

System Analysis and Design IT3120E

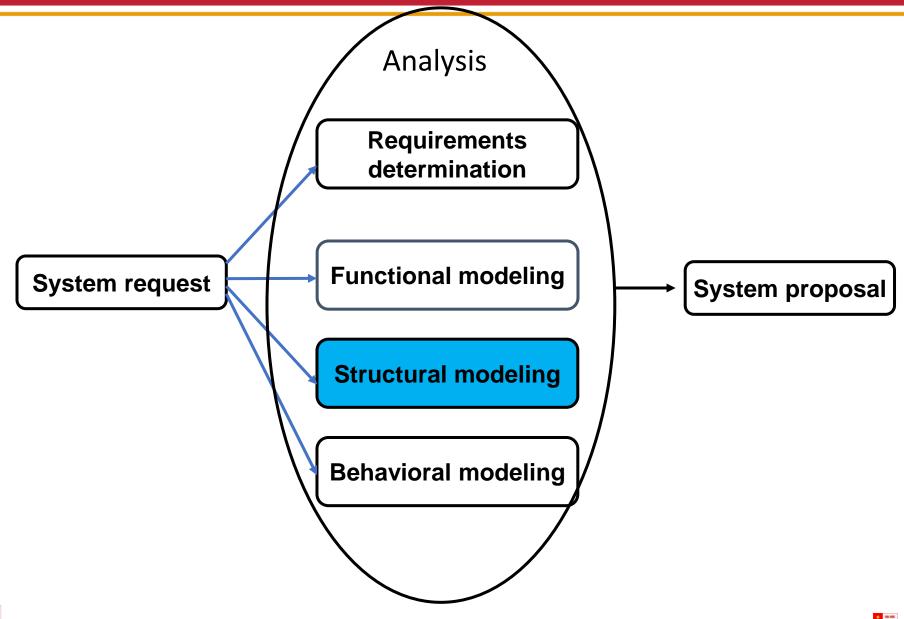
ONE LOVE. ONE FUTURE.



Part 2: System analysis

Chapter 6: Structural Modeling

Part 2: System analysis







Chapter syllabus

- 6.1. The structural models
 - Class
 - CRC cards
 - Class diagrams
 - Object diagrams
- 6.2. Creating structural models
 - Introduction
 - Object identification in use-case
 - Analysis-level class diagram





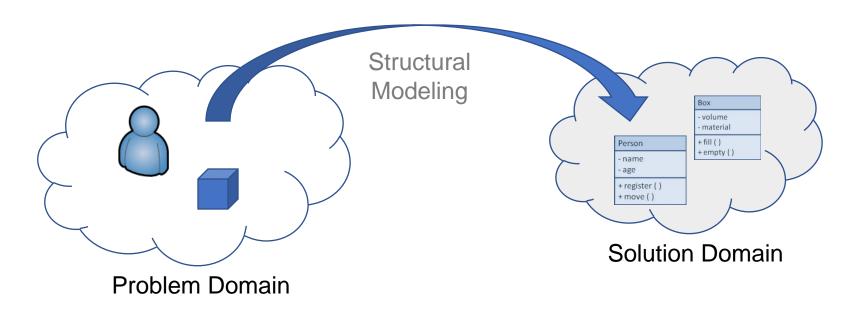
6.1. The structural models





6.1. Structural model

- Structural modeling describes the structure of the objects that supports the business processes in an organization
- A formal way of representing the objects that are used and created by a business system



Main goal: to discover the key data/objects contained in the problem domain and to build a structural model of the objects





6.1. Structural model

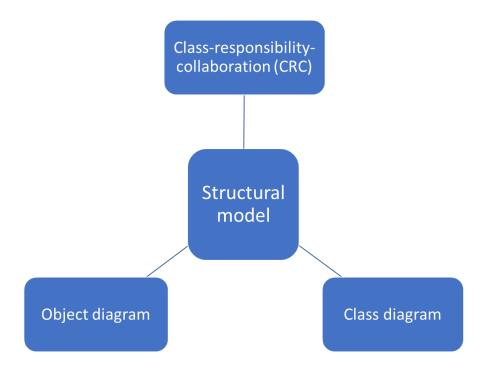
- Drawn in two steps:
 - First drawn in a conceptual, business-centric way
 - Then refined in a technology-centric way describing the actual databases and files
- In a conceptual way, structural model shows the logical organization of the objects without indicating how the objects are stored, created, or manipulated.
 - Classes created during analysis are not the classes that programmers develop during implementation
 - The refinement comes later
 - Characteristics: free from any implementation or technical details





6.1. Structural model

Typical structural models







6.1.1 Class

Class Classification

- Domain classes, user-interface classes, data structure classes, file structure classes, operating environment classes, document classes, multimedia classes etc.
- Concrete classes >< Abstract classes

Class members

- Attribute: information relevant to the description of the class within the application domain
- Operation: The actions to which the instances of the class are capable of responding





6.1.1 Class

Relationship between classes

- Association: instances of class could be either linked or combined logically or physically to the others into some aggregation.
 - Aggregation: the whole-part relationship, also known as the "a-part-of", "has parts" relationship.
 - Composition: a form of aggregation, the part cannot exist if the whole is destroyed.
- Generalization
 - Inheritance relationship, "a-kind-of" relationship
 - Classes inherit attributes and operations of other classes
 - The subclass should be capable of substituting for the superclass anywhere that uses the superclass
- Dependency: one class provides some information to the other class.
 - Modification of the supplier class may impact the client class



6.1.2 Class Responsibility Collaborator (CRC) Cards

- CRC cards to represent Responsibilities and Collaborations of classes
- Purpose: discovering objects, attributes, relationships, and operations.
- Responsibilities: what the class can do?
 - Knowing responsibilities
 - An instance of a class typically knows the values of its attributes and its relationships
 - Doing responsibilities
 - Things that an instance of a class must be capable of doing
 - An instance of a class can execute its operations, or it can request a second instance to execute one of its operations
- Collaborations: what other classes can interact with?
 - Most use case involve a set of several classes
 - These classes form Collaborations





6.1.2 Class Responsibility Collaborator (CRC) Cards

- Each CRC card represents a class
- On the card save:
 - The name of the class.
 - Class responsibilities
 - Class collaborator

Class Order							
Responsibility	Collaboration						
PriceStockValid Payment	• Customers • Order line 						

Class Deliver	γ
Responsibility	Collaboration
Item identityCheck ReceiverOrder No.Total Qty	Partner Clients



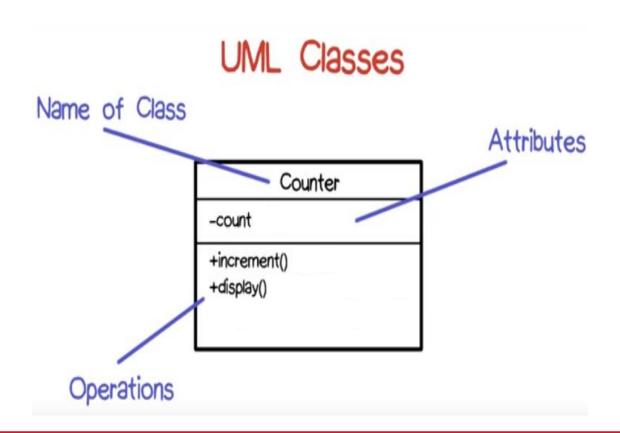
6.1.2 Class Responsibility Collaborator (CRC) Cards

• Sample CRC Card (in details)

Front:	,	,			
Class Name: Old Patient	ID: 3		Type: Concrete, D	Oomain	
Description: An individual who medical attention	needs to receive or	has received	Associated Use C	Cases: 2	
Responsibilities			Collaborators		
Make appointment		Appointme	nt		
Calculate last visit					
Change status					
Provide medical history	Medical history				
		Back:			
		-1	nt (double) nce carrier (text)		
		Relationsl Gener	hips: alization (a-kind-of):	Person	
		Aggre	gation (has-parts):	Medical History	
auso:		Other	Associations:	Appointment	
CH KHAIL					

6.1.3. Class diagram

- The static view of a system
- Shows all or part of the class structure of a system.
- Show the existence of classes and their relationships in the logical design of a system

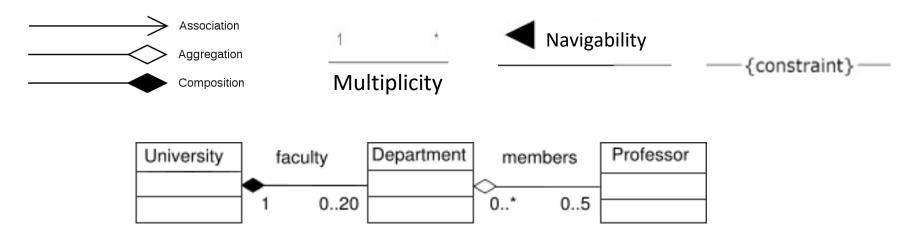


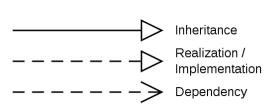


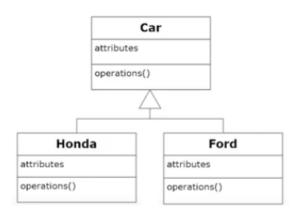


Class diagram symbols

• Relationship between classes:



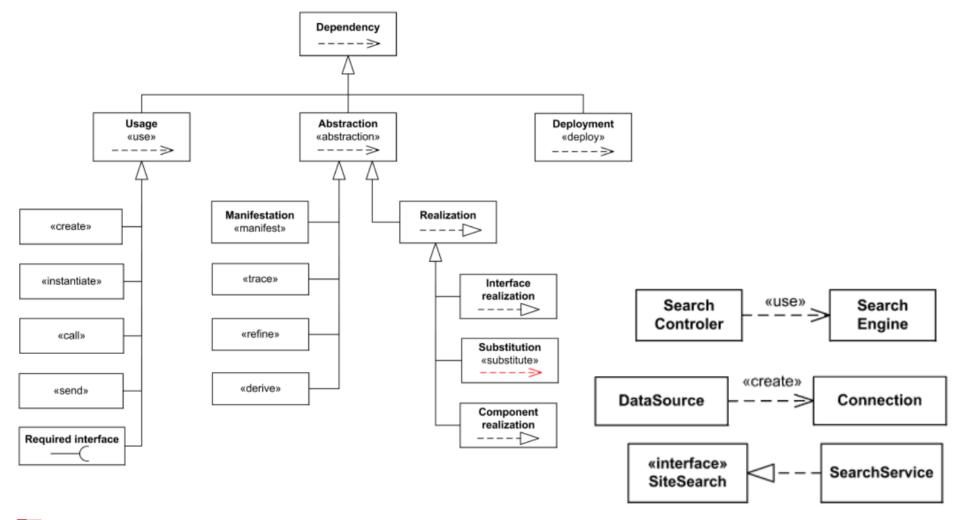






Class diagram symbols

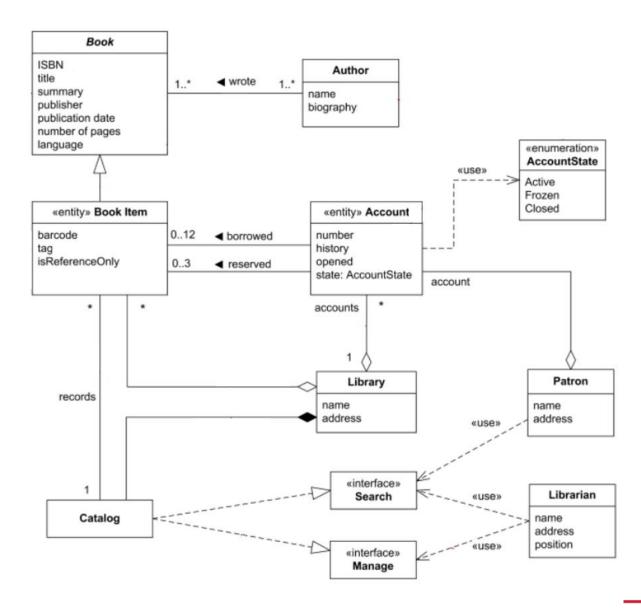
• Relationship between classes:





6.1.3. Class diagram

Example

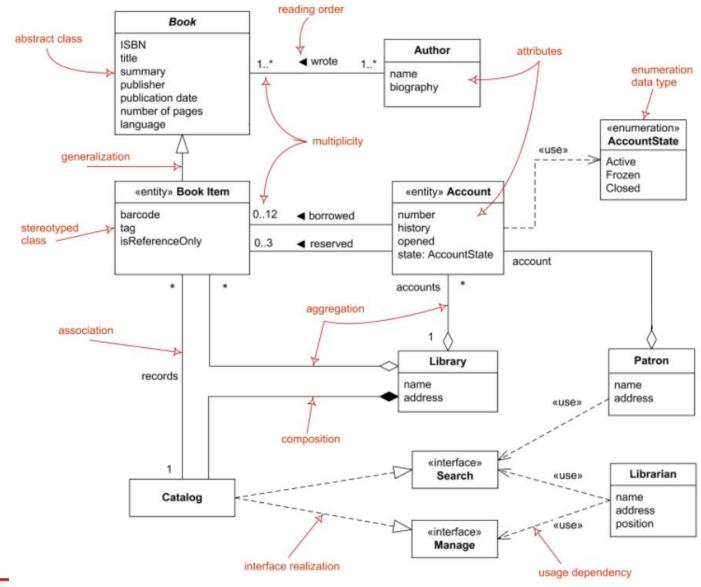






6.1.3. Class diagram

Example



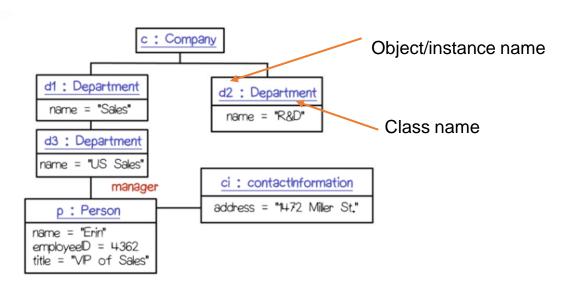


6.1.4. Object diagram

 Represent actual instances of classes and the relationships between them

Object Diagram

Conveys objects and links instead of classes and relationships



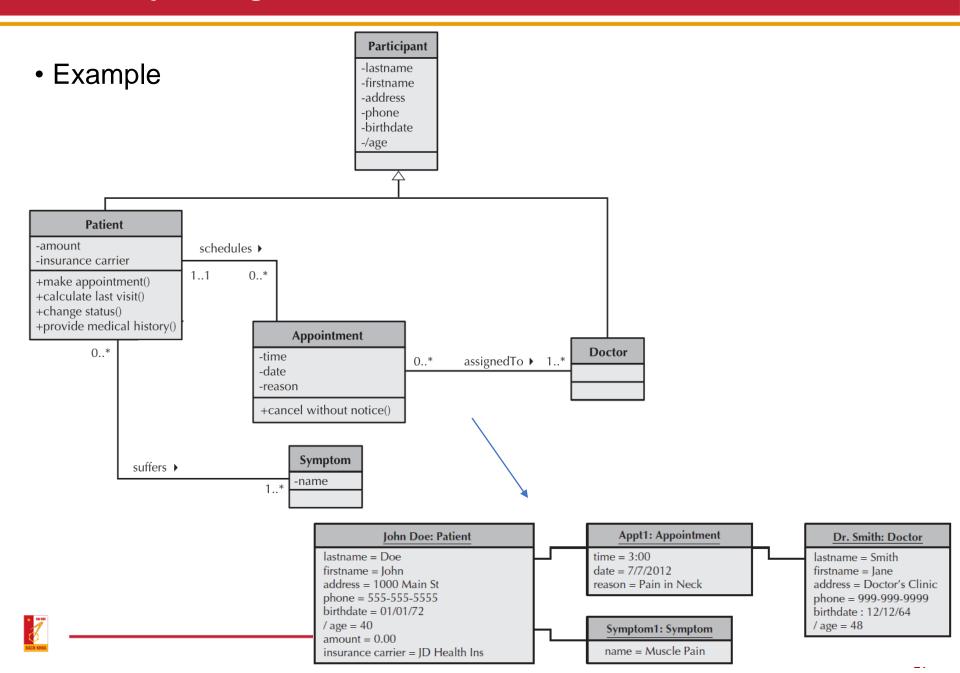


6.1.4. Object diagram

- When to use?
 - Document the structure of the classes with additional information
 - Trying to uncover details of a class.
 - It is easier to think in terms of concrete objects (instances) rather than abstractions of objects (classes)



6.1.4. Object diagram

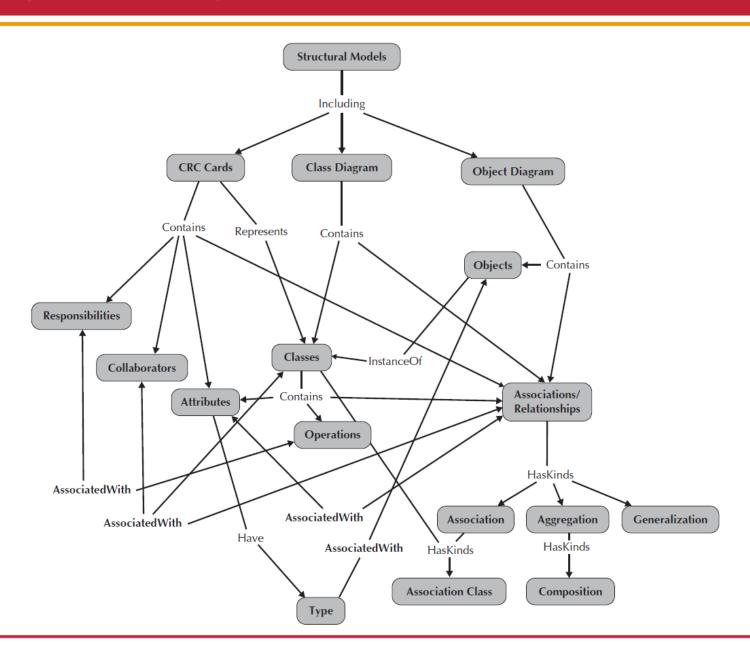


Verifying and Validating the Structural Model

- CRC cards and Class Diagrams can be used to describe the structural models of the system
 - Every CRC card should be associated with a class on the CD, and vice versa.
 - Responsibilities listed on CRC card must be included as operations in a class on a CD
 - Attributes of CRC card must be included as attributes in a class on a CD, and vice versa
 - Collaborators of the CRC card imply some type of association that is connected to the associated class on CD
 - Object type of the attributes on a CD implies an association from the class to the class of the object type



Verifying and Validating the Structural Model



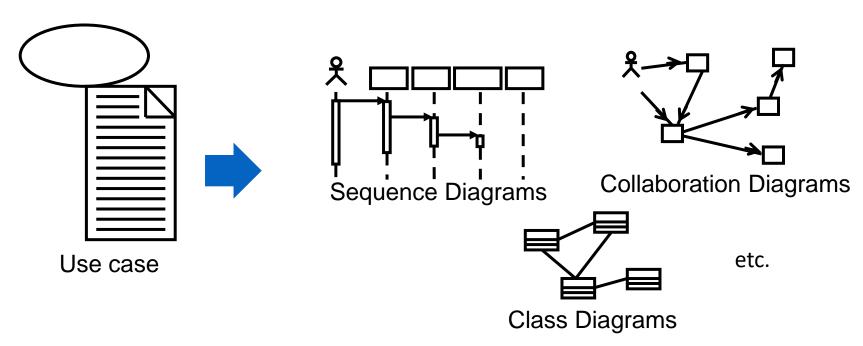


6.2. Creating structural models



6.2.1. Introduction

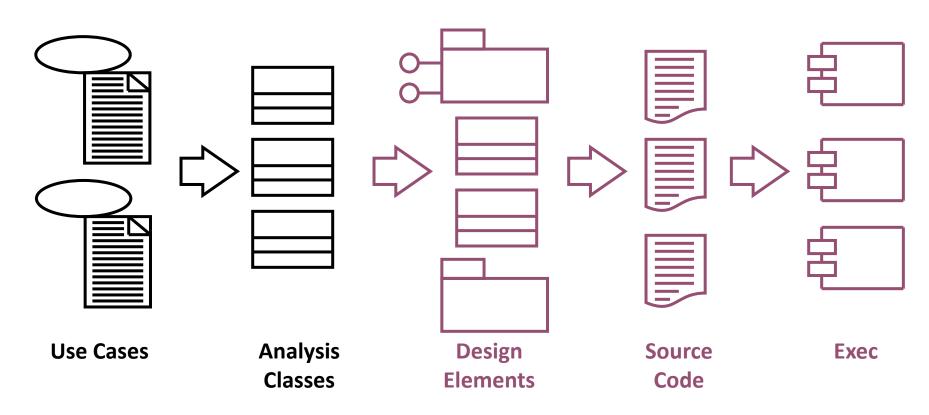
- Analysis and Design process is use-case driven
 - Use cases defined for a system are the basis for the entire development process.
 - Benefits of use cases:
 - Concise, simple, and understandable by a wide range of stakeholders.
 - Help synchronize the content of different models





6.2.1. Introduction

 To bridge the gap from analysis to design (and then to code), need to link use cases to objects.



Classes which perform a use-case flow of events



6.2.1. Introduction

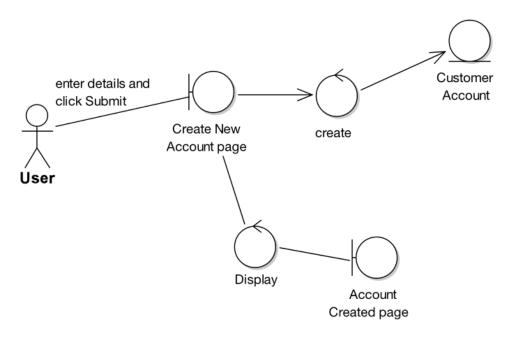
Steps:

- Domain model + Functional models (Use case models) => 1st draft of Class model (analysis-level class model): Robustness analysis
- Behavior analysis: Update 1st draft of Class model => Final Class model (next chapter)
- Robustness analysis:
 - Analyze use case text and identify set of objects (analysis classes) for each use case
 - → Object identification in a use-case



6.2.2. Object identification in a use-case

- For each Use-Case
 - From the use case text, draw an *object picture of the use case* (robustness diagram)
 - Collection of objects in the use case
 - Communication associations between them (data flow or control flow, don't worry much about the direction)



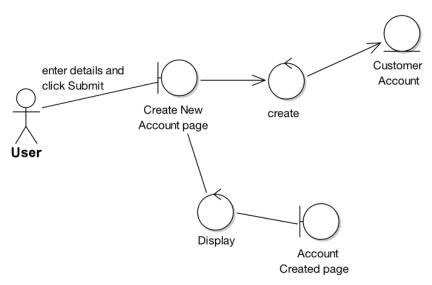


Robustness diagram

Type of objects:

- Boundary Object
 - Model interaction between the system and the outside
 - Typically are screens or web pages that the actor interacts with
- $\left(\cdot \right)$

- Entity Object
 - Store and manage information in the system (domain objects, etc.)
- Control Object
 - Coordinate the use-case behavior, "glue" between the boundary are entity objects.





Robustness diagram

- From use case text
 - Boundary objects or entity objects: noun
 - Controller objects: verbs
- The legal Communication:
 - Boundary ➤ Controller ➤ Entity: ~ noun-verb-noun pattern in use case text
 - Controller ➤ Controller
 - Boundary/Entity object can not "talk" to Boundary/Entity object

Robustness diagram Steps

From EACH use case description:

- Work through the use case a sentence at a time
- Take entity classes from the domain model, and add any that are missing.
- Make a boundary object for each screen, and name it unambiguously
- Controllers are only occasionally real control objects but typically used as a placeholder for a software function.
- Create communication associations between objects
 - There's a direct 1:1 correlation between the flow of action in the robustness diagram and the steps described in the use case text.
- Must scan not just the basic course, but all the alternate courses as well



"Write Customer Review" Use Case

BASIC COURSE:

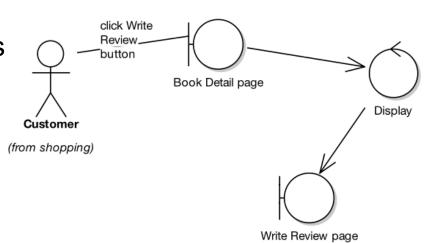
The Customer clicks the Write Review button for the book currently being viewed, and the system shows the Write Review page. The Customer types in a Book Review, gives it a Book Rating out of 5 stars, and clicks the Send button. The system ensures that the Book Review isn't too long or short, and that the Book Rating is within 1-5 stars. The system then displays a confirmation page, and the review is sent to a Moderator ready to be added.



• "Write Customer Review" Use Cas

BASIC COURSE:

The Customer clicks the Write Review button for the book currently being viewed, and the system shows the Write Review page. The Customer types in a Book Review, gives it a Book Rating out of 5 stars, and clicks the Send button. The system ensures that the Book Review isn't too long or short, and that the Book Rating is within 1-5 stars. The system then displays a confirmation page, and the review is sent to a Moderator ready to be added.

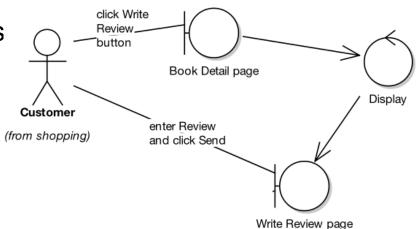




"Write Customer Review" Use Cas

BASIC COURSE:

The Customer clicks the Write Review button for the book currently being viewed, and the system shows the Write Review page. The **Customer types in a Book** Review, gives it a Book Rating out of 5 stars, and clicks the Send button. The system ensures that the Book Review isn't too long or short, and that the Book Rating is within 1-5 stars. The system then displays a confirmation page, and the review is sent to a Moderator ready to be added.



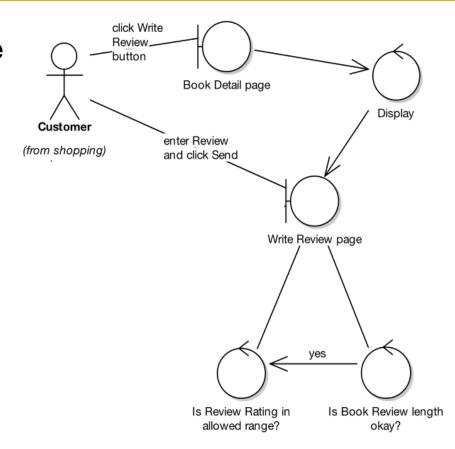


Write Customer Review Use Case

BASIC COURSE:

The Customer clicks the Write Review button for the book currently being viewed, and the system shows the Write Review page. The Customer types in a Book Review, gives it a Book Rating out of 5 stars, and clicks the Send button. The system ensures that the **Book Review isn't too long** or short, and that the Book Rating is within 1-5 stars.

The system then displays a confirmation page, and the review is sent to a Moderator ready to be added.

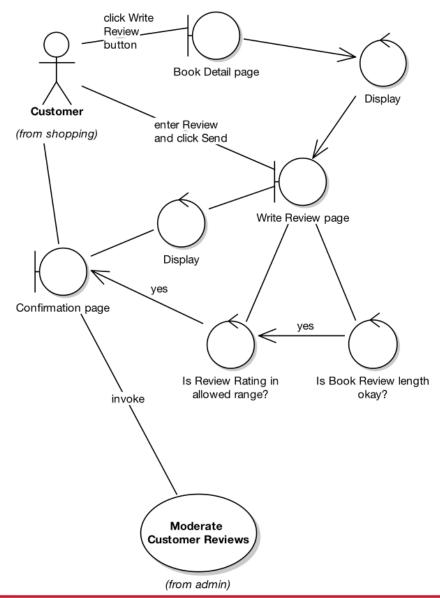




Write Customer Review Use Case

BASIC COURSE:

The Customer clicks the Write Review button for the book currently being viewed, and the system shows the Write Review page. The Customer types in a Book Review, gives it a Book Rating out of 5 stars, and clicks the Send button. The system ensures that the Book Review isn't too long or short, and that the Book Rating is within 1-5 stars. The system then displays a confirmation page, and the review is sent to a Moderator, ready to be added.





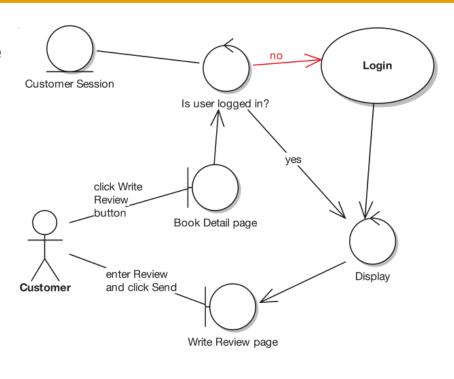
"Write Customer Review" Use Case

ALTERNATE COURSES:

- + User not logged in: The user is first taken to the Login page, and then to the Write Review page once they've logged in.
- + The user enters a review which is too long (text > 1MB): The system rejects the review, and responds with a message explaining why the review was rejected.
- + The review is too short (< 10 characters): The system rejects the review.

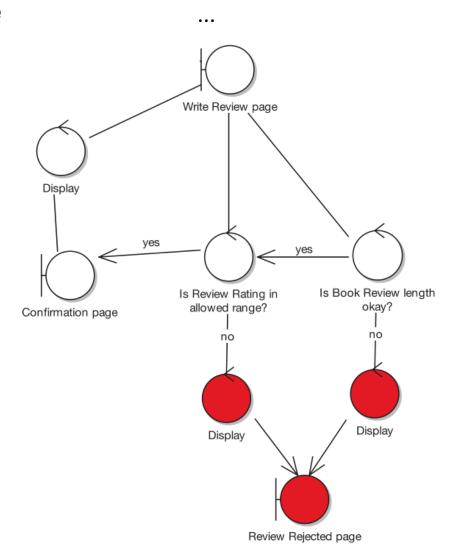


- "Write Customer Review" Use Case ALTERNATE COURSES:
 - + User not logged in: The user is first taken to the Login page, and then to the Write Review page once they've logged in.
 - + The user enters a review which is too long (text > 1MB): The system rejects the review, and responds with a message explaining why the review was rejected.
 - + The review is too short (< 10 characters): The system rejects the review.





- "Write Customer Review" Use Case **ALTERNATE COURSES:**
 - + User not logged in: The user is first taken to the Login page, and then to the Write Review page once they've logged in.
 - + The user enters a review which is too long (text > 1MB): The system rejects the review, and responds with a message explaining why the review was rejected.
 - + The review is too short (< 10 characters): The system rejects the review.





CASE STUDY: INTERNET BOO

"Write Customer Review" Use Case

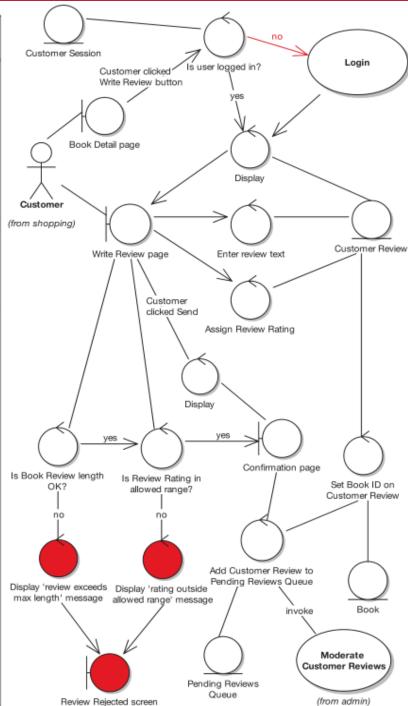
- Improve to make it clearer
- Full robust diagram

BASIC COURSE: On the Book Detail screen for the book currently being viewed, the Customer clicks the Write Review button. The system checks the Customer Session to make sure the Customer is logged in, and then displays the Write Review screen. The Customer types in a Book Review, gives it a Book Rating out of 5 stars, and clicks the Send button, The system ensures that the Book Review isn't too long or short, and that the Book Rating is within 1-5 stars. The system then displays a confirmation screen, and the review is added to the Pending Reviews Queue for moderation (this will be handled by the Moderate Customer Reviews use case).

ALTERNATE
COURSES:
User not logged in:
The user is first taken
to the Login screen,
and then to the Write
Review screen once
they've logged in.

The user enters a review which is too long (text > 1MB): The system rejects the review, and responds with a message explaining why the review was rejected.

The review is too short (< 10 characters): The system rejects the review.





6.2.3. Analysis-level class diagram

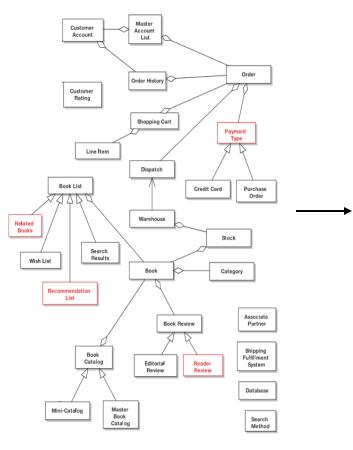
- During the Robustness analysis:
 - Correct ambiguities
 - Update the use case text if needed
 - Update the domain model if new object/domain class is added
 - Identify attributes of the domain objects (classes later)
 - Assign operations to the classes can be completed later (after behavior modeling step)
- Draw all robustness diagrams of the system.
- Take a sweep through all of the robustness diagrams, update the domain model to the 1st draft class diagram (or analysis-level class diagram)

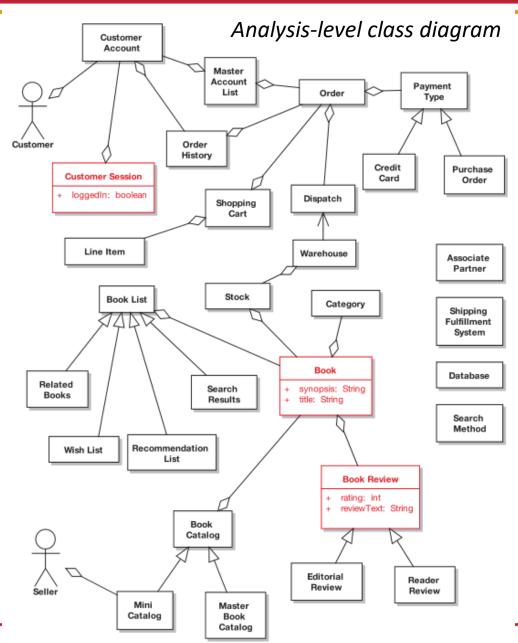




Analysis-level class diagram

Domain model







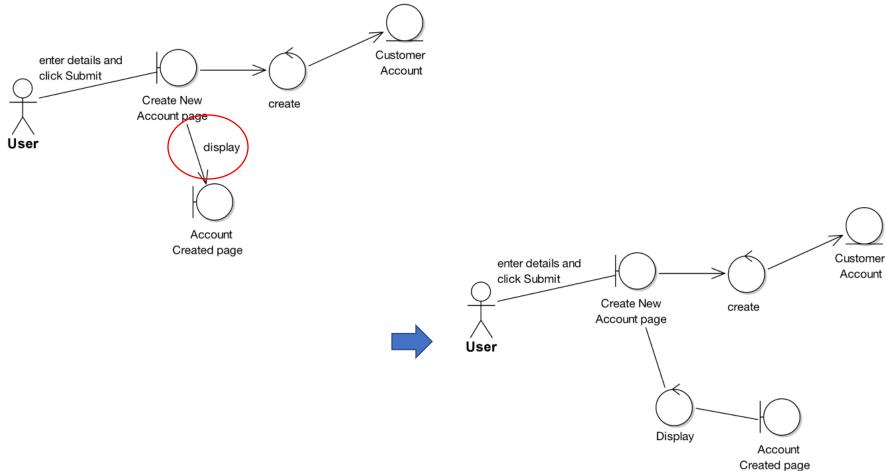
6.2.3. Analysis-level class diagram

Verification:

- For each use case, make sure the use case text matches the robustness diagram
- Don't forget the alternate courses
- Make sure you haven't violated the syntax rules for robustness analysis
- Make sure that all the entities on all robustness diagrams appear within the analysis-level class diagram



Question 1







Question 2

Add External Books to Catalog:

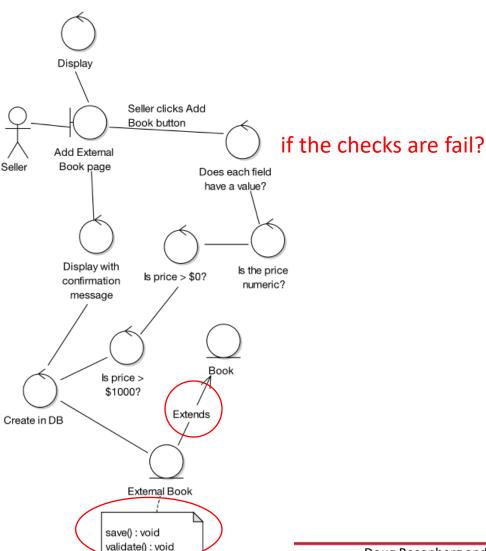
BASIC COURSE:

The system displays the Add External Book page. The Seller types in the External Book details (title, ISBN, price etc) and clicks the Add Book button. The system checks that each field has a value. and that the price is numeric, isn't a negative value or > \$1,000, then creates the External Book in the database. The system then displays the Add External Book page again (so the user can add another External Book), along with a message confirming that the External Book was successfully added.

ALTERNATE COURSES:

SBN wasn't found in he Bookstore database: The Add External Book page is redisplayed, with a message that the ISBN wasn't found.

Price was invalid: Page is redisplayed with "Invalid price" message.





Add External Books to Catalog:

BASIC COURSE:

The system displays the Add External Book page. The Seller types in the External Book details (title, ISBN, price etc) and clicks the Add Book button. The system checks that each field has a value. and that the price is numeric, isn't a negative value or > \$1,000, then creates the External Book in the database. The system then displays the Add External Book page again (so the user can add another External Book), along with a message confirming that the External Book was successfully added.

ALTERNATE COURSES:
ISBN wasn't found in the
Bookstore database: The Add
External Book page is
redisplayed, with a message that
the ISBN wasn't found.

Price was invalid: Page is redisplayed with "Invalid price" message.

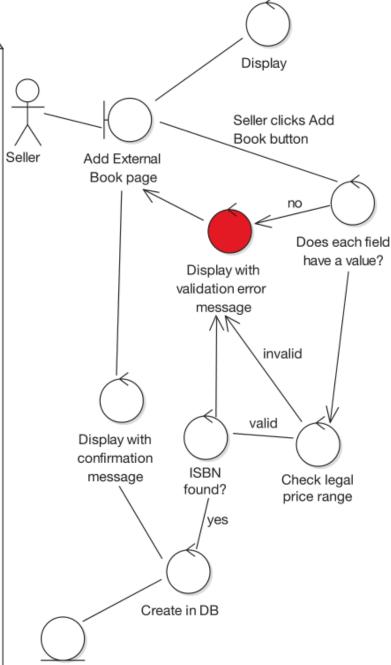


BASIC COURSE:

The system displays the Add External Book page. The Seller types in the External Book details (title, ISBN, price etc) and clicks the Add Book button. The system checks that each field has a value. and that the price is numeric, isn't a negative value or > \$1,000. The system also checks that there's a matching ISBN in the Bookstore database. The system then creates the External Book in the database. The system then displays the Add External Book page again (so the user can add another External Book), along with a message confirming that the External Book was successfully added.

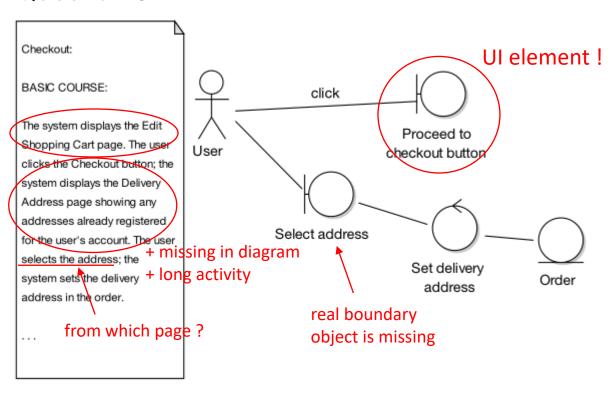
ALTERNATE COURSES: ISBN wasn't found in the Bookstore database: The Add External Book page is redisplayed, with a message that the ISBN wasn't found.

Price was invalid: Page is redisplayed with "Invalid price" message.





Question 3







Checkout:

BASIC COURSE:

The system displays the Edit Shopping Cart page. The user clicks the Checkout button; the system displays the Delivery Address page showing any addresses already registered for the user's account. The user selects the address; the system sets the delivery address in the order.

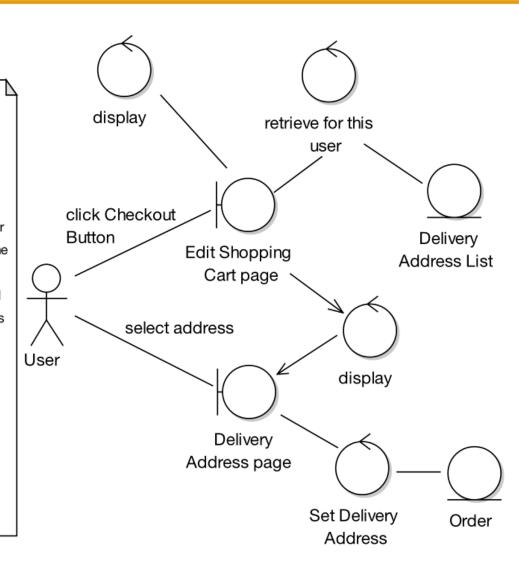
. . .

Checkout:

BASIC COURSE:

The system displays the Edit
Shopping Cart page. The user
clicks the Checkout button; the
system retrieves the Delivery
Addresses already registered
for this user, and then displays
the Delivery Address page.
The user selects the address
to use from the Customer
Address list selector, and
clicks the Select button. The
system sets the Delivery
Address in the Order.

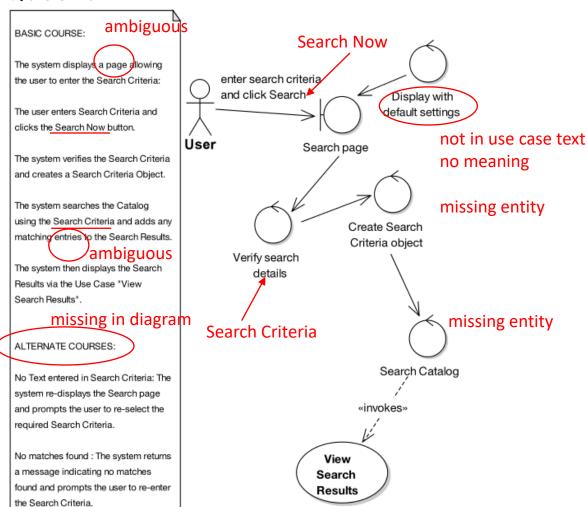
. . .







Question 4







Quiz

BASIC COURSE:

The system displays a page allowing the user to enter the Search Criteria:

The user enters Search Criteria and clicks the Search Now button.

The system verifies the Search Criteria and creates a Search Criteria Object.

The system searches the Catalog using the Search Criteria and adds any matching entries to the Search Results.

The system then displays the Search Results via the Use Case "View Search Results".

ALTERNATE COURSES:

No Text entered in Search Criteria: The system re-displays the Search page and prompts the user to re-select the required Search Criteria.

No matches found: The system returns a message indicating no matches found and prompts the user to re-enter the Search Criteria.

BASIC COURSE:

The system displays the Search Page; the user enters their Search Criteria and clicks the Search Now button.

The system checks that some text has been entered into the Search Criteria, and then creates the Search Criteria.

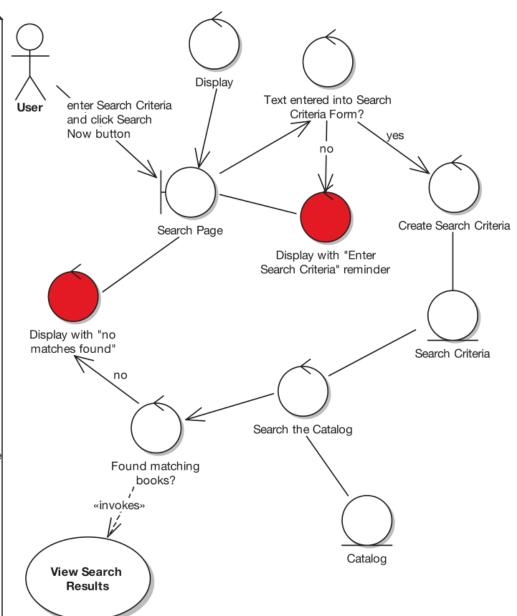
The system searches the Catalog using the Search Criteria and adds any matching Books to the Search Results.

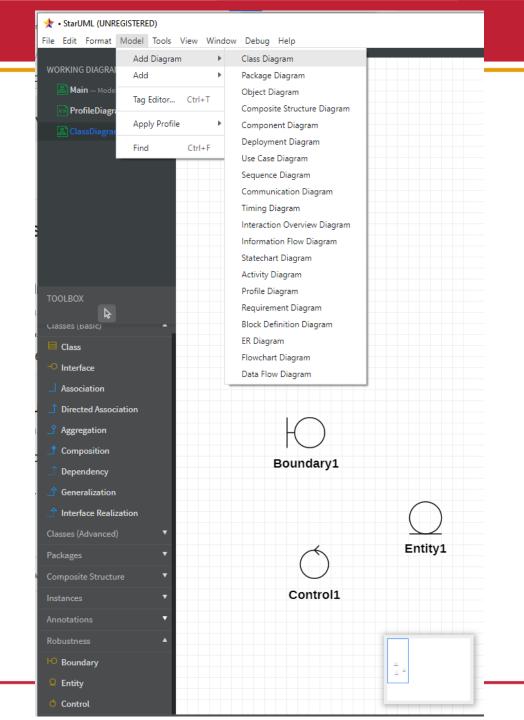
The system then displays the Search Results via the Use Case "View Search Results".

ALTERNATE COURSES:

No text entered in Search Criteria: The system redisplays the Search Page and prompts the user to reselect the required Search Criteria.

No matches found: The system returns a message indicating no matches found and prompts the user to reenter the Search Criteria.







Exercise

- Draw the Robustness diagram for your project
- Build the Analysis-level class diagram for your project

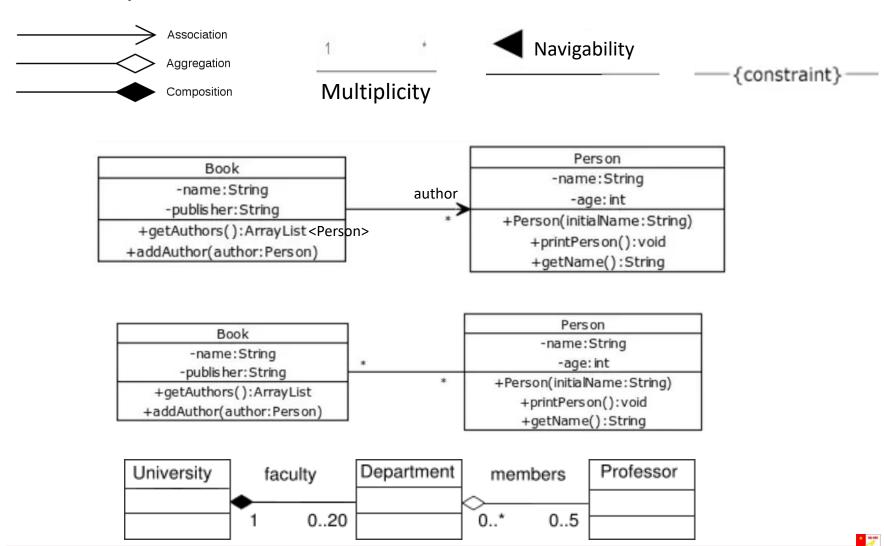






Class diagram symbols

Relationship between classes:







Class diagram symbols

- Class name, attributes, operations/actions/methods:
 - Italic: abstract members
 - <u>Underline</u>: static members
 - ALL_CAPS: final members
- Access modifier: (+, -, #, ~)

Class Name

- private attributes
- + public attributes
- # protected attributes
- private operation
- + public operation
- # protected operation



Class diagram symbols

- Derived property: /
- [Optional] Inherited Property: ^

Patient title: String givenName: String middleName: String familyName: String /name: FullName birthdate: Date admitted: Date /age: Integer

