

Knowledge Graphs

Lecture 4 - Knowledge Representation with Ontologies

4.6 - How to Design your own Ontology

Prof. Dr. Harald Sack & Dr. Mehwish Alam

FIZ Karlsruhe - Leibniz Institute for Information Infrastructure

AIFB - Karlsruhe Institute of Technology

Autumn 2020



Leibniz-Institut für Informationsinfrastruktur

Knowledge Graphs

Lecture 4: Knowledge Representation with Ontologies

4.1 A Brief History of Ontologies

4.2 Why we do need Logic

Excursion 4: A Brief Recap of Essential Logics

Excursion 5: Description Logics

4.3 First Steps in OWL

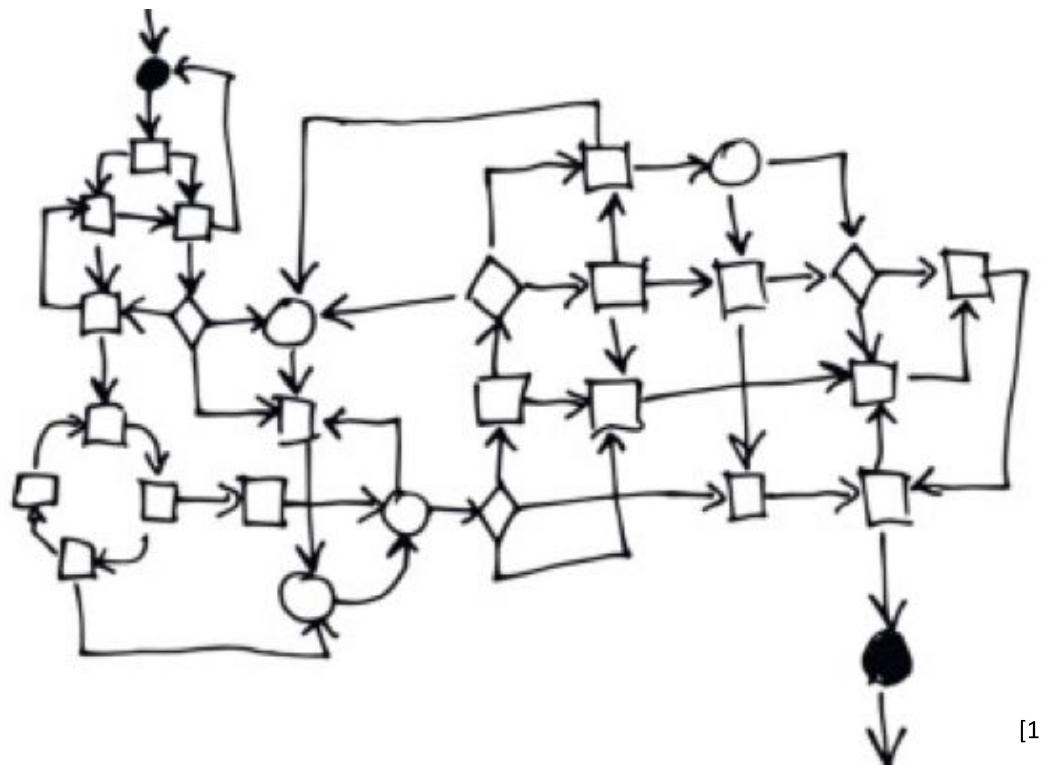
4.4 More OWL

4.5 OWL and beyond

4.6 How to Design your own Ontology

The Ontology Development Process

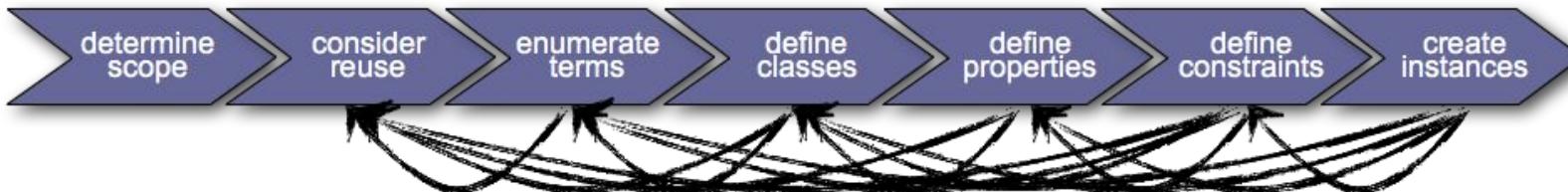
SOMETHING



Great Ontology

Ontology Development 101

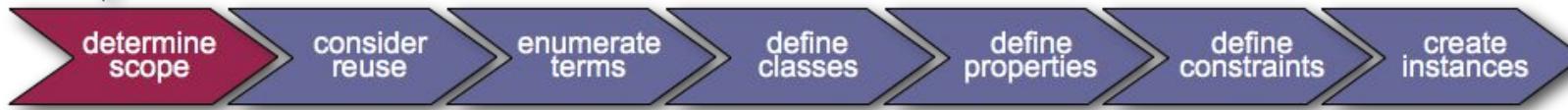
(Noy, McGuinness, 2000)



- In practice an **iterative Process** that **repeats continuously** and improves the ontology
- There are always **different approaches** for modelling an ontology
- In practice the designated application decides about the modelling approach

„There is no one correct way to model a domain.
There are always viable alternatives.“

Determine Domain and Focus



- Which **Domain** should be covered by the ontology?
- **What** should the ontology be used **for**?
- What types of **Questions** should be answered by the knowledge represented in the ontology?
- **Who** will use and maintain the ontology?
- Formulation of **Competence Questions**

Determine Domain and Focus

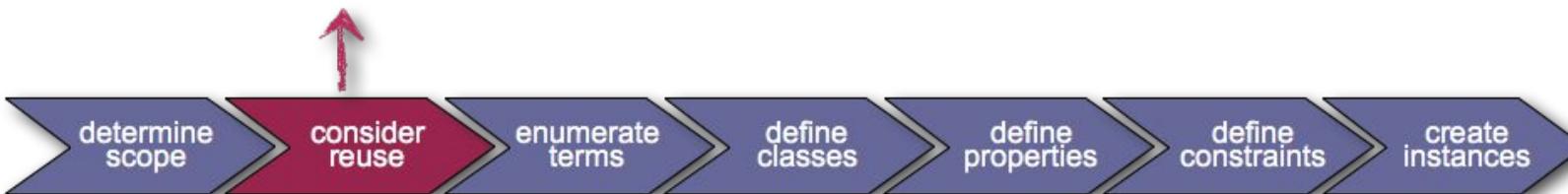


Competence Questions (Example: Wine Ontology)

- Which properties of the wine should be considered for modelling?
- Is Bordeaux a white wine or a red wine?
- Does a Sauvignon Blanc match with fish?
- Which wine matches best for grilled vegetables?
- Which properties of a wine do influence whether it matches with a specific dish?
- Does the bouquet of a wine change over time?
- Does the taste of a wine change over time?
- ...

These Questions might change
within the ontology life cycle

Consider Reuse



- Why should we consider reuse?
 - In order to save **cost**
 - In order to apply **tools** that are applied for other existing ontologies also for our own ontology
 - In order to reuse ontologies that have been validated by their application

If you don't find a suitable ontology or if the adaption is too complex then create a new ontology!

Develop a Terminology



- About which **concepts** are we talking?
- Which **properties** have these concepts?
- **What do we want to say** about these concepts?

Example: Wine Ontology

- *wine, grape, winery, location,...*
- *a wine's color, body, flavor, sugar content,...*
- *subtypes of wine: white wine, red wine, Bordeaux wine,...*
- *types of food: seafood, fish, meat, vegetables, cheese,...*
- ...

Develop Classes and Class Hierarchies

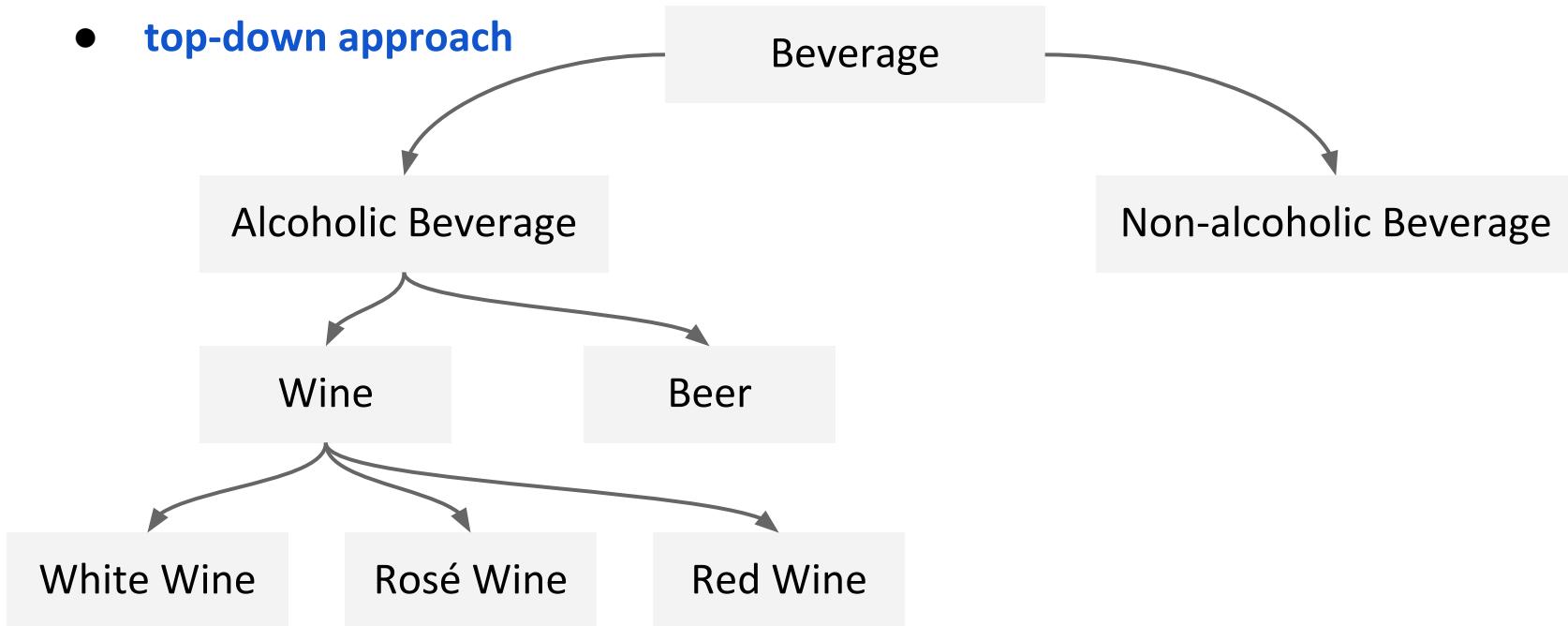


- **Classes** are concepts in the designated domain
 - *class of wines*
 - *class of wineries*
 - *class of red wines*
 - ...
- Classes are collections of objects with **similar properties**.
- Choose a **top-down / bottom-up / middle-out** approach to model class hierarchies.

Develop Classes and Class Hierarchies



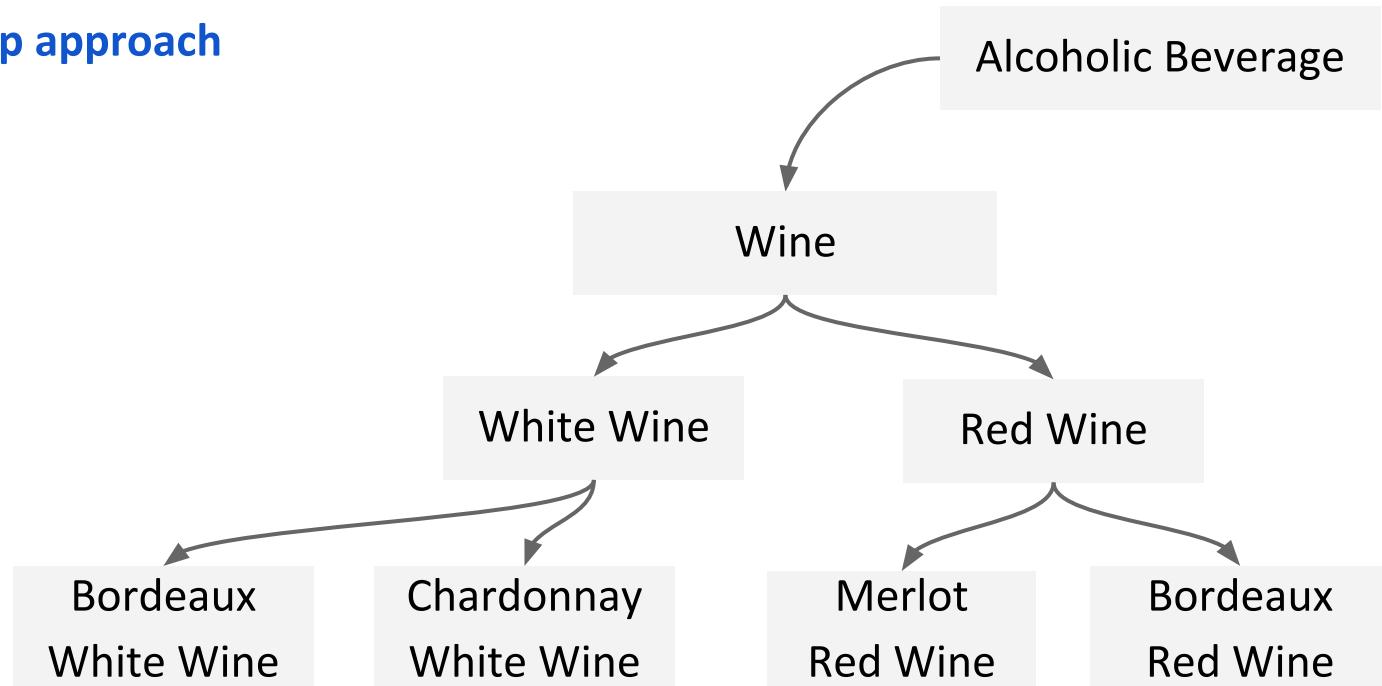
- **top-down approach**



Develop Classes and Class Hierarchies



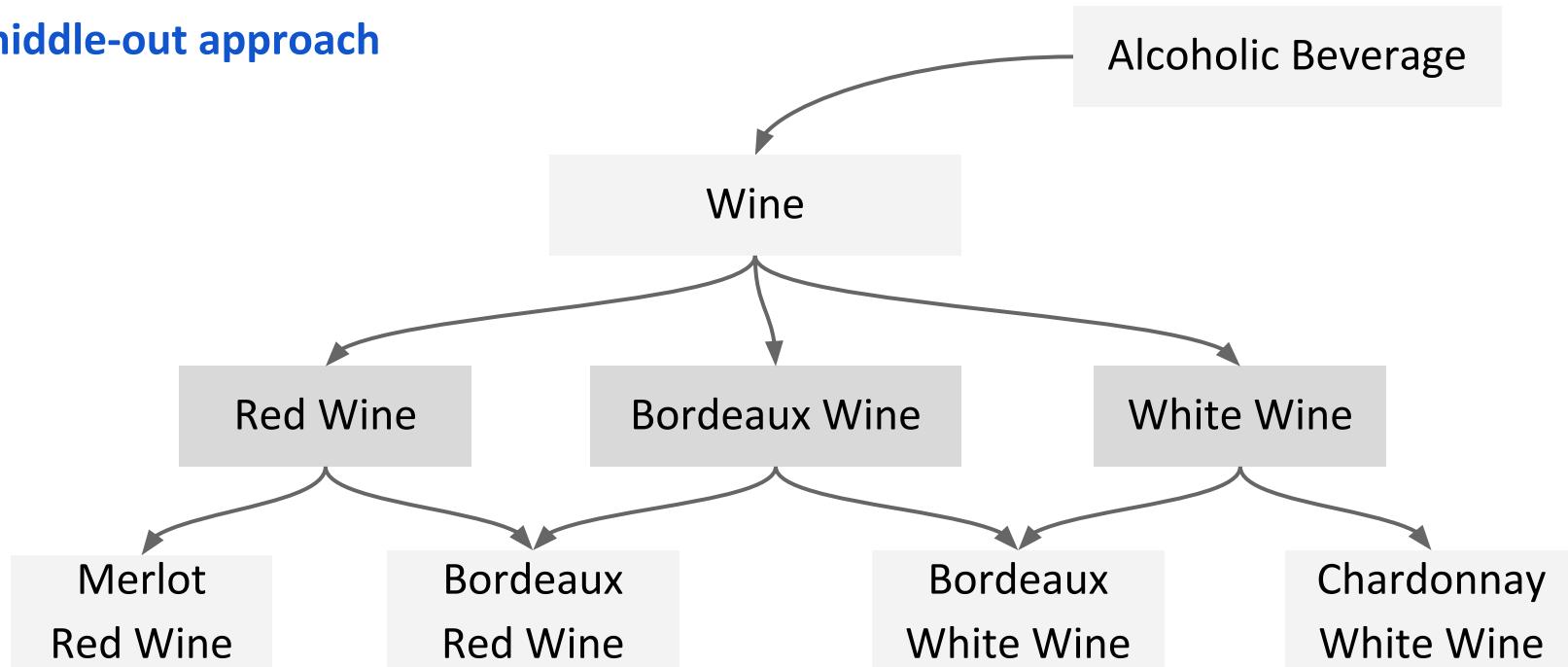
- **bottom-up approach**



Develop Classes and Class Hierarchies



- middle-out approach



Define Properties



- Properties in a class definition describe attributes of instances
 - every wine has a color, residual sugar, producer, etc...*

P Properties and Restrictions

	D	O	P	L	R	X
▶ <input type="checkbox"/> hasColor (single WineColor) (cardinality 1)						
▶ <input type="checkbox"/> hasWineDescriptor (multiple WineDescriptor)						
▶ <input type="checkbox"/> madeFromGrape (multiple WineGrape) (minCardinality 1)						
▶ <input type="checkbox"/> hasBody (single WineBody) (cardinality 1)						
▶ <input type="checkbox"/> hasFlavor (single WineFlavor) (cardinality 1)						
▶ <input type="checkbox"/> hasMaker (allValuesFrom Winery, cardinality 1)						
▶ <input type="checkbox"/> hasSugar (single WineSugar) (cardinality 1)						
▶ <input type="checkbox"/> locatedIn (multiple Region) (someValuesFrom Region)						
▶ <input type="checkbox"/> food:madeFromFruit (multiple food:SweetFruit, food:NonSweetFruit)						

Define Property Constraints



- **Property constraints** (restrictions) describe or restrict the set of possible property values
 - *The name of a wine is a String*
 - *The producer is an instance of Winemaker...*

P Properties and Restrictions

	D	O	P	R	X
▶ <input type="checkbox"/> hasColor (single WineColor) (cardinality 1)					
▶ <input type="checkbox"/> hasWineDescriptor (multiple WineDescriptor)					
▶ <input type="checkbox"/> madeFromGrape (multiple WineGrape) (minCardinality 1)					
▶ <input type="checkbox"/> hasBody (single WineBody) (cardinality 1)					
▶ <input type="checkbox"/> hasFlavor (single WineFlavor) (cardinality 1)					
▶ <input type="checkbox"/> hasMaker (allValuesFrom Winery, cardinality 1)					
▶ <input type="checkbox"/> hasSugar (single WineSugar) (cardinality 1)					
▶ <input type="checkbox"/> locatedIn (multiple Region) (someValuesFrom Region)					
▶ <input type="checkbox"/> food:madeFromFruit (multiple food:SweetFruit, food:NonSweetFruit)					

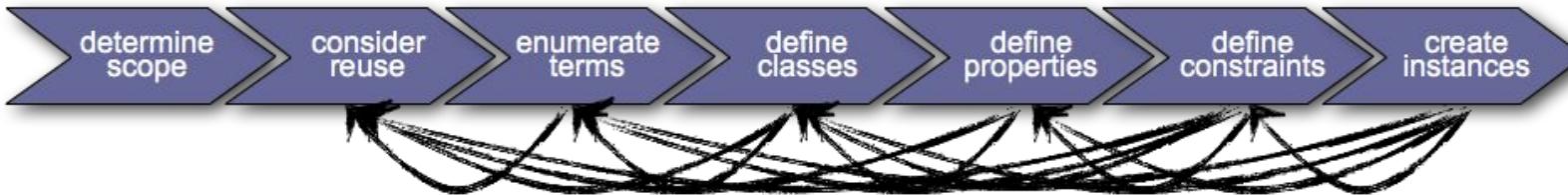
Definition of Instances



- Create **instances for the classes**
 - Every **class** directly becomes the **type** of its instances.
 - Every **superclass** of a direct type is also type of its instances.
- Create **instances for properties**, i.e. assignment of property values for the instances according to the given constraints
- „*The glass of red wine that I drank last supper...*“

Ontology Development 101

(Noy, McGuinness, 2000)



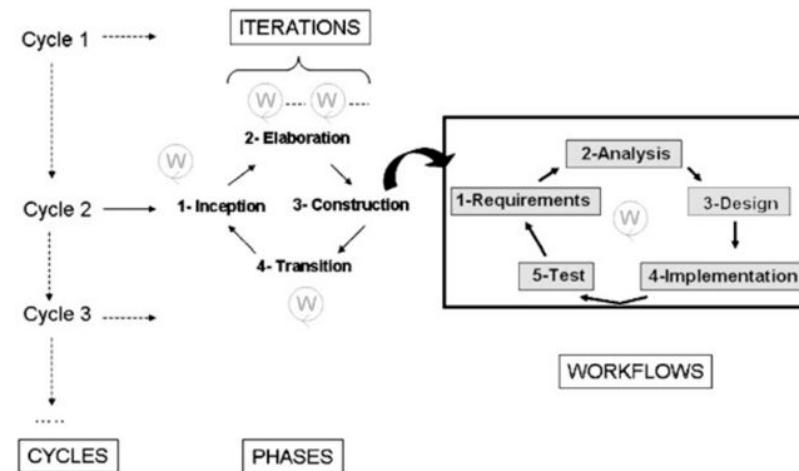
- Ontology development in practice is an **iterative process that repeats continuously** and improves the ontology.

http://protege.stanford.edu/publications/ontology_development/ontology101.pdf

Unified Process for Ontology Building

De Nicola, Missikoff, Navigli (2005)

- Development is divided into **Cycles**, which are subdivided into 4 **Phases** of **Iterations** (*Inception, Elaboration, Construction, Transition*). Each iteration results in a new prototype

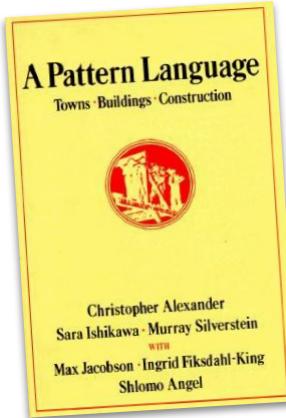


- Each iteration consists of 5 workflows (*Requirements, Analysis, Design, Implementation, Test*)

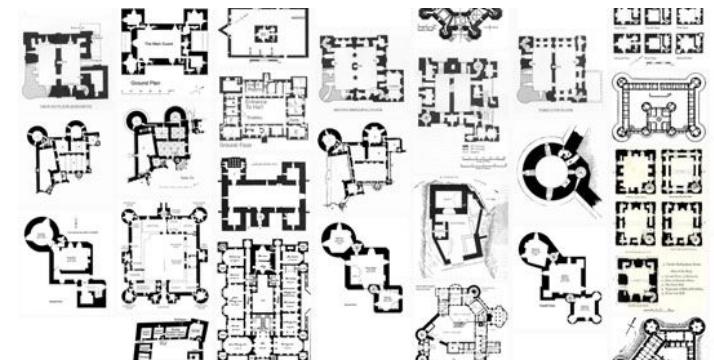
Ontology Design Patterns

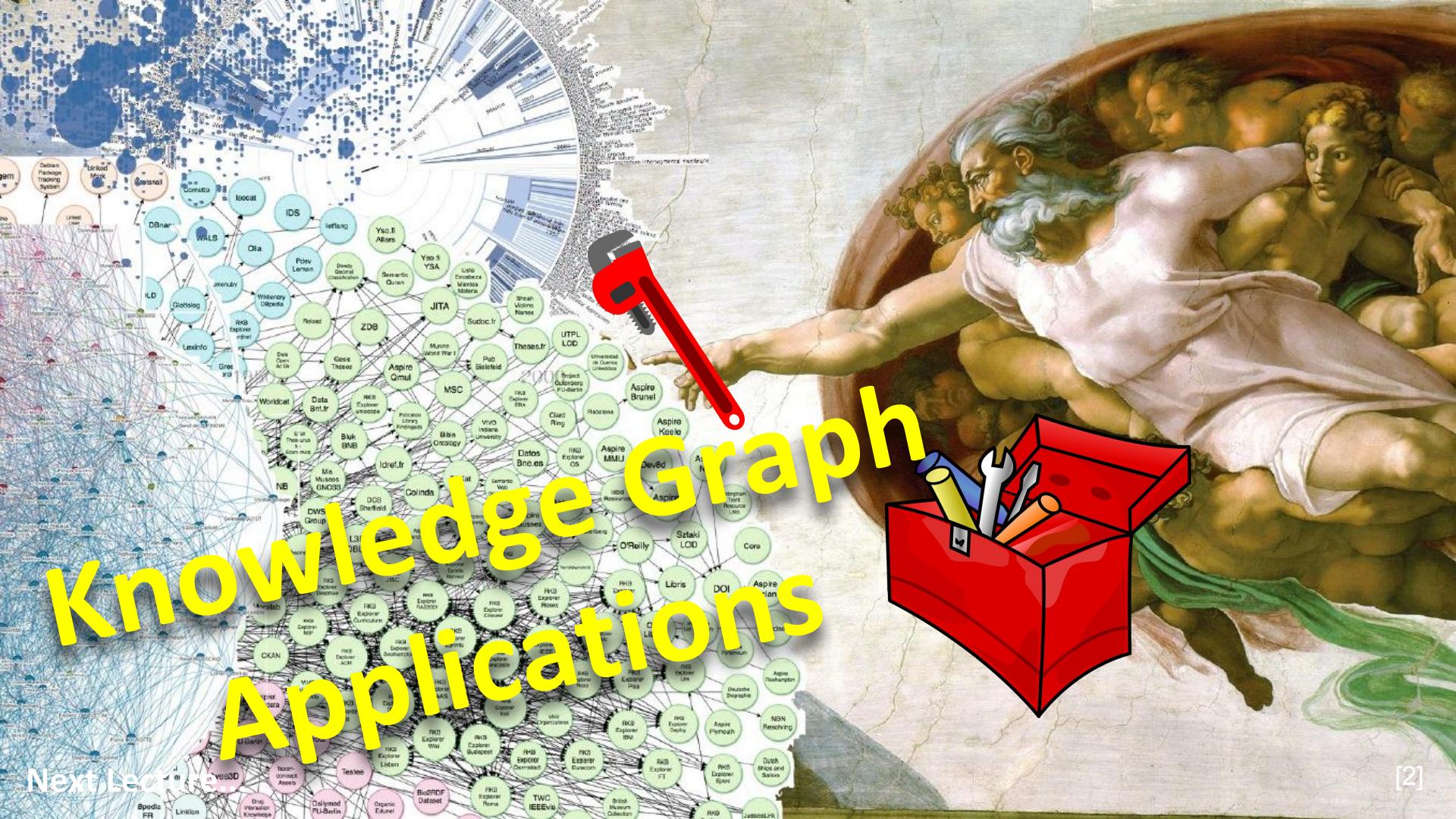
Gangemi (2005)

- Adapting a design idea originally from architecture
 - Recurring modeling problems
 - Providing a set of adaptable standard solutions
 - *A “pattern” is a solution to a problem in a given context* (Christopher Alexander)



The elements of this language are entities called patterns. Each pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice. — *Christopher Alexander*





Picture References:

- [1] The Software Development Process, Geek & Poke,
<http://geekandpoke.typepad.com/geekandpoke/2012/01/simply-explained-dp.html>
- [2] Rembrandt, The Anatomy Lesson of Dr. Nicolaes Tulp, 1632 [Public Domain]
https://commons.wikimedia.org/wiki/File:Rembrandt_-_The_Anatomy_Lesson_of_Dr_Nicolaes_Tulp.jpg