTransferBetweenDeparment
Participating actor instances	David: Director Scott: Director Mickey: Staff
Flow of events	1. David has an active campaign of creating advertisements. 2. During the campaign, David discovers a network failure in his creation department. 3. David gives a note to Scott about David's need for Scott's department staff. 4. Scott receives the note, and check schedules in his department. 5. Scott notices Mickey has free time. 6. Scott assigns Mickey to David, and sends an acknowledgment back to David.

Table 5-3-3 StaffTransferBetweenDepartment scenario

4. Identified use cases for the problem (at least 03 use cases). These use cases should be presented both as UML use case diagram and as textual description (see p. 163)

Answer:

UML use case:

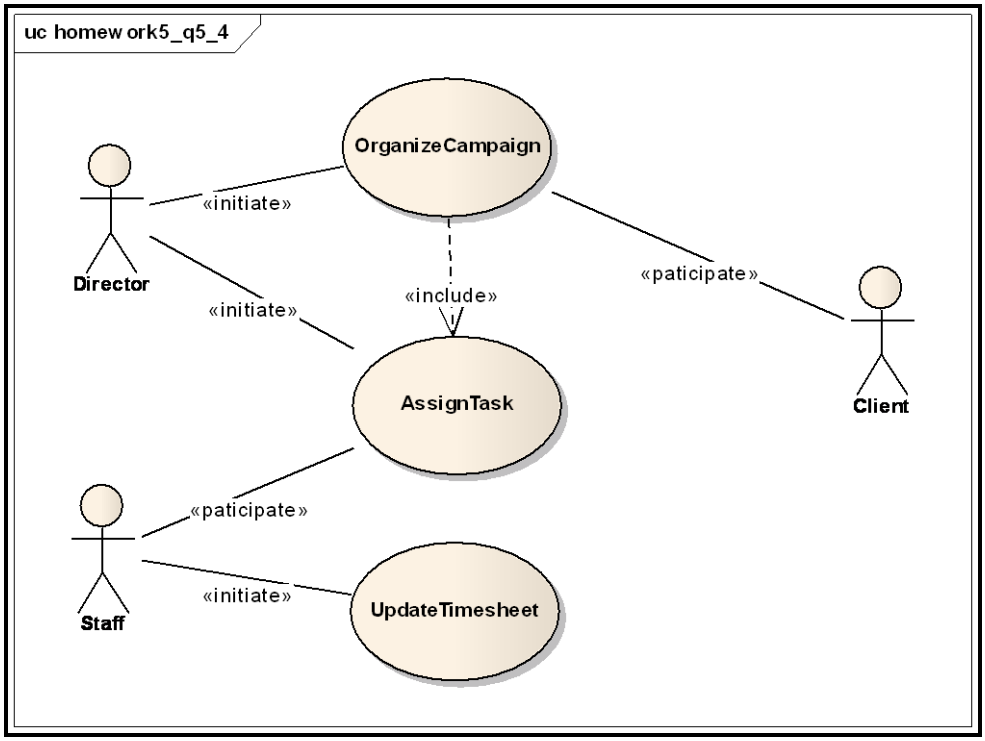

Figure 5-4-1 Draft use case

Textual Description:

Name	AssignTask
Participating Actors	Initiated by **Director** Communicates with **Staff**
Flow of events	1. The Director requests a new assignment of Staff. 2. The system presents the Director with an allocation form. 3. The Director specifies the campaign and the Staff to be involved in the campaign. 4. The system checks if the Staff is the department staff of current Director, if not, the system send a request note to the actual Director that the Staff belongs to, and wait for acknowledgement. Otherwise, the system updates the schedule of the Staff directly, and sends a popup message to inform the Staff. 5. In waiting the acknowledgement, if the system gets no reply, it sends a popup message to warn the current issuing Direct, if replied, the system update the schedule of the Staff, and sends a popup message to inform the Staff.
Entry Condition	● The Director is logged into system.
Exit Conditions	● The schedule of Staff is updated. ● The Staff is informed by the change in schedule.
Quality Requirements	● The system waits for the acknowledgement from another Director no more than 5 minutes, if no reply, the system assumes the quest of Staff is refused.

Table 5-4-1 AssignTask use case

Name	UpdateTimesheet
Participating Actors	Initiated by **Staff**
Flow of events	1. The Staff requests the list of his/her own schedule. 2. The system presents the Staff with updates for schedules and new appointments. 3. The Staff specifies the amount of hours of consultancy performed, further classified as in-office and out-of-office. 4. The system updates the timesheet by add the hours given by the Staff, and the system automatically calculates the skill level of Staff based on the total workload time.
Entry Condition	● The Staff is logged into system.
Exit Conditions	● The timesheet is updated.
Quality Requirements	● The system should list out all Staff that involved in certain campaign.

Table 5-4-2 UpdateTimesheet use case

Name	**OrganizeCampaign**
Participating Actors	Initiated by **Director** Communicates with **Client**
Flow of events	1. The Director requests the creation of a campaign. 2. The system presents the Director with a negotiation form. 3. The Client specifies the Client information, type of campaign needed, resources, and campaign budget information, start time and end time. 4. The system checks the previous campaign records of the Client, if no record found, the system creates a new entry for the Client. 5. Otherwise, the system appends the current campaign after previous records. 6. The Director locates Staff to campaign (include use case AssignTask). 7. The system compares the current global time to start time of campaign, if matched, the campaign becomes active immediately.
Entry Condition	● The Director is logged into system.
Exit Conditions	● The ownership of the campaign is settled. ● Client receives a copy of negotiation template.
Quality Requirements	● The negotiation template is presented from a secure channel. ● The Director should be able to cancel settled campaign, as required by the Client.

Table 5-4-3 OrganizeCampaign use case

5. Refinement of the use case diagrams

Answer:

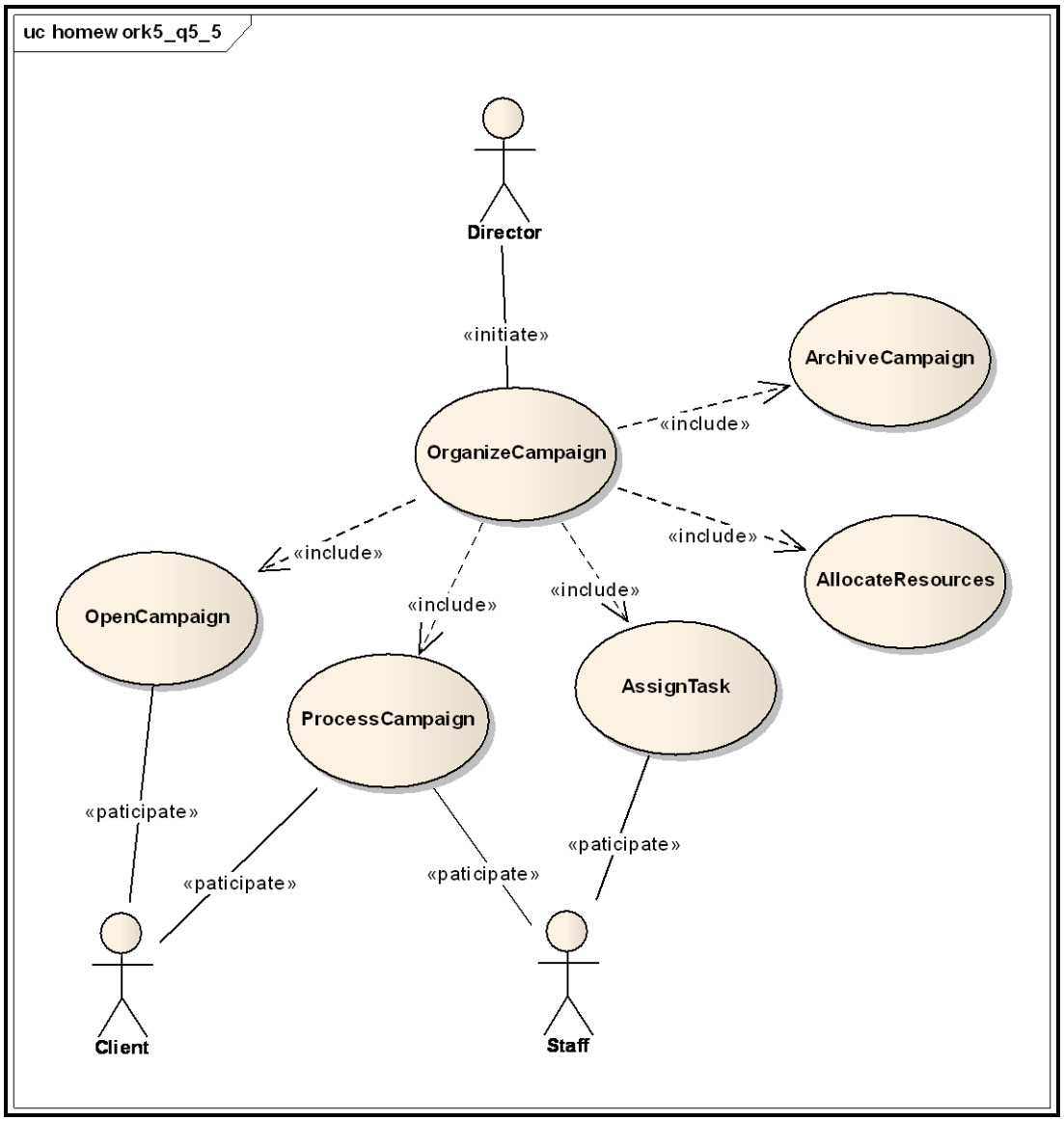

Figure 5-5-1 Refined use case

We do not show the complete refinement, we start by identifying one detailed use case for each step of the flow

6. One overview use case diagram with all the base use cases and all actors

Answer:

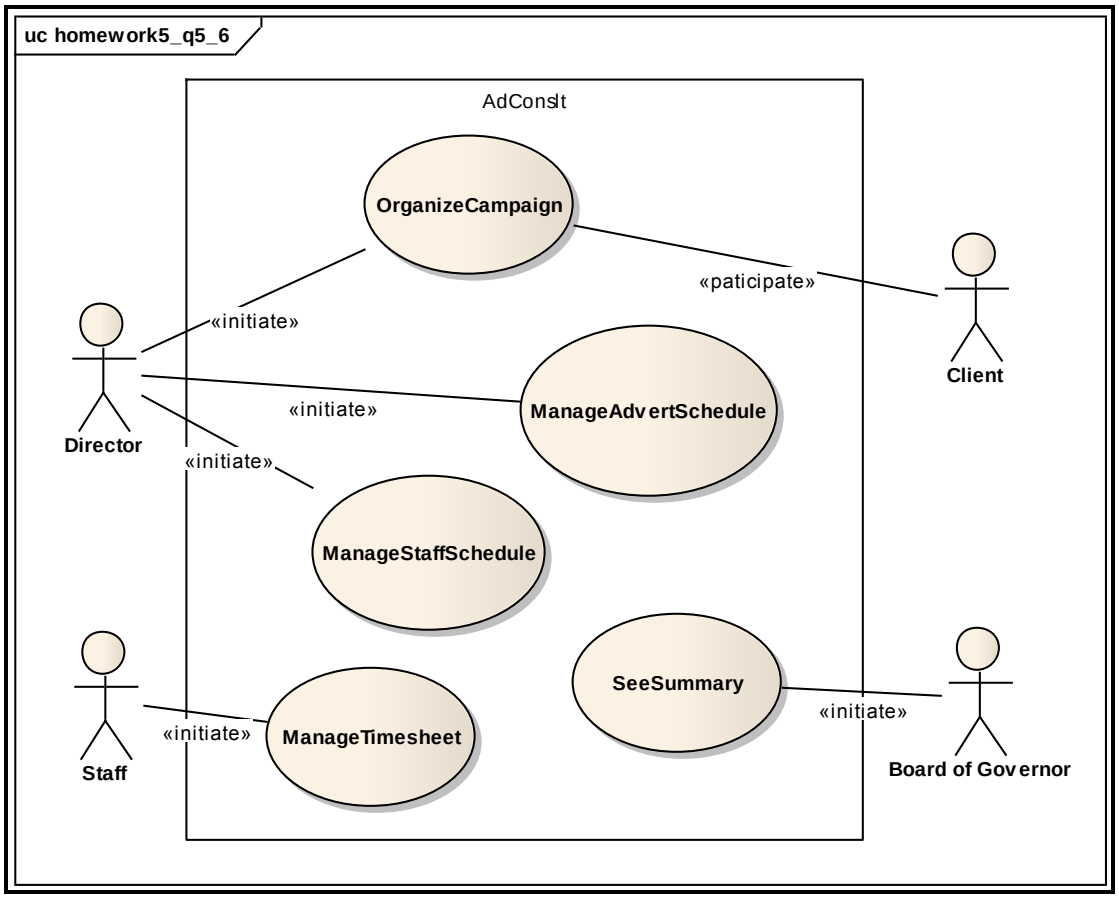

Figure 5-6-1 Overview use case

Figure 5-6-1 gives high-level use cases identified for AdConslt (Advert Consultancy Software System), it focuses on primarily tasks accomplished by the actor.

OrganizeCampaign

Direct uses negotiation template to discuss with Client to create a campaign, and Direct allocates Staff and other resources to the campaign, and current record of campaign is kept in the system for further references.

ManangeAdvertSchedule

Direct changes the advert scheduling information according to the needs from Client, as well as Direct views current active advert scheduling information to plan resources for future activities.

ManageStaffSchedule

Direct allocates Staff to a campaign, if the Staff is not within the current Direct's department, this Direct has to contact another department director about the need for staff.

ManageTimesheet

Staff sees the updates for schedules and new appointments, and Staff has to full out the amount of work as in-office or out-of-office latest by the end of the week.

SeeSummary

Board of Governor sees a summary of all active campaigns, utilization of staff, and current negotiations in process.

7. Extensions of the use case diagrams with exceptions handling

Answer:

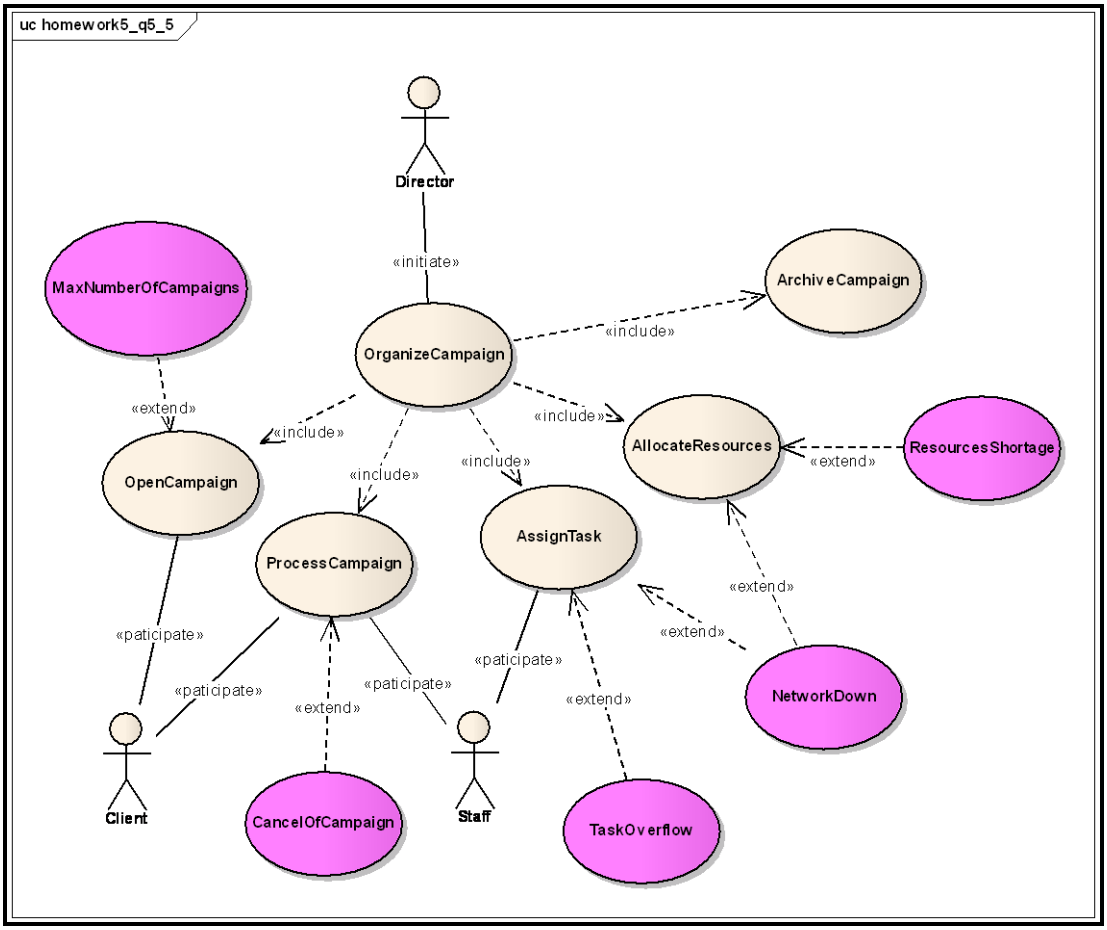

Figure 5-7-1 Refined use case with exceptional handling

We define five exceptions that could occur during the detailed use case OrganizeCampaign in respect of resource constraints and application domain constraints.

MaxNumberOfCampaigns	The system has achieved the maximum limit of campaign that it can handle, the OpenCampaign use case is terminated.
CancelCampaign	The client cancel the ongoing campaign, the system asks director to confirm the cancellation first, and remove the campaign from current active campaigns list, release the tasks defined in the campaign, and inform the previously assigned staff about the changes.
TaskOverflow	Too many tasks are assigned to staff, or the system discovers a conflict in the schedule of staff, the system skips current task assignment and informs director to conduct a new task assignment.
NetworkDown	Network failure or crisis happens during the processing of task and resource assignments, the system tries to repeat the allocations in later time.
ResoucesShortage	The system detects a anemia of resources, the system attempts to arrange resource from other departments.

8. Description of non-functional requirements

Answer:

Category	Nonfunctional Requirements
Usability	● The system should be shipped with user manual. ● The system should not throw technical error message to user.
Reliability	● Records are kept for all clients. ● The system automatically backups all records monthly.
Performance	● The system supports handling up to hundreds of ongoing active campaigns. ● The system generates list of schedules of campaigns or staffs with 5 seconds.
Supportability	● The director must be able to add new campaign. ● No modification of the existing system should be required.
Implementation	● The system should be implemented in Java programming language. ● The system is based on the combination of Struts, Spring and Hibernate.
Operation	● The director should not be able locate more resources in a campaign than a fixed limit agreed beforehand with the client during negotiation.
Legal	● The negotiation contents and the contract of the campaign should be treated as top credential. ● The director should be able to dismiss the consultancy agreement within a fixed period, as required by local laws.

Table 5-8-1 Nonfunctional Requirements of Advert Consultancy Software System