Comp 320 – Computer Systems Analysis and Design

Dr. William L. Honig
Associate Professor
Department of Computer Science

Completion of System Analysis

- Chapter Five in Text
  - Domain Models (Concepts) and Contracts
  - Aka Objects and Methods
  - Abstract or Logical Components
  - Aot Instances
- Set the stage for design and programming
  - System level operation is clear
  - Internal, physical, implementation remains

Overview

Chapter 5 discusses Steps 5 and 6 of the process for object-oriented systems analysis. It introduces the domain model and system operation contracts.

There is one domain model for the system – a static model showing the conceptual scope of the entire system. Its components are concepts, their attributes, and associations between concepts. It also shows hierarchies of concepts.

Overview
(continued)

It is helpful to construct the domain model one use case at a time in order to understand which concepts, attributes, and associations are relevant to each use case.

The final step in object-oriented systems analysis is to write a contract for each system operation. The contracts are derived from the use case narratives and the domain model.

Overview
(continued)

Each contract specifies what changes in the state of the system are required after the system operation has executed successfully. These system operation contracts will be the basis for object-oriented design.

Procedure for Object-Oriented Systems Analysis

Step 1. Identify the business events and make an event table.

Step 2. Identify the use cases and produce a use case diagram for the system.

Step 3. Write a use case narrative describing the system's response to each business event.

Procedure for Object-Oriented Systems Analysis (continued)

Step 4. Draw a system sequence diagram for each use case scenario.

Step 5. Produce a domain model showing the concepts, attributes and associations in the problem domain of the system.

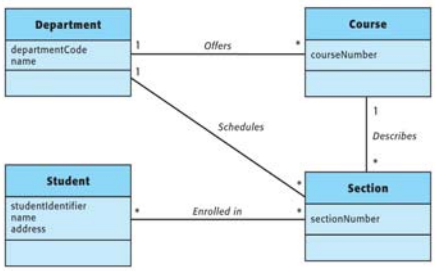
Step 6. Write a contract for each system operation.

Step 5 of Object-Oriented Systems Analysis

Produce a domain model showing the concepts, attributes, and associations in the problem domain of the system.

Domain Model

FIGURE 5.1



Concepts, Attributes, and Associations

- A concept is an abstraction of a thing, a person, or an idea. It is represented by a rectangle.
- An attribute is a characteristic of a concept which may have a value. Attribute names appear in the lower compartment of the concept rectangle.
- An association is a significant connection between concepts. It is represented by a line connecting a pair of concepts.

Concepts

Identifying and adding concepts to the domain model is Step 5a of the process for object-oriented systems analysis.

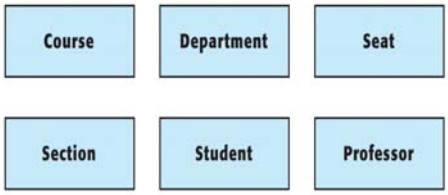
Finding Concepts

1. Look for nouns or noun phrases describing the problem domain.
2. Use a checklist of concept categories. (See Figure 5.2 in text.)

Include a concept in the domain model when the system needs to store data about the concept to respond to a future event.

Concepts

FIGURE 5.4



Show Off Time

LOYOLA UNIVERSITY CHICAGO

- Who here is a good analyst?
  - Willing to answer a few questions about the Red Line system?
- What is some one key concept in thinking about the system?
  - Imagine a use case being run through the narrative and sequence diagram

Attributes

Identifying and adding attributes to the domain model is Step 5b of the process for object-oriented systems analysis.

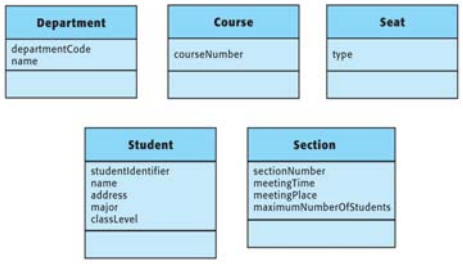
Attributes
(continued)

Attributes describe concepts.

Concept	Student	Professor	Section
Attributes:	studentIdentifier	professorIdentifier	number
	studentName	professorName	meetingTime
	studentAddress	professorAddress	meetingPlace
	major classLevel	title	maximum NumberOf Students

Attributes
(continued)

FIGURE 5.5



Attributes and Instances

An instance of a concept is a specific occurrence of a concept type.

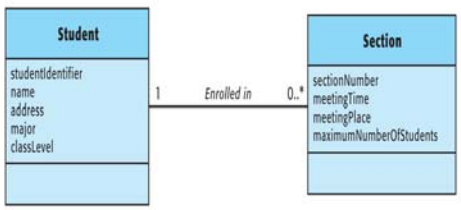
Student Concept:	Student Instance 1:	Student Instance 2:
Attributes	*Values*	*Values*
studentIdentifier	40168	82704
studentName	Louella Fernbee	Mortimer Snow
studentAddress	123 Any St.	456 Some St.
major	CIS	CS
classLevel	Junior	Sophomore

Associations

Identifying and adding associations to the domain model is Step 5c of the process for object-oriented systems analysis.

Associations
(continued)

FIGURE 5.7



Associations
(continued)

Always model associations explicitly; never use an attribute to imply an association.

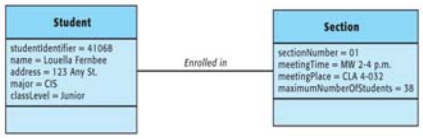
Instances of Associations

There are instances of associations as well as instances of concepts.

Association: Enrolled In

Instance: studentIdentifier = 41068 associated with sectionNumber = CIS-4-01

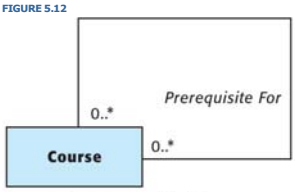
FIGURE 5.6



Reflexive Associations

A concept may be associated with itself.

FIGURE 5.12

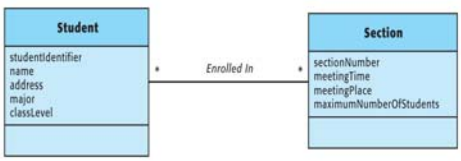


A course is a prerequisite for zero or more other courses.
A course has as prerequisites zero or more other courses.

Multiplicity of Associations

The multiplicity of an association is the number of instances of a concept which can be associated with one instance of another concept.

FIGURE 5.9



Multiplicity of Associations
(continued)

Each end of an association is labeled with the minimum and maximum values of its multiplicity.

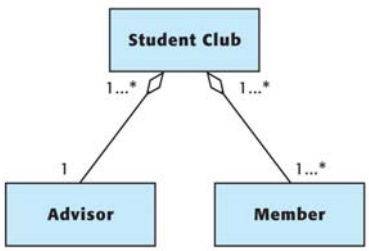
0 .. 1
1 .. 1
.. * signifies unlimited (*more* or *many*)
* alone means *zero or more*

Whole-to-Part Associations

The UML provides ways to model two types of whole-to-part associations – aggregation and composition.

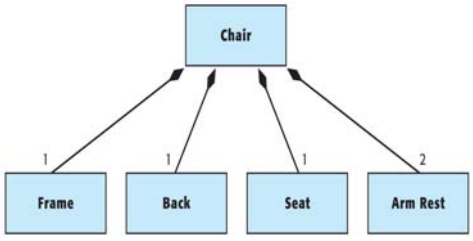
Example of an Aggregation

FIGURE 5.14



Example of a Composition

FIGURE 5.15



Aggregation and Composition

FIGURE 5.16

	AGGREGATION	COMPOSITION
Name of whole	Aggregate	Composite
Name of part	Constituent	Component
Parts	May be of different types	Usually of the same type
Existence	May exist without its parts	Does not exist without its parts
Number of wholes to which a part may belong	Part may belong to more than one aggregate at a time	Part may belong to only one composite at a time

Categories of Whole-to-Part Associations

- **Assemblies of parts**
- **Members of groups**
- **Containers and their contents**

Associations and Generalization-Specialization Hierarchies

Identifying and adding associations and generalization-specialization hierarchies to the domain model is Step 5c of the process for object-oriented systems analysis.

Show Off Time

LOYOLA UNIVERSITY CHICAGO

- Who here is a good analyst?
  - Willing to answer a few questions about the Red Line system?
- What is some one key association between concepts?
  - Imagine how information changes and grows or shrinks during system operation

Generalization-Specialization Hierarchies

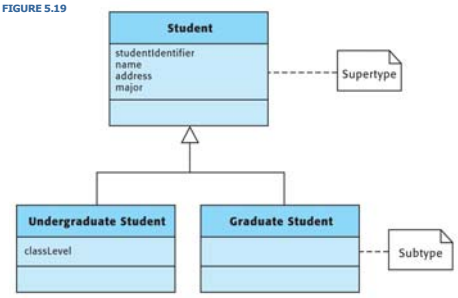
A generalization-specialization hierarchy classifies a type of concept into its subtypes.

Every instance of a subtype must also be an instance of its supertype.

Subtypes have the same set of attributes as their supertype. These attributes are not duplicated in the domain model.

Generalization-Specialization Hierarchies (continued)

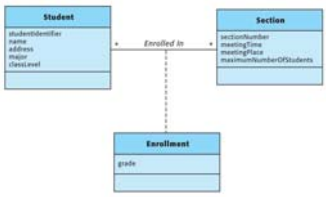
FIGURE 5.19



Association Concepts

Add an association concept to a domain model when you need to include an attribute which depends upon an association.

FIGURE 5.21



Creating a Domain Model

Create a domain model one use case at a time.

Then, merge them to form a complete domain model for the entire system.

Step 6 of Object-Oriented Systems Analysis

Write a contract for each system operation.

System Operation Contracts

A system operation is an operation which the system carries out in response to a system input.

The system input and the system operation have the same name.

This relationship will be an important link between the system analysis models and the system design models.

Creating System Operation Contracts

- Identify each system operation in a system sequence diagram.
- Write the responsibilities of that operation in the contract.
- Write the preconditions in terms of the required changes in the domain model.
- Add the postconditions and exceptions.

Postconditions for System Operation Contracts

- What instances of concepts must be created or deleted?
- What attributes have their values modified? To what new values?
- Which instances of associations must be added or deleted?

Use the past tense and the passive voice.

System Operation Contracts (continued)

FIGURE 5.28

Contract	
Name:	**enterStudentIdentifier** (studentIdentifier)
Responsibilities:	Accept and validate the Student's identifier. Verify that the Student is eligible to register.
Type:	System
Exceptions:	If the student identifier is not valid, indicate that it was an error. If the Student is not eligible to register, inform the Student.
Preconditions:	Student is known to the system.
Postconditions:	None

System Operation Contracts (continued)

FIGURE 5.27

Contract	
Name:	**requestSection** (departmentCode, courseNumber, sectionNumber)
Responsibilities:	Enroll the Student in the Section.
Type:	System
Exceptions:	If the combination of department code, course number and section number is not valid, indicate that it was an error. If no seats are available, inform the Student.
Output:	
Preconditions:	Department and Section are known to the system.
Postconditions:	A new instance of the Enrolled In association was created, linking the Student and the Section.

Summary

Step 5 of object-oriented systems analysis produces a domain model for the system.

- Step 5a adds concepts to the domain model.
- Step 5b adds attributes to the domain model.
- Step 5c adds associations and generalization-specialization hierarchies.

Step 6 of object-oriented systems analysis writes a contract for each system operation.

Learning Objectives

- Discover concepts used in the domain model.
- Identify attributes of concepts.
- Derive the associations between concepts.
- Distinguish an instance of a concept from a concept.

Learning Objectives
(continued)

- Understand the use of multiplicity in associations.
- Know when to use whole-to-part associations such as aggregations and compositions.
- Find generalization-specialization hierarchies.
- Complete system operation contracts.