Analysis

Phase 2

Analysis

- During analysis we try to create a model of the problem domain as a collection of interacting objects/dependent classes
    - This is the **domain model** (aka analysis model)
- Have to find...
    - the classes
    - the relations (associations) between classes
    - to a lesser degree; possible attributes of the classes, possible behavior (methods) of the classes, the objects life cycles (should they survive the execution of the program)
- Model expressed in UML
    - Class diagram (a static view)

Finding the Domain Model

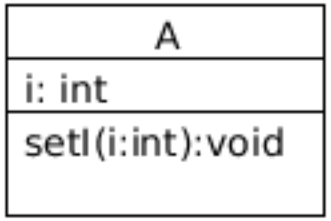
- Have the RAD use it....!
- Simple method
    - Underline **nouns** in use cases, will be classes
    - Underline **verbs**, will be methods
    - Find attributes/relationships from text (has, uses, is a, owns, knows, sends, receives, ...)
    - **Include as much as possible**. Easy to skip later, ...
- This is a **critical activity**
    - If model wrong, not complete, inconsistent, ...
    - ... **BIG** trouble later!!!
- This is a **creative activity**, few (no) rules ...

Automate a mess and you get an automated mess!
// The 21 laws of programming

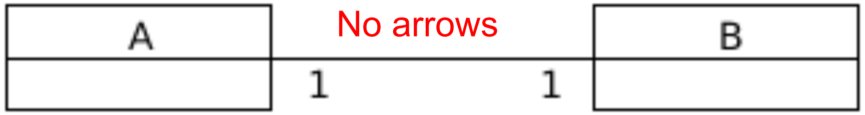
Picking Nouns and Verbs for MP

- From UC Move
  - Dice
  - Piece
  - Board
  - Space
  - Jail
  - Card
  - Property
  - Player
  - Balance

Class Diagram for Domain Model

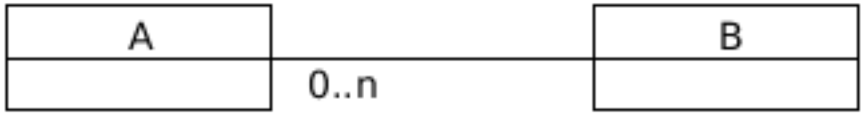


Class named A with attribute i and method setI()

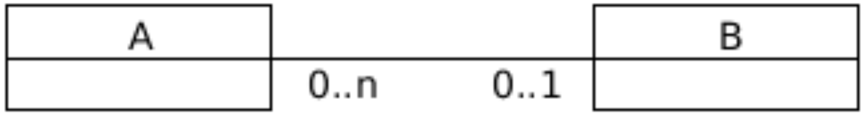


No arrows

One instance of A is associated with one instance of B

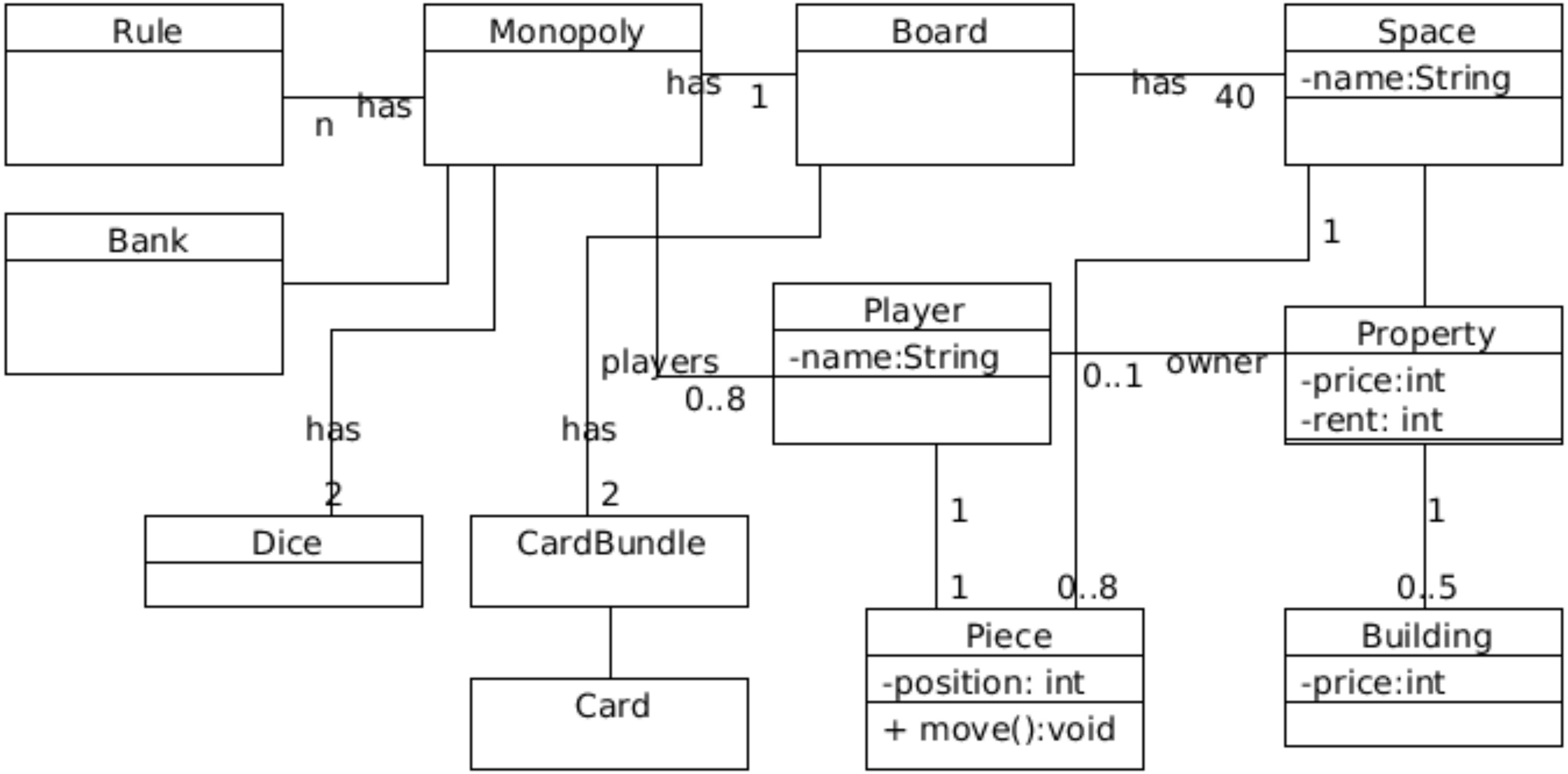


Zero or many instances of A is associated with B



Zero or many instances of A is associated with zero or one B

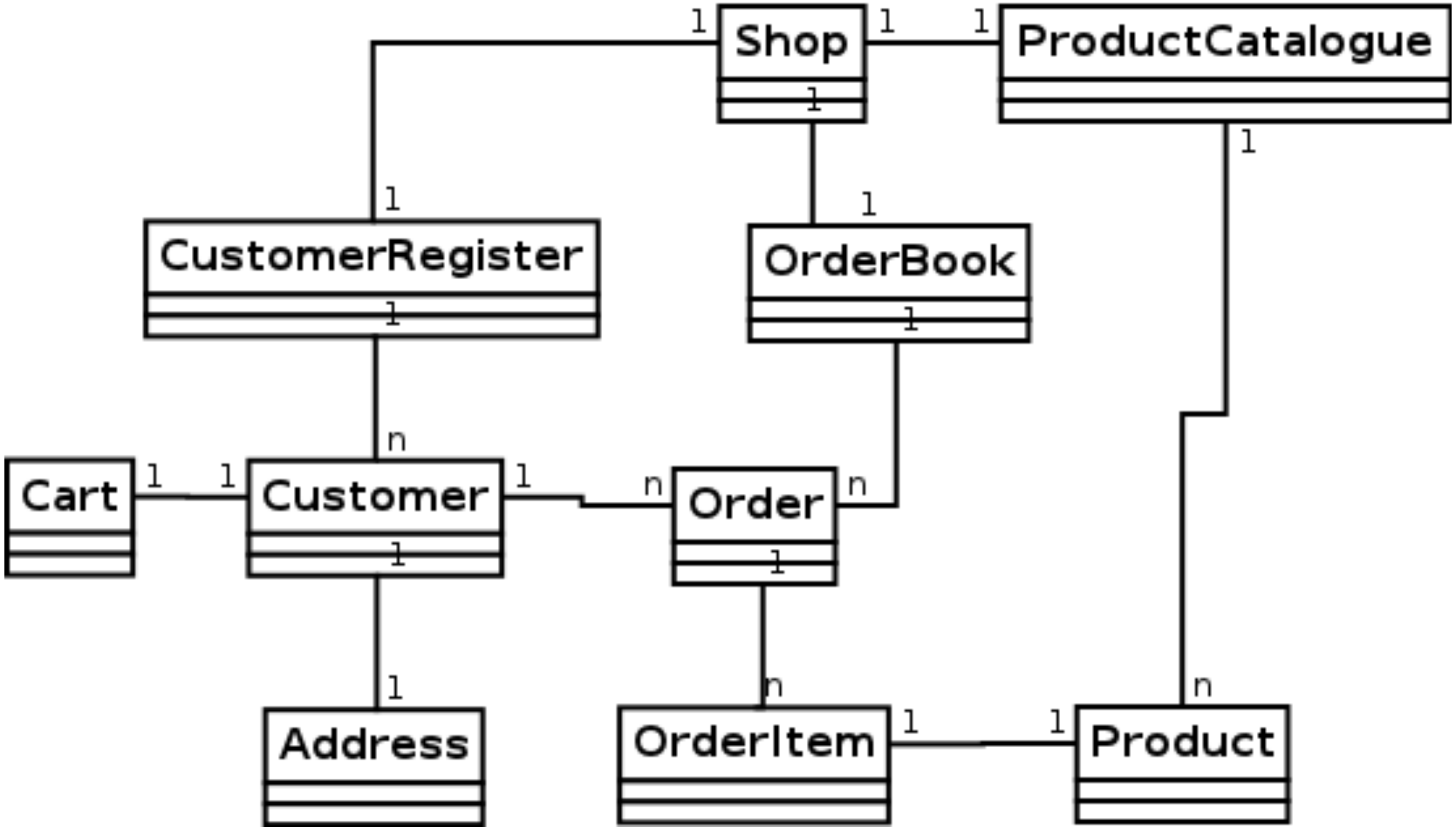
Domain Model for MP



This is **not** the ultimate truth, it's a starts of a solution...

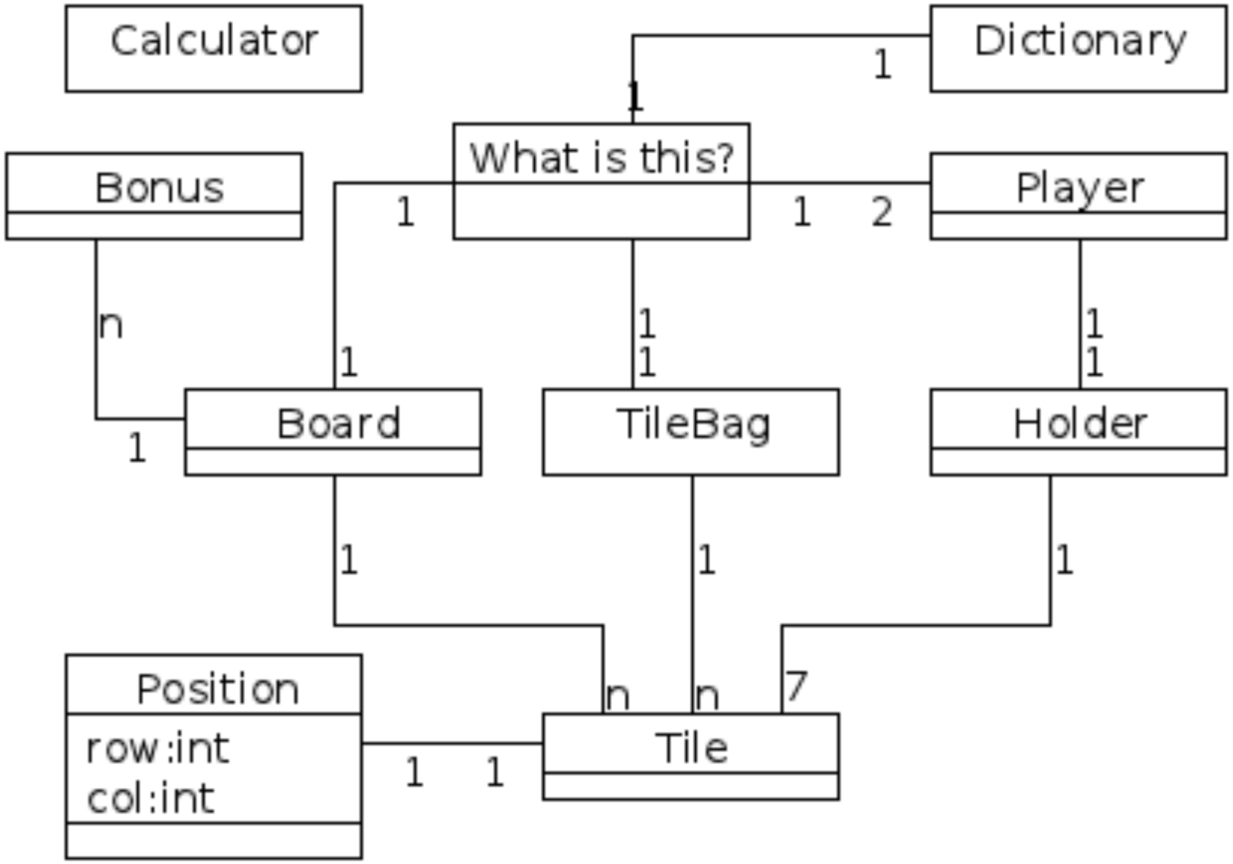
Other Example of Domain Model

A Store



Yet Other Example of Domain Model

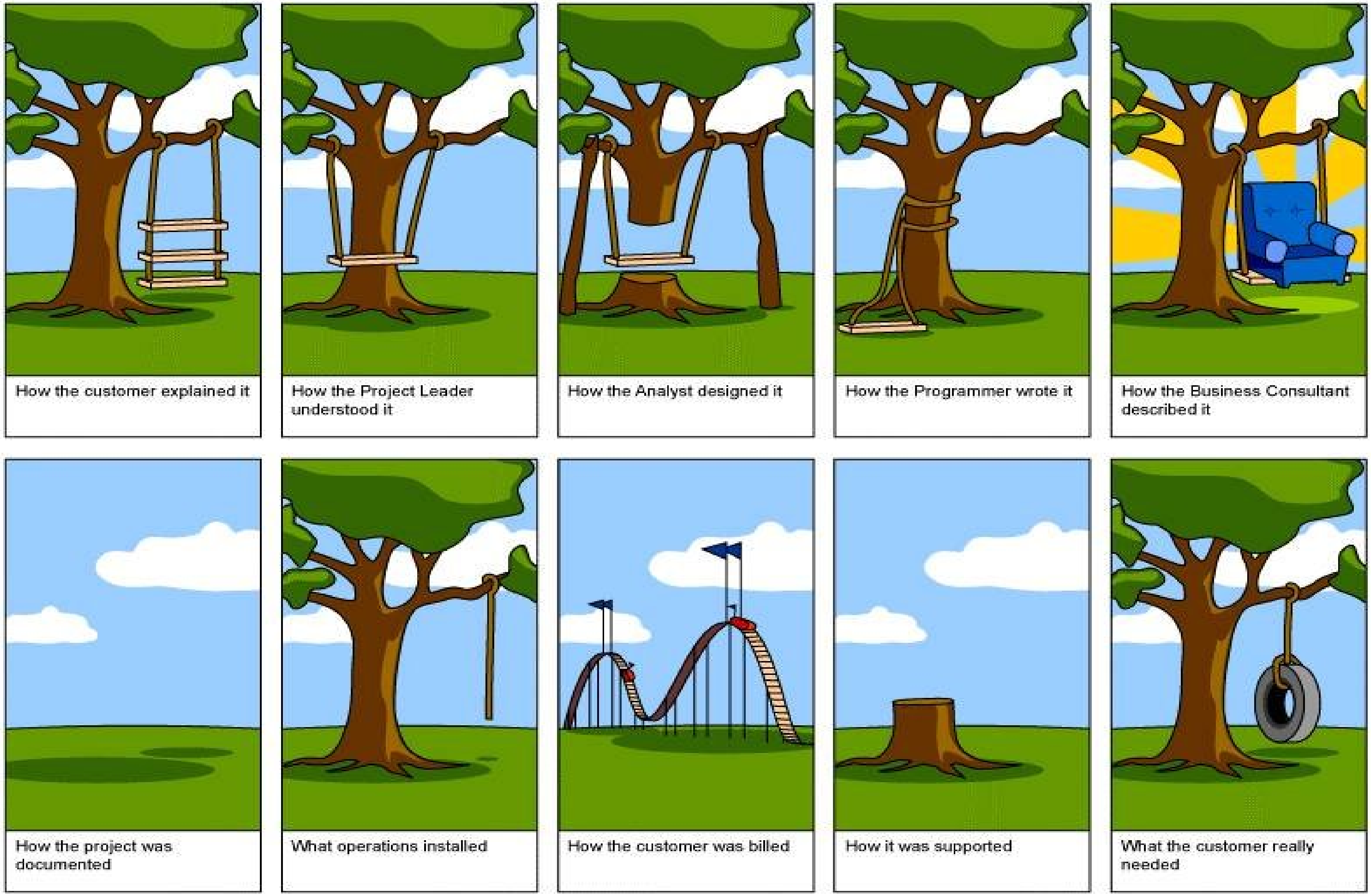
A domain model of what?



Documenting the Analysis

- Analysis is documented in RAD
- Include the domain model diagram (possible updated later)
  - Optimal is to first draw on white board!
  - Later: Tools to draw UML
    - UMLet plugin to Eclipse, fastest possible (I use)
    - Linux : Dia
    - Mac/Win? ...

Hmmm...


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Parallel Development

- During this phase you can, in parallel develop the GUI
- We will not use it for a while (and it will probably change a lot in response to the design and implementation of the model)

Technical Notes for GUI

- The GUI shouldn't be monolithic (i.e one huge JFrame)
- GUI is composed of many preferably small panels
- Let panels act act actionListeners and observers of model
- GUI-drawing tool often generate horrible code
    - Possible many times as much code...
    - Checkout before using in full scale
    - Also: Separate construction code (JButton b = new JButton()) and event code (listeners)
    - Of course no model logic whatever in GUI
    - Possible use a builder class to construct the GUI
    - Support GUI with non-visible classes, Options, Adapters to model (delivering some interface from model to GUI)

GUI Tools

- NetBeans GUI editor
  - Have to move classes from NetBeans to Eclipse, :-(
- Using XML to define the GUI
  - SwiXML
  - BerylXML
  - Links from course page
- Others...???

Summary

- We used the requirements (use cases) to find a model
- We expressed the model as an UML-class diagram
- Documented in RAD
- In parallel we develop (and probably constantly) modify a GUI

Next: From domain model to design model, i.e. design phase