

Example: Domain Model Using CRC Cards

Steven Zeil

September 11, 2013

Contents

1	Problem Statement	2
2	Identifying Candidate Classes and Responsibilities	3
3	Assign Responsibilities to Classes	4
4	Probable Inheritance Hierarchies	4
5	Fill in the Candidate Responsibilities	5
6	The Story So Far	11
6.1	That's Far Enough for Now	12

Recording

These slides accompany a recorded video: *Play Video*

.....

Example of Early Analysis

This lesson works through an example of the early stages of analysis.

- Available documentation is natural language, fairly general in nature
 - Natural language is always tricky to work with. Ambiguities and contradictions are common.
 - You must read carefully and critically.
- We're working on a domain model
 - But we don't yet have enough info for a complete model
- What do we hope to accomplish?
 - Learn as much as possible from the info provided material
 - Reveal questions for later, more detailed follow-up

- Mistakes will be made!

One of my pet peeves about reading how to do analysis and design in textbooks is that they always make the right decisions at each step.

It's important to realize that designers do make mistakes and need to back up and reconsider things. (That's why we have the "V" in the ADIV workflow!)

So I try to be honest and record my analysis examples as a stream-of-consciousness of what I actually went through considering the problem for the first time, including the mistakes.

.....

1 Problem Statement

Problem Statement

ODU offers a number of courses via the internet. A common requirement among these courses is for a system of online assessment. An assessment is any form of graded question-and-answer activity. Examples include exams, quizzes, exercises, and self-assessments. In preparation for automating such a system, our group has undertaken a study of assessment techniques in traditional classrooms.

An assessment can contain a number of questions. Questions come in many forms, including true/-false, single-choice from among multiple alternatives, multiple choices, fill-in-the-blank, and essay. There may be other forms as well.

Example: Domain Model Using CRC Cards

Students take assessments that are administered by instructors. The students' responses to each question are collected by the instructor, who grades them by comparison to a rubric for each question. The instructor may also elect to provide feedback (written comments), particularly about incorrect responses.

A total score for the assessment is computed by the instructor. If this is a self-assessment, the score is for informational purposes only. For other kinds of assessments, the instructor records the score in his/her grade book.

Information is returned to the student about their performance. At a minimum, the student would learn of their score and any instructor-provided feedback. Depending upon the instructor, students may also receive the questions, a copy of their own responses, and the instructor's correct answer.

.....

2 Identifying Candidate Classes and Responsibilities

Identifying Candidate Classes and Responsibilities

For the initial list, mark up the description, looking for *noun phrases* and *verb phrases*.

ODU offers a number of courses via the Internet. A common requirement among these courses is for a system of on-line assessment. An *assessment* is any form of graded question-and-answer activity. Examples include *exams*, *quizzes*, *exercises*, and *self-assessments*. In preparation for automating such a system, our group has undertaken a study of assessment techniques in traditional classrooms.

An assessment can *contain* a number of *questions*. Questions come in many forms, including *true/false*, *single-choice* from among multiple alternatives, *multiple choices*, *fill-in-the-blank*, and *essay*. There may be other forms as well.

Students *take* assessments that are *administered* by *instructors*. The students' *responses* to each question are *collected* by the *instructor*, who *grades* them by comparison to a *rubric* for each question. The instructor may also elect to *provide feedback* (written comments), particularly about incorrect responses.

A total *score* for the assessment is *computed* by the instructor. If this is a self-assessment, the score is for informational purposes only. For other kinds of assessments, the instructor *records* the score in his/her *grade book*.

Information is *returned* to the student about their *performance*. At a minimum, the student would learn of their score and any instructor-provided feedback. Depending upon the instructor, students may also receive the questions, a copy of their own responses, and the *instructor's correct answer*.

.....

Candidate Classes

assessment, exams, quizzes, exercises, self-assessments, questions, true/false question, single-choice

Example: Domain Model Using CRC Cards

question, multiple choices question, fill-in-the-blank question, essay question, students, instructors, responses, rubric, feedback, score, grade book, information, performance, instructor's answer

.....

Candidate Responsibilities

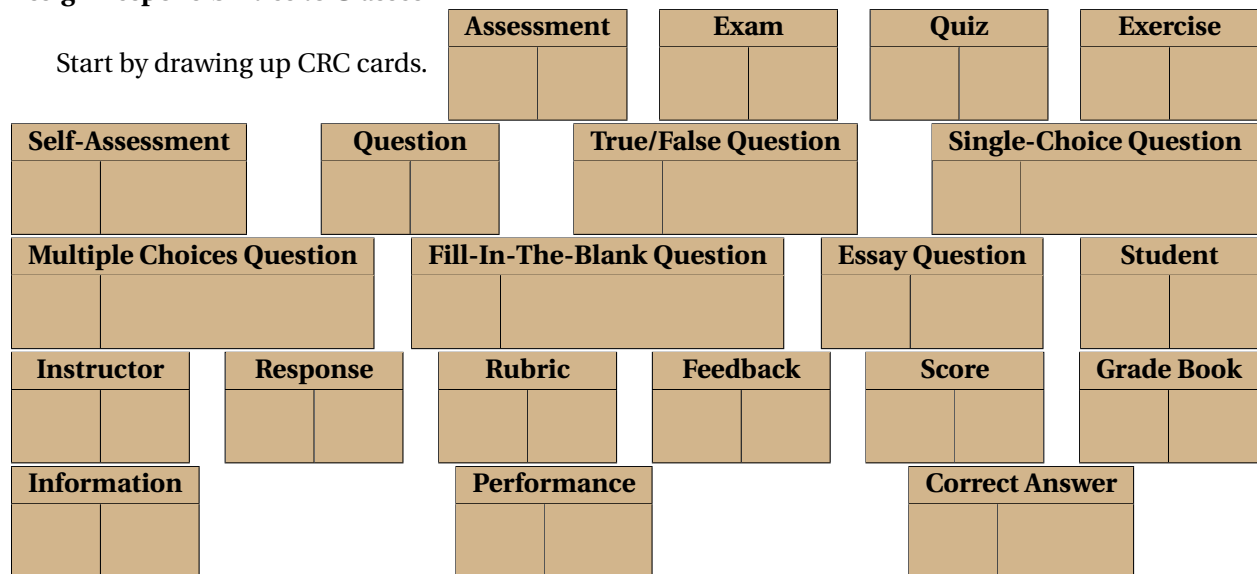
contain (questions), take (assessment), administer, collect (responses), grade, provide (feedback), compute (score), record (score), return (information)

.....

3 Assign Responsibilities to Classes

Assign Responsibilities to Classes

Start by drawing up CRC cards.



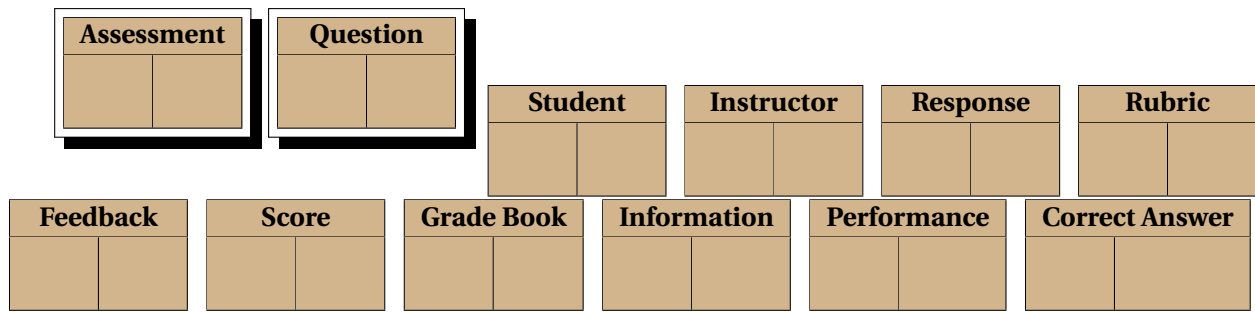
4 Probable Inheritance Hierarchies

Probable Inheritance Hierarchies

Since all of the various kinds of assessments are likely to have similar responsibilities and collaborators, let's stack their cards for now and treat them as a unit.

We'll do the same with the various kinds of questions.

Example: Domain Model Using CRC Cards



5 Fill in the Candidate Responsibilities

Fill in the Candidate Responsibilities

Now fill in the operations known so far:

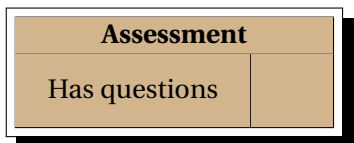
- contain (questions)
- take (assessment),
- administer,
- collect (responses),
- grade,
- provide (feedback),
- compute (score),
- record (score),
- return (information),

.....

contain questions

An assessment can contain a number of *questions*.

- This is really a statement about attributes of an assessment



.....

taking and administering assessments

Students take assessments that are administered by *instructors*.

- Is this really two separate operations?

The language (plurals) is a bit tricky.

- Instructors administer an assessment to an entire class.
- Each student individually takes the assessment.

Instructor	
administer assessment to group of Students	

.....

Taking or Administering?

- Surprised that I put that in Instructor?

Remember the basic rule: if A does B to C, then "do B" is usually a responsibility of C

- It would not be a responsibility of the assessment
 - Tests don't administer themselves ITRW
- Could it be a responsibility of the Student?
 - No, the statement says that students "take" assessments,
 - But is "take assessment" simply a synonym for "accept administration of an assessment"?

.....

What's involved in administering an assessment?

The problem statement tells us:

Students take assessments that are administered by instructors. The students' responses to each question are collected by the instructor, who grades them ... The instructor may also elect to provide feedback (written comments), particularly about incorrect responses. A total score for the assessment is computed by the instructor... Information is returned to the student about their performance.

We're

looking at the instructor's *method* for administering an assessment.

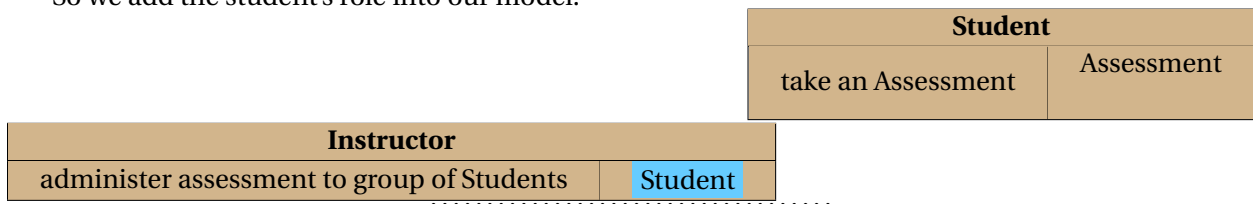
Example: Domain Model Using CRC Cards

- a strong suggestion that administration of an assessment is far more involved than simply having a student “take” it.
 - and that they are, therefore, separate responsibilities.

.....

taking an assessment

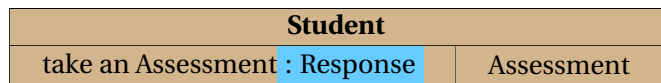
So we add the student’s role into our model:



collecting responses

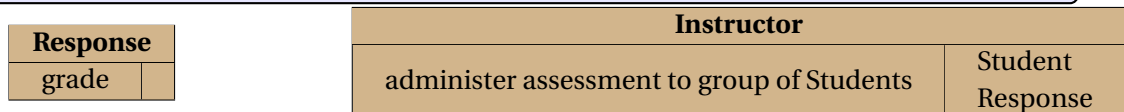
The students’ *responses* to each question are collected by the *instructor*

- This is really just describing the output from the request sent to students asking them to take the assessment.



grading responses

the *instructor*, who grades them by comparison to a *rubric* for each question.



- not a responsibility of the instructor, because we are still tracing out the steps that constitute the instructor’s method for administering an assignment

.....

Example: Domain Model Using CRC Cards

Working with Rubrics

We are told there is a separate rubric for each question. So the “comparison” is between a response to a single question and a rubric.

- This highlights the distinction between the response to an assessment and the responses to individual questions.
 - We don’t know what the proper terminology here would be, so we use a placeholder and make a note to consult the domain experts.

Response	
has QuestionResponses? grade all question responses via a seq of Rubrics	Rubric

QuestionResponse?
grade(Rubric): score

.....

Wait a minute...

Response	
has QuestionResponses? grade all question responses via a seq of Rubrics	Rubric

QuestionResponse?	
grade(Rubric): score	

At this point, sanity reasserts itself

- ITRW, when a student returns an exam sheet or a bluebook (the Response), those things don’t grade themselves.
 - They’re just paper
 - And while anthropomorphism is common in OO modeling, that may be going a little too far.

.....

What are our options?

Response	
has QuestionResponses? grade a seq of Questions via a seq of Rubrics	Rubric

QuestionResponse?	
grade(Rubric): score	

- First thought: the instructor does the grading
 - but it’s not really a responsibility of the instructor because no one in this model tells the instructor to take this step
 - it’s part of the Instructor’s “administer an assessment” method

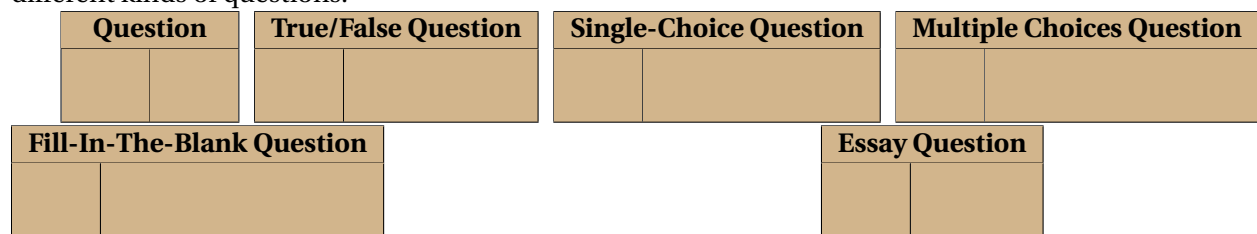
Example: Domain Model Using CRC Cards

- But not mentioning it all in the model seems wrong
 - So let's think about what we want to eventually capture

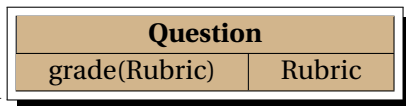
.....

Looking for Variant Behavior

One reason that we really want to model the grading process is that we know that we have many different kinds of questions:



and we suspect that the grading method varies from one type of question to another.



- So perhaps we should say

.....

But I Don't Like That Either

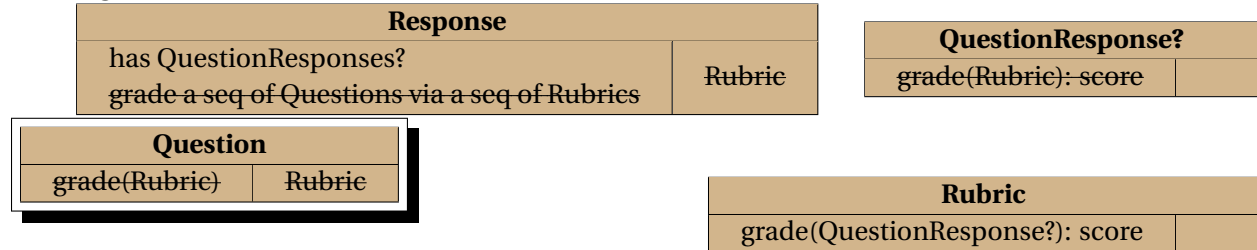
...and here's why:

- Essay questions can cover all kinds of things.
 - “English” essays where grammar and sentence construction are graded
 - General essays where content is more important than form
 - Mathematical proofs
 - Code an algorithm, and so on
- Are these really the same kind of question?
 - They share lots of behaviors (e.g., they are represented the same way on the printed page, students use the same mechanism for answering them)
 - But the rubrics for grading them are very different
- So maybe it's the rubrics that capture this behavior

.....

Example: Domain Model Using CRC Cards

Grading - revised



- I'm much happier with that
- Can rubrics be “intelligent” or is this unacceptable anthropomorphism again?
 - ITRW, rubrics for essay question, in particular, are often expressed in ways that assume a human intelligence
- We're probably going to wind up with a small deck of Rubric cards for different variants.

.....

provide feedback

The instructor may also elect to provide feedback (written comments), particularly about incorrect responses.

GradedQuestionResponse?	
Has score, feedback	

- Another case of needing to consult the domain experts to find the proper name for a graded question response.

.....

computing scores

A total *score* for the assessment is computed by the instructor.

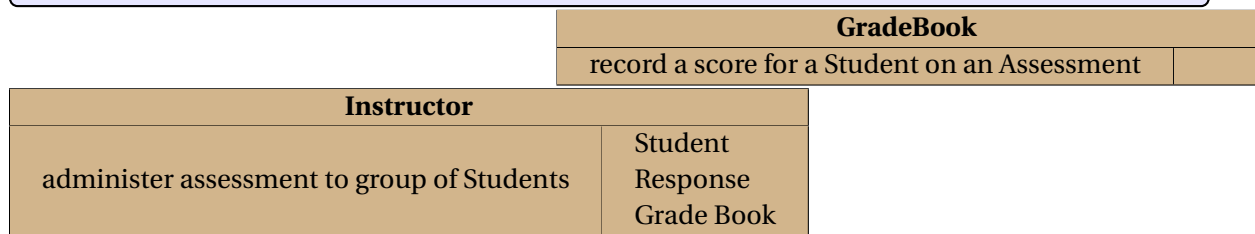
GradedResponse?	
has GradedQuestionResponses compute total score	

.....

Example: Domain Model Using CRC Cards

recording grades

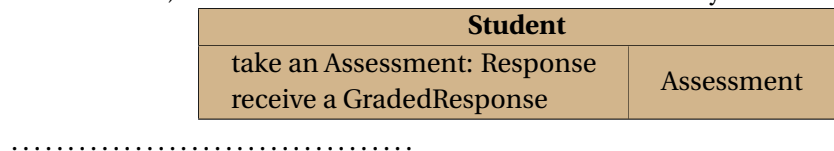
the instructor records the score in his/her *grade book*.



returning information

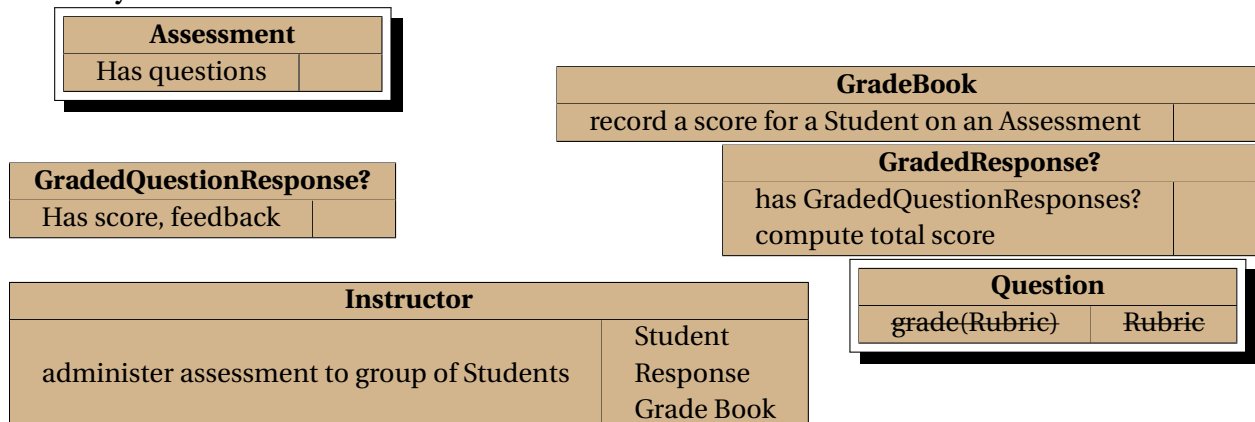
Information is returned to the student about their *performance*.

- It's a pretty good bet that we *don't* want a class with as vague a name as "Information".
 - But we've already encountered the concept under a better name
- The clue is the description: "At a minimum, the student would learn of their score and any instructor-provided feedback."



6 The Story So Far

The Story So Far



Example: Domain Model Using CRC Cards

<table> <tr><th colspan="2">QuestionResponse?</th></tr> <tr> <td>grade(Rubric): score</td><td></td></tr> </table>	QuestionResponse?		grade(Rubric): score		<table> <tr><th colspan="2">Response</th></tr> <tr> <td>has QuestionResponses?</td><td rowspan="2">Rubric</td></tr> <tr> <td>grade a seq of Questions via a seq of Rubrics</td></tr> </table>	Response		has QuestionResponses?	Rubric	grade a seq of Questions via a seq of Rubrics
QuestionResponse?										
grade(Rubric): score										
Response										
has QuestionResponses?	Rubric									
grade a seq of Questions via a seq of Rubrics										
<table> <tr><th colspan="2">Rubric</th></tr> <tr> <td>grade(QuestionResponse): score</td><td></td></tr> </table>	Rubric		grade(QuestionResponse): score		<table> <tr><th colspan="2">Student</th></tr> <tr> <td>take an Assessment: Response</td><td rowspan="2">Assessment</td></tr> <tr> <td>receive a GradedResponse</td></tr> </table>	Student		take an Assessment: Response	Assessment	receive a GradedResponse
Rubric										
grade(QuestionResponse): score										
Student										
take an Assessment: Response	Assessment									
receive a GradedResponse										

We might (cautiously) question whether some of the empty cards represent classes that we need to retain in the model.

- But we're still very early in the discovery process

.....

6.1 That's Far Enough for Now

That's Far Enough for Now

Right now, we have as many questions as answers.

- but finding useful questions is part of the process
- We can't go much further without more info
 - ... and it's very dangerous to start making stuff up based on intuition about how we think the program could work.

.....