

Use Cases

Steven Zeil

October 5, 2013

Contents

1	Scenarios	2
2	Use Cases	3
2.1	Evolving a Collection of Use cases	4
3	Organizing Use Cases	12
3.1	Includes	12
3.2	Generalization	14
3.3	Extends	16
4	How to Use a Use Case	17

1 Scenarios

Scenarios

Most OOAD is scenario-driven.

- Scenarios are "stories" that describe interactions between a system and its users.
- Usually in informal natural language
 - Often with many variants on a common theme to accommodate errors, exceptional cases, alternative processing, etc.
- Gathered (during Elaboration) from documents and interviews with domain experts

.....

Example: Assessment Scenarios

1. An instructor prepares three new questions for an upcoming test. One is an essay question, one is a multiple-choice question, and the third calls for students to perform a physical experiment to submit tabulated data from that experiment. The instructor proceeds to compose a test from these three questions plus an existing question from his personal store.
2. A student wishes to attempt a test. She approaches the instructor who provides a copy of the test on which the student is supposed to write her answers.. The instructor authorizes the student to begin the assessment session.
3. Same as #2, but the student is given a copy of the questions and a separate form on which to record her responses.
4. Same as #3, but the student brings a set of blue books that are approved by the instructor as a separate form on which to record her responses.
5. A student wishes to attempt a test. He approaches the instructor, but the instructor will not authorize the assessment session because the test is not available until next day.
6. A student begins an assessment session. For each question on the test, the student reads the question and takes note of the expected response types. The first question is an essay question. The student prepares a few paragraphs of text as her response. The second question is a multiple-choice question. The student selects one choice as her response. The third question calls for the student to conduct an activity outside of the current session, so the student suspends the assessment session, with the intention of resuming it after the external activity has been completed.

7. An instructor begins to process the responses of a collection of students. All responses are for the same assessment. For each question in that assessment, the instructor obtains the question's rubric. Then for each response, the instructor obtains the response for that question, applies the rubric, and records the score as part of the assessment result. Once all questions have been scored, the instructor computes and records the assessment score in the assessment result.
8. An instructor is processing an question response to an essay question. The rubric consists of a series of concepts that could be considered valid, coupled with a score value for each. The instructor reads the response, keeping a running total of the value of all rubric concepts recognized. That total is compared to the question's normal minimum and normal maximum and adjusted, if necessary, to fall within those bounds. The adjusted total is recorded as the question score for that question response.

If the question score is less than the normal max, then the instructor adds a list of the rubric concepts not contained in the response as feedback within the question outcome.

.....

2 Use Cases

Use Cases

A *use case* is

- a collection of scenarios
- related to a common user goal
- expressed in step-by-step form

.....

Where do use cases come from?

Typically these arise from analysis of documents provided to the analysis team or from interviews conducted with the domain experts. (This might be a good time to review the [different stages of analysis and modeling](#) (7) that we have discussed.)

.....

What are use cases good for?

What do we do with use-cases once we have them?

- Use them to guide analysis and design
 - Validate models by walking through use cases to see if model supports the scenario

- Sometimes used as requirements statements

.....

2.1 Evolving a Collection of Use cases

Evolving a Collection of Use cases

1. Start with a collection of scenarios
2. Group by common user goals
3. Define the Actors
4. Give brief descriptions of use cases
5. Give a detailed description of the basic path
6. Add alternative paths

.....

Start with a collection of scenarios

The informal scenarios are your starting point.

.....

Example 1. Assessment Scenarios

- Consultation with domain experts has revealed a standard set of terminology (QTI) related to assessments, particularly automated assessments. Accordingly we replace certain informal terms by their more domain-appropriate equivalents ("question" ⇒ "item", "student" ⇒ "candidate") and the role of "instructor" is split into distinct classes of "author", "proctor", and "scorer".
1. An author prepares three new items for an upcoming test. One is an essay question, one is a multiple-choice question, and the third calls for students to perform a physical experiment to submit tabulated data from that experiment. These items become part of that author's personal item bank. The author proceeds to compose a test from these three items plus an existing item (a fill-in-the-blank question) from the item bank.
 2. A candidate wishes to attempt a test. She approaches the proctor who provides a copy of the test on which the candidate is supposed to write her answers.. The proctor authorizes the candidate to begin the assessment session.

3. Same as #2, but the candidate is given a copy of the questions and a separate form on which to record her responses.
 4. Same as #3, but the candidate brings a set of blue books that are approved by the proctor as a separate form on which to record her responses.
 5. A candidate wishes to attempt a test. He approaches the proctor, but the proctor will not authorize the assessment session because the test is not available until next day.
 6. A candidate attempts a test. She approaches the proctor who provides a copy of the test and a (possibly separate) empty response form. The proctor authorizes the student to begin the assessment session. For each item on the test, the candidate reads the item body and takes note of the item variables. The first item is an essay question. The candidate prepares a few paragraphs of text as her item response. The second item is a multiple-choice question. The student selects one choice as her item response. The third question calls for the candidate to conduct an activity outside of the current session, so the student suspends the assessment session, with the intention of resuming it after the external activity has been completed.
 7. A scorer begins to process the responses of a collection of candidates. All responses are for the same assessment. For each item in that assessment, the scorer obtains the item's rubric. Then for each response, the scorer obtains the item response for that item, applies the rubric, and records the item outcome as part of the assessment result. Once all items have been scored, the scorer computes and records the assessment outcome in the assessment result.
 8. A scorer is processing an item response to an essay question. The rubric consists of a series of concepts that could be considered valid, coupled with a score value for each. The scorer reads the response, keeping a running total of the value of all rubric concepts recognized. That total is compared to the item's normal minimum and normal maximum and adjusted, if necessary, to fall within those bounds. The adjusted total is recorded as the item outcome for that item response.

If the item outcome is less than the normal max, then the scorer adds a list of the rubric concepts not contained in the response as feedback within the item outcome.
- *CRC cards and class relationship diagrams should also be updated accordingly.*

Group by common user goals

A use case describes a collection of scenarios related by common use goals.

Use Cases

- Note that a given scenario might lack a clear goal or might span multiple goals.

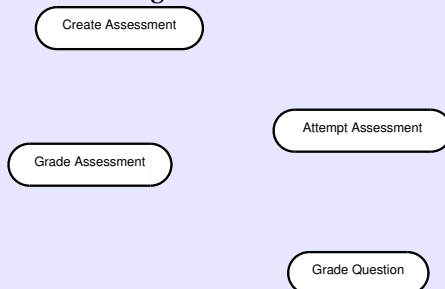
.....

Group by common user goals

Example 2. Assessment User Goals

- Create an assessment
- Attempting an assessment
- Grading an assessment
- Grading a question

We can summarize these with the initial stage of a UML *Use case Diagram*:



.....

Define the Actors

An *actor* (in this context) is a role that a user plays in this system.

An actor is usually an entity that behaves spontaneously to initiate an interaction.

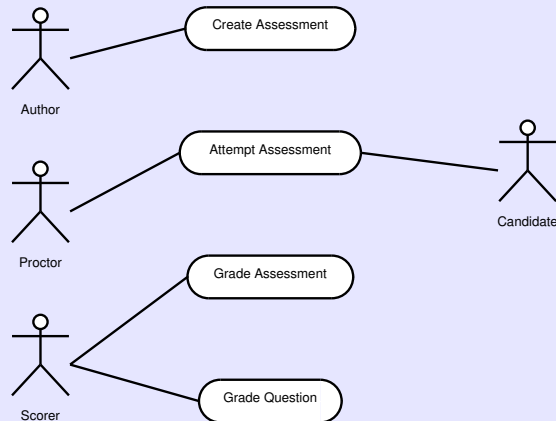
- Often include the major users and stakeholders in the system
- Relating use cases to actors helps identify the domain experts who need to be consulted on each case to obtain details, validate for accuracy, etc.

Later, these relations help characterize user interfaces.

.....

Defining the Actors

Example 3. Assessment Actors



The connections between actors and use cases indicate that the actor participates in the use case.

.....

Give brief descriptions of use cases

For each of the use cases, describe briefly what goes on in this case.

You can pull heavily on the original text of the scenarios, but generalize to pull in the identified actors. You may abstract some details.

.....

Example 4. Assessment UC Brief Descriptions

Create Assessment An author creates a new assessment, using a variety of different types of items (questions). Some of these items are created specifically for this assessment. Others are pulled from the author's item bank.

Attempt Assessment A candidate attempts an assessment. The proctor provides a copy of the assessment and a (possibly separate) empty response document. The proctor authorizes the student to begin the assessment session and can end the session when the candidate is finished or upon expiration of a time limit. During the session, the candidate prepares responses to items in the assessment. At the end of the session, the student returns the response document to the proctor.

Grade Assessment A scorer begins to process the response documents of a collection of candidates. All response documents are for the same assessment. The scorer may opt to process responses on an item by item basis (grading all student responses to a single item) or on a candidate by candidate basis (grading all item responses for a given candidate). Once all items have been scored, the scorer computes and records the assessment outcome in the

assessment result.

Grade Item A scorer begins to grade an item response within a response document to an assessment. The scorer obtains the item's rubric for that assessment, applies that rubric to the item response, and records the resulting score and, optionally, feedback as part of the assessment result.

Give a detailed description of the basic path

The basic path through your use case is the normal, everything-is-going-right, processing.

- This is expressed as a numbered sequence of steps.
- Eventually you want enough detail to model this as a set of messages exchanged between objects, but you may need multiple iterations to get there.

.....

Example 5. Assessment UC Basic Paths

Create Assessment

1. Author creates items of varying kinds, adding each to his/her item bank.
2. Author creates an empty assessment, adding a title and indicating the grading policy (graded immediately or in batch, grades recorded in grade book or not), policy on re-attempts permitted, dates & time of availability, and security policy (suspend & resume permitted, open/closed book, etc.).
3. Author selects questions from item bank and adds them to the assessment, assigning point value as he/she does so.
4. Author concludes creation of the assessment, making it available to the appropriate proctors.

Attempt Assessment

1. A candidate indicates to a proctor their desire to take an assessment. (This may involve simply showing up at a previously-schedule time and place when the assessment is going to be available.)
2. The proctor provides the candidate with a copy of the assessment and an empty response document.
3. The proctor authorizes the start of the assessment session.

4. The candidate reads the assessment, and repeatedly selects items from it. For each selected item, the candidate creates a response, adding it to the response document.
5. The proctor ends the assessment session after time has expired.
6. The candidate returns the response document to the proctor.

Grade Assessment

1. The scorer begins with an assessment and a collection of response documents.
2. For each item in the assessment, the scorer obtains the item's rubric. Then for each response document, the scorer goes to the item response for that same item, grades the response using that rubric, and adds the resulting core and (if provided) feedback to the result document for that response document.
3. When all items have been graded, then the scorer computes a total score for each results document.
4. The scorer add the score from each result document to the grade book.

Grade Item

Given an assessment item, its rubric, and an item response,

1. The scorer applies the rubric to the item response to obtain a raw score and, possibly, feedback text for the item.
2. The raw score is scaled to match the point value in the rubric to the point value of the item in the assessment. If need be, the scaled score is subjected to the minimum/maximum point constraints on the item.
3. The results is returned as the result score, together with any feedback.

Add alternative paths

A use case is a *collection* of scenarios. The basic path constitutes a single scenario.

- The basic path constitutes a single scenario.
 - not necessarily one of the original scenarios
- Additional scenarios are introduced as alternative paths

.....

Example 6. Assessment UC Alternate Paths Create Assessment

1. Author creates items of varying kinds, adding each to his/her item bank.
2. Author creates an empty assessment, adding a title and indicating the grading policy (graded immediately or in batch, grades recorded in grade book or not), policy on re-attempts permitted, dates & time of availability, and security policy (suspend & resume permitted, open/closed book, etc.).
3. Author selects questions from item bank and adds them to the assessment, assigning point value as he/she does so.
4. Author concludes creation of the assessment, making it available to the appropriate proctors.

Alternative: *Edit assessment*

1. Author edits an existing assessment, possibly altering one or more of the assessment properties.



Alternative: *Save assessment*

1. Author concludes this work session, saving the assessment for later editing without releasing it to the proctors.

Attempt Assessment

1. A candidate indicates to a proctor their desire to take an assessment. (This may involve simply showing up at a previously-schedule time and place when the assessment is going to be available.)
2. The proctor provides the candidate with a copy of the assessment and an empty response document.
3. The proctor authorizes the start of the assessment session.
4. The candidate reads the assessment, and repeatedly selects items from it. For each selected item, the candidate creates a response, adding it to the response document.
5. The proctor ends the assessment session after time has expired.
6. The candidate returns the response document to the proctor.

Alternative: *Unavailable*

- CS330 11  
1. The proctor determines that the candidate is not eligible to take the assessment at this time (the assessment is not available or the candidate has not fulfilled assessment requirements such as being enrolled in the course).
The use case terminates immediately.

Alternative: *Variant Assessments*

3 Organizing Use Cases

Organizing Use Cases

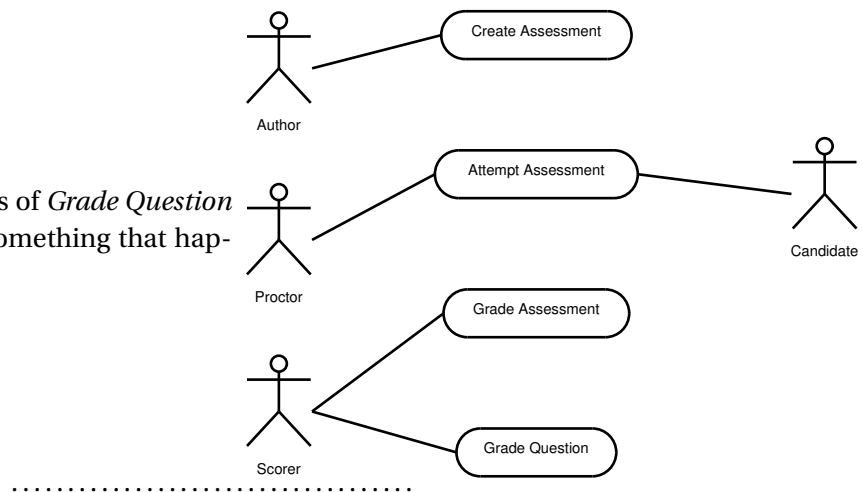
- A 10 person-year project may typically have a dozen use cases (Fowler) with lots of alternative paths.
- In some cases, it's preferred to break some of the alternative paths to have simpler, related use cases.
 - UML recognizes 3 relations between use cases.
- The collection of use cases, actors, and the relations among them constitutes our *use case model*.

.....

3.1 Includes

One Use Case Invoking Another

In our current model, the steps of *Grade Question* are actually a description of something that happens inside *Grade Assessment*.



.....

Example

Grade Assessment

1. The scorer begins with an assessment and a collection of response documents.
2. For each item in the assessment, the scorer obtains the item's rubric. Then for each response document, the scorer goes to the item response for that same item, **grades the response using that rubric**, and adds the resulting score and (if provided) feedback to the result document for that response document.

3. When all items have been graded, then the scorer computes a total score for each results document.
4. The scorer add the score from each result document to the grade book.

Alternative: *Candidate by Candidate Scoring*

1. For each candidate, the scorer goes through each of the items. For each item, the scorer obtains the item's rubric, grades the item response using that rubric , adds the resulting score and (if provided) feedback to the result document for that response document.

Grade Item

Given an assessment item, its rubric, and an item response,

1. The scorer applies the rubric to the item response to obtain a raw score and, possibly, feedback text for the item.
2. The raw score is scaled to match the point value in the rubric to the point value of the item in the assessment. If need be, the scaled score is subjected to the minimum/maximum point constraints on the item.
3. The results is returned as the result score, together with any feedback.

Alternative: *Essay Question Scoring*

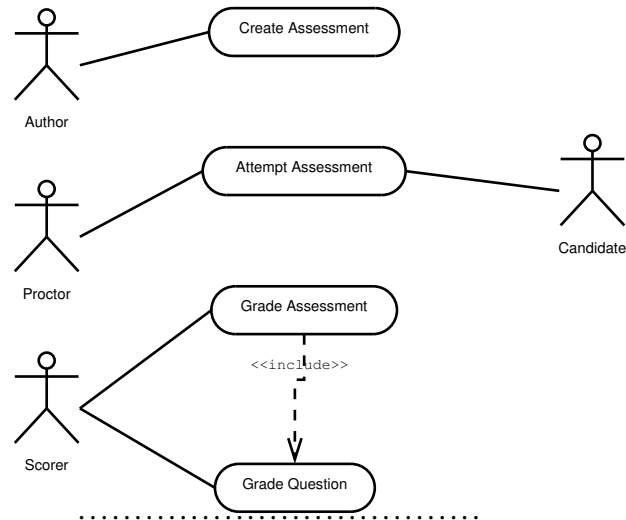
1. The item is an essay question.

The rubric consists of a series of concepts that could be considered valid, coupled with a score value for each.

- (a) The scorer reads the response, keeping a running total of the scores of all rubric concepts recognized.

The Includes Relation

This is an example of an *includes* relation between use cases.



3.2 Generalization

The Generalization Relationship

The *generalization* relation between use cases indicates that one use case meets the same user goal as the other but overrides one or more steps of the procedure for meeting that goal.

- For example, we will shortly look at sequence diagrams as a way of mapping our analysis & design models onto use cases.
 - It would be hard to document the two paths in the following use case in that fashion:

.....

Example

Grade Assessment

- The scorer begins with an assessment and a collection of response documents.
- For each item in the assessment, the scorer obtains the item's rubric. Then for each response document, the scorer goes to the item response for that same item, grades the response using that rubric, and adds the resulting score and (if provided) feedback to the result document for that response document.
- When all items have been graded, then the scorer computes a total score for each results document.
- The scorer add the score from each result document to the grade book.

Alternative: *Candidate by Candidate Scoring*

1. For each candidate, the scorer goes through each of the items. For each item, the scorer obtains the item's rubric, grades the item response using that rubric, adds the resulting score and (if provided) feedback to the result document for that response document.

Splitting a Use Case

So we might choose to break this into two distinct cases:

.....

Example

Grade Assessment

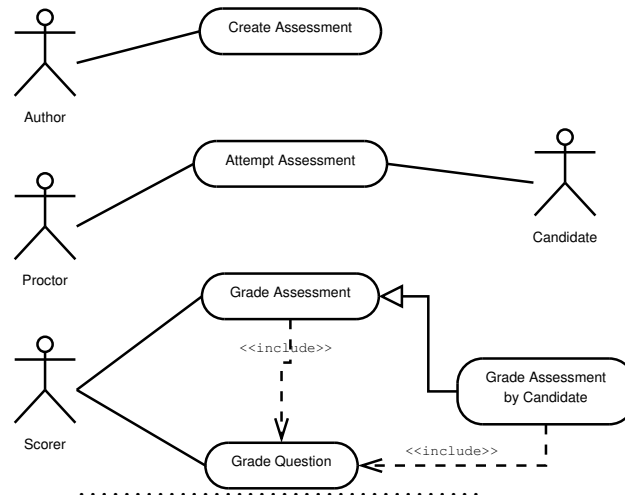
1. The scorer begins with an assessment and a collection of response documents.
2. For each item in the assessment, the scorer obtains the item's rubric. Then for each response document, the scorer goes to the item response for that same item, grades the response using that rubric, and adds the resulting score and (if provided) feedback to the result document for that response document.
3. When all items have been graded, then the scorer computes a total score for each results document.
4. The scorer add the score from each result document to the grade book.

Grade Assessment by Candidate

1. The scorer begins with an assessment and a collection of response documents.
2. For each candidate, the scorer goes through each of the items. For each item, the scorer obtains the item's rubric, grades the item response using that rubric, adds the resulting score and (if provided) feedback to the result document for that response document.
3. When all items have been graded, then the scorer computes a total score for each results document.
4. The scorer adds the score from each result document to the grade book.

Generalization

The relation between these two use cases is shown like this:



3.3 Extends

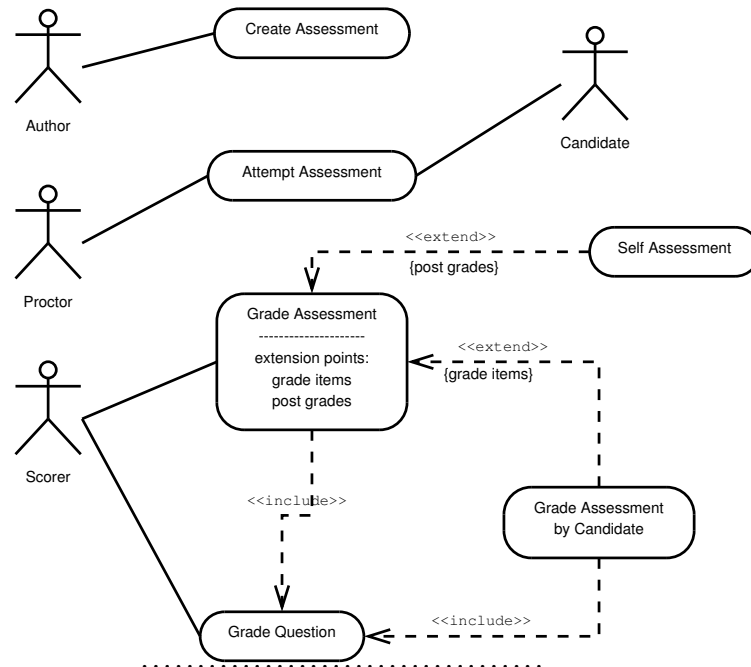
Extends

A more "disciplined" version of generalization.

- The base use case specifies specific *extension points* at which other use cases may override steps.
- The derived use cases announce which extension points they are going to override.

.....

Extends



4 How to Use a Use Case

How to Use a Use Case

- As an aid to analysis: read through the case, looking for objects and interactions not listed in the current model.
- As an aid to validation: Model the use case as a series of passed messages, recording the interactions in a *sequence diagram*. Update the model as necessary so that a cause-and-effect chain can be demonstrated from the start of the use case to its end.
- As a statement of requirements: some companies now use use-cases in place of older forms of requirements statements