



Competency Questions

John Beverley

Assistant Professor, University at Buffalo

Co-Director, National Center for Ontological Research

Affiliate Faculty, Institute of Artificial Intelligence and Data Science

Outline

- Review of Design Pattern Submissions
- Varieties of Competency Question
- Not a Drill, Drill

Outline

- Review of Design Pattern Submissions
- Varieties of Competency Question
- Not a Drill, Drill

Domain: Tracing Corporate Projects, and their Deliverables

CQs:

- What are all the deliverables fed by the data set X?
- What deliverables and tools feed deliverable Y?
- What is the lineage of a particular data source?
- Who are the participant team members working on project of similar data domains?
- What processing tools are leveraged by more than one deliverable?
- What data sources are leveraged by more than one customer?
- What different deliverables are funded by the same customer organization?
- Which projects have overlapping domains? (data domains/expertise domains)
- What domain does a particular data source belong to?
- Who is the owner of the data that feeds a particular deliverable?
- Which projects overlap in data domains?

Classes:

- Organization
- Project
- Team Member
- Project Team
- Deliverable
- Data Source
- Data Product
- Data Processing Tool
- Customer
- Data Domain
- Data Owner
- Act of Data Processing
- Act of Deliverable Production
- Act of Investment

Relations:

- Participates in / Has participant
- Agent in / Has agent
- Has input
- Has output
- Has process part

Domain: Tracing Corporate Projects, and their Deliverables

CQs:

- What are all the deliverables fed by the data set X?
- What deliverables and tools feed deliverable Y?
- What is the lineage of a particular data source?
- Who are the participant team members working on project of si
- What processing tools are leveraged by more than one deliverab
- What data sources are leveraged by more than one customer?
- What different deliverables are funded by the same customer or
- Which projects have overlapping domains? (data domains/expe
- What domain does a particular data source belong to?
- Who is the owner of the data that feeds a particular deliverable?
- Which projects overlap in data domains?

Classes:

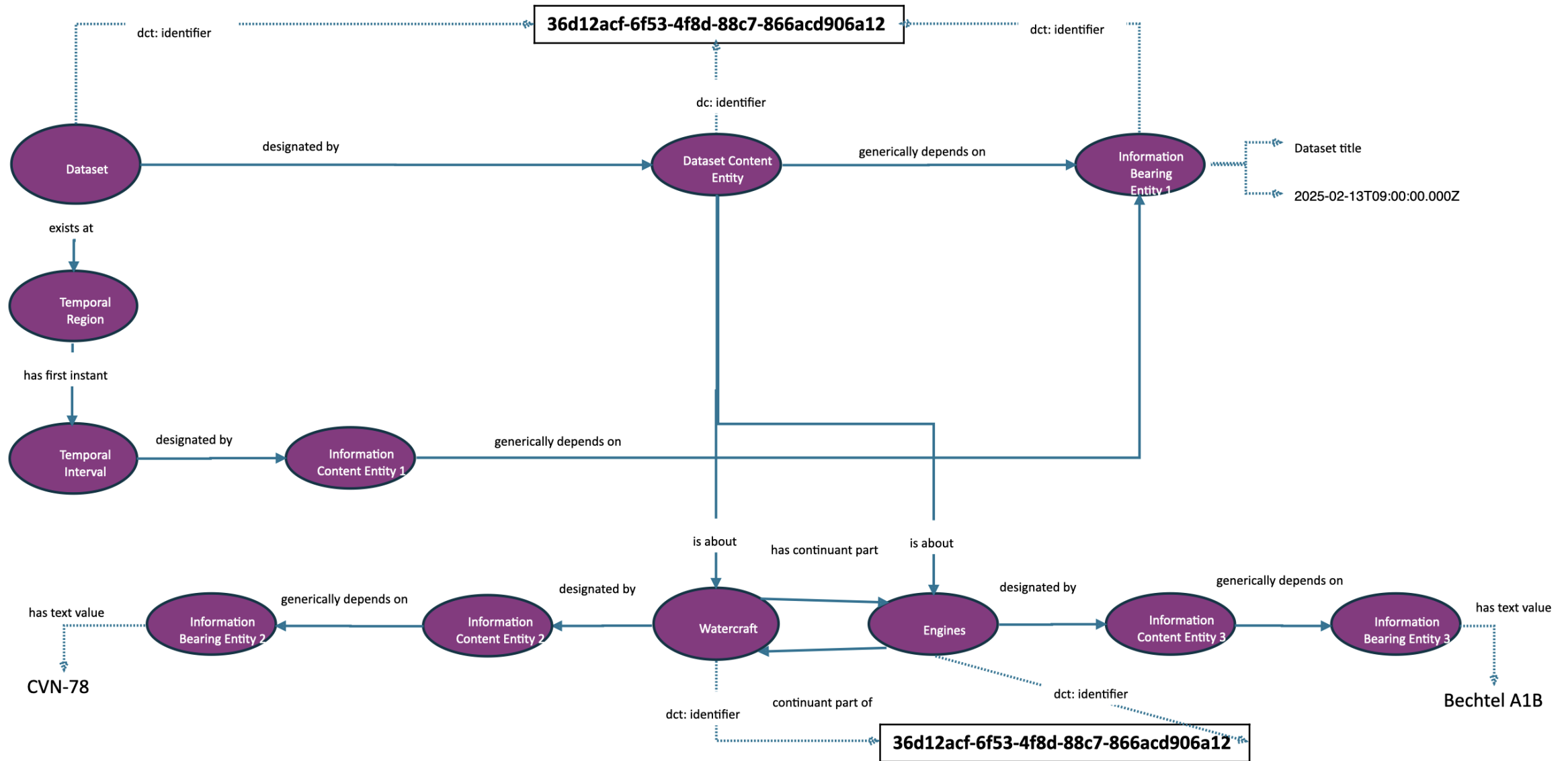
- Organization
- Project
- Team Member
- Project Team
- Deliverable
- Data Source
- Data Product
- Data Processing Tool
- Customer
- Data Domain
- Data Owner
- Act of Data Processing
- Act of Deliverable Production
- Act of Investment

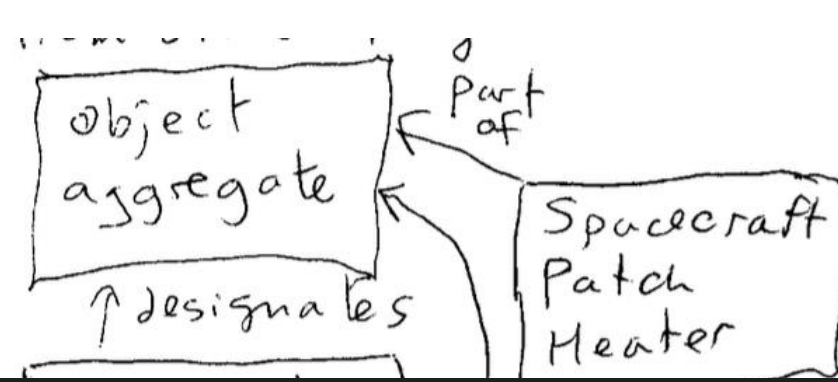
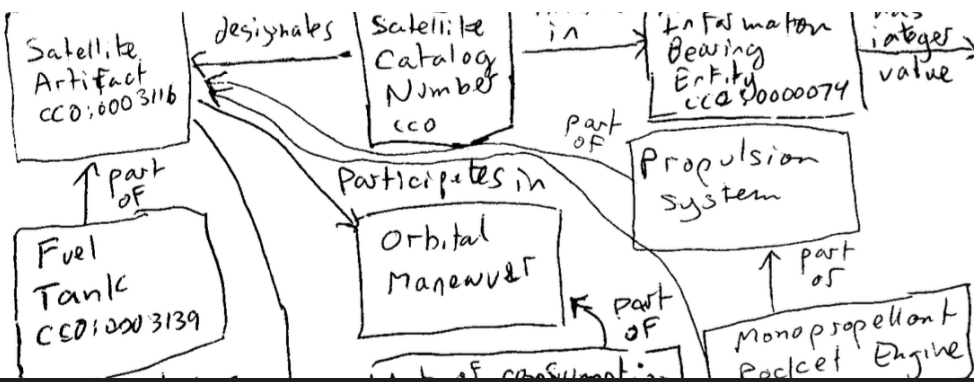
Relations:

- Participates in / Has participant
- Agent in / Has agent
- Has input
- Has output
- Has process part

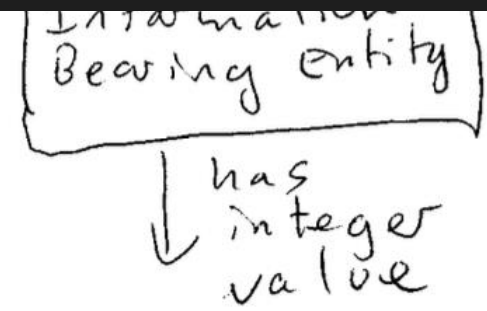
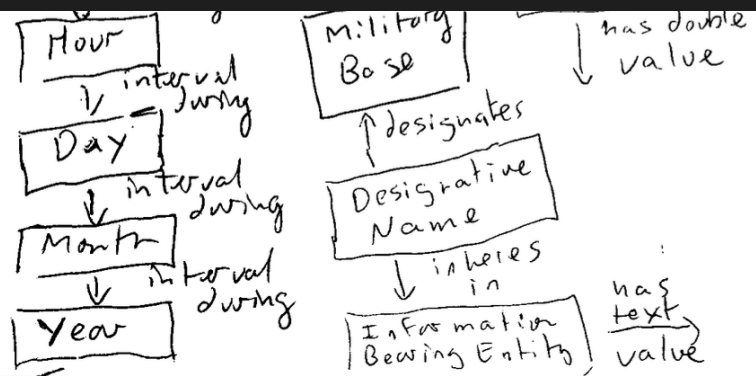


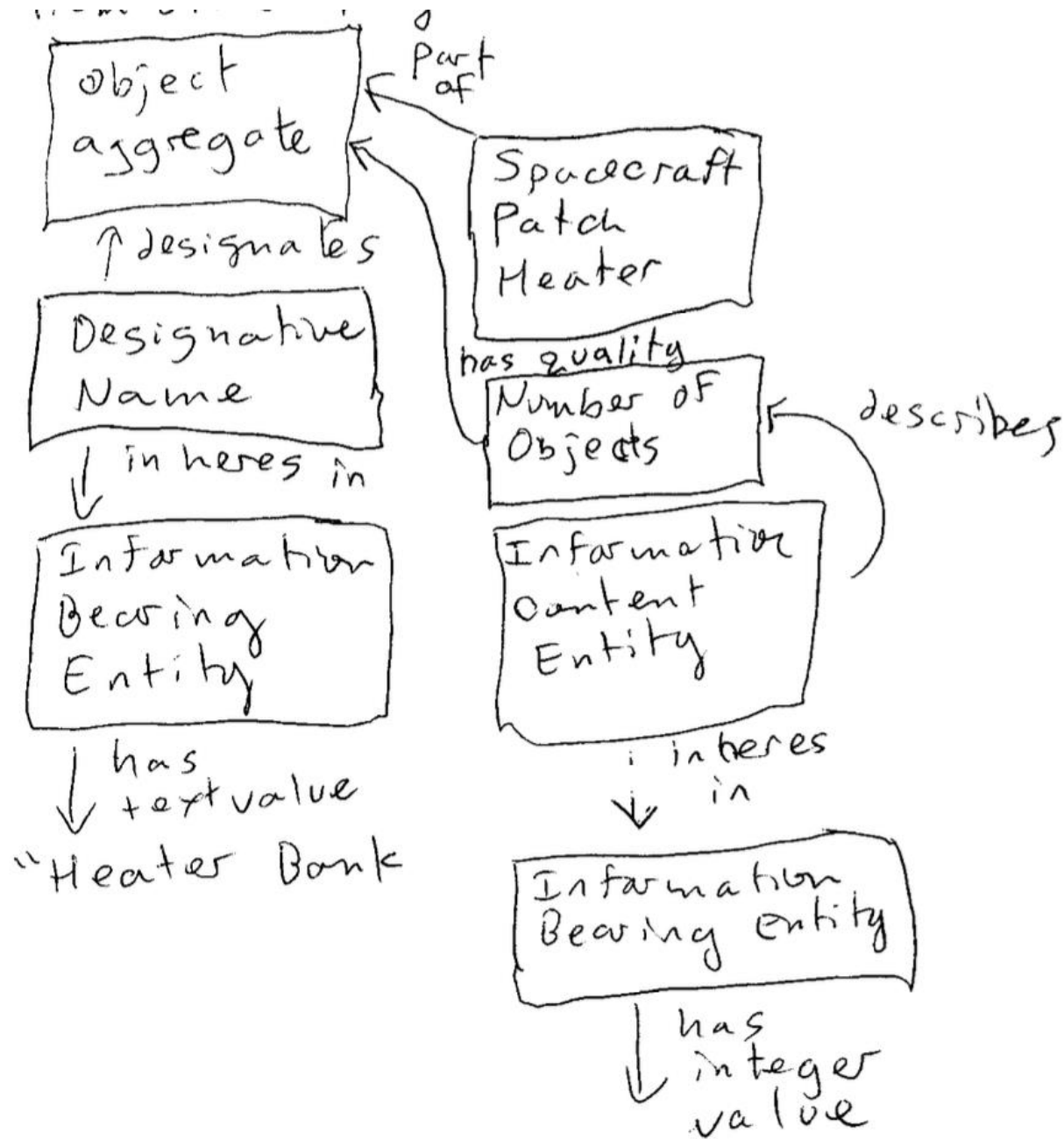
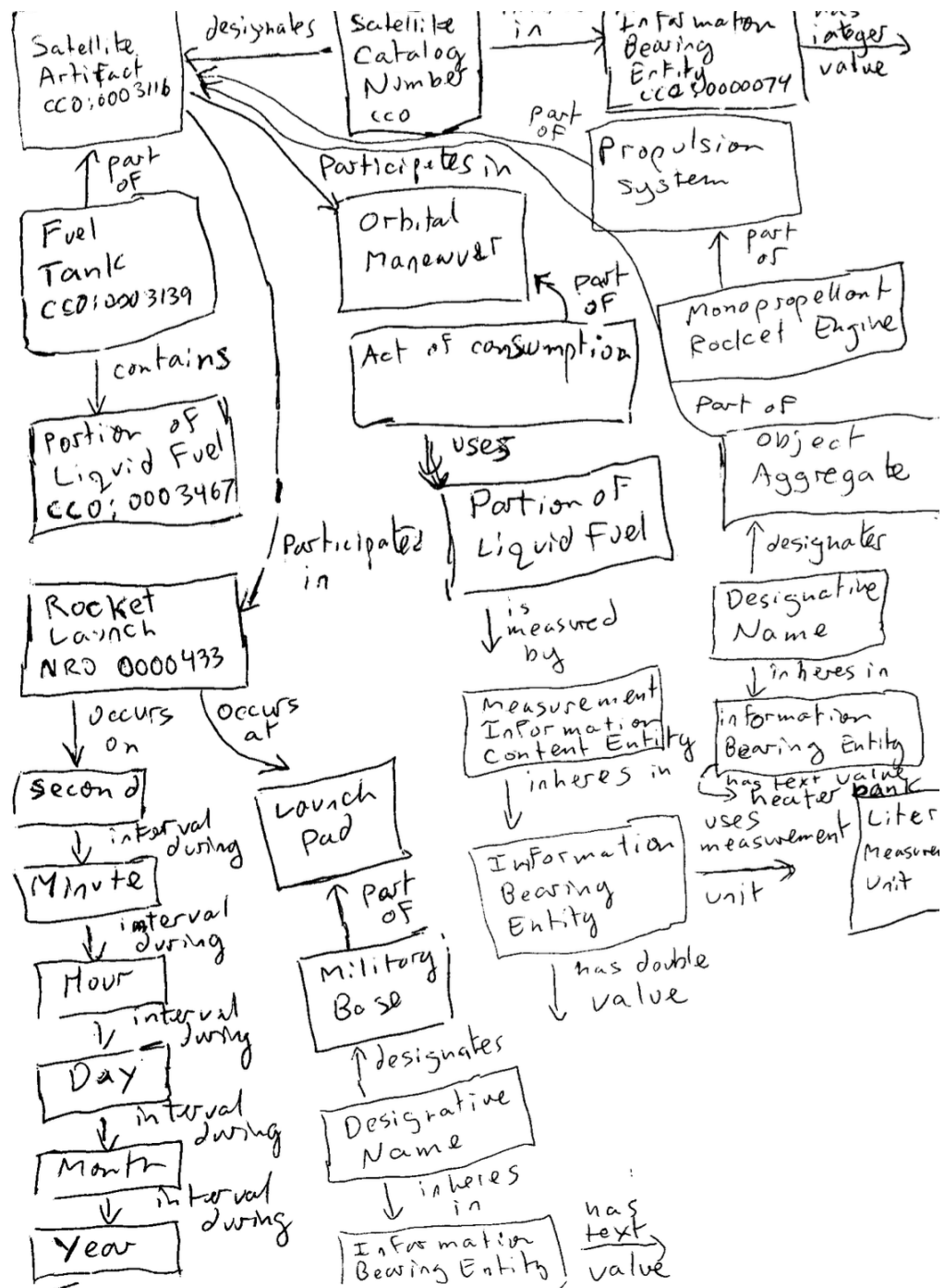
What engine is used by the by the Gerald R. Ford Aircraft Carrier (CVN-78)?





1. What kind of a propulsion system does a satellite have
2. How many redundant heaters does a satellite have
3. Where and when did the satellite launch from
4. How much fuel does the satellite have
5. How much fuel will a maneuver consume for the satellite





CQ's :

When was the game released?

What videogame system/platform is it on?

Who made the videogame?

What sort of genre of game is it?

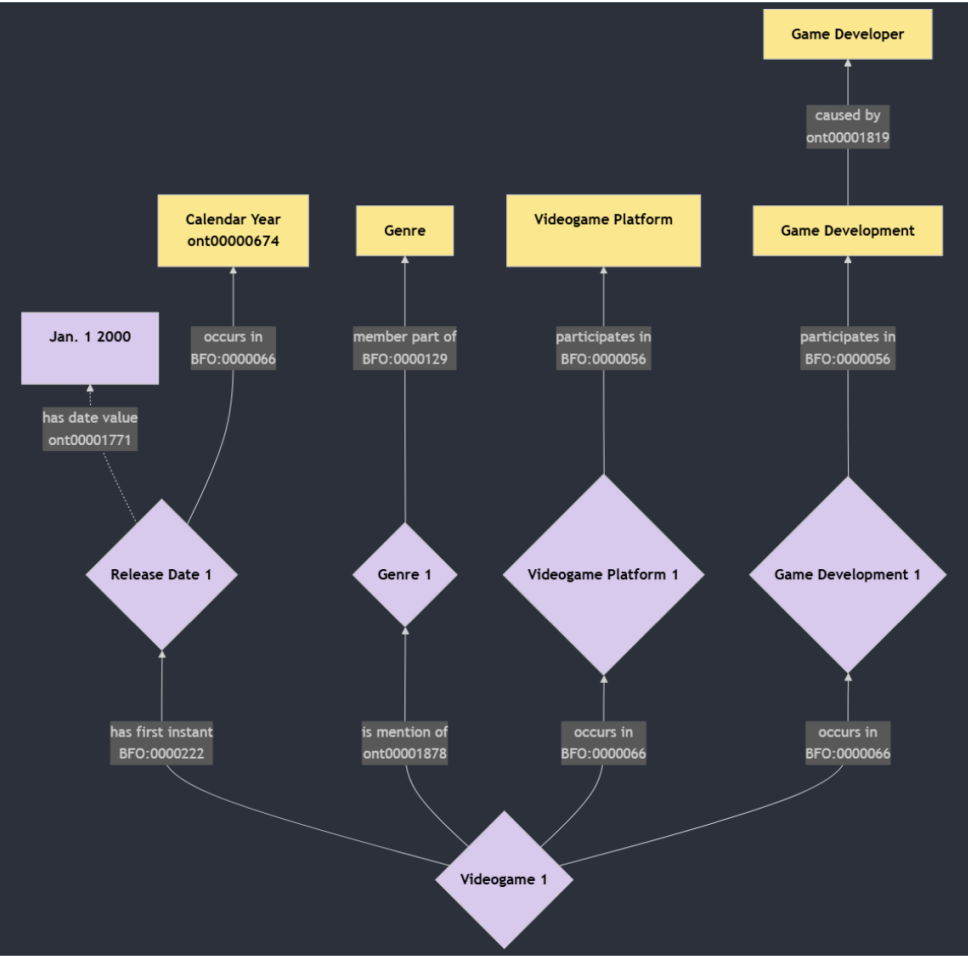
How many copies were sold?

Classes:

Release Date, Genre, Videogame systems, Game Developer, Copies sold

Relations:

has first instant, is mention of, occurs in, participates in, member part of, caused by, has date value



MERMAID CODE:

```
graph BT
    A["A<br><b>Videogame 1</b></span><br>"] -->|has first instant<br>BFO:0000222| B["B<br><b>Release Date 1</b></span>"]
    B -->|has date value<br>ont00001771| C["C<br><b>Jan. 1 2000</b></span><br>"]
    B -->|occurs in<br>BFO:0000066| D["D<br><b>Calendar Year</b></span><br>ont0000674"]
    A -->|is mention of<br>ont00001878| E["E<br><b>Genre 1</b></span><br>"]
    A -->|member part of<br>BFO:0000129| F["F<br><b>Genre</b></span>"]
    A -->|occurs in<br>BFO:0000066| G["G<br><b>Videogame Platform 1</b></span><br>"]
    A -->|participates in<br>BFO:0000056| H["H<br><b>Videogame Platform</b></span><br>"]
    A -->|occurs in<br>BFO:0000066| I["I<br><b>Game Development 1</b></span><br>"]
    A -->|participates in<br>BFO:0000056| J["J<br><b>Game Development</b></span><br>"]
    A -->|caused by<br>ont00001819| K["K<br><b>Game Developer</b></span><br>"]

    classDef yellow fill:#ffef80
    classDef purple fill:#d8b4fe
    class J,F,G,I,K,H yellow
    class A,B,C,D,E,I purple
```

I didn't like my example because it's boring, so I'm choosing something more fun! * Physics *

Identify a domain: Physics

- Competency Questions:
1. What are the fundamental forces of nature?
 2. What are the SI units for different physical quantities?
 3. What is the relationship b/w force, mass, & acceleration?
 4. What are the fund. equations?
 5. classical vs. quantum mechanics
 6. Particles in the standard model?
 - 7.
 - 8.

I don't really know how to do this ;)

Classes:

1. Physical Quantities:

- scalar: mass, energy, temp
- vector: force, velocity, momentum

2. Measurement units

- SI units: meter, seconds, kg
- derived units: Newton, Joule, pascal

3. Forces (fundamental)

- gravitational
- electromagnetic
- strong nuclear
- weak nuclear

4. Particles

- elementary (electron, photon, quark)
- composite (proton, neutron)

5. probably some sort of laws?

6. some fundamental equations?

Relations:

Physical quantity \rightarrow unit of measurement

Laws \rightarrow equations

Force \rightarrow object

equation \rightarrow laws

Disambiguation:

- could clear up a physical equation
v.s a theoretical one?
- difference b/w force vs. interaction?

Below is my input for the question you posed to the group this morning in the Design Pattern class:

- Exercise:
 - Identify a domain to model: radiometry
 - Identify competency questions:
 - What sources of light exist within a natural scene?
 - How many photons arrive at a pixel for a given time?
 - What wavelengths do the photons have?
 - How does the atmosphere affect the travel of a photon?
 - How does sensor material impact the measurement of photons?
 - How does a lens collect light?

Exercise:

1. Identify a domain to model
 1. Identified Domain: Human Emotion
 1. Subdomain: Fear
2. Identify competency questions
 1. What are the physical manifestations of fear?
 2. How does fear manifest differ in physical symptoms?
 3. Does socioeconomic status affect how fear manifests?
 4. What are the differences (if any) between physical, emotional, and psychological fear?
 5. Does a change in socioeconomic status change how fear is experienced (i.e. low income to high income and vice versa)?
3. Identify classes and relations satisfying cqs
 1. Classes: person, emotion, fear, capability, income, education, occupation, wealth, social class, physical symptoms, heart rate, respiratory rate, digestive status
4. Disambiguate

Material entities within scope, i.e. Material Entity

Person, heart, digestive tract, lungs (?)

Qualities these material entities have, i.e. Quality

Physical symptoms, heart rate, digestive status, respiratory rate, education, occupation, income, emotions

What these material entities could do, i.e. Realizable Entity

Physical symptomology, react

What these material entities actually do, i.e. Process

Experience fear, engaging event,

Where these material entities and boundaries are located, i.e. Immaterial Entity

Within the person

When these entities exist, i.e. Temporal Region

Various time frames

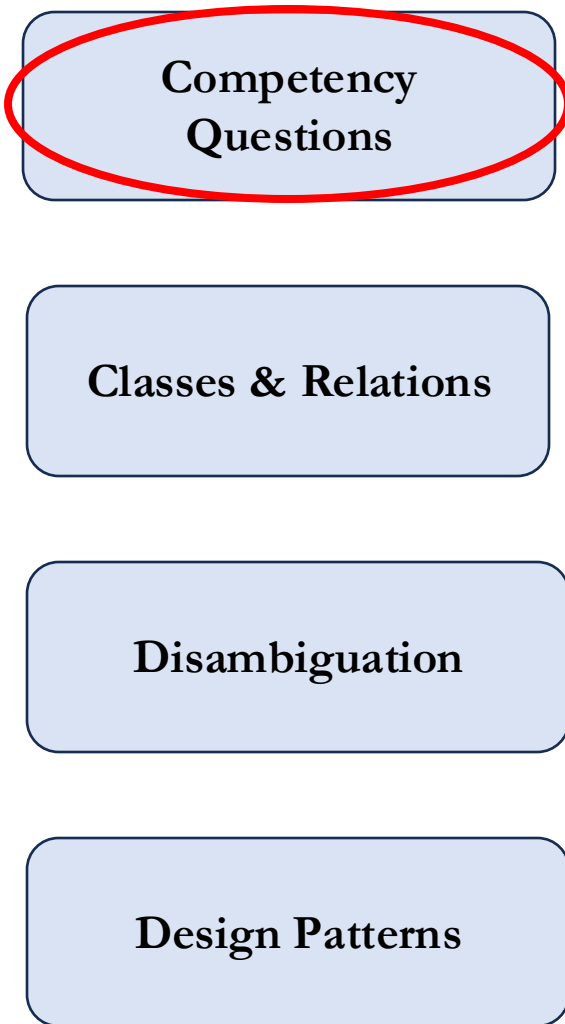
Information we use to talk about 1-6, i.e. Generically Dependent Continuant

Outline

- Review of Design Pattern Submissions
- Varieties of Competency Question
- Not a Drill, Drill

Guidance

- Competency questions are **used to guide ontology development** and **generate unit tests** to ensure ontologies are sufficiently well-developed
- Identify a preliminary list of competency questions **first**
- They will help you scope your project



Competency
Questions

Classes & Relations

Disambiguation

Design Patterns

CQ Types

- **Scoping CQ**: Used to define the scope of a domain ontology
- These help determine what the ontology is intended to be about, i.e. what will be included and what will be excluded

CQ Types

- **Scoping CQ**: Used to define the scope of a domain ontology
- These help determine what the ontology is intended to be about, i.e. what will be included and what will be excluded

At what speed in knots can a patrol boat move on open water?

CQ Types

- **Validation CQ**: Used to validate the content of an ontology by checking if the ontology adheres to its intended meaning
- This is a domain-focused CQ for evaluating whether the ontology reflects the relevant domain

CQ Types

- **Validation CQ**: Used to validate the content of an ontology by checking if the ontology adheres to its intended meaning
- This is a domain-focused CQ for evaluating whether the ontology reflects the relevant domain

Do rifles have cartridges?

CQ Types

- **Foundational CQ**: Used to align a domain entity to an entity within a top-level ontology.
- This helps ensure coherence between a top-level and a domain ontology that extends it

CQ Types

- **Foundational CQ**: Used to align a domain entity to an entity within a top-level ontology.
- This helps ensure coherence between a top-level and a domain ontology that extends it

Are rifles material entities?

CQ Types

- **Relation CQ**: Used to evaluate logic of relationships asserted in an ontology
- For example, domain/range constraints as well as role characteristics

CQ Types

- **Relation CQ**: Used to evaluate logic of relationships asserted in an ontology
- For example, domain/range constraints as well as role characteristics

continuant part of has domain and range continuant
occurrent part of is transitive

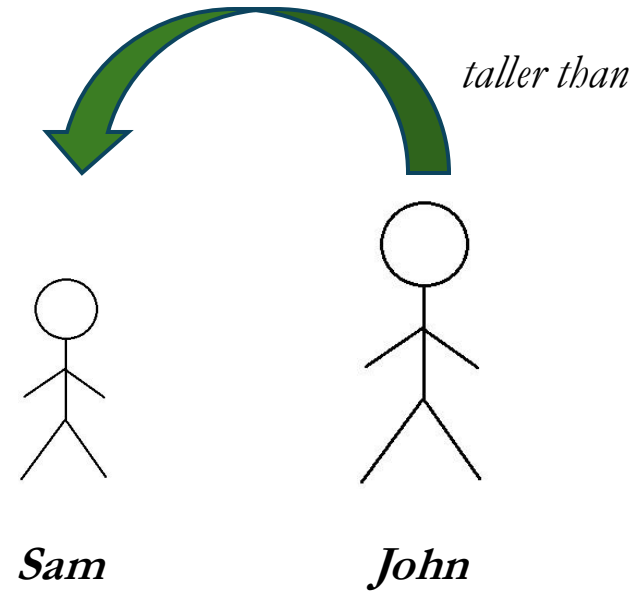
CQ Types

- **Relation CQ**: Used to evaluate logic of relationships asserted in an ontology
- For example, domain/range constraints as well as role characteristics

continuant part of has domain and range continuant
occurrent part of is transitive

Relations

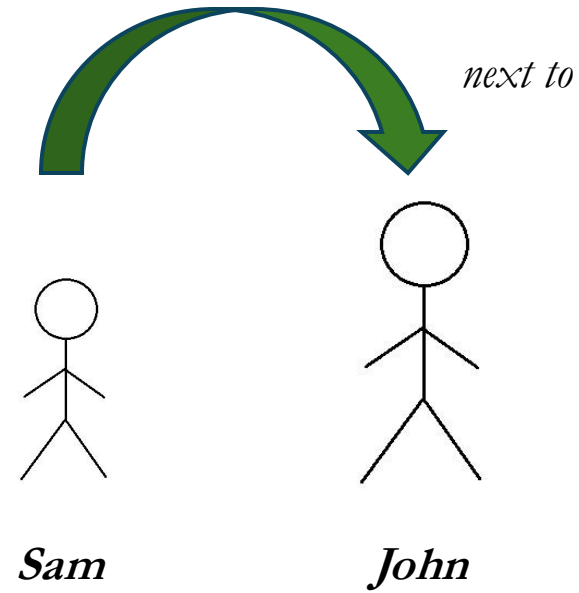
- Perhaps more interesting, are *two-placed* properties, or what are known as *relations*
- For example:
 - John is *taller than* Sam



Relations

- Perhaps more interesting, are *two-placed* properties, or what are known as *relations*

- For example:
 - John is *taller than* Sam
 - Sam is *next to* John

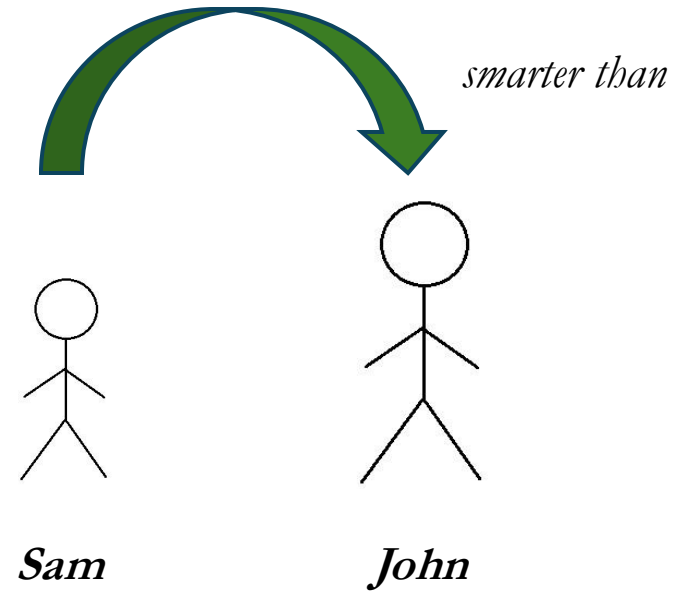


Relations

- Perhaps more interesting, are *two-placed* properties, or what are known as *relations*

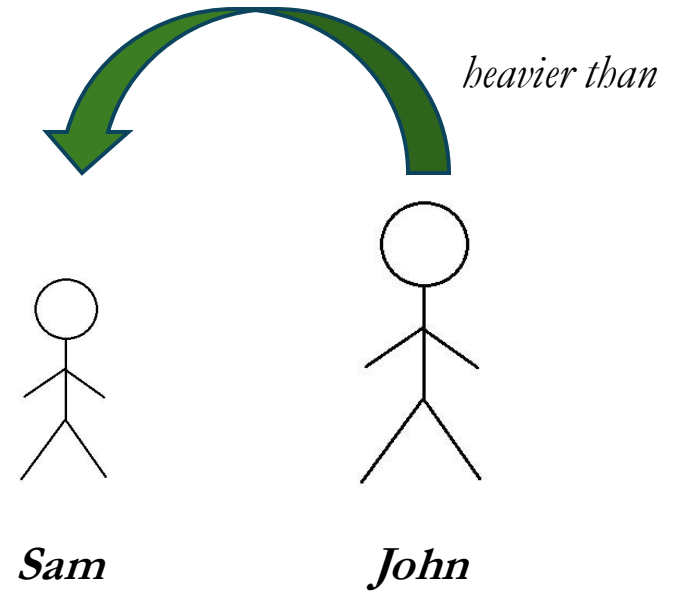
- For example:

- John is *taller than* Sam
- Sam is *next to* John
- Sam is *smarter than* John



Relations

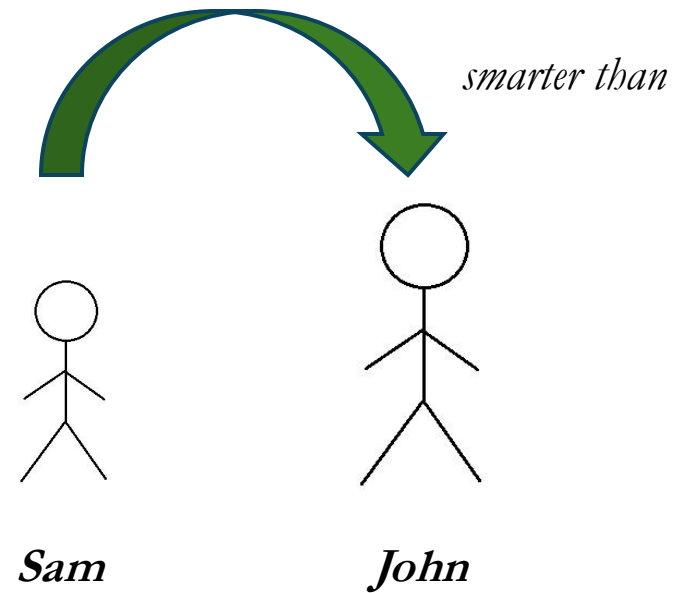
- Perhaps more interesting, are *two-placed* properties, or what are known as *relations*
- For example:
 - John is *taller than* Sam
 - Sam is *next to* John
 - Sam is *smarter than* John
 - John is *heavier than* Sam



Relations

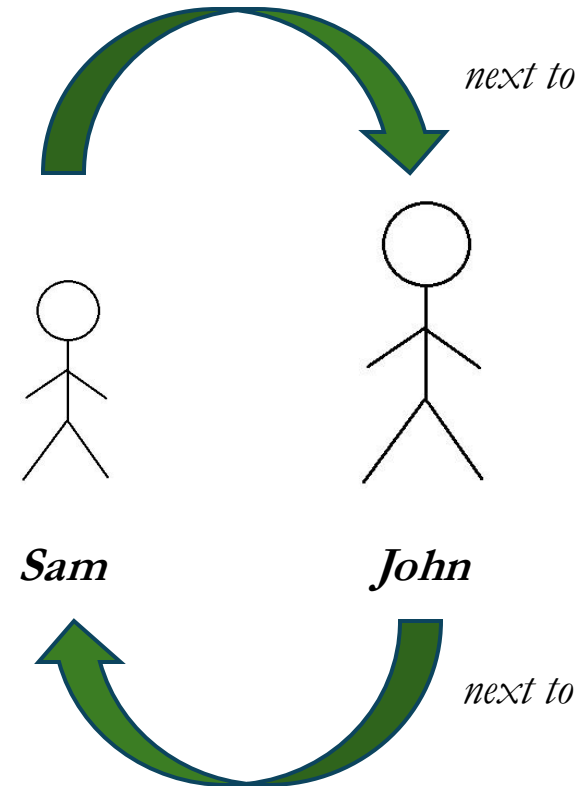
- Perhaps more interesting, are *two-placed* properties, or what are known as *relations*

- For example:
 - John is *taller than* Sam
 - Sam is *next to* John
 - Sam is *smarter than* John
 - John is *heavier than* Sam
 - Sam is *the smartest* person in the room



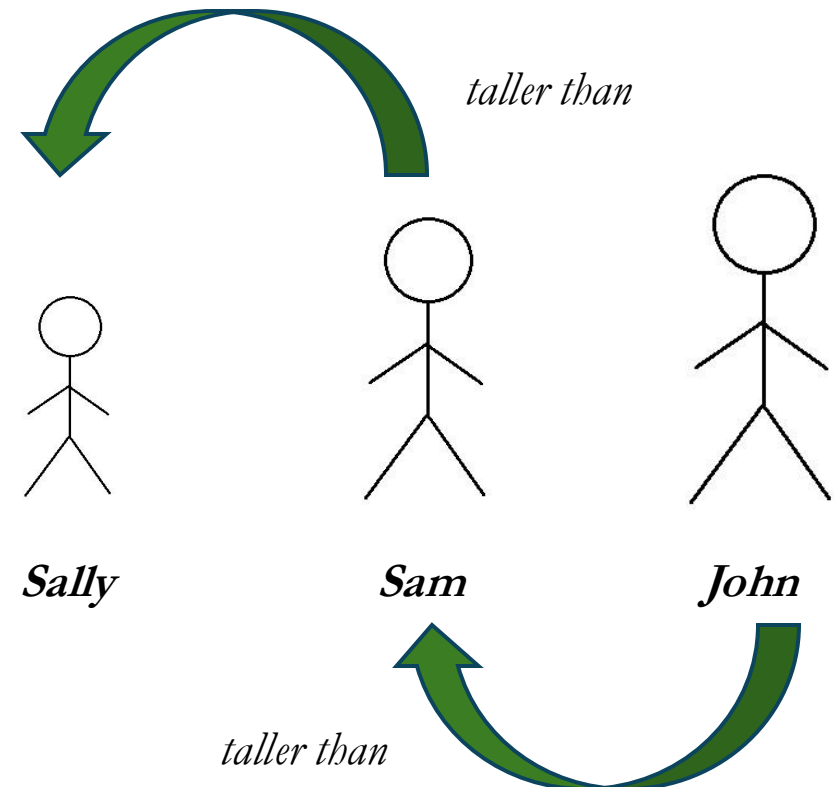
Relations

- Relations *themselves* have properties in natural language, which can reveal lots of information
- For example:
 - John *next to* Sam, means Sam is *next to* John



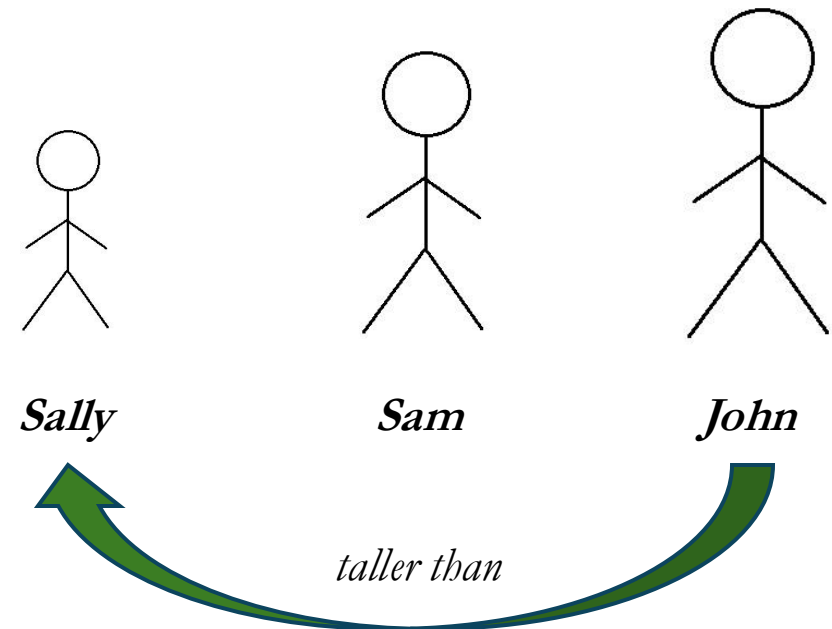
Relations

- Relations *themselves* have properties in natural language, which can reveal lots of information
- For example:
 - John *next to* Sam, means Sam is *next to* John
 - John *taller than* Sam and Sam *taller than* Sally, means John is *taller than* Sally



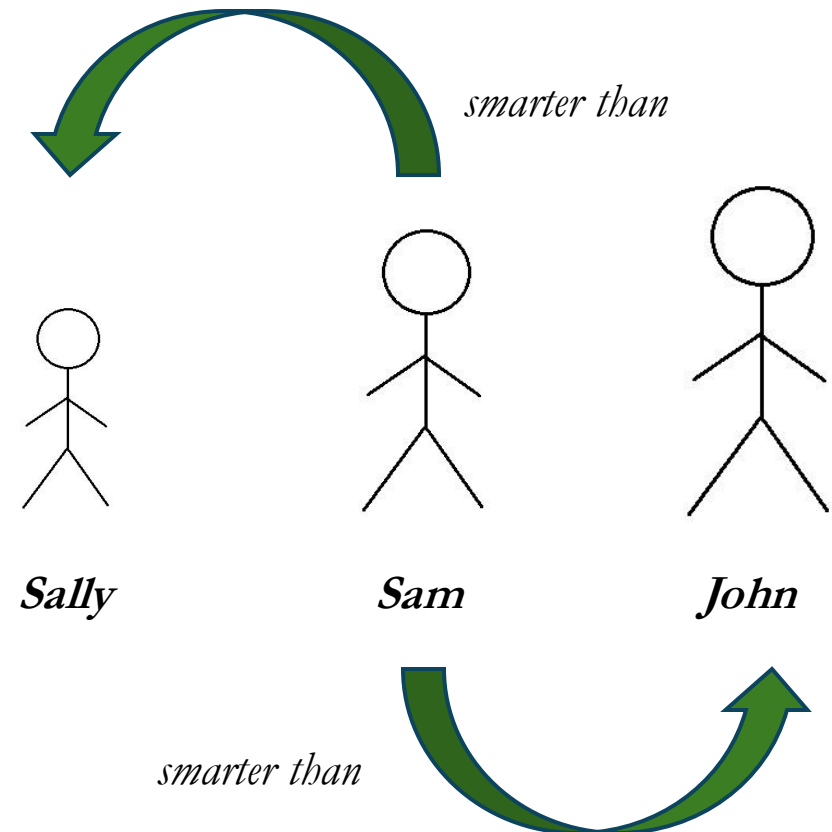
Relations

- Relations *themselves* have properties in natural language, which can reveal lots of information
- For example:
 - John *next to* Sam, means Sam is *next to* John
 - John *taller than* Sam and Sam *taller than* Sally, means John is *taller than* Sally



Relations

- Relations *themselves* have properties in natural language, which can reveal lots of information
- For example:
 - John *next to* Sam, means Sam is *next to* John
 - John *taller than* Sam and Sam *taller than* Sally, means John is *taller than* Sally
 - Sam as *the smartest person* in the room, means Sam is *smarter than* anyone in the room



Reflexivity

- A relation is *reflexive* just in case for anything it might apply to, that thing is related to itself
- For example, the relation *identical to* is reflexive
- This is because any x is always *identical to* itself

Symmetry

- A relation is *symmetric* just in case if one thing is related to another, the reverse relation holds as well
- For example, the relation *next to* is symmetric
- If x is *next to* y then y is *next to* x

Symmetry

- Some natural language relations have more than one property
- For example, the relation *is identical to* is both symmetric **and** reflexive
- That is, if *x is identical to y*, then *y is identical to x*, **and** *x is identical to x* **and** *y is identical to y*

Transitive

- A relation is transitive just in case if an object is related to another, which itself is related to a third, then the first is also related to the third
- For example, the relation *is taller than* is transitive
- If x is *taller than* y and y is *taller than* z , then x is *taller than* z

Transitive

- The relation *is identical to* is also transitive
- Any relation that has all three of these properties, reflexive, symmetric, and transitive, is called an *equivalence relation*
- Spotting equivalence relations allows you to group things in a coarse-grained manner

Equivalence Relation

- For example, consider the relation *has the same father as*
- Now suppose there are several people who bear this relation to each other
- If x *has the same father as* y , then y *has the same father as* x
- Anyone in this group *has the same father as* themselves, etc.

Parthood

- Which of the following seem true:
- Everything is part of itself...the maximal part!
- Consider just John's leg...now combine this with John's torso, elbows, etc., until you have all John's (proper) parts...what's the result?

Parthood

- Now, *is part of* does not seem symmetric
- In fact, it seems intuitive to say that *is part of* is **not** symmetric
- We might capture this as: If P_{xy} then it is not the case that P_{yx}
- But this cannot work...

Parthood

- SUPPOSE: If P_{xy} then it is not the case that P_{yx}
- The ‘x’ and ‘y’ can be *anything* in our domain; they’re unrestricted
- But that means whatever is being picked out by ‘x’ might be the same thing as whatever is being picked out by ‘y’
- So we can replace “ P_{xy} ” with “ P_{xx} ”

Parthood

- SUPPOSE: If P_{xy} then it is not the case that P_{yx}
- Which means with our supposition we have: If P_{xx} then it is not the case that P_{xx}
- But that's a **contradiction**!
- So we have to take back our supposition

Parthood

- Still, we don't want to say that *is part of* is symmetric; that's crazy
- We get around this by introducing another property:
- Anti-symmetric: If P_{xy} and P_{yx} then $x=y$
- This is consistent with reflexivity

Contained In

Leibniz's Law: If $x=y$ then any property x has, y has

- This applies to properties of relations...
- We've said *is part of* is reflexive, antisymmetric, and transitive
- Is the relation *is contained in* the same relation as *is part of*?

Contained In

Leibniz's Law: If $x=y$ then any property x has, y has

- Consider whether *is contained in* is transitive...antisymmetric...reflexive...

Contained In

Leibniz's Law: If $x=y$ then any property x has, y has

- Consider whether *is contained in* is transitive...antisymmetric...**reflexive**...

Contained In

Leibniz's Law: If $x=y$ then any property x has, y has

- Consider whether *is contained in* is transitive...antisymmetric...**reflexive**...
- It is *not* reflexive...and that's enough to show *is contained in* is not the same relation as *is part of*

Properties of Attitudes

- What goes for relations goes for attitudes
- By “attitude” I mean something like *belief, knows, wishes, desires*
- In other words, *cognitive* attitudes between an individual and some content, e.g. “A cheesecake”, “The Torah by heart”, “To be a millionaire”, “To get a job”

Properties of Attitudes

- Attitudes bear properties too...
- If John *knows* viruses are obligate parasites
- Then John *believes* viruses are obligate parasites

RATIONAL: If *S knows P* then *S believes P*

Properties of Attitudes

- If John *wishes* viruses to be obligate parasites
- Then John *desires* viruses to be obligate parasites

CONATIVE: If S *wishes* P then S *desires* P

Properties of Attitudes

- Can John imagine a round square?
- No.

CONTRADICTION: If “P” is a contradiction, then it is not the case that S can *imagine* P

Properties of Attitudes

- Can John know there is a round square?
- No.

CONTRADICTION: If “P” is a contradiction, then it is not the case that S can *know* P

Properties of Attitudes

- But John could *suppose* there is a round square
- And John could *wish* there is a round square

Neither of these attitudes satisfy CONTRADICTION

Properties of Attitudes

- It doesn't make sense to say:

“I *know* it's raining and I don't believe it”

- This is just belief and knowledge being logically connected

Properties of Attitudes

- And it's okay to say:

“I *believe* it's raining but I don't *know* it”

“I *believe* raining outside but I *suppose* it isn't”

“I *know* raining outside but I *imagine* it isn't”

“I *know* raining outside but I *wish* it wasn't”

Properties of Attitudes

- It doesn't sound right to say:

“I have the coffee and I want the coffee”

- S can continue to *want to keep* P, but that's different...

OWNERSHIP: If S has P, then S cannot desire P

Summary

- You may find it helpful to bucket types of CQs for your work
- Scoping questions are most likely where you will focus
- Because the true expressibility of our work is captured in relationships, you should become comfortable with the logic of relations

Outline

- Review of Design Pattern Submissions
- Varieties of Competency Question
- Not a Drill, Drill