System Analysis and Design

Eighth Edition

Alan Dennis, Barbara Wixom, Roberta M. Roth

Chapter 5

Structural Modeling

Objectives

- Understand the rules and style guidelines for creating CRC cards, class diagrams, and object diagrams.
- Understand the processes used to create CRC cards, class diagrams, and object diagrams.
- Be able to create CRC cards, class diagrams, and object diagrams.
- Understand how to verify and validate the structural models.
- Understand how to verify and validate the structural models with the functional models.

Introduction

- A ***structural model*** is a formal way of representing the objects that are used and created by a business system.
  - It illustrates people, places, or things about which information is captured and how they are related to one another.
  - The structural model is drawn using an iterative process in which the model becomes more detailed and less conceptual over time.
- In analysis, analysts draw a ***conceptual model***, which shows the logical organization of the objects without indicating how the objects are stored, created, or manipulated.
  - Because this model is free from any implementation or technical details, the analysts can focus more easily on matching the model to the real business requirements of the system.

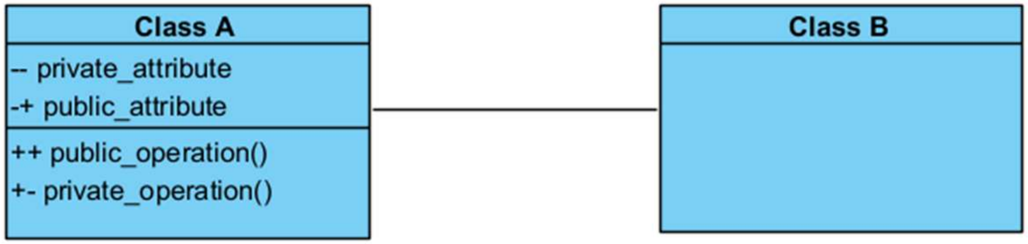
Structural Models

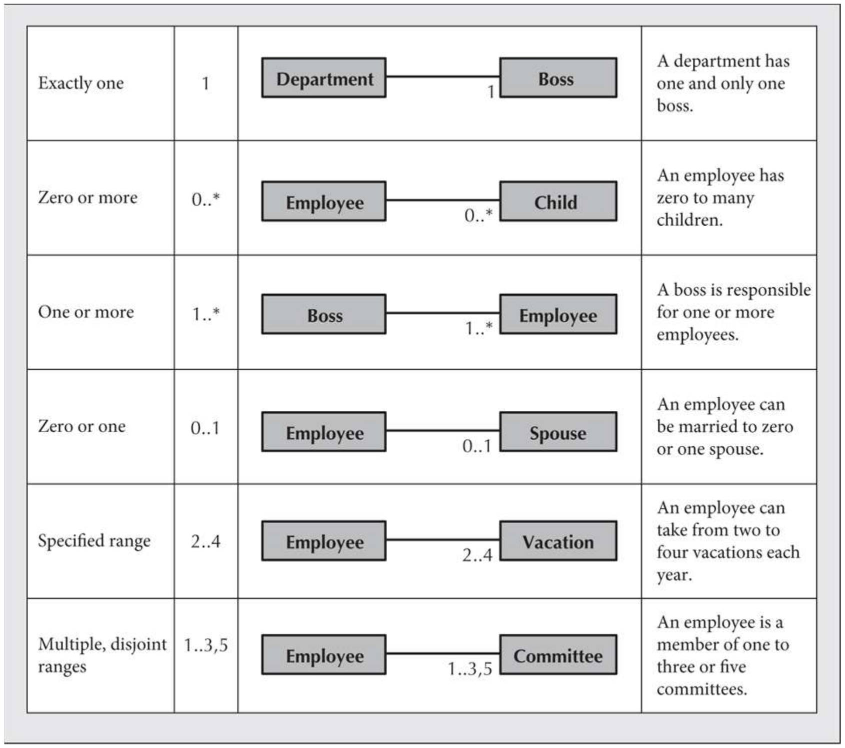
- The goal of the analyst is to discover the key objects contained in the problem domain and to build a structural model. Object-oriented modeling allows the analyst to reduce the semantic gap between the underlying problem domain and the evolving structural model.
- An exact mapping between the structural model and the problem domain may not be possible. In fact, it might not even be desirable.
- One of the primary purposes of the structural model is to create a vocabulary that can be used by the analyst and the users.
- Structural models represent the things, ideas, or concepts contained in the domain of the problem, and their relationships. By creating a structural model of the problem domain, the analyst creates the vocabulary necessary for the analyst and users to communicate effectively.
- It is important to remember that at this stage of development, the structural model does not represent software components or classes in an object-oriented programming language, even though the structural model does contain analysis classes, attributes, operations, and the relationships among the analysis classes.

Class/Object Diagrams

- Class diagrams can be used to depict the structural model.
- They can be used at many different levels of analysis:
  - System Level Context Diagram
  - Software Level Class designs
- The classes on the diagram provide the structural element necessary for that level of analysis.
  - On system diagrams, the classes represent subsystems, that are too high level to actually implement
  - On software diagrams, the classes represent actual things that the software engineer codes.

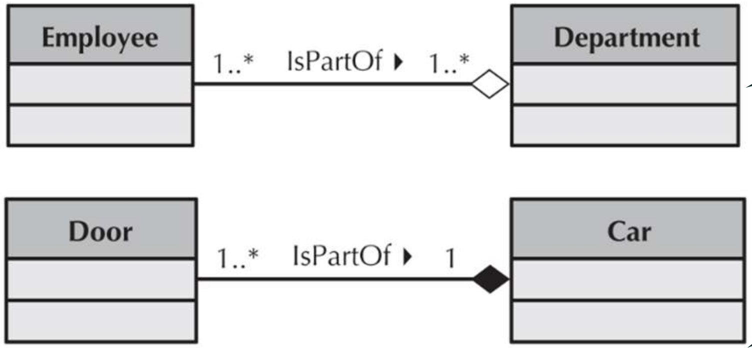
Classes, Attributes, Operations





Exactly one	1	Department 1 Boss	A department has one and only one boss.
Zero or more	0..*	Employee 0..* Child	An employee has zero to many children.
One or more	1..*	Boss 1..* Employee	A boss is responsible for one or more employees.
Zero or one	0..1	Employee 0..1 Spouse	An employee can be married to zero or one spouse.
Specified range	2..4	Employee 2..4 Vacation	An employee can take from two to four vacations each year.
Multiple, disjoint ranges	1..3,5	Employee 1..3,5 Committee	An employee is a member of one to three or five committees.

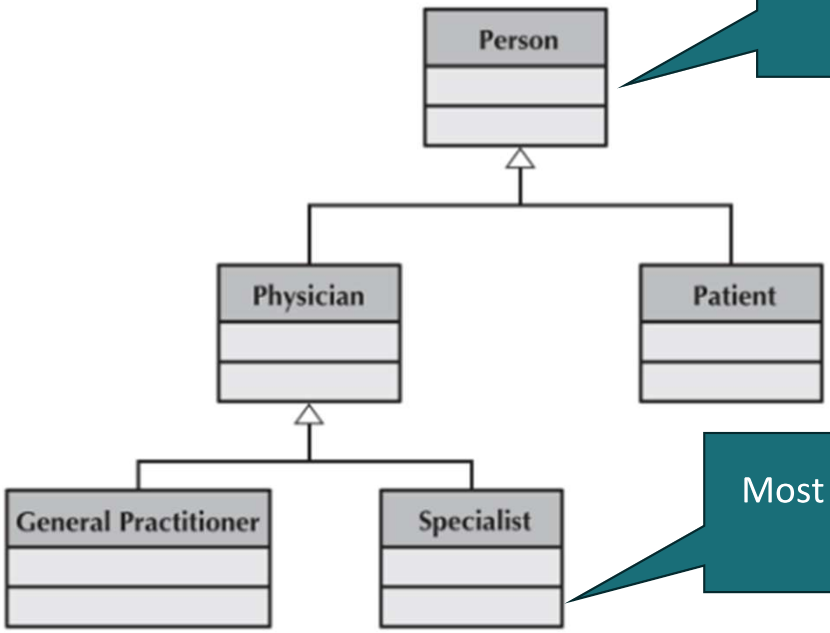
Class Associations - Aggregation



Loose Aggregation
Employee has a Life
Span < Department

Strict Aggregation
(Composition)
Door Life Span = Car

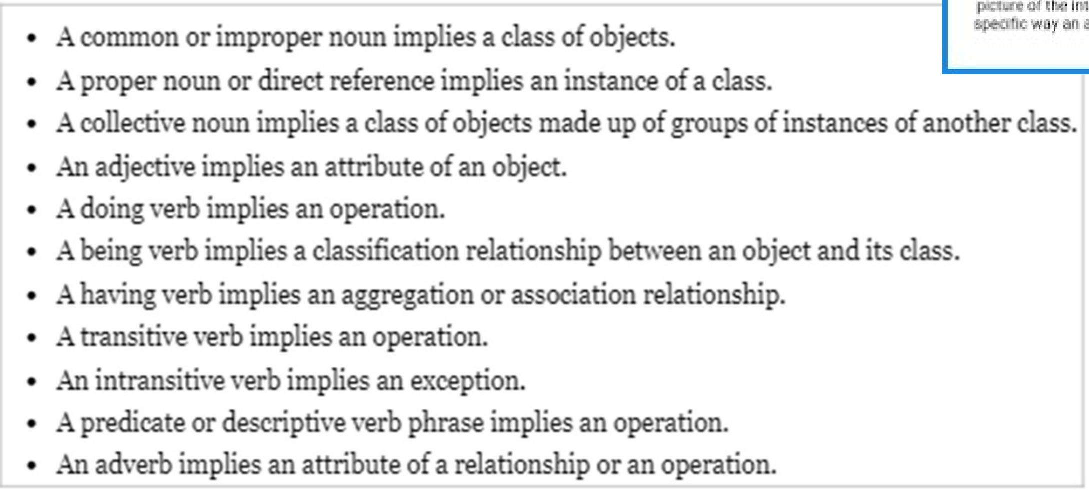
Class Associations – Generalization (Inheritance)



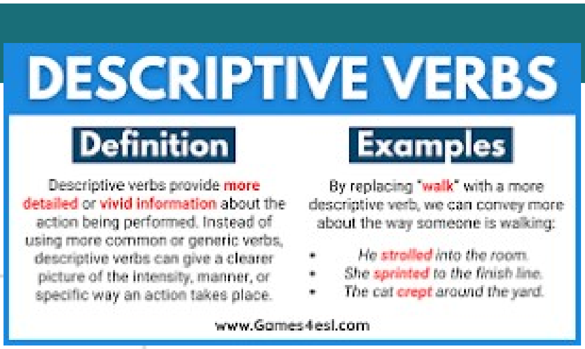
Most General Base Class

Most Specific Derived Class

Brainstorming Classes



- A common or improper noun implies a class of objects.
- A proper noun or direct reference implies an instance of a class.
- A collective noun implies a class of objects made up of groups of instances of another class.
- An adjective implies an attribute of an object.
- A doing verb implies an operation.
- A being verb implies a classification relationship between an object and its class.
- A having verb implies an aggregation or association relationship.
- A transitive verb implies an operation.
- An intransitive verb implies an exception.
- A predicate or descriptive verb phrase implies an operation.
- An adverb implies an attribute of a relationship or an operation.



DESCRIPTIVE VERBS

Definition

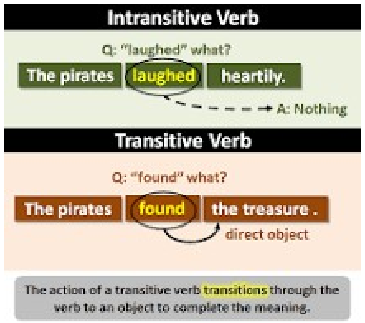
Descriptive verbs provide more detailed or vivid information about the action being performed. Instead of using more common or generic verbs, descriptive verbs can give a clearer picture of the intensity, manner, or specific way an action takes place.

Examples

By replacing "walk" with a more descriptive verb, we can convey more about the way someone is walking:

- *He strolled into the room.*
- *She sprinted to the finish line.*
- *The cat crept around the yard.*

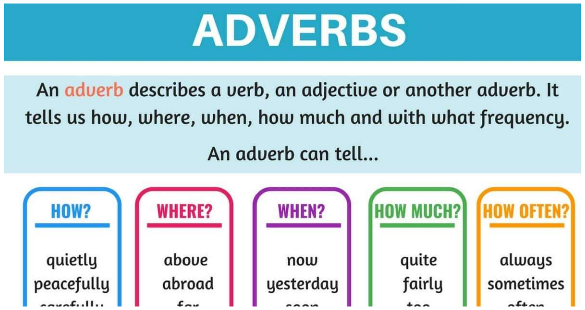
www.Games4esl.com



Intransitive Verb

Transitive Verb

The action of a transitive verb transitions through the verb to an object to complete the meaning.



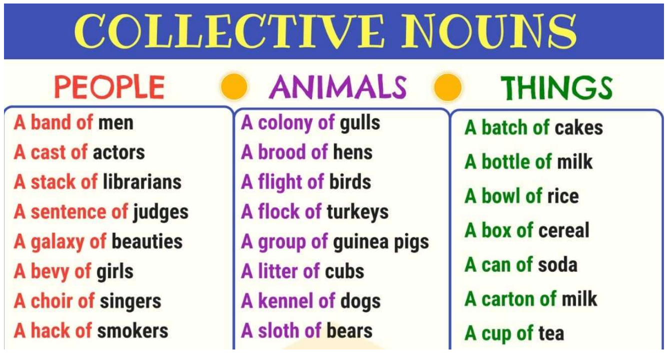
ADVERBS

An adverb describes a verb, an adjective or another adverb. It tells us how, where, when, how much and with what frequency.

An adverb can tell...

HOW?	WHERE?	WHEN?	HOW MUCH?	HOW OFTEN?
quietly	above	now	quite	always
peacefully	abroad	yesterday	fairly	sometimes
carefully	far	soon	too	often

https://7esl.com/



COLLECTIVE NOUNS

PEOPLE	ANIMALS	THINGS
A band of men	A colony of gulls	A batch of cakes
A cast of actors	A brood of hens	A bottle of milk
A stack of librarians	A flight of birds	A bowl of rice
A sentence of judges	A flock of turkeys	A box of cereal
A galaxy of beauties	A group of guinea pigs	A can of soda
A bevy of girls	A litter of cubs	A carton of milk
A choir of singers	A kennel of dogs	A cup of tea
A hack of smokers	A sloth of bears	

Chapter Review

- Describe the purpose of a structural model.
- Describe the different elements of a structural model.
- Explain the difference between abstract and concrete classes.
- Describe the three general types of relationships typically used in a structural model.
- Create a structural model using textual analysis of use-case descriptions, brainstorming, and patterns.
- Explain the purpose of a CRC card in structural modeling.
- Create a structural model using CRC cards.
- Describe the different elements of a CRC card.
- Describe how to role-play CRC cards using use-case scenarios.

Chapter Review - Cont

- Describe the different elements of a class diagram.
- Describe the four basic operations that can be represented on a class diagram.
- Explain the differences between the types of relationships supported on a class diagram.
- Create a class diagram that represents a structural model.
- Describe the different elements of an object diagram.
- Create an object diagram that represents an instantiation of a portion of a class diagram.
- Verify and validate the evolving structural model using role-playing and walkthroughs.
- Verify and validate the structural model by ensuring the consistency of the CRC cards and class diagram.
- Verify and validate both the structural and functional models by balancing the two.

Key Terms

- A-kind-of
- A-part-of
- Abstract class
- Aggregation association
- Assemblies
- Association
- Association class
- Attribute
- Brainstorming
- Class
- Class diagram
- Class-Responsibility-Collaboration (CRC)
- Client
- Collaboration
- Conceptual model
- Concrete class
- Constructor operation
- Contract
- CRC cards
- Decomposition
- Derived attribute
- Destructor operation
- Doing responsibility
- Domain classes

Key Terms - Cont

- Generalization association
- Has-parts
- Information hiding
- Instance
- Instantiation
- Knowing responsibility
- Method
- Multiplicity
- Object
- Object diagram
- Operation
- Package
- Parts
- Pattern
- Private
- Protected
- Public
- Query operation
- Responsibility
- Role-playing
- Roles
- Server
- Specialization
- State
- Static model
- Static structure diagram
- Structural model
- Subclass
- Substitutability
- Superclass
- SVDPI
- Tangible things
- Textual analysis
- Update operation
- View
- Visibility
- Wholes