

Homework #2

30 points

Part 1 due Wed, Feb 5, 2003

Part 2 due Wed, Feb 12, 2003

Contact: Ed Colbert (ecolbert@usc.edu)

Goals

1. Practice reading UML diagrams.
2. Compare the understanding of reviewers to the understanding of the architect.
(Reviewing a design common task you will be asked to do in industry.)

Description

This is an individual assignment and you may not collaborate with other students except as specified in the homework assignment. You may use some of the same material as other 577b students; however, you must submit work that is significantly different from theirs.

Pick a 577b project whose LCA SSAD you are going to review. Each person on a team should pick different project. If you are the architect of your 577b project, use your 577b team's SSAD. If did not take 577a last semester, use your 577b project's SSAD (so you get familiar with it). All other team members must pick a different project's SSAD.

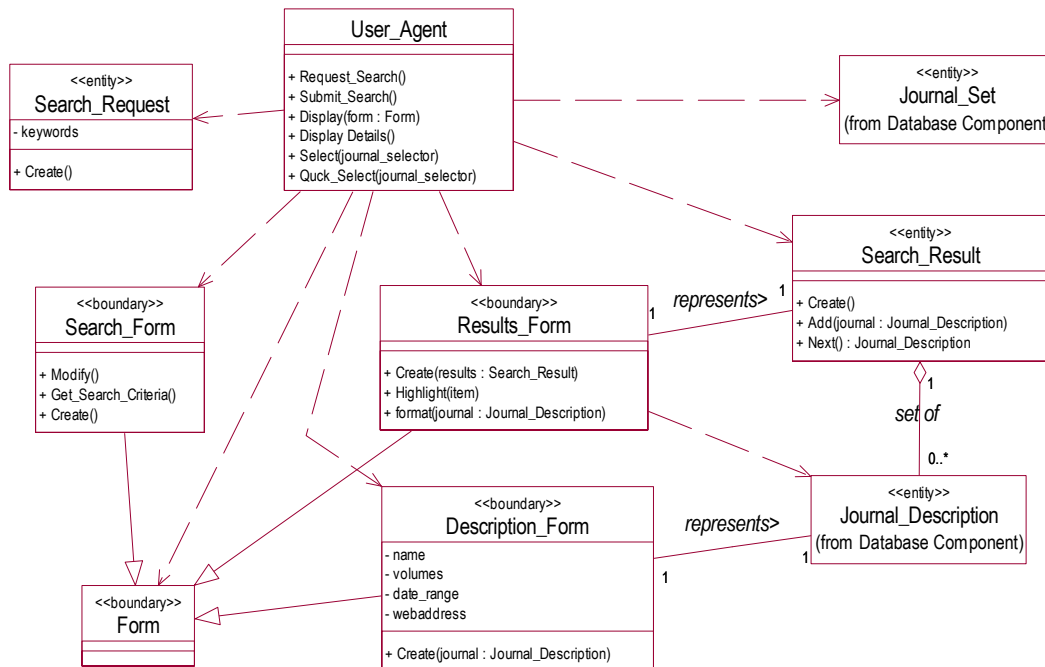
Part 1: Reading UML [20 points, Complete by February 5th]

Pick a *non-trivial* example diagram from each of the following sections of the SSAD, copy the diagram into the document that you will turn in and explain in detail what the diagrams tells (see example below). Be sure to number your statements.

1. SSAD 3.1.5 Logical Class Model
2. SSAD 3.3 Interaction Model

A non-trivial class diagram would contain 5 or more classes. A non-trivial Object-Structure Diagram would contain 5 or more objects. A non-trivial Interaction Diagram would have 3 or more objects and 3 or more messages. If you cannot find a diagram in each section that meets these criteria, then use the most complex diagram available. If no diagrams exist in a section and

- a. If you are working on your project's SSAD, then create a diagram and explain it.
- b. If you are working on another team's SSAD, then use a different project's SSAD.

Class Diagram Example:

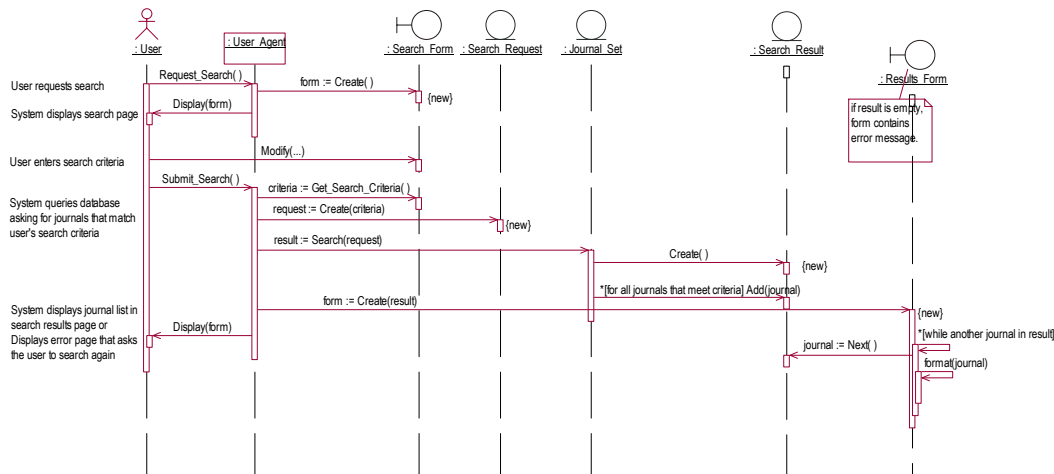
This Class Diagram tells us the following about the classes that make up the Library User Component of the Full-Text Title Database System (*Note: this is only a partial description of the diagram, showing key features that you need to describe. You **must** fully describe the diagram you are reviewing.*)

1. There is class named “User_Agent” that:
 - a. Has 6 operations
 - i. “Request” which takes no parameters and returns no result.
 - ii. “Submit_Search” which takes no parameters and returns no result.
 - iii. “Display” which takes a parameter named “form” of the type “Form” and returns no results.
 - iv. “Display_Details” which takes no parameters and returns no result.
 - v. “Select” which takes a parameter named “journal_selector” of an unnamed type, and returns no results.
 - vi. “Quick_Select” which takes a parameter named “journal_selector” of an unnamed type, and returns no results.
 - b. Depends on the entity class Search_Request;
 - c. Depends on the entity class Search_Result;
 - d. Depends on the entity class Journal_Set;
 - e. Depends on the boundary class Search_Form;
 - f. Depends on the boundary class Result_Form;
 - g. Depends on the boundary class Description_Form;

- h. Depends on the boundary class Form.
- 2. There is an entity class named "Journal_Set" that is defined in the Database component, which is used by classes in the Library User component.
- 3. There is an entity class named "Search_Request" that:
 - a. Has an attribute named "keywords" of an unnamed type;
 - b. Has an operation named "Create" which takes no parameters and returns no result.
- 4. There is an entity class named "Search_Result" that:
 - a. Has 3 operations
 - i. "Create" which takes no parameters and returns no result;
 - ii. "Add" which takes a parameter named "journal" of the type "Journal_Description", and returns no results;
 - iii. "Add" which takes no parameter, and returns a result of the type "Journal_Description".
 - b. Has an aggregate relation with the entity class "Journal_Description"
 - i. The relation is named "set of" and is read from Search_Result to Journal_Description (i.e. "Search_Result is a set of Journal_Descriptions");
 - ii. Each Search_Result instance may have 0 or more instances of Journal_Descriptions;
 - iii. Each Journal_Descriptions instance is a member of 1 instance of Search_Result. (Note: the diagram is probably in error. The relation should be "Each Journal_Descriptions instance is a member of 0 or more instances of Search_Result.")
 - c. Has an association with the boundary class "Result_Form"
 - i. The relation is named "represents" and is read from Result_Form to Search_Result (i.e. "Result_Form represents Search_Result");
 - ii. Each Search_Result instance is associated with 1 instances of Result_Form;
 - iii. Each Result_Form instance is associated with 1 instance of Search_Result.
- 5. There is a boundary class named "Result_Form" that:
 - a. Has 3 operations
 - i. "Create" which takes a parameter named "result" of the type "Search_Result", and returns no results;
 - ii. "Highlight" which takes a parameter named "item" of an unnamed type, and returns no results.
 - iii. "format" which takes a parameter named "journal" of the type "Journal_Description", and returns no results.
 - b. Has an association with the entity class "Search_Result", which is described in 5c.
 - c. Is a subclass of the boundary class named "Form".
- 6. ...

(Notice how much is said by a single diagram.)

Interaction Diagram Example:



This Interaction Diagram tells us the following about the Search for Journals behavior (*Note: this is only a partial description of the diagram, showing key features that you need to describe. You **must fully describe** the diagram you are reviewing.*)

1. There are 7 participants in this behavior.
 - a. An unnamed instance of the actor User;
 - b. An unnamed instance of the class User_Agent;
 - c. An unnamed instance of the boundary class Search_Form;
 - d. An unnamed instance of the entity class Search_Request;
 - e. An unnamed instance of the entity class Journal_Set;
 - f. An unnamed instance of the entity class Search_Result;
 - g. An unnamed instance of the boundary class Result_Form;
2. The text in the left margin is a list of the actions defined in the Use-Case Description Form. The title of the diagram (“All COAs”) indicated that this diagram shows the design for all courses of action (typical, alternate, and exceptional) described in the Use-Case Description Form.
3. The unnamed instance of the actor User starts the behavior by sending the message Request_Search with no parameters to the unnamed instance of the class User_Agent.¹
4. The unnamed instance of the class User_Agent performs its Request_Search method, which
 - a. Creates a new instance of Search_Form by calling the constructor operation “Create”;
 - b. Stores the created Search_Form in a local object named “form”;
 - c. Sends the “Display” message with the local object named “form”.

¹ This step can be described as “The unnamed instance of the actor User starts the behavior by calling the operation Request_Search of the unnamed instance of the class User_Agent with no parameters”.

5. The unnamed instance of the actor User enters search criteria by sending a “Modify” message with parameters that are not shown (they are represented by the “...”) to the unnamed instance of the boundary class Search_Form.
6. The unnamed instance of the actor User starts the search by sending a “Submit_Request” message with no parameters to the unnamed instance of the class User_Agent.
7. ...

Part 2: Analyzing UML Readings [10 points]

On 5 February, in addition to turn in your Part 1 report to the TA,

Non-architects: send a copy of your Part 1 report to the architect of the project that created the SSAD that you used, and request a copy of the architect’s report. Compare your description to that of the architect.

Architects: send a copy of your Part 1 report to each student that requests a copy. Pick one of the reports sent to you can compare with your description to that of the reviewer.

For each diagram reviewed in Part 1, make a table like that shown below.

Reviewer Statement #	Correct? (yes/no)	What’s Wrong

List reviewer statement numbers in order in the first column. In the second column, indicate whether you think the reviewer’s statement is correct or not (i.e. “yes” => correct, “no” => incorrect. If the reviewer’s statement is incorrect, explain what you think is wrong with it in the third column.

On 12 February, turn in your Part 2 report, which should include your 2 tables and the Part 1 report to which you analyzed.

Grading:

- 1) Part 1: up to 20 points
 - a. -1 point for each unexplained model element (e.g. class, object, relation, or message, attribute, operation, role, parameter, etc.).
 - b. $-\frac{1}{4}$ point for each incorrectly described model element.
 - c. -10 points for each section that you do not describe a diagram.
- 2) Part 2: up to 10 points
 - a. $-\frac{1}{2}$ point for each statement that is identified as correct (“yes” in 2nd Column), which is wrong.
 - b. $-\frac{1}{2}$ point for each statement that is identified as incorrect (“no” in 2nd Column), which is correct.
 - c. $-\frac{1}{4}$ point for each statement that is identified as incorrect (“no” in 2nd Column), which is wrong, but for a different reason than what you described.
 - d. -5 points for each table that you do not create.
 - e. -10 points for not including the report that you analyzed.