Reading Technique for Use Case Construction

Create use cases to document the functionality that users of the system should be able to perform. This will require listing the functionality that the new system will provide and the external interfaces which are involved. You will have to identify actors (sets of users) who will use the system for specific types of functionality. Remember to include all of the functionality in the system, including special/contingency conditions. Follow the procedure below to generate the use cases, using the questions provided to identify faults in the requirements:

1) Read through the requirements once, finding participants involved.

   a) Identify the participants in the new requirements. Participants are the other systems and the users that interact with the system described in the requirements – that is, they participate in the system functionality by sending messages to the system and/or receiving information and instructions from it. List these participants on Form A. You may use the following questions to help you identify participants:
      - Which user groups use the system to perform a task?
      - Which user groups are needed by the system in order to perform its functions? These functions could be its major functions, or secondary functions such as system maintenance and administration.
      - Which are the external systems that use the system in order to perform a task?
      - Which components or hardware are managed by the system?

   b) Decide which of the participants are actors. Actors represent classes of users which are external to the system and interact with it in characteristic ways. Indicate these actors on the form by checking the box next to the appropriate participants. The following guidelines may be helpful:
      - Actors represent *classes* of users, and may not be the same as the *individual* users of the system.
      - Actors represent the set of external things that need to exchange information (or otherwise interact) with the system.
      - Actors are different from other participants in that their actions are non-deterministic.

   Q1.1 Are multiple terms used to describe the same participant in the requirements?
   Q1.2 Is the description of how the system interacts with a participant inconsistent with the general requirements? Are the requirements unclear or inconsistent about this interaction?
   Q1.3 Have necessary participants been omitted? That is, does the system need to interact with another system, a piece of hardware, or a type of user which is not described?
   Q1.4 Is an external system or a class of "users" described in the requirements which does not actually interact with the system?

2) Read through the requirements a second time, identifying the product functions.

   a) Identify low-level functionality in the requirements. Jot these functionalities down on a piece of scrap paper.

   b) Group functionality together to form higher-level tasks. Each of these high-level tasks will be a use case or scenario that represents, at a high-level, what kind of system functionality will be used by an actor. You may find it helpful to consider the following guidelines when group functionality together to create use cases:

      i) Group functionalities that are involved with the same subsystem.

ii) Identify processing phases – is there a sequence or ordering to the functionality? Group together functionality which is consecutive.
iii) Group functionalities that lead to a definite goal state for the system or for an actor.

c) For each group of functionality, decide how the lower-level pieces are related. On a copy of Form B, use labeled boxes to represent the lower-level pieces in a particular group. Draw arrows between the boxes to identify the flow of system control ("First this functionality happens, then this…") and use branching arrows to signify places where the flow of control could split. Remember to include exception functionality in the use case. Check the appropriate functional requirements to ensure that you have the details of the processing and flow of control correct.

d) For each use case you create, remember to signify the actor who would use the functionality, as well as the action that starts the functionality (e.g. "selecting option 2 from the main menu" might start a use case for deleting records from a database). There are spaces for recording both of these pieces of information on Form B. Finally, give the use case a descriptive name that conveys the functionality it represents. Enter the name both on the associated Form B as well as on Form A.

Q2.1 Are the start conditions for each use case specified at an appropriate level of detail?
Q2.2 Has any functionality that should appear in a use case been omitted?
Q2.3 Has the functionality been described sufficiently so that you understand how the low-level functionalities are related to each other? Does the description allow more than one interpretation as to how the functionalities are related?
Q2.4 Have necessary functionalities been omitted, according to your domain knowledge or the general description?

3) Match the actors to all of the use cases in which they participate. (Remember that if two actors participate in all of the same use cases, they probably represent a single unique actor and should be combined.) Use the third column on Form A to cross-reference the actors with the appropriate use case from the list.

Q3.1 Is it clear from the requirements which actors are involved in which use cases?
Q3.2 Based on the general requirements and your knowledge of the domain, have all of the necessary use cases been specified for each actor?
Q3.3 Are use cases described for an actor which are incompatible with the actor's description in the general requirements?
Q3.4 Have necessary actors been omitted (that is, are there use cases for which no actor is specified)?

Acknowledgements
These guidelines owe much to the following works:

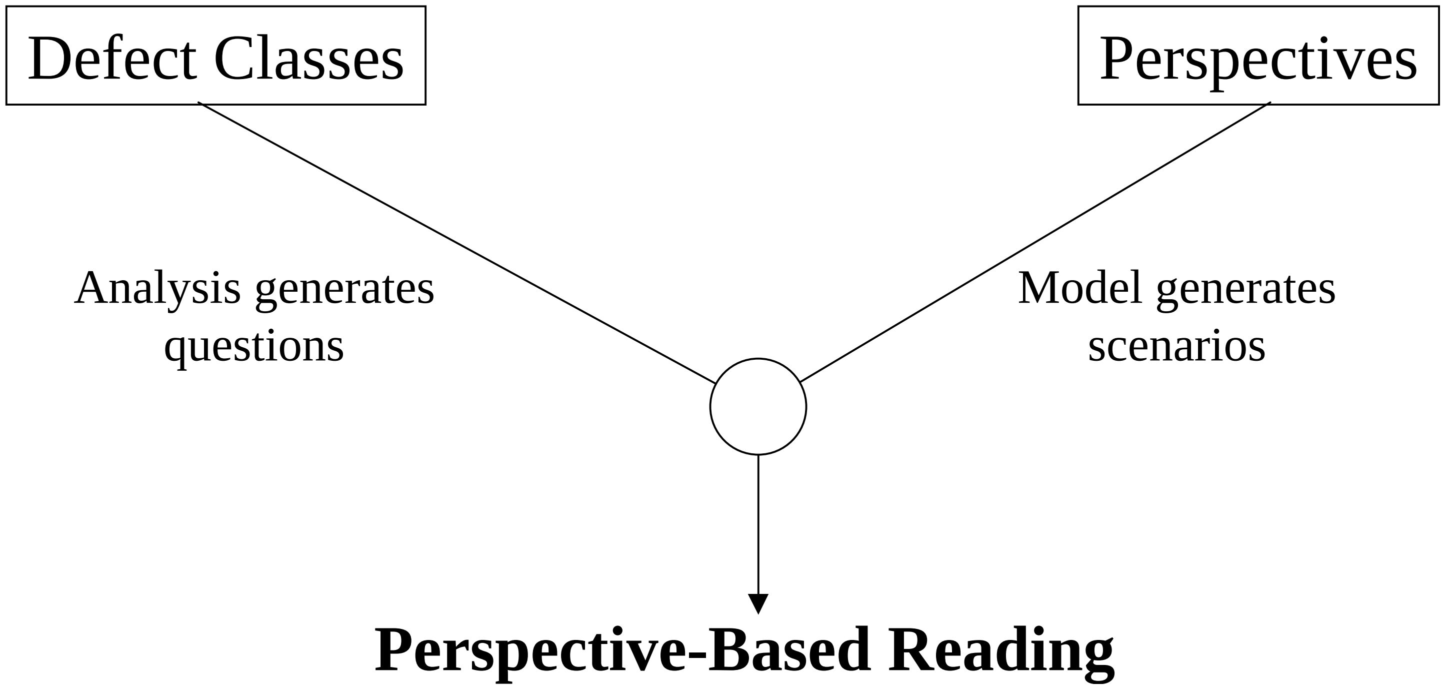
Andersson, Michael, and Johan Bergstrand. (1995) *Formalizing Use Cases with Message Sequence Charts.* Masters Thesis for the Department of Communication Systems at Lund Institute of Technology.
Jacobson, Ivar, *et al.* (1992) *Object-Oriented Software Engineering: A Use Case Driven Approach.* Addison-Wesley Publishing Company.

And to the work of Curt Zimmerman during his stay with the Empirical Software Engineering Group.

Perspective-Based Reading

- Example users of a requirements document:
  - a designer, who uses it to produce a system design;
  - a tester, who produces a test plan to ensure that the system meets the requirements;
  - a requirements analyst, who has to translate the functionality into a manual for the user;
  - …
- Each of them creates a model of the system described by the requirements.

Perspective-Based Reading


Defect Classes
Perspectives
Analysis generates questions
Model generates scenarios
Perspective-Based Reading

Perspective-Based Reading

- Goal for each participant:
  - **Analyze** the documents provided
  - **for the purpose of** evaluation
  - **with respect to** defect detection
  - **from the point of view of** designer (or tester, user, …)
  - **in the context of** a specific technique for creating system designs (or test plans, user manuals, …)

- Results:
  - defect report forms
  - a high-level model of the system

Example using Use Case Procedure

- **As an example in these notes we will use excerpts from a requirements document which describes a system for running a video store. Here are some excerpts from the general description of this system:**
- **"**Customers select at least one video for rental. The maximal number of tapes that a customer can have outstanding on rental is 20. The customer's account number is entered to retrieve customer data and create an order. Each customer gets an ID card from ABC for identification purposes. This ID card has a bar code that can be read with the bar code reader. Bar code IDs for each tape are entered and video information from inventory is displayed. The video inventory file is updated. When all tape IDs are entered, the system computes the total bill. Money is collected and the amount is entered into the system. Change is computed and displayed. The rental transaction is created, printed and stored. The customer signs the rental form, takes the tape(s) and leaves. "

Example using Use Case Procedure

- Here is a brief description of the users of this system:
- "**User Characteristics:** The system will be used by ABC Video management, clerks, and indirectly by customers. From the point of view of the system, clerks and managers are identical. Some system operations are only accessible to managers (such as printing daily reports) and are protected by password. . ."

User Perspective

- Focuses on creating **use cases**, a means of documenting requirements for ease of communication with the customer

- Terminology:
  - **participant:** any external system or user that interacts with the system described in the requirements
  - **actor:** classes of users which are external to the system and interact with it in characteristic, non-deterministic ways
  - **use case**: a scenario that describes the set of events that results when an actor uses a particular system functionality

User Perspective

- **1. Identify the participants in the requirements, and decide which are actors.**
  - Are multiple terms used to describe the same participant in the requirements?
  - Is the general description of how the system interacts with a participant inconsistent with the general requirements? Are the requirements unclear or inconsistent about this interaction?
  - Have necessary participants been omitted? That is, does the system need to interact with another system, piece of hardware, or type of user which is not described?
  - Is an external system or a class of “users” described in the requirements which does not actually interact with the system?

Form A - Participants and use cases in new requirements

Reviewer Name:

Document Reviewed:

Participants	Actor?	Act in which use cases?
1. **clerks**	**X**	|___|___|___|___|
2. **managers**	**X**	|___|___|___|___|
3. **customers**		|___|___|___|___|
4. **bar code reader**		|___|___|___|___|
5. **cash drawer**		|___|___|___|___|
6.		
7.		
8.		
9.		
10.		
11.		|___|___|___|___|
12.		|___|___|___|___|
13.		|___|___|___|___|
14.		|___|___|___|___|

Note that we are here listing all of the users and other systems with which the system we are building will interact. The guidelines on the handout and the class lecture should help you determine which are the actors. Note that 'customers' should probably not be considered actors (actually, it is questionable whether they should even be included as participants) because they never interact with the system directly - only indirectly, through clerks and managers.

Use Cases
1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.

User Perspective

- **2. Identify product functions in the requirements, and use them to construct use cases.**
  - Are the start conditions for each use case specified at an appropriate level of detail?
  - Has any functionality that should appear in a use case been omitted?
  - Has the functionality been described sufficiently so that you understand how the low-level functionalities are related to each other? Does the description allow more than one interpretation as to how the functionalities are related?
  - Have necessary functionalities been omitted, according to your domain knowledge or the general description?

User Example

- **You may find it easiest just to jot down low-level functionalities as you read through the requirements, then group related functionalities into use cases. (see next slide for an example)**
- **Then on Form B, diagram how the functionalities are ordered.**

User Example - System Functionalities

System displays main menu

Clerk selects ‘return videos’ option

Clerk scans tape bar code

System retrieves video records

System marks video records with return date

System computes any money owed

Clerk collects money and enters amount

Functionalities concerned with returning tapes

Clerk selects ‘new customer’ option

Clerk enters necessary customer info

System creates account and assigns ID

System prints account bar code

Clerk glues bar code to card

Clerk gives card to customer

Functionalities concerned with adding new customers

Manager selects ‘deletion’ option

Manager enters account number of video or customer

System deletes appropriate account

System displays whether operation successful

. . .

Functionalities concerned with deletion

It’s usually a little bit harder than this to see how to group related functionalities, but this should give you some idea of the goal of this step. The point is to identify tasks that can be accomplished using the system, and all the steps that are involved in completing those tasks.

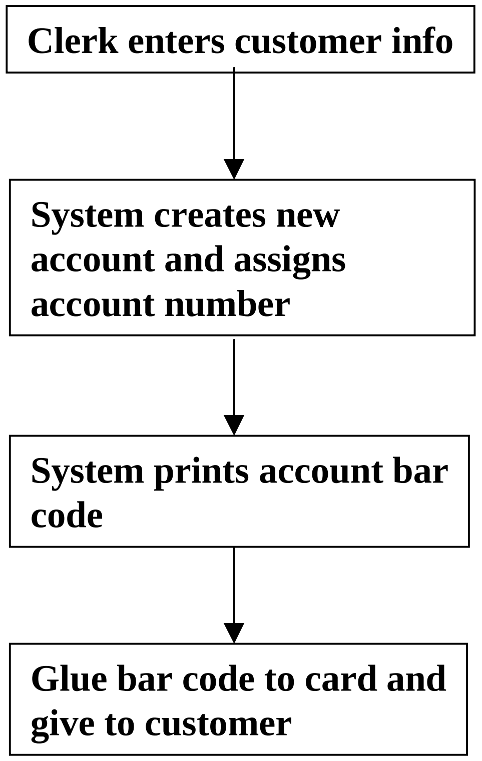
Form B - Use Case

Reviewer Name: Document Reviewed:

Use Case: **Add new customer**

Actor: **clerk**

Start: **select 'new customer' option from main menu**

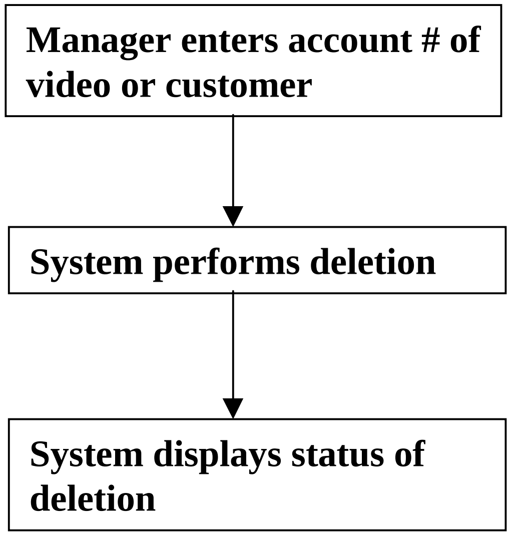

Clerk enters customer info
System creates new account and assigns account number
System prints account bar code
Glue bar code to card and give to customer

This is a really simple example, but it should give you the idea of how to specify the sequence of functionality. Usually, the diagram may be a little more complicated, with the possibility of branches and exception conditions.

Form B - Use Case

Reviewer Name:	Document Reviewed:

Use Case:	**Deletion**
Actor:	**manager**
Start:	**select ‘deletion’ from main menu**


Manager enters account # of video or customer
System performs deletion
System displays status of deletion

Another simple, contrived example that shows the type of errors that might be found using this procedure. Q2.2 asks you to look over the use cases as you build them, looking for omissions - in this case, you should remember from the general requirements that there should be some functionality that restricts deletion to managers only. Since there was no low-level functionality to do this described in the document, you can see that it is missing from this use case.

User Perspective

- **3. Match the actors to all of the use cases in which they participate.**
  - Is it clear from the requirements which actors are involved in which use cases?
  - Based on the general requirements and your knowledge of the domain, have all of the necessary use cases been specified for each actor?
  - Are use cases described for an actor which are incompatible with the actor’s description in the general requirements?
  - Have necessary actors been omitted (that is, are there use cases for which no actor is specified)?

Form A - Participants and use cases in new requirements

Reviewer Name:

Document Reviewed:

	Participants	Actor?	Act in which use cases?			
1.	**clerks**	**X**	**1**	**2**		
2.	**managers**	**X**	**3**			
3.	**customers**					
4.	**bar code reader**					
5.	**cash drawer**					
6.						
7.						
8.						
9.						
10.						
11.						
12.						
13.						
14.						

	Use Cases
1.	**Return tapes**
2.	**Add new customer**
3.	**Deletion**
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
12.	
13.	
14.	