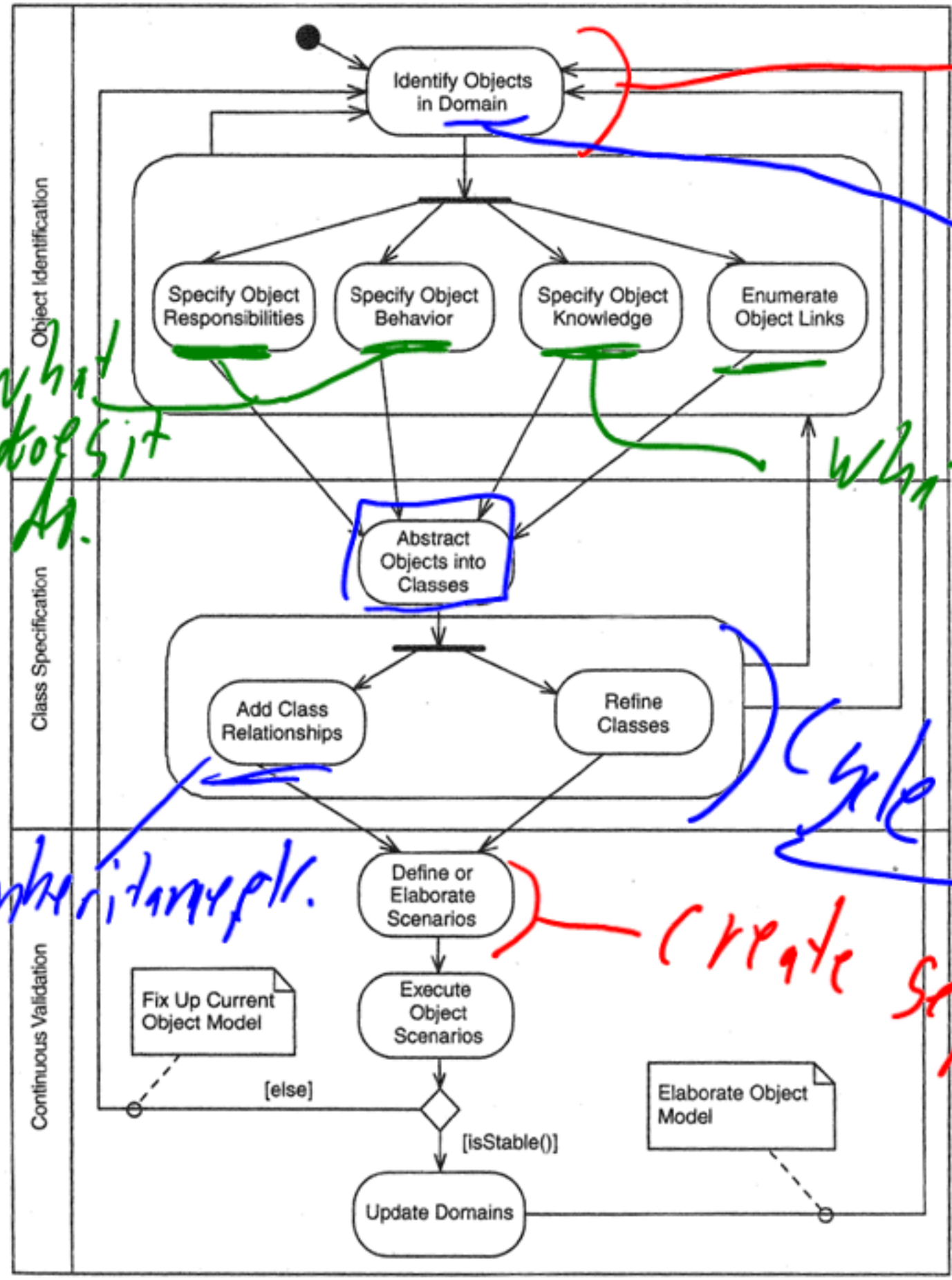
Domain Model

Objectives

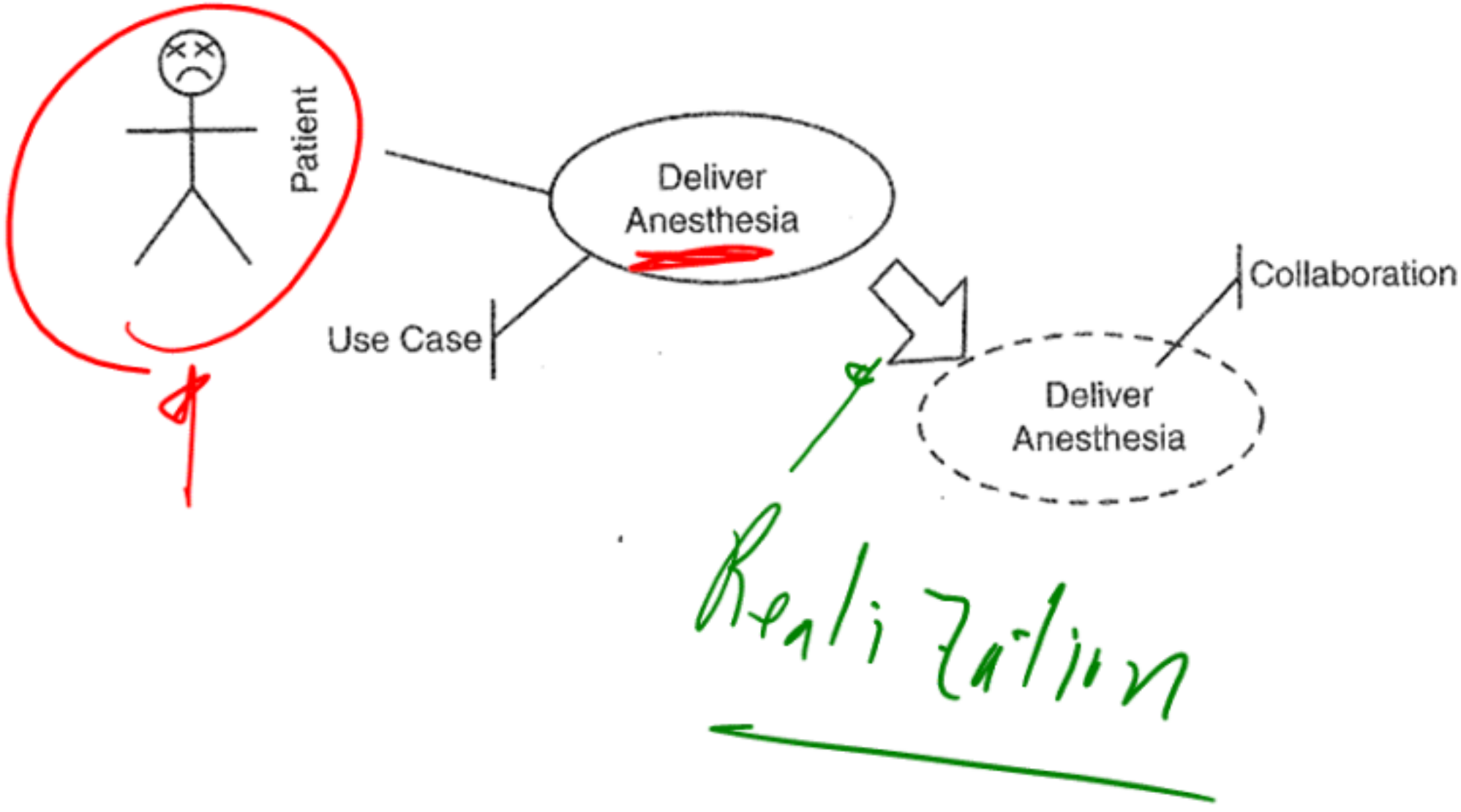
- Explain the purpose for Object Domain Analysis.
- Explain the relationship between the use case model and the Object Model
- Explain the concept of UML packages and Domains
- Explain the mechanism used to connect object domain models with use case models
- Apply key strategies for identifying objects within a problem domain
  - Noun Strategy
  - Services
  - Physical Devices
  - Persistent Information

Ropes Domain Analysis

Process Flow



Civil War Surgery


Patient
Deliver Anesthesia
Use Case
Collaboration
Deliver Anesthesia
Realization

Domain Model

Object Model Realizing the Use Case

Knob

selects

sets

Ventilator

provides gas flow info

Vaporizer

setAnesthesiaDepth
getAnesthesiaDepth
setAgentPercent
getAgentPercent
setAgentType
getAgentType

Agent Reservoir

Agent Type
Level

select
getLevel

Actor

Patient

VaporizerView

shows

AlarmingClass

issues alarms

Alarm Manager

silences

monitors depth

How deep asleep is the person?

EMG Monitor

shows

EMGView

raises

Alarm

shows

AlarmView

Button

Display

Monitoring

Display

Control

Display

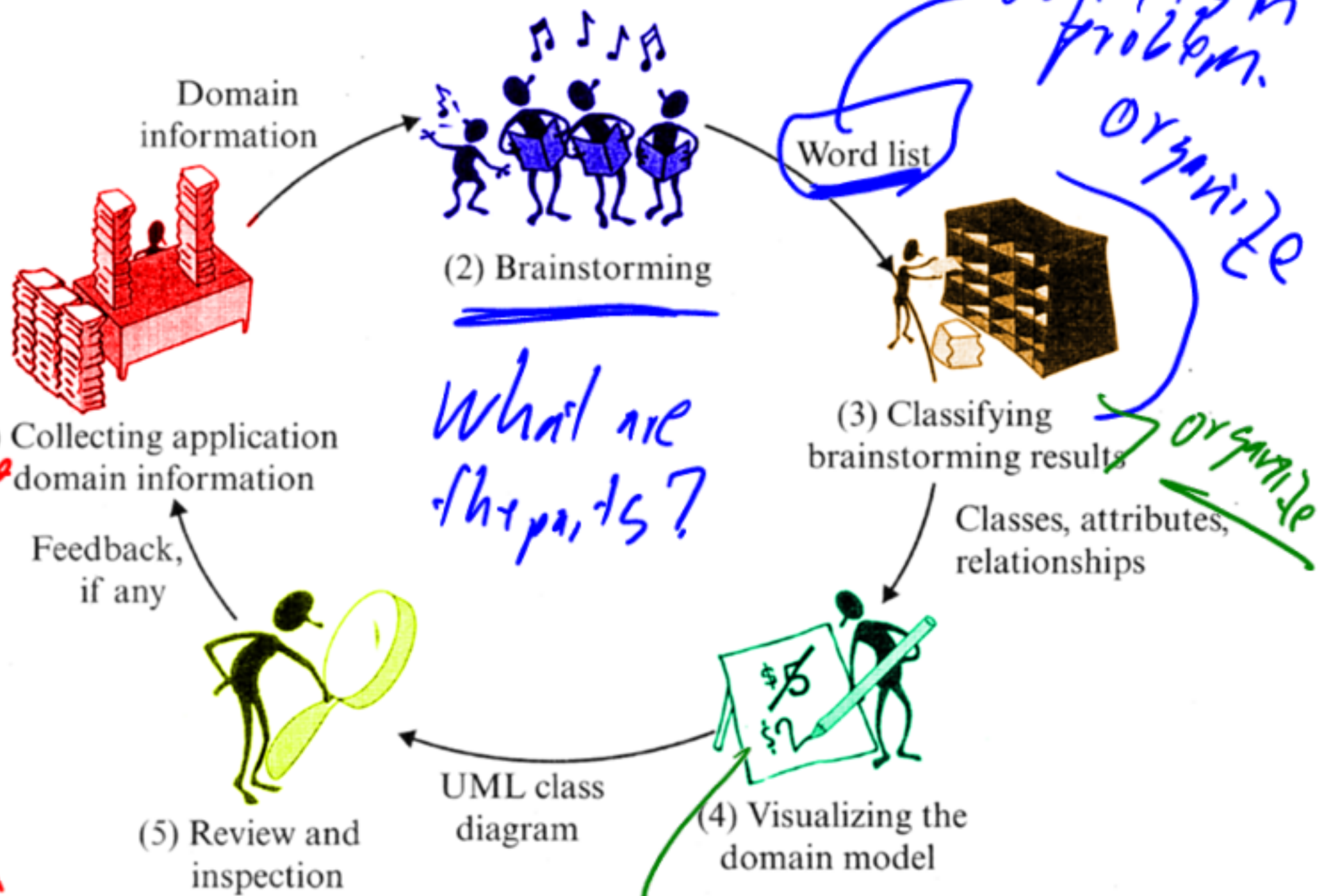
User input

Alarms / things that can go wrong.

Deals w/ managing sets of Alarms

Domain Modeling

(Kung: Object oriented Software Engineering)



How do we define the Candidate Objects

How do we define the Objects

- Underline the noun
- Identify causal objects
- Identify services
- Identify messages and information flow
- Identify real world items
- Identify physical devices
- Identify key concepts
- Identify transactions
- Identify persistent information
- Identify visual elements
- Identify control elements
- Apply scenarios

Strategies

things that have actions

things that exist

data

UI

what / how we use sw.

Noun Identification

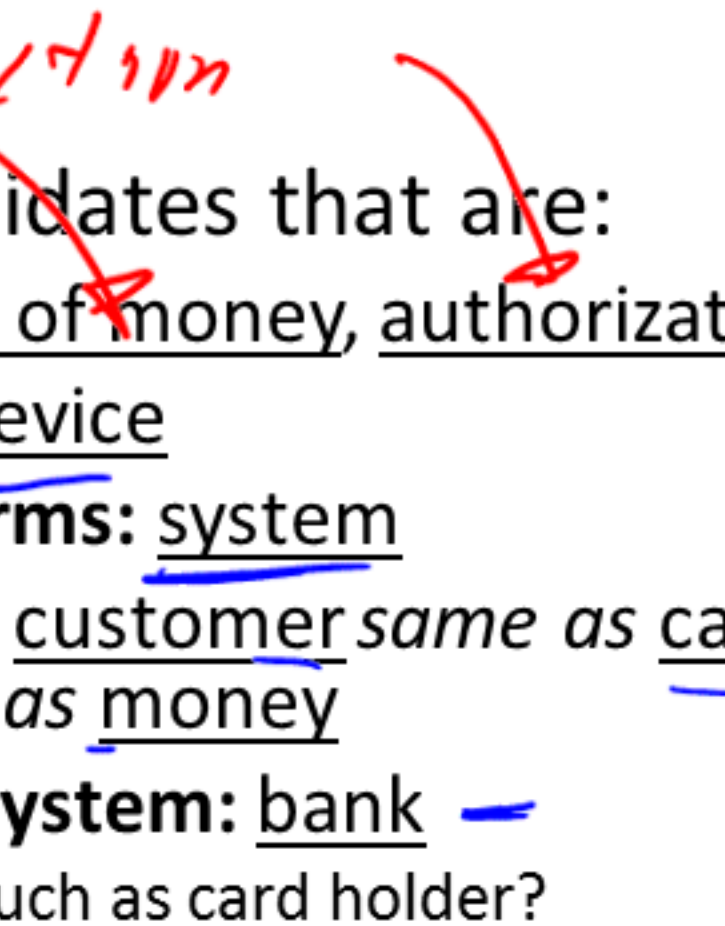
- A way of finding "enduring domain objects":
- A "data driven development" (DDD) method
  - Useful in early stages of development (analysis)
  - Provides "first-cut" set of problem classes
- Useful in analysis of requirements:
  1. Pick out all nouns/noun phrases in requirements documentation
  2. Filter out 'inappropriate' expressions
  3. Use remaining phrases as candidate classes

"An ATM is an electronic device designed for automated dispensing of money. A card holder can get a balance, or withdraw money from an account after authorization. Customers interact with the system through a simple interface consisting of a display screen, a card reader and a numeric keypad. Cash is obtained from a dispenser. Bank members are card holders who can access additional functions such as ordering a statement from the bank"

ATM Case Study: Finding Candidate Classes

Stage 2: Discard candidates that are:

- **Events:** dispensing of money, authorization
- **Vague:** function, device
- **Meta-language terms:** system
- **Redundant terms:** customer *same as* card holder; cash *same as* money
- **Outside scope of system:** bank
  - but retain actors such as card holder?
- **Constitute everything:** ATM (system)
  - Avoid monolithic design!



ATM Case Study: Finding Candidate Classes

- Remaining nouns represent candidate domain objects/classes:
  - **Actor/Role entities**: card holder, bank member
  - **Boundary (interface) objects**: interface, card reader, keypad, display screen, cash dispenser
  - **Entity (domain) objects:** balance, money, account, statement
- This analysis provides the initial set of classes in the ATM class model.

- http://thenxtstep.blogspot.com/2011/05/nxt-elevator.html

Watch a video

Underline the noun

A software system *must control a* set of eight Acme elevators *for a* building *with 20* floors. *Each* elevator *contains a* set of buttons, *each corresponding to a desired* floor. *These are called* floor request buttons, *since they indicate a* request *to go to a specific* floor. *Each* elevator *as well has a* current floor indicator *above the* door. *Each* floor *has two* buttons *for requesting* elevators *called* elevator request buttons, *because* they *request an* elevator.

Each floor *has a sliding* door *for each* shaft *arranged so that two* door halves *meet in the center when closed. When the* elevator *arrives at the* floor, *the* door *opens at the same time the* door *on the* elevator *opens. The* floor *does have both* pressure and optical sensors *to prevent closing when an* obstacle *is between the two* door halves. *If an* obstruction *is detected by either* sensor, *the* door *shall open. The* door *shall automatically close after a* timeout period *of 5 seconds after the* door *opens. The detection of an* obstruction *shall restart the* door closure time *after an* obstruction *is removed. There is a* speaker *on each* floor *that pings in response to the arrival of an* elevator.

On each floor (except the highest and lowest), *there are two* elevator request buttons, one for UP *and* one for DOWN. *On each* floor, *above each* elevator door, *there is an* indicator *specifying the* floor *that the* elevator *is currently at and another* indicator *for its current* direction. *The* system *shall respond to an* elevator request *by sending the nearest elevator that is either idle or already going in the requested* direction. *If no elevators are currently available, the* request *shall pend until an* elevator *meets the above-mentioned* criterion. *Once pressed, the* request buttons *are backlit to indicate that a* request *is pending. Pressing an* elevator request button *when a* request *for that* direction *is already pending shall have no* effect. *When an* elevator *arrives to handle the* request, *the* backlight *shall be removed. If the* button *is pressed when an* elevator *is on the* floor *to*

Underline the noun

system (1)	elevator (8)	building (1)
floor (20)	request	button
floor request button (8*20)	elevator request button (20*2)	current floor indicator (8)
door (20*8 + 8)	optical sensor	obstruction
pressure sensor	speaker	UP button
door half	elevator door	indicator
DOWN button	floor door	secondary pressure sensor
internal door set	OPEN button	CLOSE button
elevator control panel	alarm	central station
EMERGENCY CALL button	elevator request	door closure timer
electrical power	telephone	elevator occupants
Stop-Run Switch	switch	message
emergency locks	alarm area	mechanical locking clamp
pressure sensor	tracks	electrical power source

of the system. Emergency

MSOE

Identifying the causal Objects

- Behavioral Active Objects
  - Produce / control action
  - Provide interfaces to people
  - Store information
  - Provide services to others

Do things.

Elevator Request button

Servi

- Objects which provide passive contributions to the system
  - Storage
  - Sensors

Monitoring the system.

↳ Door sensor.

Real World Items

- SSN
- Name
- Address
- Sex
- Etc.

"Data items"

Information

Physical Devices

- Hardware within your system
- Physical Devices

- Objects that must persist for a finite amount of time and represent interactions between objects

Transactions

Things that happen n/ time.

Example

- Play CD Use Case
  - This use case allows a user to play an audio CD on a car radio.
- Lets do a a Domain model for this

System Description

- A system is to be built which will enable a driver to hear an audio CD. The CD may contain up to 99 tracks or up to 79 minutes of music. The driver will have the ability to fast forward or rewind tracks as well as skipping undesirable tracks by hitting buttons on the bezel. The volume of the audio can be adjusted using a volume knob, and the bass and treble settings can be adjusted as well to suite the driver's preference. The track number and current elapsed time shall be displayed on a display for the driver.