SWE 621

FALL 2018

DESIGN AS DOMAIN MODELING

IN CLASS EXERCISE

- As you come in and take a seat, consider the following:

- What is a domain?

WHAT IS A DOMAIN?

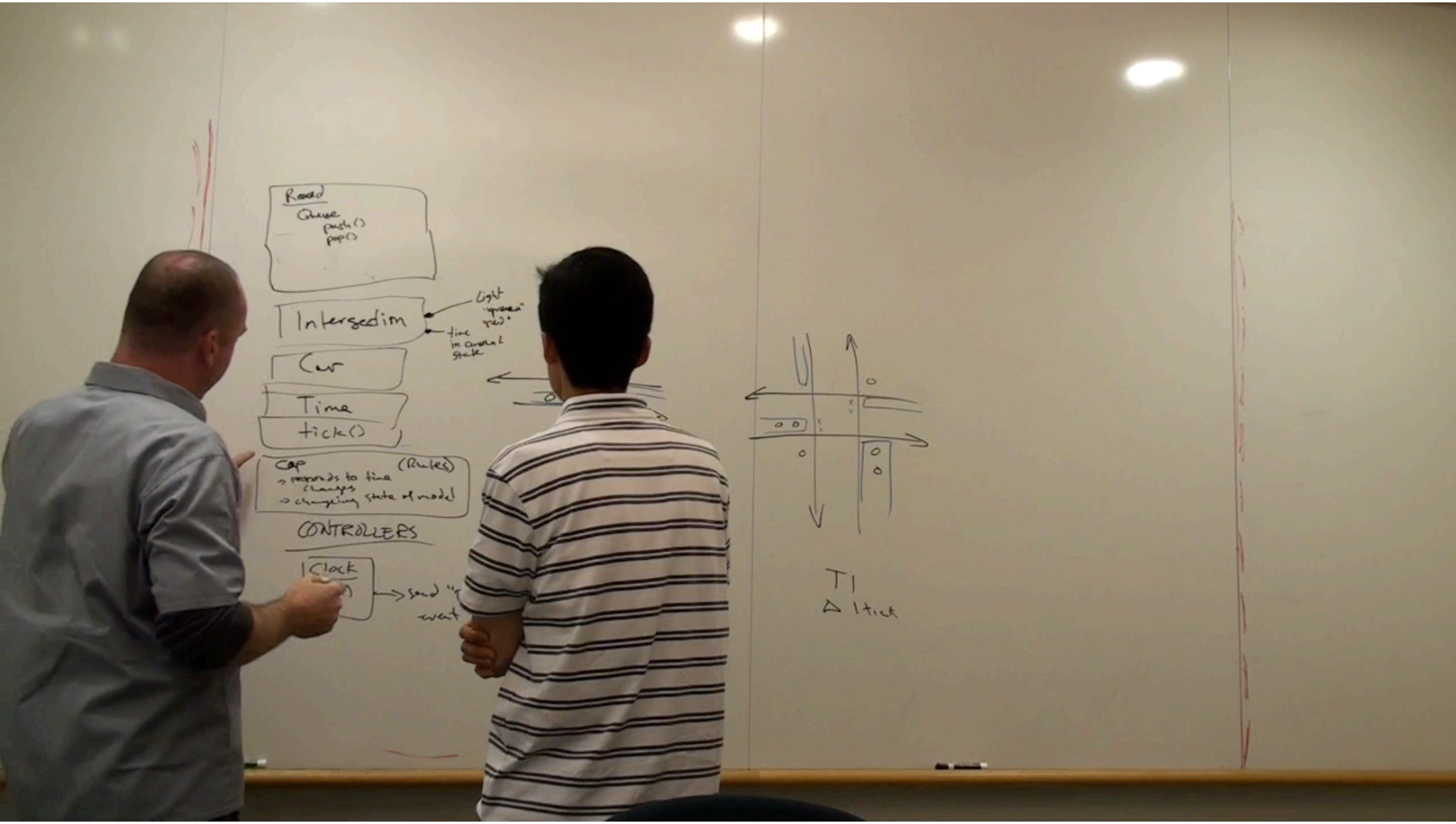
- What the software is "about"
  - The program is designed to manipulate information in the domain
- For systems that model real world, domain is aspect of real world that is being modeled
  - A shipping management system ---> how the shipping business works
  - A todo application --> how todos are manipulated
- For technical systems, system may be its own domain model
  - An operating system --> an operating system (?)

LOGISTICS

- HW1 due next week

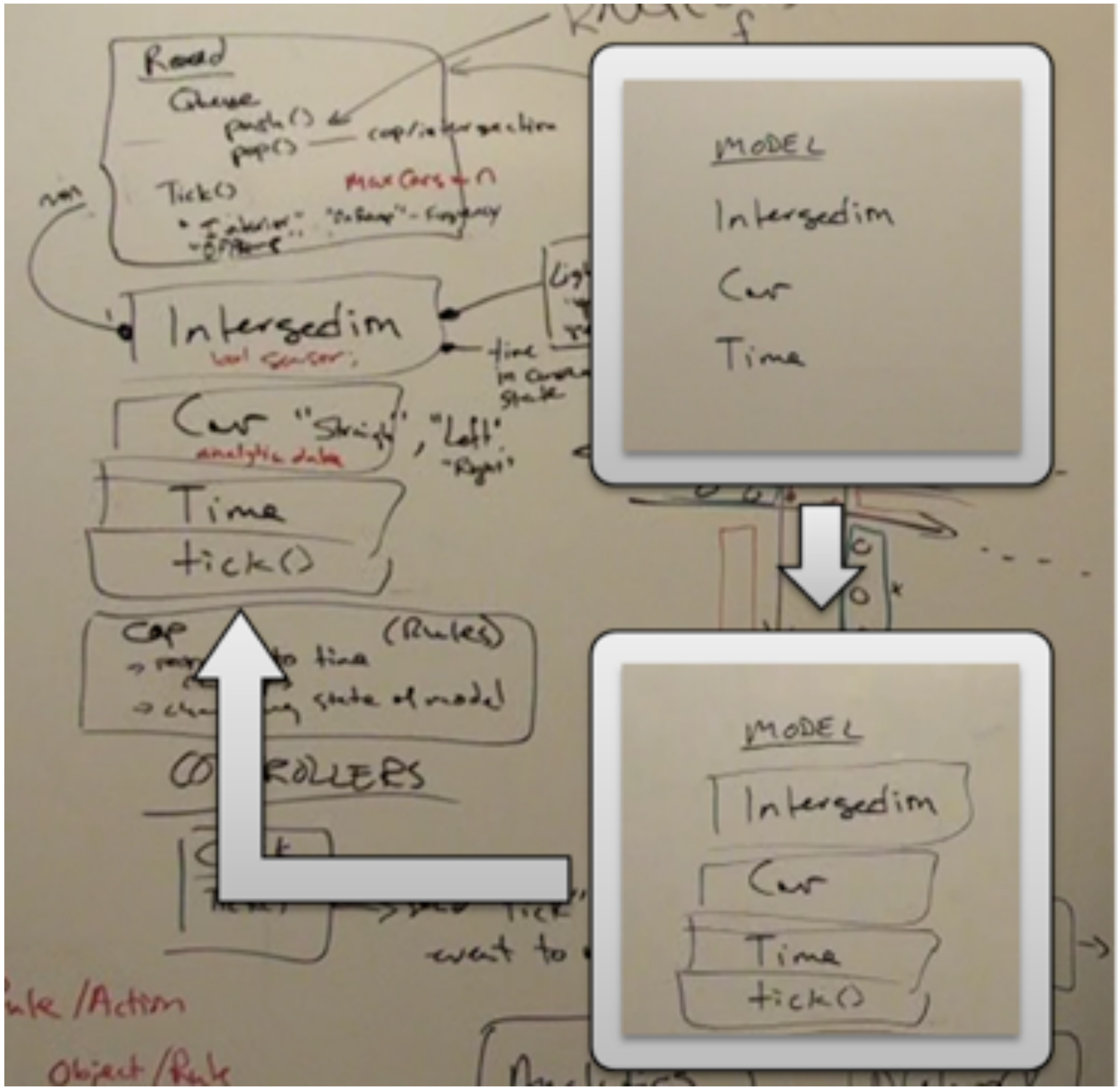
WHAT IS IT THAT EXPERTS ARE DOING WHEN THEY'RE DESIGNING?

HOW DO EXPERT DEVELOPERS DESIGN?

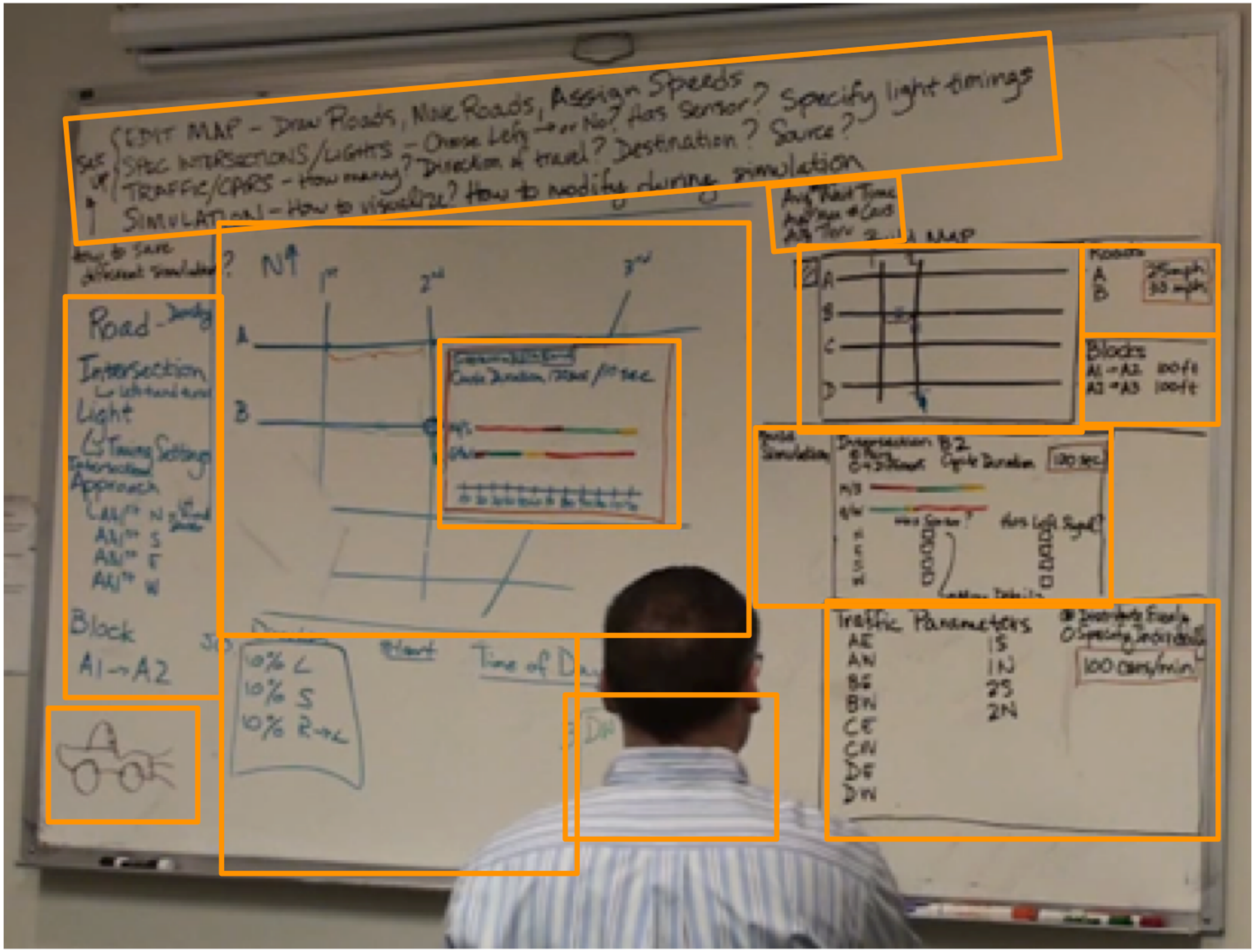


Mangano, LaToza, Petre, van der Hoek. (2015). How software designers interact with sketches at the whiteboard. *Transactions on Software Engineering.*

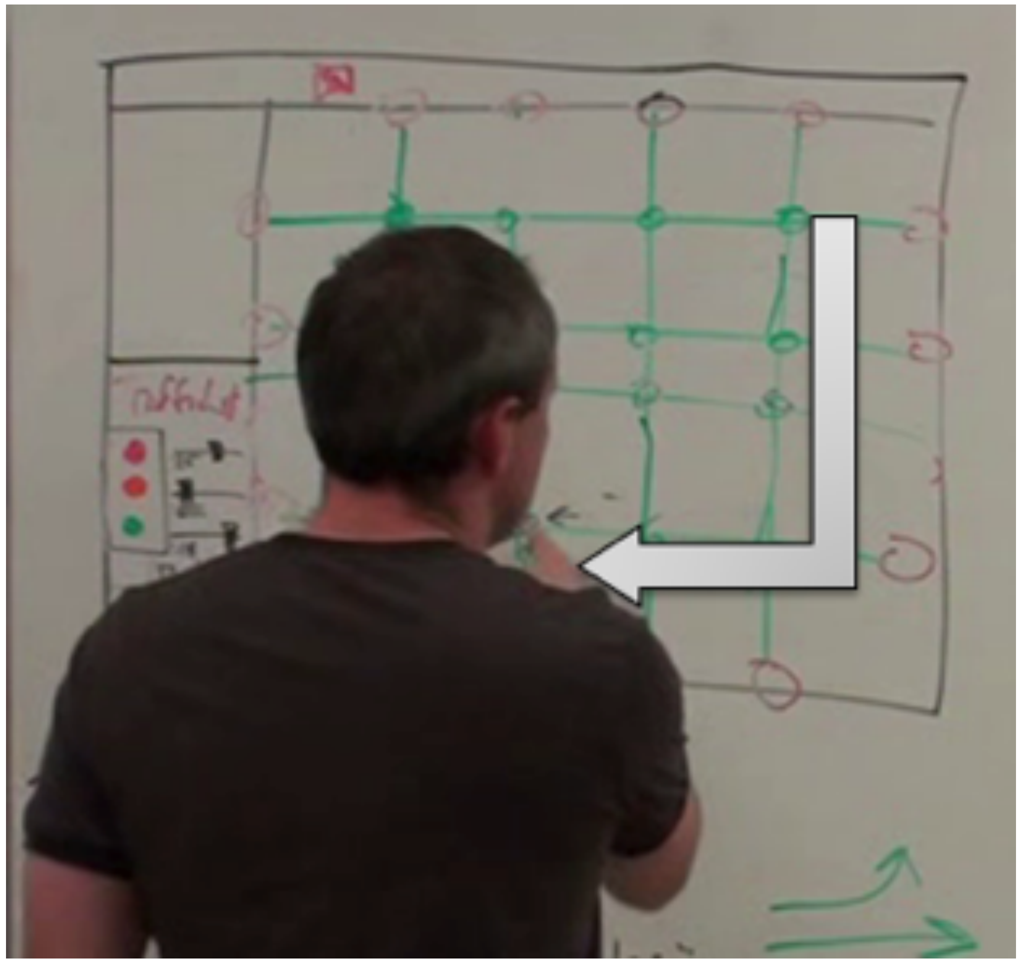
SKETCHES SUPPORT DESIGN CONVERSATIONS



DESIGNERS MODEL THE DOMAIN


EDIT MAP - Draw Roads, Nice Roads, Assign Speeds
SPEC INTERSECTIONS/LIGHTS - Choose Left turn or No? Has Sensor? Specify light timings
TRAFFIC/CARS - How many? Direction of travel? Destination? Source?
SIMULATION - How to visualize? How to modify during simulation
How to save different simulations?
Avg Wait Time
Avg/Max # Cars
Road
Intersection
Light
Timing Settings
Approach
Block
A1 → A2
N
1st
2nd
3rd
A
B
C
D
Cycle Duration
10% L
10% S
10% R
Time of Day
Roads
A 25mph
B 30mph
Blocks
A1 → A2 100ft
A2 → A3 100ft
Intersection B2
Cycle Duration
100 sec
Traffic Parameters
AE 1S
AN 1N
BE 2S
BN 2N
CE
CN
DE
DN
100 cars/min

DESIGNERS PERFORM MENTAL SIMULATIONS



Simulated

—data and control flow through system

—how model changed over time

83 ± 17% supported by sketches

30 ± 30% involved edits

9 ± 11% involved creating new sketches

Sketches used as an external medium for illustrating scenarios

Reused for multiple scenarios rather than created special purpose

Appropriate old sketches to discuss new designs

DESIGNERS WORK WITH SMALL GROUPS OF SKETCHES

25 ± 6% of focus periods involved <3 sec momentary references:

— 43 ± 14% quick glances - gather information or seek confirmation

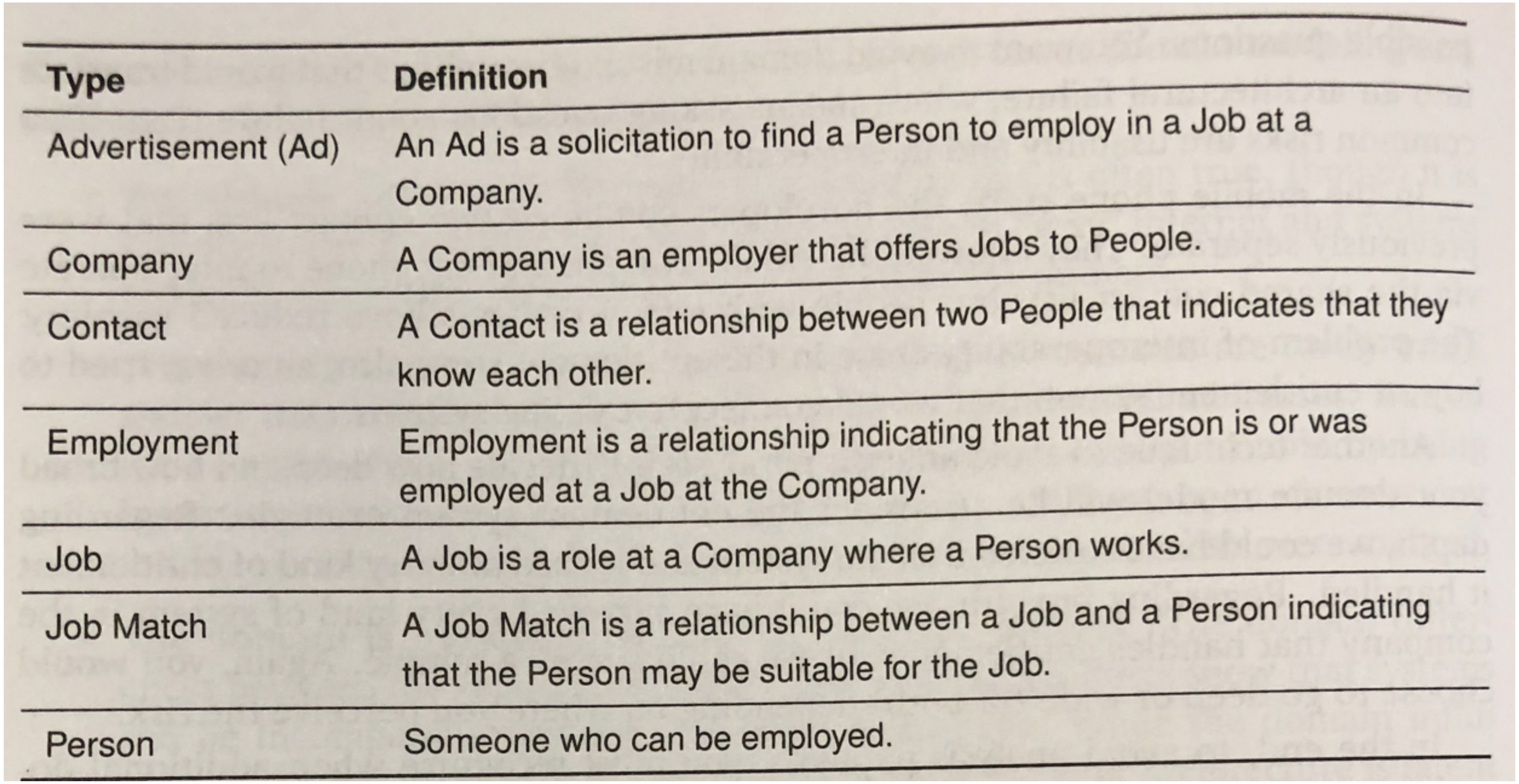
— 37 ± 14% pointing - guide attention to explain, review, or simulate

— 20 ± 10% split focus - reasoning using multiple sketches

WHY MODEL A DOMAIN?

- Where do elements in your design come from (e.g., classes, packages, namespaces, folders, etc.)?
- How should computation be distributed to these elements?
- How should these elements be named?

EXAMPLE: TEXTUAL MODEL OF DOMAIN



Type	Definition
Advertisement (Ad)	An Ad is a solicitation to find a Person to employ in a Job at a Company.
Company	A Company is an employer that offers Jobs to People.
Contact	A Contact is a relationship between two People that indicates that they know each other.
Employment	Employment is a relationship indicating that the Person is or was employed at a Job at the Company.
Job	A Job is a role at a Company where a Person works.
Job Match	A Job Match is a relationship between a Job and a Person indicating that the Person may be suitable for the Job.
Person	Someone who can be employed.

UBIQUITOUS LANGUAGE

- Domain models are often a bridge between your understanding of the problem and domain experts understanding of the problem
- Ensure that the terminology you use for element identifiers, state, and relationships matches terminology domain experts use
  - Makes it easier to walk through domain description and check with soma model
  - Makes it easier for domain experts to inspect
- This domain model can then serve as a starting point for your design model

IDENTIFYING ELEMENTS

- Can just pick them from the domain
  - Find the nouns! These are classes. Verbs are operations.
- But…
  - What about processes?
  - What about operations that might be swapped out for other operations?

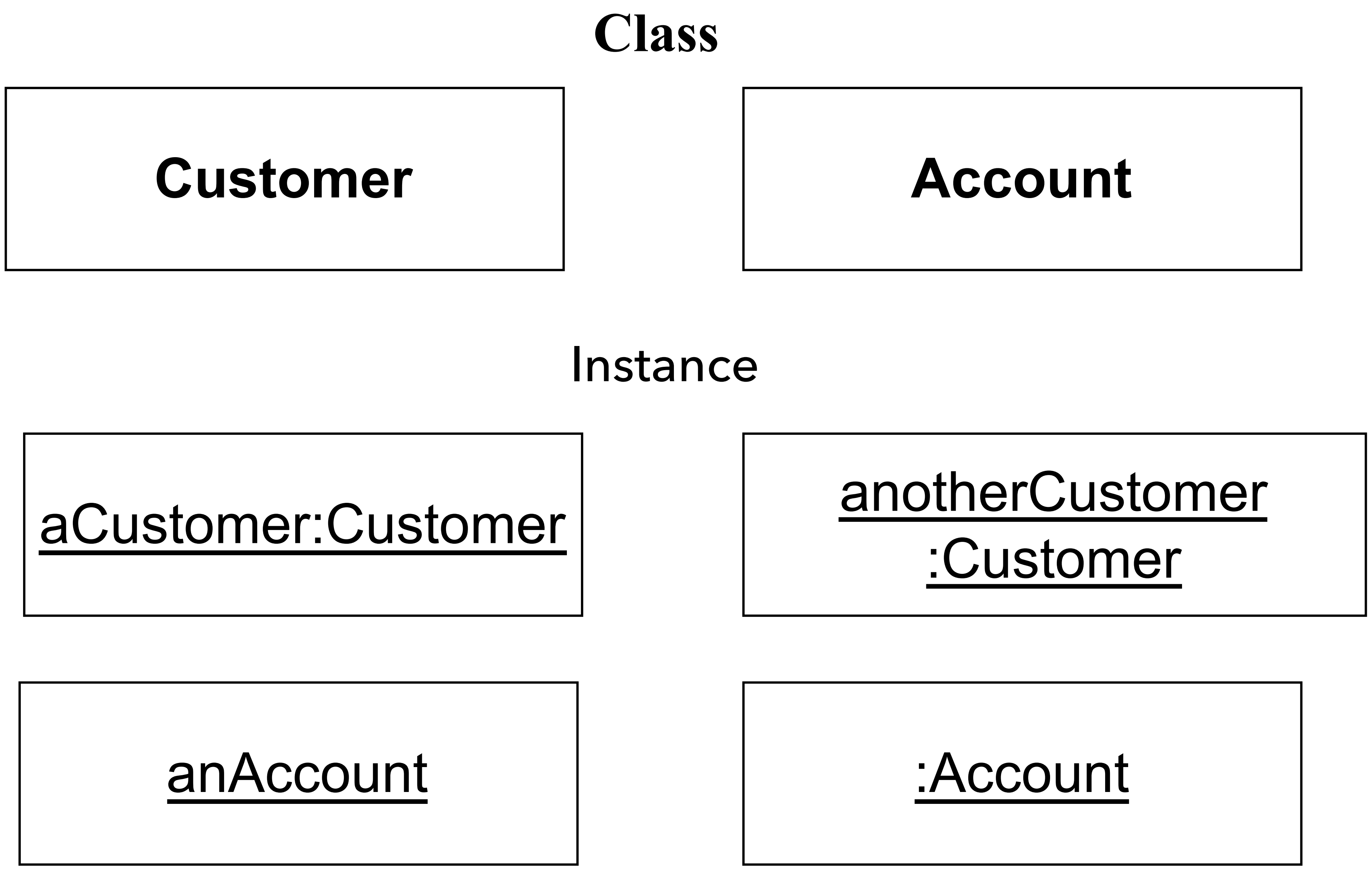
IDENTIFYING ELEMENTS, TAKE 2

- Elements are things that have identity
  - Has state, stored in attributes
  - Has operations
  - Has associations with other elements

SOME SIMPLE NOTATION

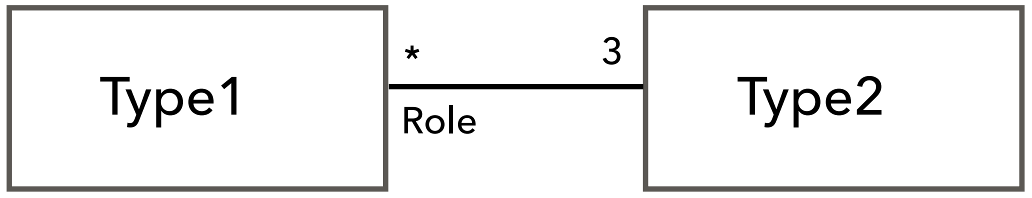
- Can use UML class diagrams for modeling domain
- Modeling your understanding of how the domain works
    - Not yet a model of how it will be designed or implemented in code
- Trying to make more precise how to think about the domain
    - What elements exist?
    - How are these elements related?
    - Differentiate between types of elements and individual instances of elements

SIMPLIFIED UML CLASS NOTATION: ELEMENTS



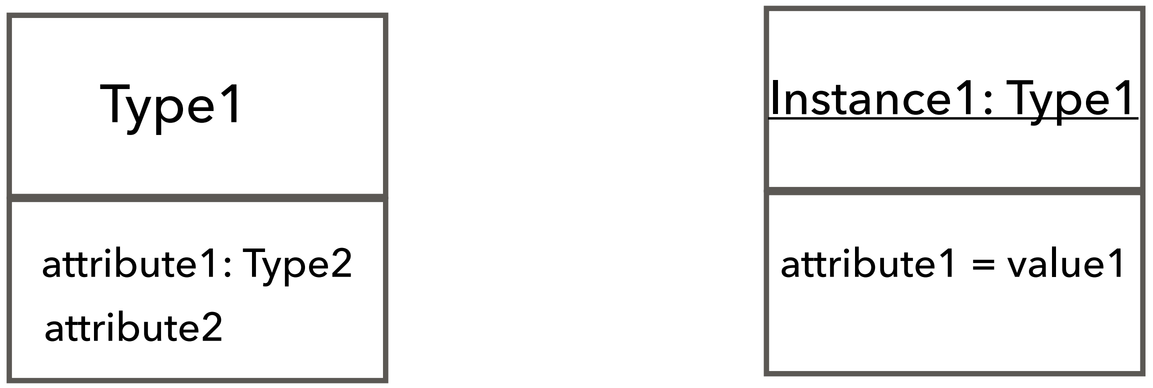
- Class vs. instance

SIMPLIFIED UML CLASS NOTATION: ASSOCIATIONS



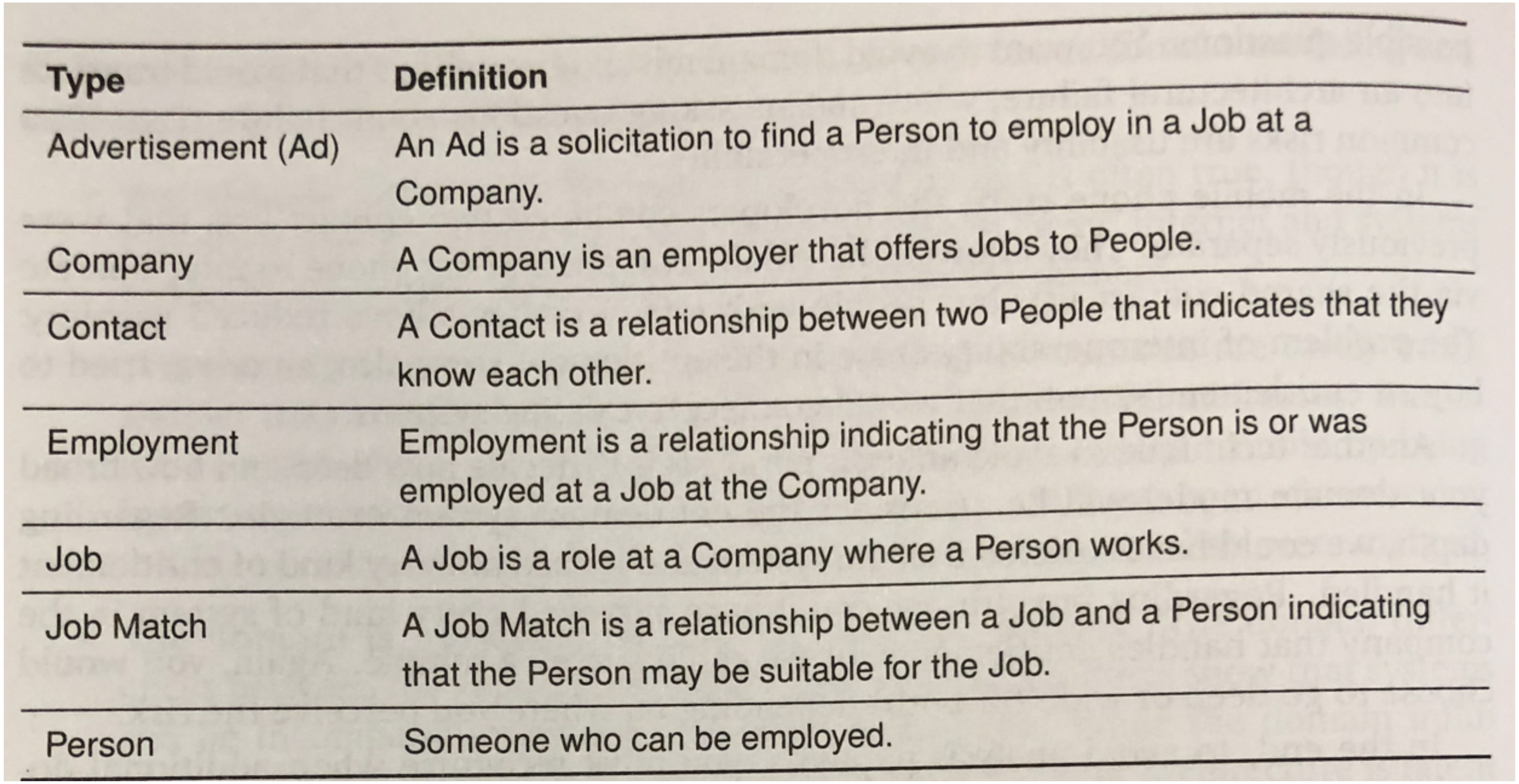
- Associations describe navigability
  - Can navigate from Type1 to Type2
- Multiplicities specify relationships between instances
  - Each Type2 instance is associated with 0, 1, ..., n Type1 instances (named Type2's Role)
  - Each Type1 instances is associated with 3 Type2 instances

SIMPLIFIED UML CLASS NOTATION: ATTRIBUTES



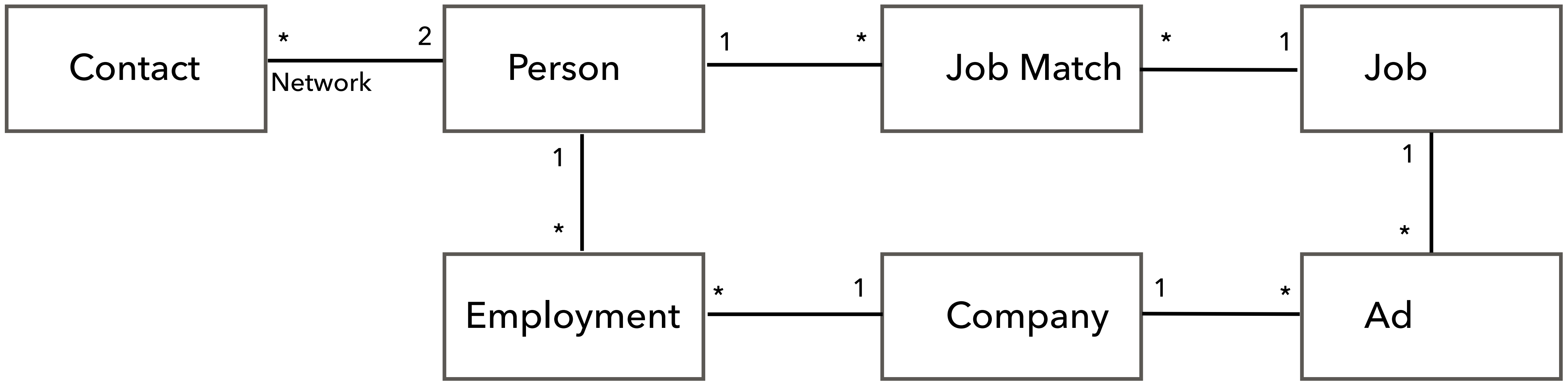
- Classes can have attributes, which may have types
- Instances may have values

ACTIVITY: BUILD A DOMAIN MODEL



Type	Definition
Advertisement (Ad)	An Ad is a solicitation to find a Person to employ in a Job at a Company.
Company	A Company is an employer that offers Jobs to People.
Contact	A Contact is a relationship between two People that indicates that they know each other.
Employment	Employment is a relationship indicating that the Person is or was employed at a Job at the Company.
Job	A Job is a role at a Company where a Person works.
Job Match	A Job Match is a relationship between a Job and a Person indicating that the Person may be suitable for the Job.
Person	Someone who can be employed.

DOMAIN MODEL, SIMPLIFIED UML CLASS DIAGRAM NOTATION



INTERROGATING A DOMAIN MODEL

- How do you know if your domain model works well?
  - Can try simulating a scenario with the model
- Scenario simulates a series of changes to the information model
  - Sequence of steps describing actions that occur that mutate state in the domain model

SCENARIO: EXAMPLE

- Initial state: Bradley is employed at Widgetron

1. Owen and Bradley meet, exchange business cards, and become part of each other's network of contacts.
2. Bradley's company, Widgetron, posts an Ad for a software developer job.
3. Bradley matches Owen to the job
4. Owen is hired by Widgetron for the software developer job

INTERROGATING A DOMAIN MODEL

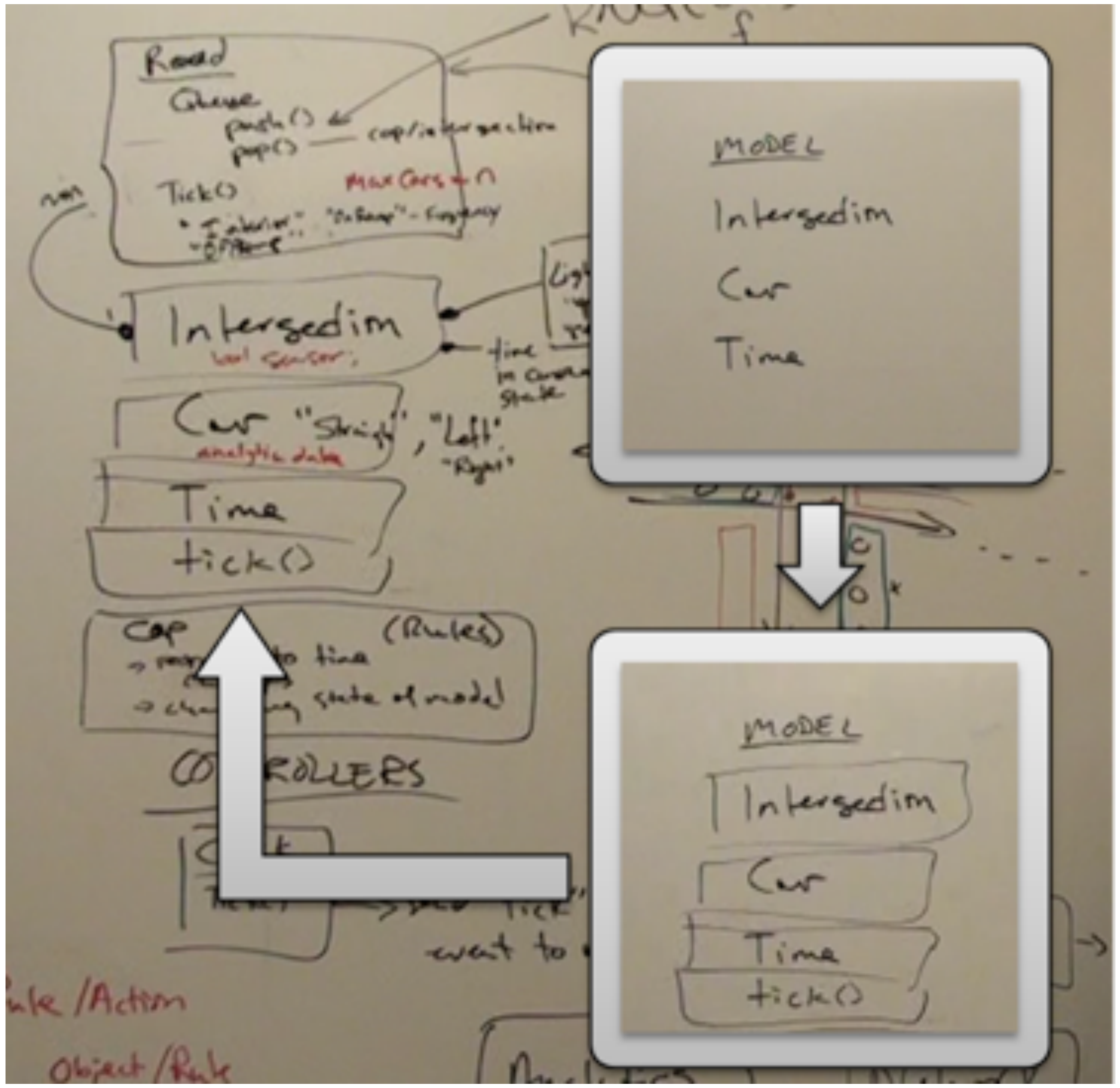
- Each step should correspond to different snapshot of domain model
- If state change is described in step, can domain model capture each step?
- Is there state necessary to determine what to do next that is not captured in the domain model?
- Are there additional elements needed?
- Would the model be simpler if some elements were combined into a single element or split into multiple elements?

ADVICE ON MODELING

- Before you start modeling, build a list of questions your model will answer
  - What risk(s) are you trying to address?
  - What decisions are you trying to make?
- Recognize when additional modeling is not providing additional value
- Only focus on aspects of the problem necessary
  - **Do not** need to build a model of everything
  - Leads to analysis paralysis
  - Model may change once you implement it

NOTATIONS <-> DECISIONS

- More formal notation makes more decisions
  - v1: list of 3 things, with a title
  - v2: each of these is an element, last element has operation
  - v3: relationship between some of these elements
- If you don't need to make the decision, don't need that formality in your model
  - Most popular notation: list

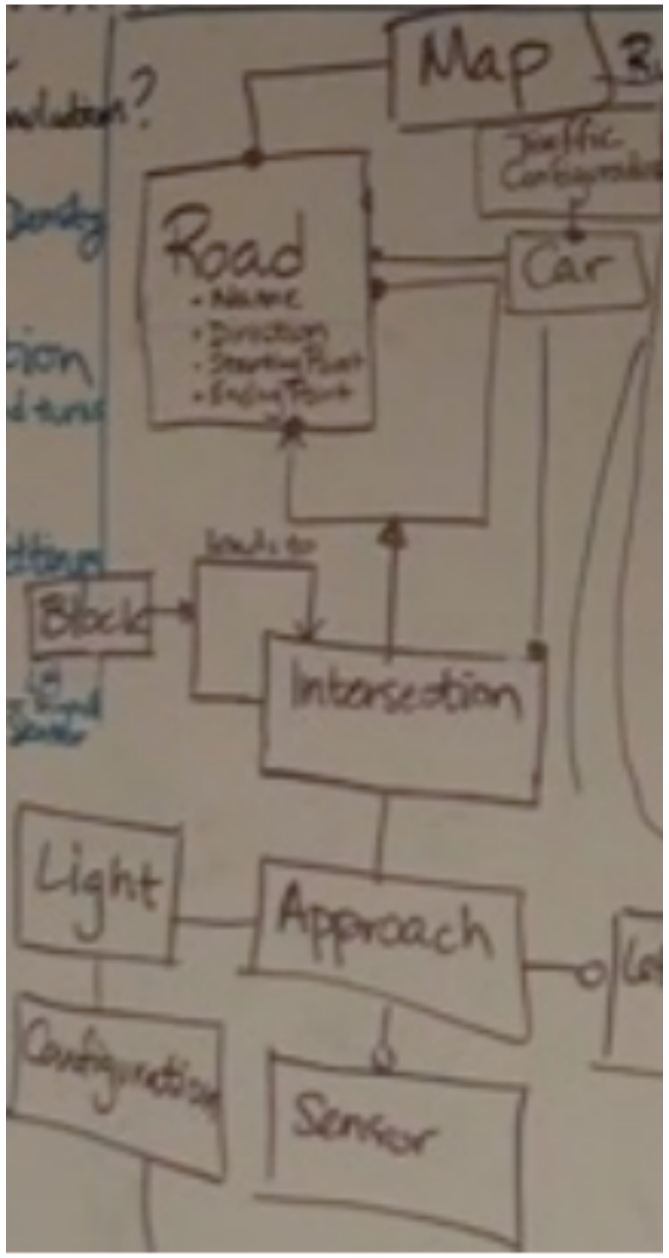


OTHER NOTATIONS

- Understanding aspects of domain often requires notations other than simplified uml class diagrams
  - Sometimes want notation that represents additional aspects of the domain

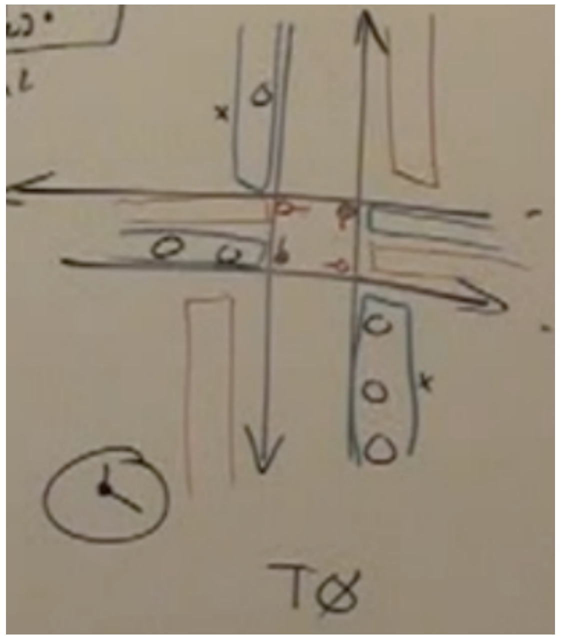
EXAMPLE: TRAFFIC

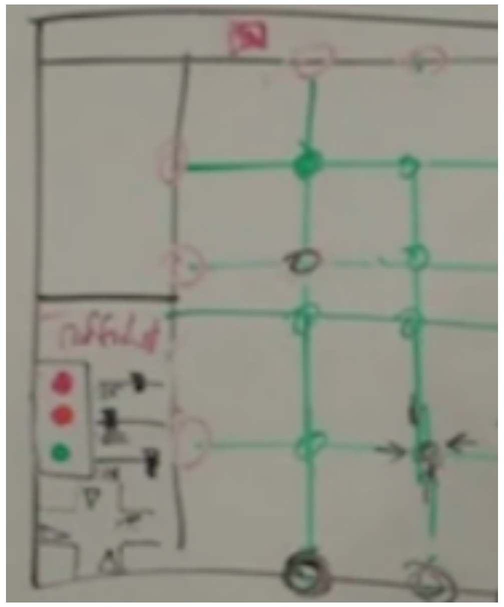
- Imagine trying to model how traffic flows through a network of roads
- Could describe elements (roads, intersections, lights, sensors, etc.)
- But there's a natural, expressive, and compact notations for describing traffic….



EXAMPLE: INTERSECTION AND MAP

- Intersection
  - Look at configuration of how cars come and go through intersection
- Map
  - Look at connectivity between roads in map





EXAMPLES OF NOTATIONS

- list
- map
- table
- GUI mockup
- simplified class diagram
- drawing
- array structure

CHOOSING NOTATIONS

- Experts choose the right notation for the question at hand
  - If you're trying to understand how traffic light and cars interact, build a notation that lets you construct examples of that
- May extend existing notation to add (or omit) details that are important (or unimportant) to the situation at hand
  - e.g., underline elements in a table to show elements that are all related to another element

SUMMARY

- Domain modeling helps you understand the "real world" aspects of your problem, independent of the eventual implementation of the model in your application
- Often used to understand what entities exist, what they should be named, how they are related, and what state they have
- Can also be used to explore arbitrary questions about the domain, particularly when driven by risks of complex or poorly understood domain
- Important to interrogate model through scenarios, updating model as necessary
- Choose which notational elements to include (even made up ones) based on questions to answer rather than including every possible notational element

http://www.bytesmiths.com/pubs/DesignFest97/problems/expense.html

IN CLASS ACTIVITY: BUILD DOMAIN MODEL

- A better system (better than an Excel spread sheet) is required for an employee making a claim for reimbursement of incurred expenses. A claim is an itemization of expenses where the following information must be recorded for each item:
  - the type of expense (eg. travel, meals, taxes)
  - the project to which the expense is to be charged
  - who paid for the expense (ie. employee or the company)
  - explanation for unusual or abnormal expenses (eg. lost ticket charges or two lunches)
  - identification of the supporting documentation (eg. receipts)
- This is complicated by the following:
  - a receipt may have several different items (eg. hotel & meals) which must be reported on separate lines
  - a receipt may have items which must be reported in different reporting periods (eg. return flight could end up in a different reporting period from the outbound flight)
  - under certain conditions the taxes on an item must be identified and tracked
  - an item may be split between two or more projects (and therefore must appear on two or more expense reports)
  - a separate expense report must be submitted for each project (which might also have different reporting periods)
  - an expense report may contain items with different currencies (for those employees who travel internationally)
  - an expense might include a personal portion which must be deducted (eg. extra rent on the automobile when the employee stayed away for personal time)
- It is important that:
  - an expense report can be audited easily (by the company, the client or the government)
  - that the employee can easily verify that all expenses have been claimed (given that some items may appear on different expense reports in different reporting periods)
  - the manual labor involved in producing and auditing expense claims be reduced
- Finally, it would be nice (although not required) if the system "knew" which expenses were expected (based on a trip) and could alert the employee to missing expenses (eg. a missing meal) and could flag expense items that are outside of the guidelines (and require further explanation).

DESIGN ACTIVITY: STEP 2

- Join together with another group
- Compare the models you built
  - Are differences due to different assumptions? Different focus?

DESIGN ACTIVITY: DISCUSSION

- What did you learn about the practice of design from this activity?