

# Knowledge Engineering Pitfalls

# Knowledge Engineering Pitfalls

Which one is better to represent “Pizza margherita” ?

(A)  $Pizza(x) \wedge Margherita(x)$

(B)  $Pizza(x) \wedge \exists y. (hasType(x, y) \wedge PizzaMargherita(y))$

Which one is better?

(A)  $ThinkPad(TSeries)$

(B)

$\forall x. TSeries(x) \rightarrow ThinkPadModel(x)$

$\forall x. ThinkPad(x) \wedge \exists y. (hasModel(x, y) \wedge ThinkPadModel(y))$

Which one is better?

(A)

$\forall x. \text{DiskDrive}(x) \rightarrow \text{ComputerPart}(x)$

$\forall x. \text{Memory}(x) \rightarrow \text{ComputerPart}(x)$

$\forall x. \text{Computer}(x) \wedge \exists y. (\text{hasPart}(x, y) \wedge \text{ComputerPart}(y))$

(B)

$\forall x. \text{DiskPart}(x) \rightarrow \text{ComputerPart}(x)$

$\forall x. \text{MemoryPart}(x) \rightarrow \text{ComputerPart}(x)$

$\forall x. \text{DiskPart}(x) \rightarrow \text{DiskDrive}(x)$

$\forall x. \text{MemoryPart}(x) \rightarrow \text{Memory}(x)$

$\forall x. \text{Computer}(x) \wedge \exists y. (\text{hasPart}(x, y) \wedge \text{ComputerPart}(y))$

# Instantiation Pitfalls

- ▶ Does this ontology mean that “My ThinkPad is a ThinkPad Model”?

$$T21(mythinkpad123)$$
$$\forall x. T21(x) \rightarrow ThinkPadModel(x)$$

- ▶ Question: What ThinkPad models do you sell?
- ▶ Answer should NOT include My ThinkPad – nor yours.

$$\mathcal{K} \models ThinkPadModel(mythinkpad123)$$

## Instantiation Pitfalls (cont.)

- ▶ Corrected version

*NotebookComputer(mythinkpad123)*  
*hasModel(mythinkpad123, T21)*  
*TSeries(T21)*  
 $\forall x. TSeries(x) \rightarrow ThinkPadModel(x)$

# Composition Pitfalls

$\forall x. \text{MicroDrive}(x) \rightarrow \text{DiskDrive}(x)$

$\forall x. \text{DiskDrive}(x) \rightarrow \text{Computer}(x)$

$\forall x. \text{Memory}(x) \rightarrow \text{Computer}(x)$

- ▶ Question: What Computers do you sell?
- ▶ Answer should NOT include Disk Drives or Memory

## Composition Pitfalls (cont.)

- ▶ Corrected version

$\forall x. \text{MicroDrive}(x) \rightarrow \text{DiskDrive}(x)$

$\forall x. \text{DiskDrive}(x) \wedge \exists y. (\text{partOf}(x, y) \wedge \text{Computer}(y))$

$\forall x. \text{Memory}(x) \wedge \exists y. (\text{partOf}(x, y) \wedge \text{Computer}(y))$

# Disjunction Pitfalls

- ▶ Unintended model: flashcard110 is a computer-part

*hasPart(camera15, flashcard110)*

*Memory(flashcard110)*

$\forall x. \text{Computer}(x) \wedge \forall y. (\text{hasPart}(x, y) \rightarrow \text{ComputerPart}(y))$

$\forall x. \text{Memory}(x) \rightarrow \text{ComputerPart}(x)$

$\forall x. \text{DiskDrive}(x) \rightarrow \text{ComputerPart}(x)$

## Disjunction Pitfalls (cont.)

- ▶ Corrected version

*hasPart(camera15, flashcard110)*

*FlashMemory(flashcard110)*

*Camera(camera15)*

$\forall x. \text{Camera}(x) \rightarrow \neg \text{Computer}(x)$

$\forall x. \text{Computer}(x) \wedge \forall y. (\text{hasPart}(x, y) \rightarrow \text{ComputerPart}(y))$

$\forall x. \text{ComputerPart}(x) \leftrightarrow (\text{MemoryPart}(x) \vee \text{DiskPart}(x) \vee \dots)$

# Polysem Pitfalls

$\forall x. \text{Book}(x) \rightarrow \text{PhysicalObject}(x)$

$\forall x. \text{Book}(x) \rightarrow \text{AbstractEntity}(x)$

$\text{Book}(b1), \dots, \text{Book}(b5000)$

- ▶ Question: How many books do you have on Hemingway?
- ▶ Answer: 5,000

## Polysem Pitfalls (cont.)

- ▶ Corrected version

$\forall x. \textit{BookSense1}(x) \rightarrow \textit{PhysicalObject}(x)$

$\forall x. \textit{BookSense2}(x) \rightarrow \textit{AbstractEntity}(x)$

$\textit{BookSense2}(b_1), \dots, \textit{BookSense2}(b_{5000})$

## Constitution Pitfalls (WordNet)

$\forall x. \text{Metal}(x) \rightarrow \text{AmountOfMatter}(x)$

$\forall x. \text{Clay}(x) \rightarrow \text{AmountOfMatter}(x)$

$\forall x. \text{Computer}(x) \rightarrow \text{PhysicalObject}(x)$

$\forall x. \text{PhysicalObject}(x) \rightarrow \text{AmountOfMatter}(x)$

$\forall x. \text{AmountOfMatter}(x) \rightarrow \text{Entity}(x)$

- ▶ Question: What types of matter will conduct electricity?
- ▶ Answer should NOT include computers.

# Constitution Pitfalls (cont.)

► Corrected version

$\forall x. \text{Metal}(x) \rightarrow \text{AmountOfMatter}(x)$

$\forall x. \text{Clay}(x) \rightarrow \text{AmountOfMatter}(x)$

$\forall x. \text{Computer}(x) \rightarrow \text{PhysicalObject}(x)$

$\forall x. \text{PhysicalObject}(x) \wedge \exists y. (\text{constitutedBy}(x, y) \wedge \text{AmountOfMatter}(y))$

$\forall x. \text{AmountOfMatter}(x) \rightarrow \text{Entity}(x)$

$\forall x. \text{PhysicalObject}(x) \rightarrow \text{Entity}(x)$

# Temporality Pitfalls

1963(*chris*)

$\forall x. 1963(x) \rightarrow 1960s(x)$

$\forall x. 1964(x) \rightarrow 1960s(x)$

## Temporality Pitfalls (cont.)

- ▶ Corrected version

$1963\text{Births}(\text{chris})$

$\forall x.1963\text{Births}(x) \rightarrow 1960s\text{Births}(x)$

$\forall x.1964\text{Births}(x) \rightarrow 1960s\text{Births}(x)$

## Temporality Pitfalls (cont.)

*Person(chris), bornIn(chris, 1963)*

*Year(1963), Year(1964)*

*contains(1960s, 1963), contains(1960s, 1964)*

*Decade(1960s)*

# Spatial/Containment Pitfalls

$\forall x. \text{AlsaceRegion}(x) \rightarrow \text{FrenchRegion}(x)$

$\forall x. \text{LoireRegion}(x) \rightarrow \text{FrenchRegion}(x)$

Corrected ...

*Region(alsace), Region(loire)*

*contains(france, alsace), contains(france, loire)*

*Country(france)*

# About Instances

- ▶ For every class, think about what an instance of it is
  - ▶ What is an instance of “Loire Region”?
- ▶ Classes do not describe their subclasses
  - ▶ “Regions by Country” is a class of classes
- ▶ Criteria for individuation must remain constant within a taxonomy
  - ▶ Instance of a class is also an instance of every superclass
    - ▶ Thus “Chris” is not an instance of “1963 births”
  - ▶ Explore the “boundary conditions”
    - ▶ E.g. Changes in existence, distinctions with similar classes
- ▶ “Leaf Nodes” of a hierarchy have no special significance
  - ▶ Don’t switch to instances
- ▶ Think of an instance as the keyvalue of a record in a database, while of a class as the schema (signature) of a relational table

# Common Pitfalls

- ▶ Composition (part of)
  - ▶  $\forall x. \text{Arm}(x) \rightarrow \text{Body}(x)$
- ▶ Constitution
  - ▶  $\forall x. \text{Statue}(x) \rightarrow \text{Marble}(x)$
- ▶ Disjunction
  - ▶  $\forall x. \text{Car}(x) \rightarrow (\forall y. \text{hasPart}(x, y) \wedge \text{CarPart}(y))$
  - ▶  $\forall x. \text{Engine}(x) \rightarrow \text{CarPart}(x)$
  - ▶  $\forall x. \text{Tire}(x) \rightarrow \text{CarPart}(x)$
- ▶ Spatial
  - ▶  $\forall x. \text{NewYork}(x) \rightarrow \text{US}(x)$
- ▶ Polysemy
  - ▶  $\forall x. \text{Book}(x) \rightarrow \text{PhysicalObject}(x)$
  - ▶  $\forall x. \text{Book}(x) \rightarrow \text{ConceptualCreation}(x)$
- ▶ Arbitrary organisational nodes
  - ▶  $\forall x. \text{FictionalBookbyLatinAmericanAuthor}(x) \rightarrow \text{FictionalBook}(x)$
- ▶ Instance
  - ▶  $\text{Grape}(\text{pinotnoir})$
- ▶ Temporality

# Linguistic Tests

- ▶ If  $P$  subclass  $Q$ , you should be able to say “ $P$  is a kind of  $Q$ ”
- ▶ If  $a$  instanceOf  $P$ , you should be able to say, “ $a$  is a  $P$ ”
- ▶ If  $a$  instanceOf  $P$  subclassOf  $Q$ , you should be able to say “ $a$  is a  $Q$ ”
- ▶ For every instance, there should be a class it is (rigidly) an instance of that is its natural label
- ▶ You should not find it natural to say, if  $P$  subclassOf  $Q$ , “ $P$  has  $Q$ ”, “ $P$  might be  $Q$ ”, “ $P$  was  $Q$ ”, “ $P$  is in  $Q$ ”, “ $P$  is part of  $Q$ ”

# What's in a name

- ▶ Don't argue about what specific terms mean
  - ▶ Common software architecture argument: “What is a bridge?”
- ▶ Try and find the distinctions that matter
  - ▶ Assign them labels later
- ▶ Avoid “ish”, “-thing” & “other-” classes
  - ▶ Find good names that will avoid meaning creep
  - ▶ Other- classes create a maintenance nightmare
- ▶ Classes describe their instances
  - ▶ Remember the linguistic tests
- ▶ The superclass is not part of the name
  - ▶ So don't assume it is (e.g. `Best_Practices subclassOf Document`)