CLASSIFICATION

- What is Classification?
- Classification and OOD
- Why is Classification So Hard?
- Approaches to Classification
- Object-Oriented Analysis
- Domain Analysis
- Alternate Approaches
- Key Abstractions and Mechanisms

WHAT IS CLASSIFICATION?

Classification **- the means whereby we order knowledge**

In OOD, recognizing the sameness among things allows us to expose the commonality within key abstractions and mechanisms and eventually leads to simpler designs.

However, there is no simple approach to the problem of identifying classes and objects. The selection of classes and objects for an OOD is a compromise shaped by many competing factors.

This module focuses on heuristics useful for identifying the classes and objects relevant to a particular problem.

CLASSIFICATION AND OOD

The identification of classes and objects is the hardest part of OOD.

The identification of classes and objects involves:

- **discovery, through which we recognize the key abstractions and mechanisms that form the vocabulary of our problem domain**
- **invention, through which we devise generalized abstractions and new mechanisms that regulate how objects should collaborate**

During classification, we group entities that have a common structure or exhibit a common behavior.

Classification is highly dependent upon the reason for the classification, and different observers naturally tend to classify the same thing differently.

The best classifications result when an incremental and iterative process is applied. The quality of a classification can only be meaningfully evaluated at later stages in the design, once clients have been built which use the abstractions.

WHY IS CLASSIFICATION SO HARD?

- **There is no such thing as a "perfect" classification, although some classifications are better than others.**
- **Any classification is relative to the perspective of the observer doing the classification.**
- **Intelligent classification requires a tremendous amount of creative insight.**

Why is a LASER beam like a goldfish?
Because neither one can whistle.

Creative insight or idocy?

How is a speck of dust like a thought?
Both can be conceived of.

APPROACHES TO CLASSIFICATION

- Classical categorization
- Conceptual clustering
- Prototype theory

Classical Categorization

Classical Categorization **- all entities that have a given property or set of properties in common form a category; such properties are necessary and sufficient to define the category**

-- Lakoff, G. *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*, 1987, The University of Chicago Press, Page 161

Related properties are therefore the criteria for sameness among objects in the Classical Categorization approach. One can divide objects into disjoint sets depending on the presence or absence of a particular property. Properties to be considered are domain-specific.

Marvin Minsky has suggested that "the most useful sets of properties are those whose members do not interact too much. This explains the universal popularity of that particular combination of properties: size, color, shape, and substance."

-- Minsky, M. *The Society of Mind*, 1986, Simon and Schuster, New York, Page 199

Conceptual Clustering

Conceptual Clustering **- a modern variation on the classical approach in which classes (clusters of entities) are generated by formulating conceptual descriptions of these classes and then classifying the entities according to the descriptions**

Conceptual clustering is a probabilistic clustering of objects.

Prototype Theory

Prototype Theory **- based on work in the field of cognitive psychology, a class of objects is represented by a prototypical object, and an object is considered to be a member of this class if and only if it resembles this prototype in some significant ways**

Prototype Theory is often applied when classical categorization and conceptual clustering fail. For instance, try to identify entities which fall into a class called "game" by classical categorization or conceptual clustering.

Classification and OOD Revisited

An approach proposed by Grady Booch:

1. **Identify classes and objects according to the properties relevant to the application domain.**
2. **If this fails, cluster objects by concepts.**
3. **If either (1) or (2) fail, classify by association, through which clusters of objects are defined according to how closely each resembles some prototypical object.**

These three approaches to classification provide the theoretical foundation of object-oriented analysis, domain analysis, and other methods applied to identify classes and objects in an object-oriented design.

OBJECT-ORIENTED ANALYSIS

Object-Oriented Analysis **- the process of creating a model of the problem domain by identifying the classes and objects that form the vocabulary of the problem domain**

Object-Oriented Design **- the process of inventing the abstractions and mechanisms that provide the behavior that the model created during OOA requires**

Sources of Classes and Objects

Proposed by Shlaer and Mellor:

- **Tangible things, such as cars, telemetry data, and sensors**
- **Roles, such as mother, teacher, and politician**
- **Events, such as landing, interrupt, and request**
- **Interactions, such as loan, meeting, and intersection**

Proposed by Ross (from the perspective of data modeling):

- **People - humans who carry out some function**
- **Places - areas set aside for people or things**
- **Things - tangible physical objects or groups of objects**
- **Organizations - collections of people, resources, facilities, and capabilities that have a defined mission**
- **Concepts - principles or ideas not tangible used to track activities and/or communications**
- **Events - things that happen**

Sources, Continued

Proposed by Coad and Yourdon:

- **Structure - "kind of" and "part of" relationships**
- **Other systems - external systems with which the application interacts**
- **Devices - devices with which the application interacts**
- **Events remembered - a historical event that must be recorded**
- **Roles played - the different roles users play in interacting with the application**
- **Locations - physical locations, offices, and sites important to the application**
- **Organizational units - groups to which users belong**

DOMAIN ANALYSIS

Domain Analysis **- the process of identifying the classes and objects that are common to all applications within a given domain**

Contrast Domain Analysis to Object-Oriented Analysis, which focuses on one problem at a time.

Domain Analysis is useful for pointing you to the key abstractions that have proven useful in other related systems, giving the designer ideas for the abstractions pertinent in the system under design. Domain Analysis works well because there are very few truly unique kinds of software systems.

Suggested Steps in Domain Analysis

- Construct a generic model of the domain by consulting with domain experts [a *domain expert* is simply a user or a person intimately familiar with the elements of a particular problem].
- Examine existing systems within the domain and represent this understanding in a common format.
- Identify similarities and differences between the systems by consulting with domain experts.
- Refine the generic model to accommodate existing systems.

ALTERNATE APPROACHES

OOA and Domain Analysis are the preferred techniques to use with OOD, but two alternate approaches also merit discussion:

- *Informal English Description* - Underline the nouns and verbs in a written description of the problem. The nouns represent candidate objects and the verbs represent candidate operations on them. Note that a key problem with this approach is that any noun can be verbed and any verb can be nouned, so it is easy to skew the candidate list to emphasize either objects or operations.
- *Structured Analysis* - Use the products of structured analysis as a front end to OOD. Data Flow Diagrams (DFD's) provide a reasonably formal model of the problem, and a DFD can be used to derive candidate objects from the following:
  - ❍ External entities
  - ❍ Data stores
  - ❍ Control stores
  - ❍ Control transformations

  Candidate classes derive from two sources:
  - ❍ Data flows
  - ❍ Control flows

KEY ABSTRACTIONS AND MECHANISMS

Key Abstraction **- a class or object that forms part of the vocabulary of the problem domain**

Values of identifying key abstractions:

- **Key abstractions give boundaries to the problem.**
- **Key abstractions highlight the things that are in the system and therefore relevant to an OOD.**

The identification of key abstractions involves two processes:

- ***discovery*, through which the abstractions used by domain experts are recognized**
- ***invention*, through which designers create new classes and objects that are not necessarily part of the problem domain but are useful artifacts in the design or implementation**

Refining Key Abstractions

- Evaluate the abstraction based on metrics previously discussed.
- Place the abstraction in the context of the existing class and object hierarchies. Placing classes and objects at the right levels is difficult:
  - Sometimes we employ *class promotion*, in which we find a general subclass and choose to move it up in the class structure.
  - Sometimes a *grainsize conflict* is found, wherein a class is too general, making inheritance by a subclass difficult due to the semantic gap.
- Name things properly, so that they reflect their semantics. This is important in capturing the essence of the abstractions being described. Some suggestions:
  - Objects should be named with proper noun phrases
  - Classes should be named with common noun phrases
  - Modifier operations should be named with active verb phrases
  - Selector operations should imply a query or be named with verbs of the form "to be"

Identifying Mechanisms

Once the key abstractions are identified, behaviors are added to these abstractions to derive the observable behaviors of the system.

Mechanism - any structure whereby objects work together to provide some behavior that satisfies a requirement of the problem; mechanisms represent strategic design decisions

It is socially unacceptable for objects to step outside the boundaries of the rules of behavior dictated by a particular mechanism (such as the car's lights coming on when the driver depresses the accelerator).