Analysis [Iteration 1, Phase 2]

Slide Series 3

Analysis

During analysis we try to create a model of the problem domain as a collection of interacting objects

- This is the **analysis model** aka **domain model** (I'll use both interchangeably)

Have to find...

- Objects and how they are related
- Classes for the objects
- To a lesser degree; attributes, behavior (methods)

Model represented as UML class diagram

- A static view

Abstraction

The model is often an abstraction of some problem

So what do we mean to abstract?

- Are you good at abstracting?

Remainder: Iterative Development

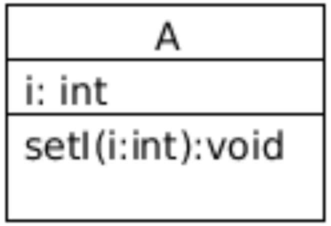
Getting a stable analysis model is a key issue

- Possibly not stable after first iteration …
- … but if not stable after 2-3 … problems!!!
- Don't assume you can fix it by coding, must solve the problem!

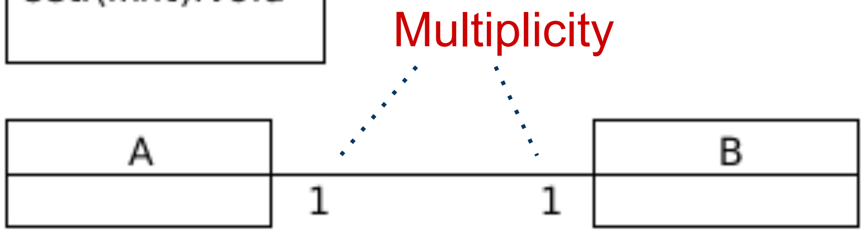
Work really hard to get model stable

Notation for Analysis Model

Make class diagram simple, just names of classes and undirected associations (no arrowheads), multiplicity is useful!

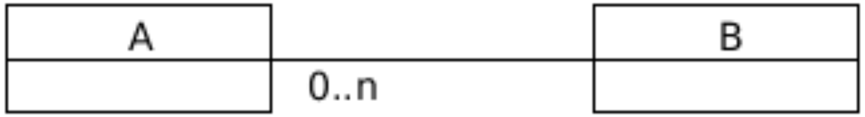


Class named A with attribute i and method setI()

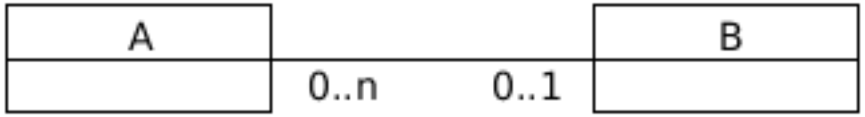


Multiplicity

One instance of A is associated with exactly one (but any) 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

Finding the Analysis Model

Have the UC's from RAD, simple method

- Underline nouns in use cases, will become be classes
- Underline verbs in use cases will become methods
- Find attributes/relationships from text (has, uses, is a, owns, knows, sends, receives, moves, rolls ...)
- **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

Efficient Modelling

Optimal is to first draw on whiteboard!

- Very fast drawing and communication
- Use phone/camera to document
- Very fast communication, everyone can participate

Later, Tools to draw UML

- When model getting more stable
- UMLet plugin to Eclipse, fastest possible (I use)
- Linux : Dia
- Mac/Win? ...

Picking Nouns for MP

From UC Move

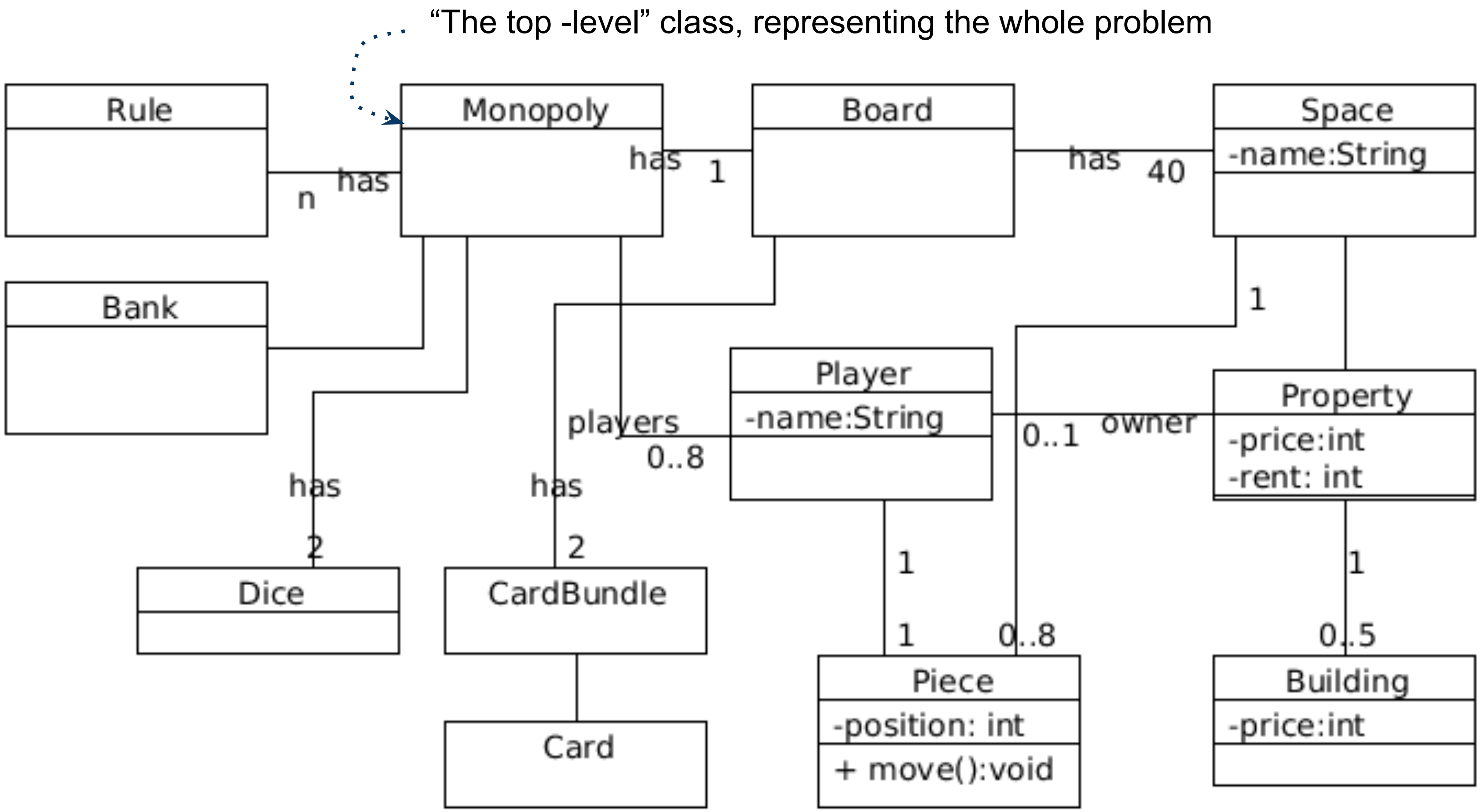
- Dice
- Piece
- Board
- Space
- Jail
- Card
- Rent
- Player
- Balance

Associations for MP

Have found many classes from nouns. How to find the associations?

- From use cases … or (also) …
- … visualize to real situation, board is associated with spaces, deck and cards are associated, also player and piece, etc.
- Skip directions for now, just note the association (next slide)

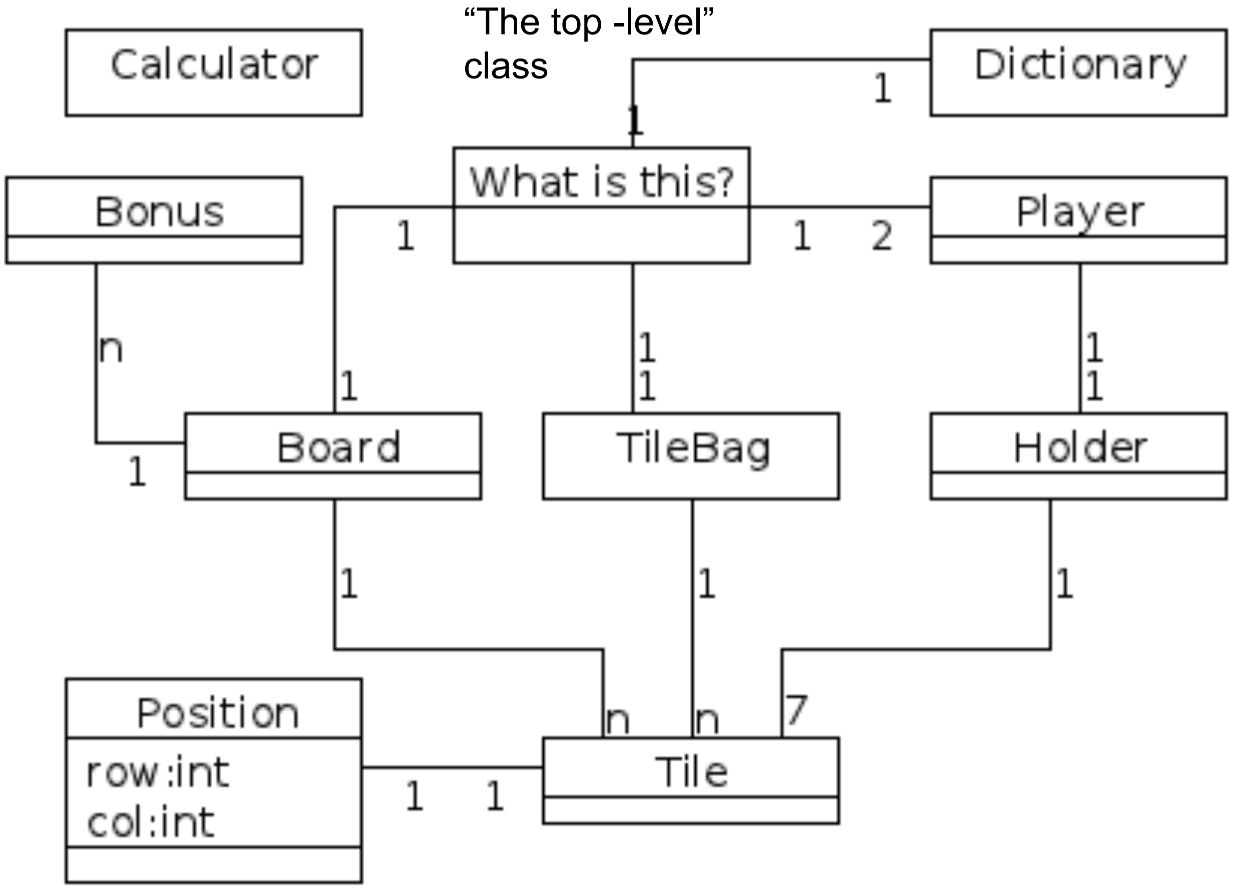
Analysis Model for MP


“The top -level” class, representing the whole problem

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

Another Example of Analysis Model

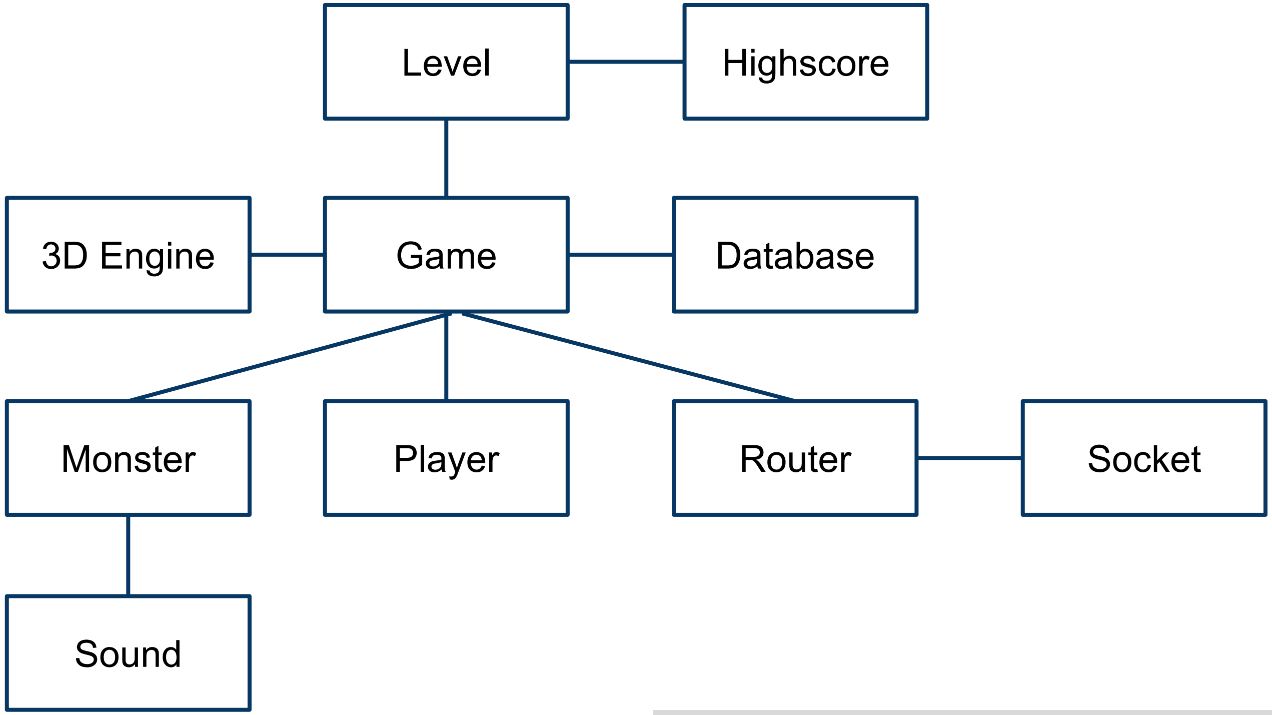
A domain model of what?



Now, let's create a Model…

for this!

How about this model?

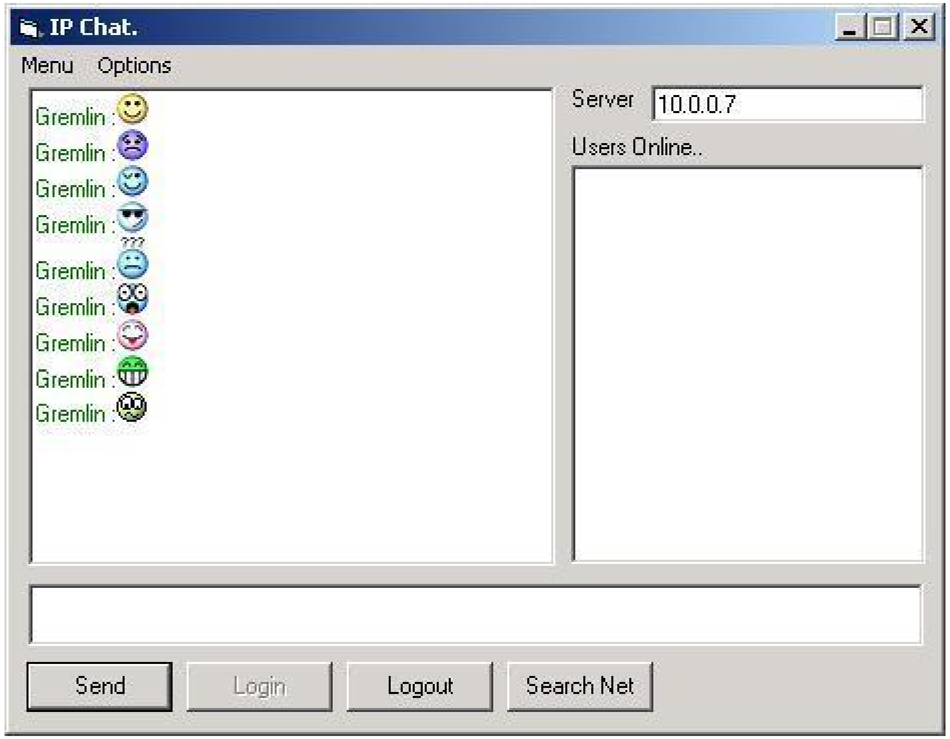


Comment: …

Pitfalls Analysis Model

Assume a chat application

- What's the first that comes to your mind?
- … (to be continued)



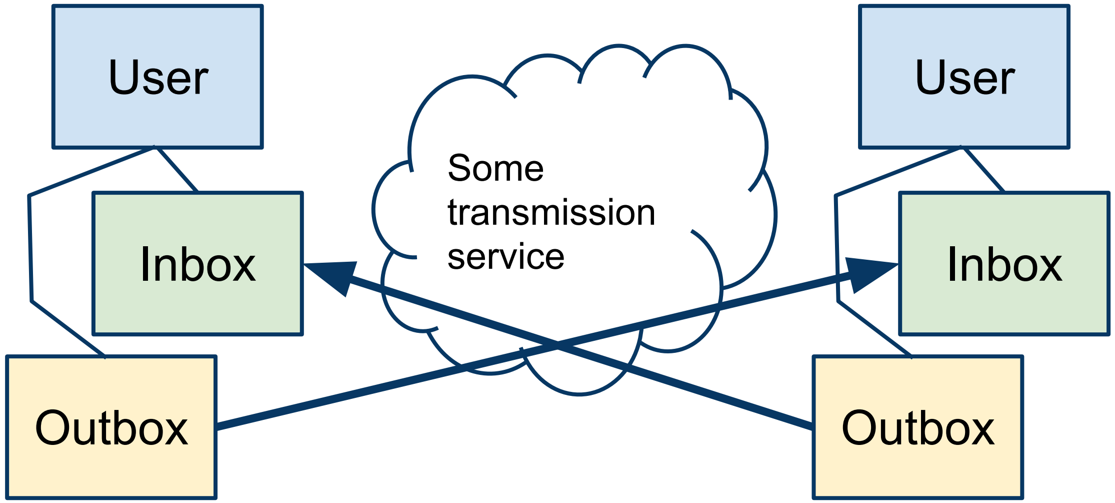
Gremlin :
Gremlin :
Gremlin :
Gremlin :
Gremlin :
Gremlin :
Gremlin :
Gremlin :
Gremlin :

Server 10.0.0.7

Pitfalls Analysis Model, cont.

Hopefully you resisted the temptation to start thinking about networking …

- A chat just deliver messages from the outbox to the inbox (some refinement). The abstraction!
- Network is a technical detail … not in analysis model



Some Object Characteristics

Some valuable characteristics to note for the model objects (if applicable)

- Identity
- Equality
- Persistence. Will any objects survive the execution of the program?
- Lifecycle. When is object created? How long does it exist? When destroyed?
- Use stereotypes to annotate (example <<Persistent>> for surviving objects)

Identity and Life Cycle for MP

Player names must be unique!
Space names must be unique!

No objects will survive … (for now)

All objects (model) created at start of game, exists until end.

More .. ?!?

Prototyping

During this phase you should start out prototyping

- Implement some part/specific aspects (in simplest possible way = quick'n'dirty)

Horizontal Prototype

- Provides a broad view of an entire system or subsystem, focusing on user interaction (mock up functionality)

Vertical Prototype

- Useful for obtaining detailed requirements for a given functionality (Example: Save game -> prototype the persistence functionality all the way from GUI to file on disk)

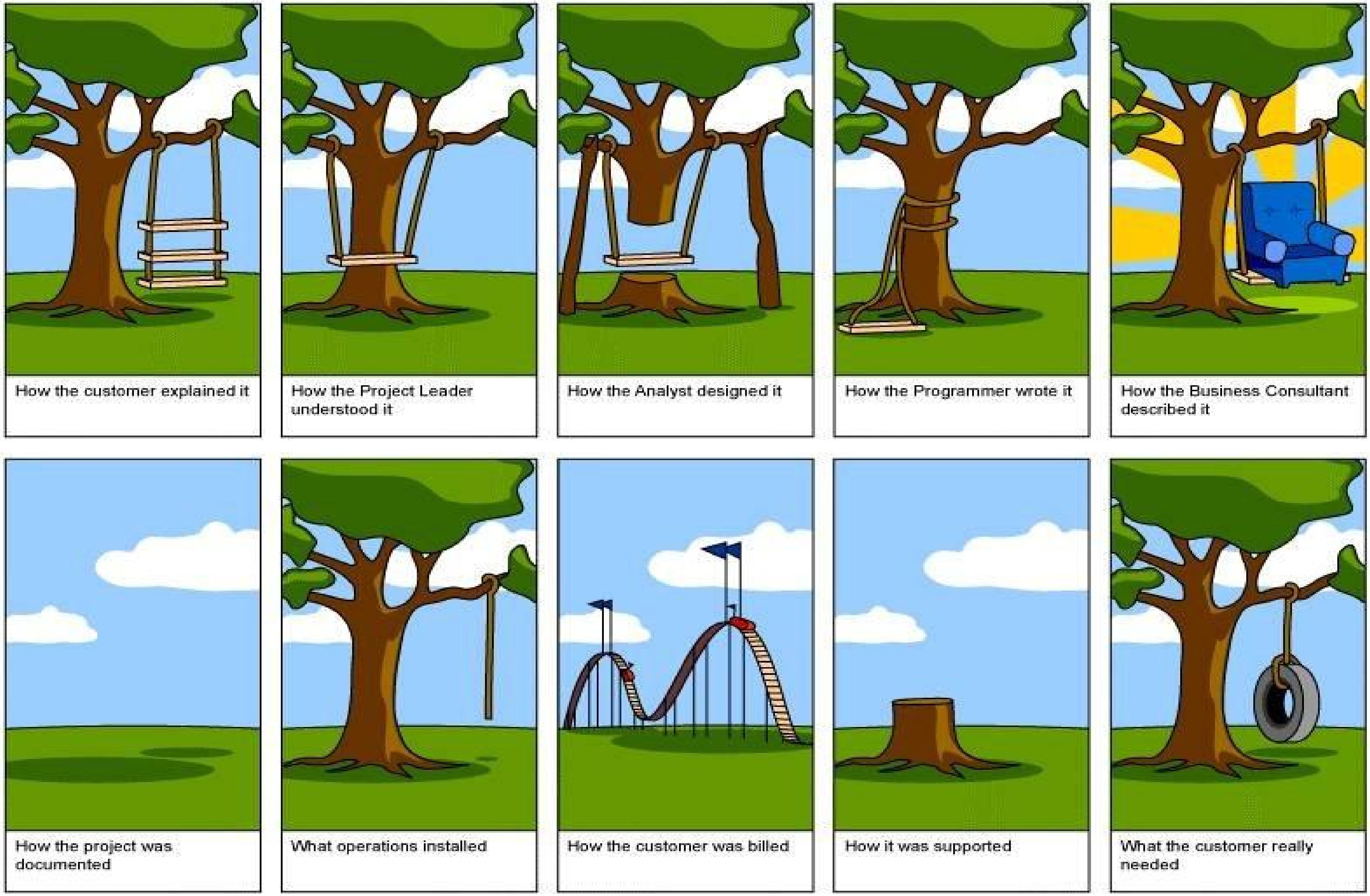
Documenting the Analysis

From previous phase we have recorded requirement elicitation in the RAD

Analysis is also documented in RAD

- Include the analysis model class diagram (possibly updated later)
- List: Identity, Persistence if complex life cycle
- Possibly updated GUI

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

Summary

Analysis focus on building an analysis (domain) model

- We used the requirements from RAD (use cases) to find a model
    - Mostly classes, not much of attributes and methods
    - Simple association with no direction
- We expressed the model as an UML-class diagram
- We documented model in RAD
- In parallel we do some prototyping

Next: From analysis model to design model and first running increment