Ch. 4: Part-A
Classification and Basic Behavioral Modeling in UML

H.S. Sarjoughian

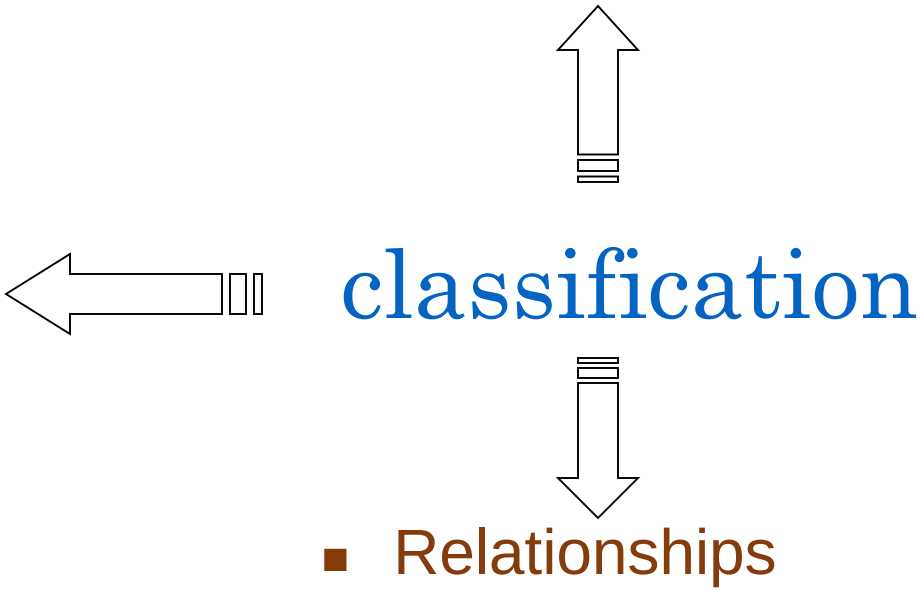
CSE 460: Software Analysis and Design

School of Computing, Informatics and Decision Systems Engineering
Fulton Schools of Engineering

Arizona State University, Tempe, AZ, USA

Basic Principles and Some Artifacts of OOAD

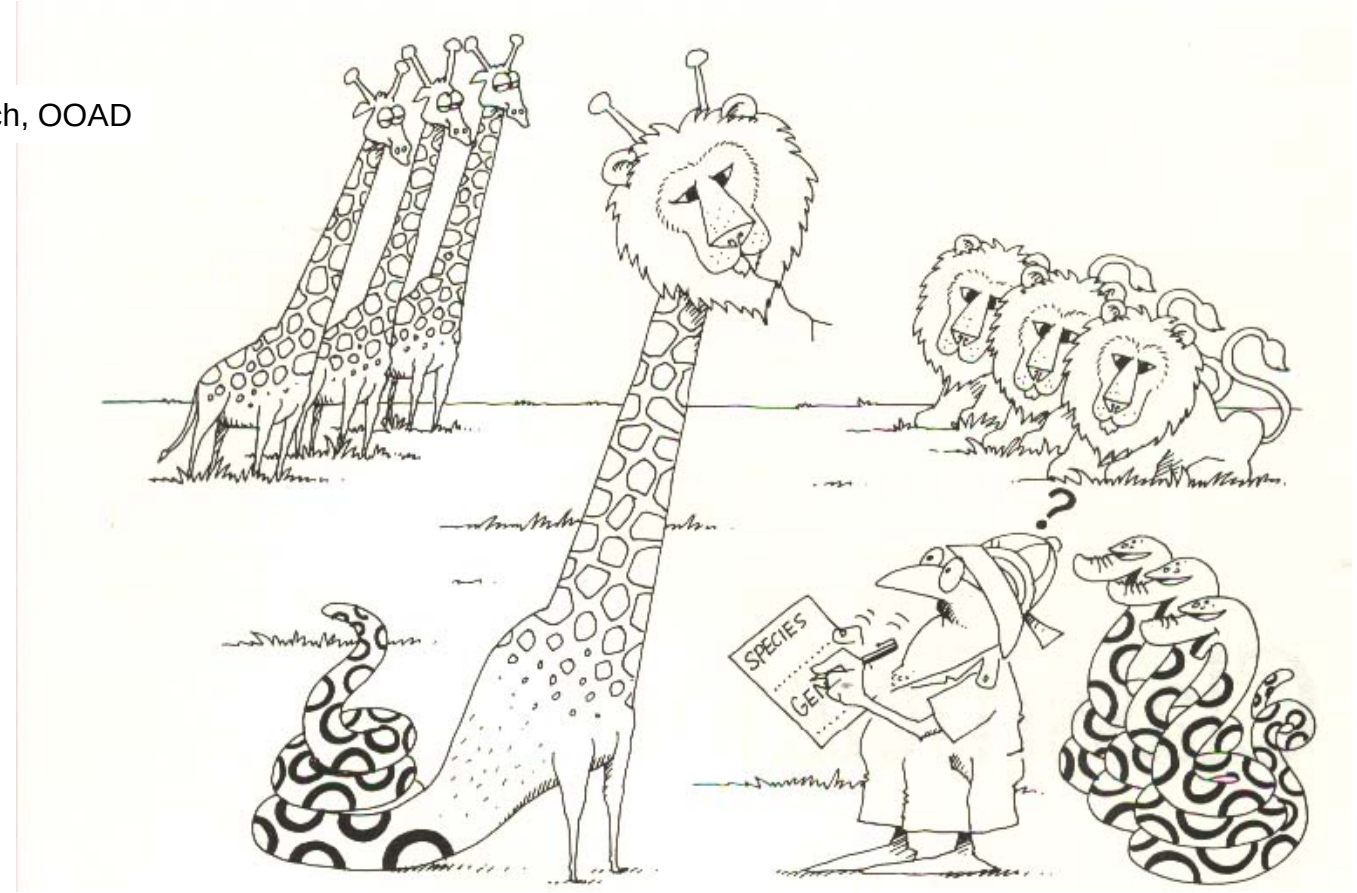
- Abstraction
- Encapsulation
- Modularity
- Hierarchy



Specifying objects and classes has to be considered in terms of specific application domains

Classification Caricature

G. Booch, OOAD



classification is the means whereby we order knowledge

Classification and Its Perils

- Classification underlies creating knowledge taxonomies
- Classification aims at finding *common aspects* of things (key abstractions and mechanisms) and therefore supporting smaller and simpler software
- Classification is **not unique**
  - Choice of one classification vs. another is largely arbitrary – depends on an observer's view for a given domain of discourse (e.g., public and private transportation)
  - No single "perfect" classification can exist – multiple classifications may be necessary to satisfy various needs of analysis and design (e.g., public transport system – traffic management vs. state/federal policies)

"the discovery of an order is no easy task. … yet once the order has been discovered there is no difficulty at all knowing it" [Descartes]

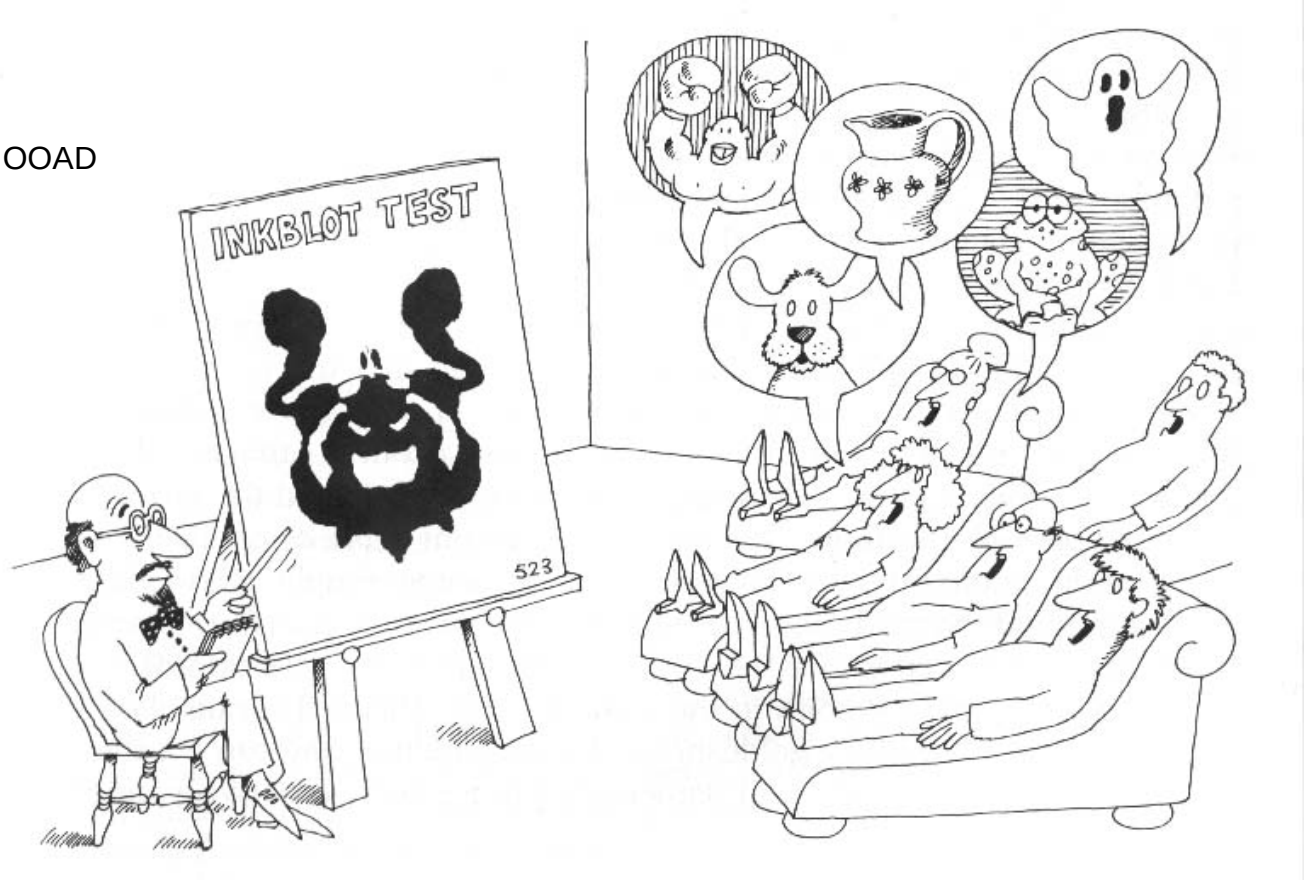
Classification and Its Perils (cont.)

- Are there general methods (automated rules and algorithms) for identifying classification of objects and classes? NO
  - "it's a Holy Grail. There is no panacea" [Stroustrup]
  - "that's a fundamental question for which there is no easy answer. I try things" [Gabriel]

there does not exist the "perfect" class structure; there does not exist the "right" set of objects – however, ***some are better than others!***

Alternative Classifications

G. Booch, OOAD



different observers will classify the same object differently

Discovery and Invention

Identification of classes and objects is one of the **hardest parts** of the object-oriented analysis and design work

DISCOVERY:

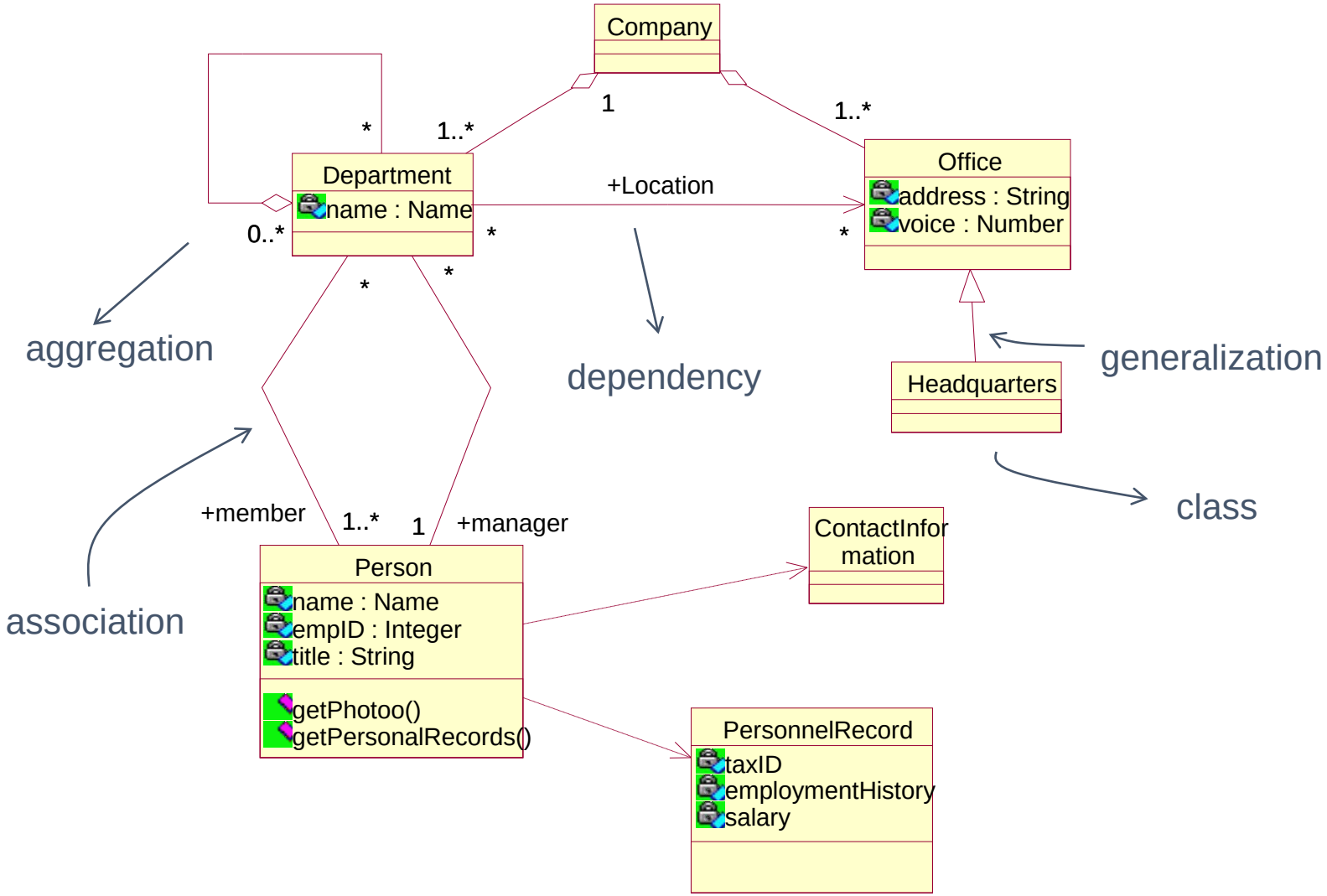
- Establishing the vocabulary of the problem and solution domains;
- Recognizing key abstractions and relationships – identifying aggregation, generalization/specializing, and dependency

INVENTION:

- Devising mechanisms through which higher-order (collaborative) behavior can be achieved
  - Aggregation, generalization/specializing, and dependency
  - Modularization
  - Performance
  - Resource allocation (local vs. distributed)
  - …

through discovery and invention we strive to develop common structures that exhibit common behavior

Company Example



Example: Classifying Trains

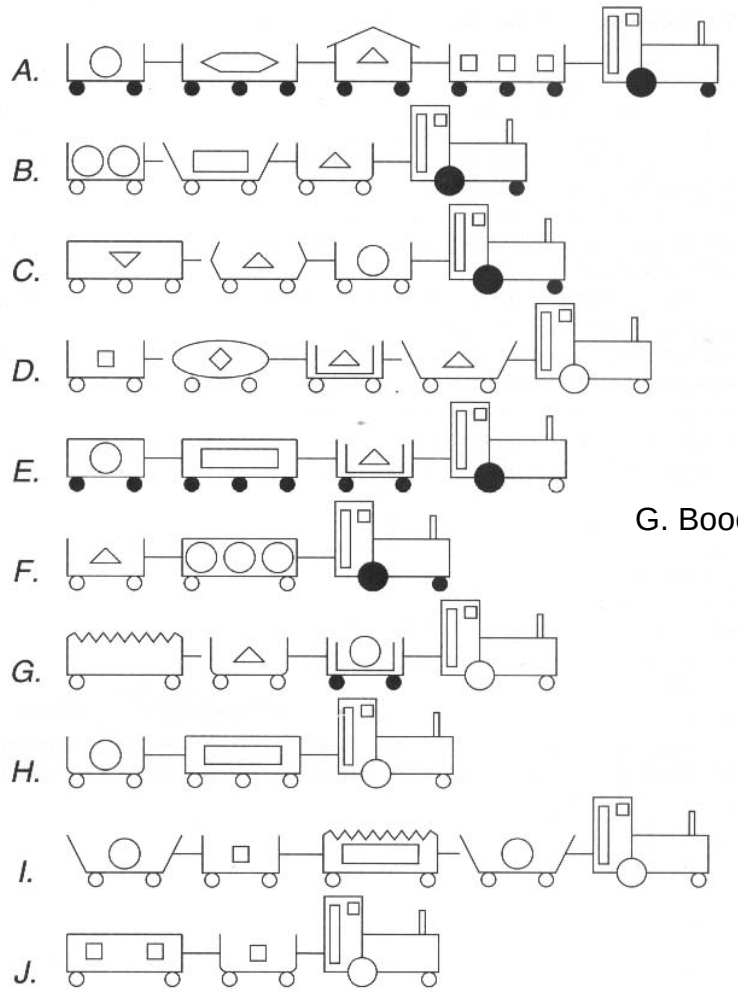
Each train has:

- engine (1)
- cars (2-4)

A large number of groups are possible

93 distinct categories – e.g.,

- trains whose engine have all black wheels
- three-car trains having identical middle car
- …



G. Booch, OOAD

Iterative and Incremental Classification

A macro classification lifecycle process for a given problem

a) Create "first-generation" classifications (abstractions)
b) Examine the quality (usefulness) of alternative classifications
c) Select most effective, suitable classifications

- size of class and object diagrams
- decomposition and composition
- extensibility

} important features

d) Evolve the classifications in an iterative and incremental fashion until appropriate classifications have been identified

Identification of Classes and Objects

- Approaches to classification originating from philosophers, linguists, mathematicians, and artists among others are
  - **Classical categorization** (classification by properties)
  - **Conceptual clustering** (classification by concepts)
  - **Prototype theory** (classification by association with a prototype)
- The above approaches offer a basis for carrying out object-oriented analysis and design.

Classical Categorization

Identify classes and objects according to the relevant properties for the problem domain of interest

- married people –
  - property: married; values: yes and no
- indigenous plants –
  - property: occurring naturally in a particular region; values: yes and no
- aircraft
  - property: fly; values: yes and no
  - property: fuel; values: yes and no

generally there does not exist a list of properties which may be used to form and distinguish the members of a category

members of a category may belong to other categories (e.g., a person who is both a musician and married)

Conceptual Clustering

- Identify collaborating objects using inexact ("best fit") criteria. Conceptual clustering based on descriptions is related to probabilistic or fuzzy theories
  - portrait picture –
    - happiness; values: jubilant, very happy, …
- Conceptual categories can be used as a basis for forming classical categories – distinct concepts are used to group things with

Prototype Theory

- Identify objects by association to prototypical objects
- Objects of a classification resemble in some significant way a prototypical object – i.e., objects belonging to the classification must share some properties, but not all, with the prototypical object
  - E.g., chair, game, …(a set of common properties does not hold among all games)
- Prototype theory suggests grouping things according to the degree of their relationship to concrete prototypes.

Object-Oriented Analysis

Object-oriented analysis is a **method** of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the **problem domain**

- Understanding problem requirements from the point of customers, users, and system/software engineering team. It **does not focus independently** on the examination of data and functionality

Object-oriented analysis, similar to structured analysis, is **primarily a discovery undertaking**

Analysis Approaches

- Object-oriented approaches
  - Classical approaches
  - Behavioral Analysis
  - Domain Analysis
  - Use-Case Analysis
  - Class/Responsibilities/Collaborators
  - Problem Description
- Classical structured approaches
- Mixed approaches

Classical Approaches

Classical approaches are founded based on the principles of "classical categorization."

- Database modeling [Ross, 1987]
  - People: humans who carry out some function
  - Places: areas allocated for people or things
  - Things: physical objects or group of objects that are tangible
  - Organizations: organized collection of people, resources, and capabilities having specific objectives
  - Concepts: principles or ideas which are not tangible
  - Events: occurrences that happen to something

Behavioral and Domain Approaches

Behavioral analysis: focuses on behavior produced by similar objects – responsibilities expected from objects and knowledge maintained. This view can help to reveal general and specialized behavior which may then be used to form hierarchical relationships.

Domain analysis: aims at identifying objects and classes (and their relationships) from knowledge gathered from domain experts. This approach is also applicable to similar applications within a given domain (e.g., banking, police, IRS, and transportation software systems are instances of the Management Information Systems). This approach focuses on identifying classes and objects across similar applications in a given domain.

References

- *Object-Oriented Analysis and Design with Applications, 3rd Edition, G. Booch, et al. 2007, Addison Wesley, 2007*
- *OMG Unified Modeling Language Specification, UML Standard, http://www.omg.org/technology/documents/formal/uml.htm, Computer Associates International, Inc., 2001*
- *The Unified Modeling Language User Guide,G. Booch, J. Rumbaugh, I. Jacobson, Addison Wesley Object Technology Series, 1999*
- *Software Engineering: A Practioner's Approach, 8h Edition, R.S. Pressman, McGraw Hill, 2000*