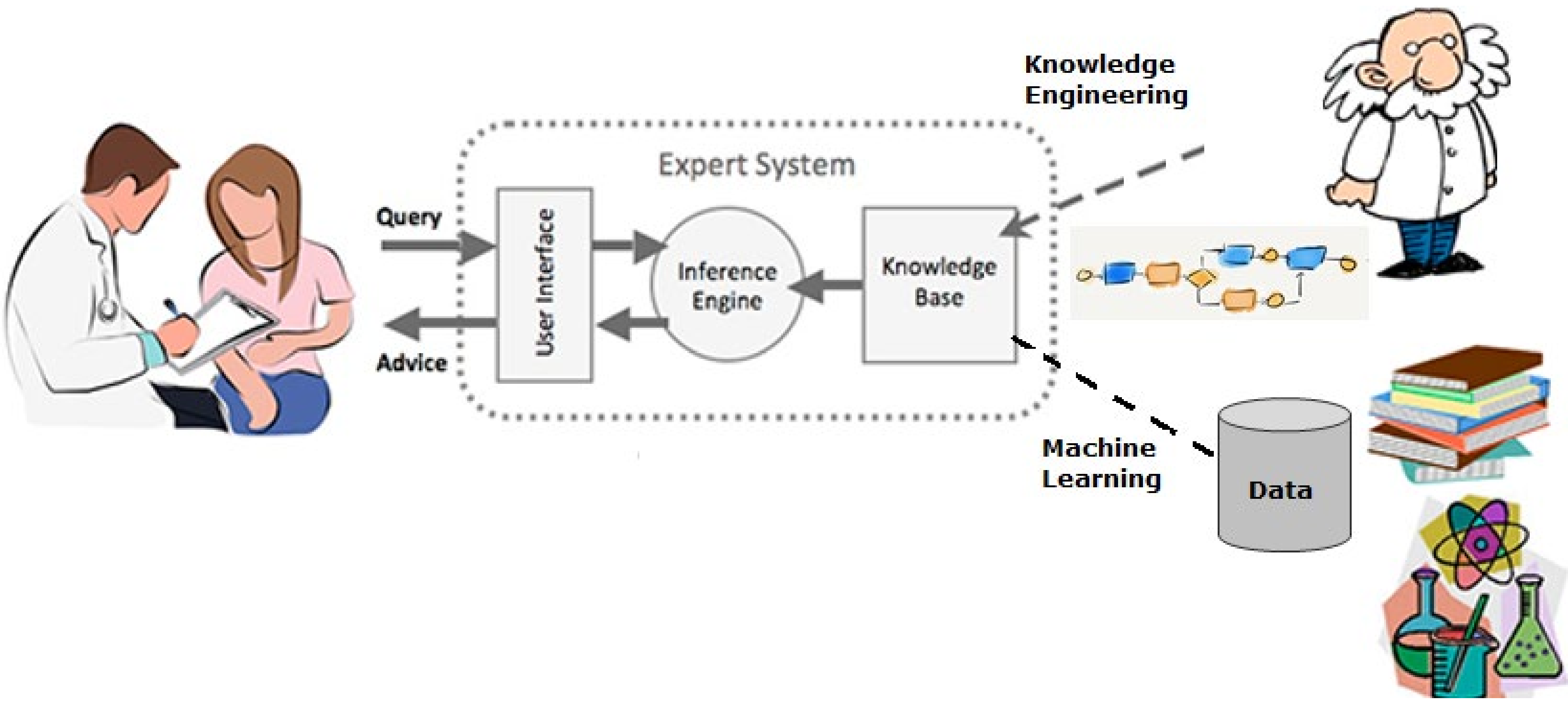
Knowledge Engineering and Business Intelligence

Repetition

Preliminary Note

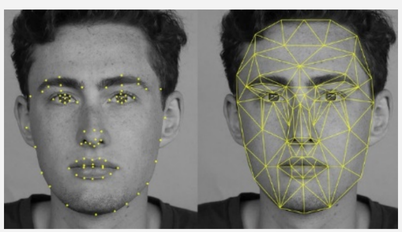
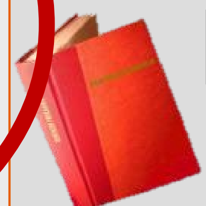
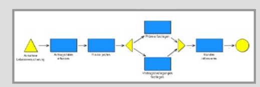
- *These slides guide through the topics of the module Knowledge Engineering and Business Intelligence*
- *It is intended as a collection of the main content that you need to know to pass the exam.*
- *Instead, it provides a structure of the module in order to stimulate discussions and questions of the students*

Knowledge-Based Systems



Knowledge Engineering

Knowledge Engineering

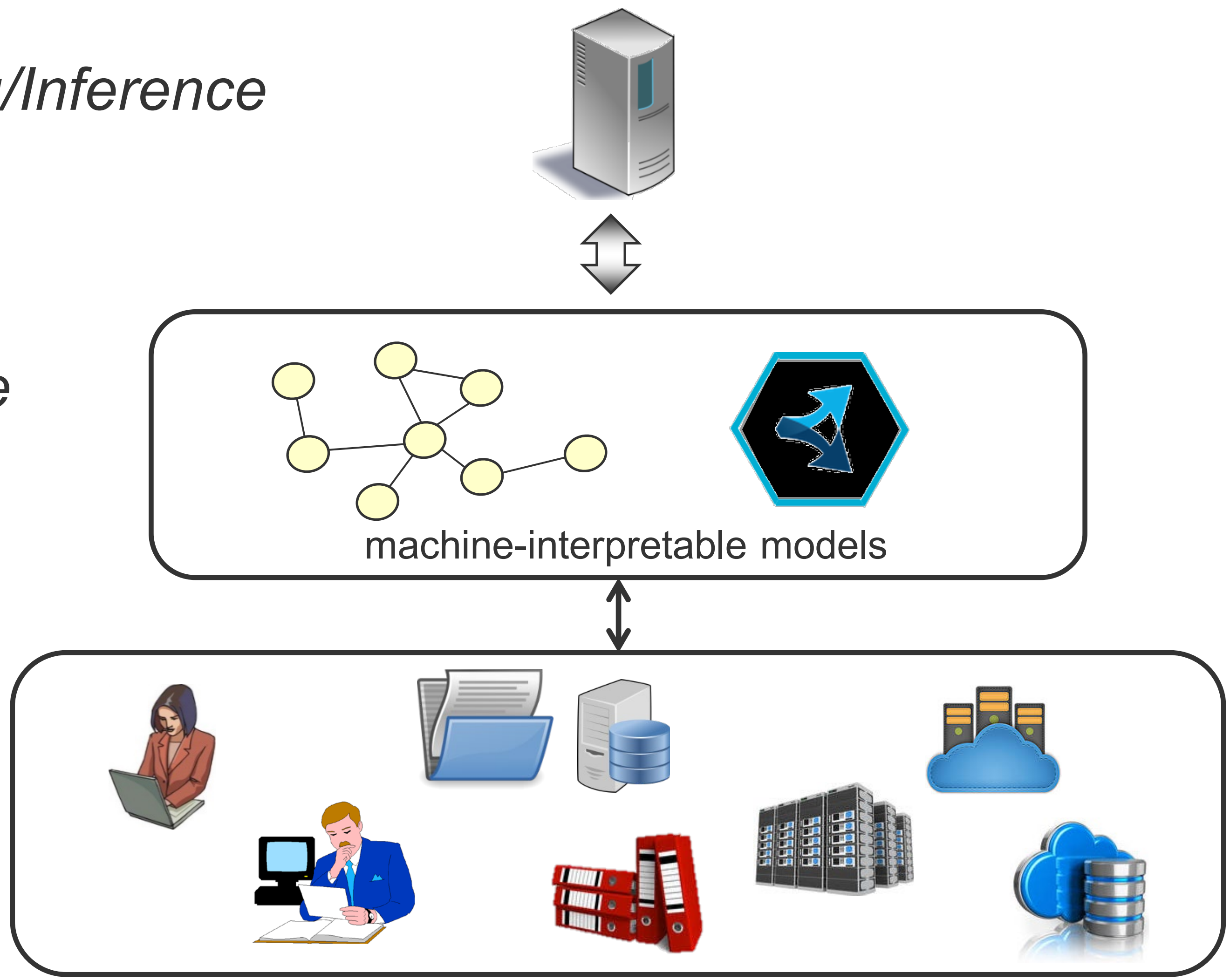
internal knowledge		external knowledge	
tacit knowledge	**self-aware knowledge**	**documented knowledge**	**automated knowledge** knowledge bases
		  $E = m c^2$	

Knowledge-Representation and Reasoning

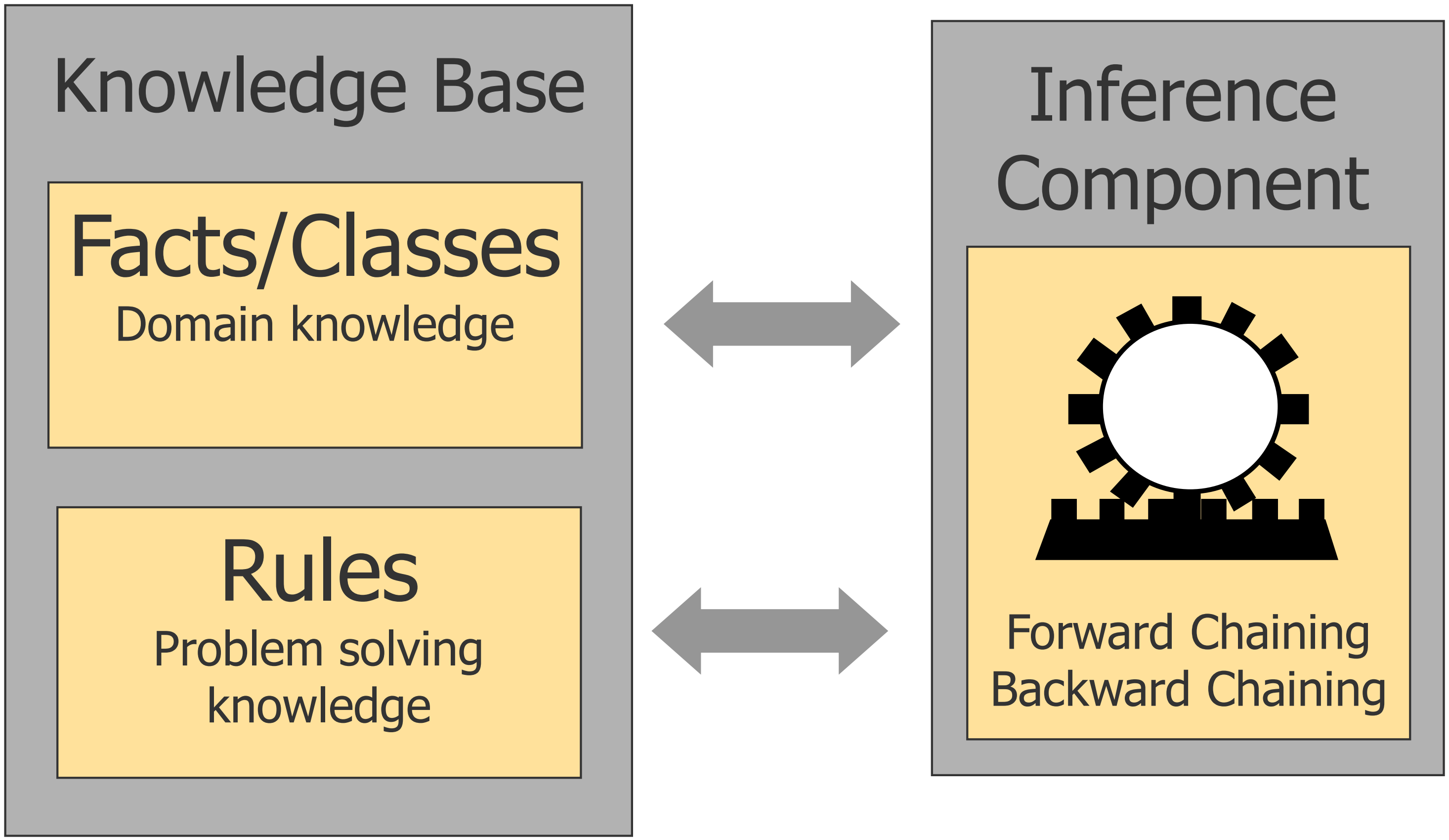
Reasoning/Inference

Knowledge Base

Reality



Rule-Based


Knowledge Base
Facts/Classes
Domain knowledge
Rules
Problem solving knowledge
Inference Component
Forward Chaining
Backward Chaining

Decision Tables and Rule-Based Systems

Eligibility			
A	Degree valid	University Registered	eligible
	yes,no	*yes, no, unclear*	*yes,no*
1	yes	yes	yes
2	no	-	no
3	-	no	no
4	-	unclear	no

- Rule 1:
  - IF Temperature = *low*
    THEN heating power is *increased*
- Rule 2:
  - IF Temperature = *normal*
    AND humidity = *low*
    THEN heating power is *normal*

Facts:

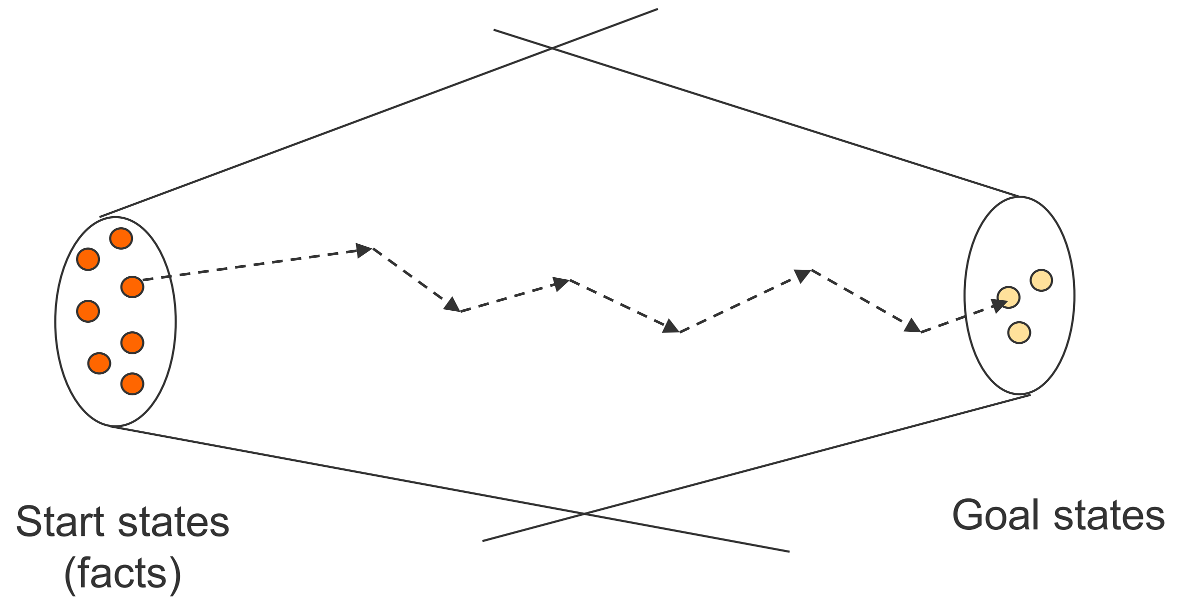
father(peter,mary)
father(peter,john)
mother(mary,mark)
mother(jane,mary)

Rules:

father(X,Y) → parent(X,Y)
mother(X,Y) → parent(X,Y)
father(X,Y) AND parent(Y,Z) → grandfather(X,Z)
mother(X,Y) AND parent(Y,Z) → grandmother(X,Z)
parent(X,Y) AND parent(X,Z) → sibling(Y,Z)

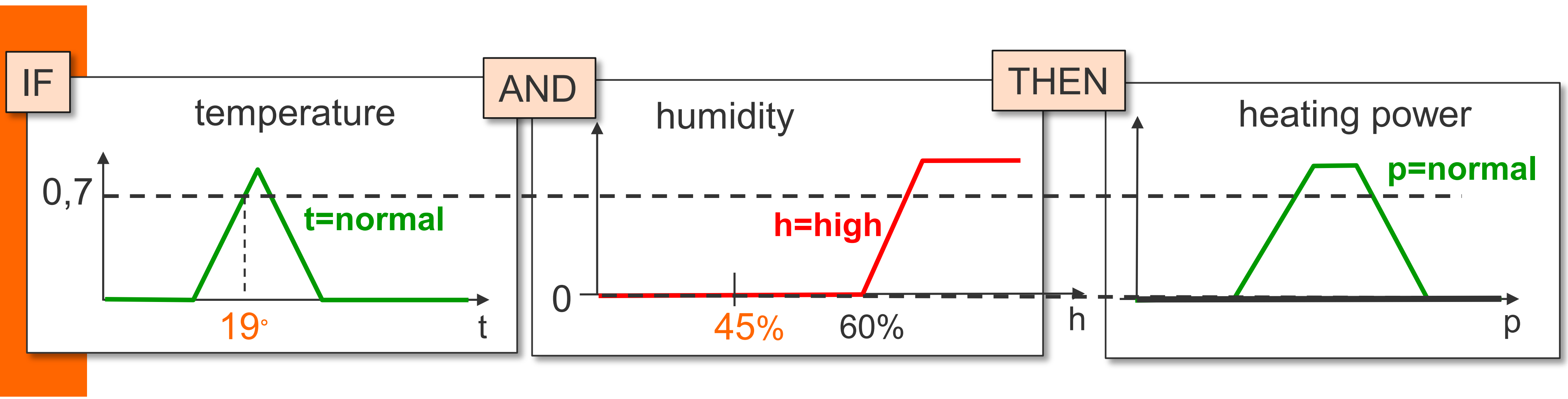
Forward and Backward Chaining

- Backward Chaining
  - If you already know what you are looking for
- Forward Chaining
  - If you don't necessarily know the final state of your solution

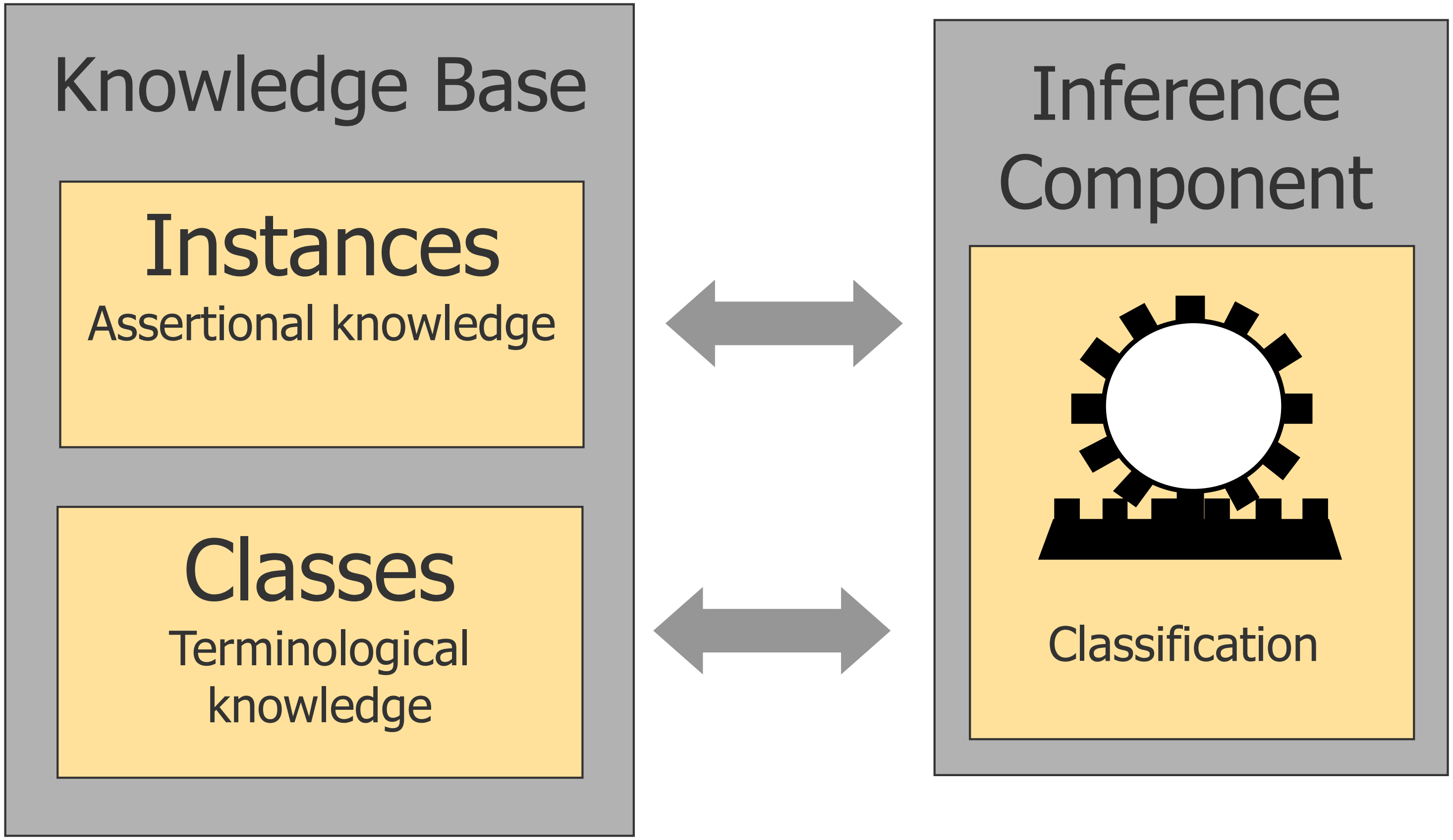


Fuzzy Logic – Vague Knowledge

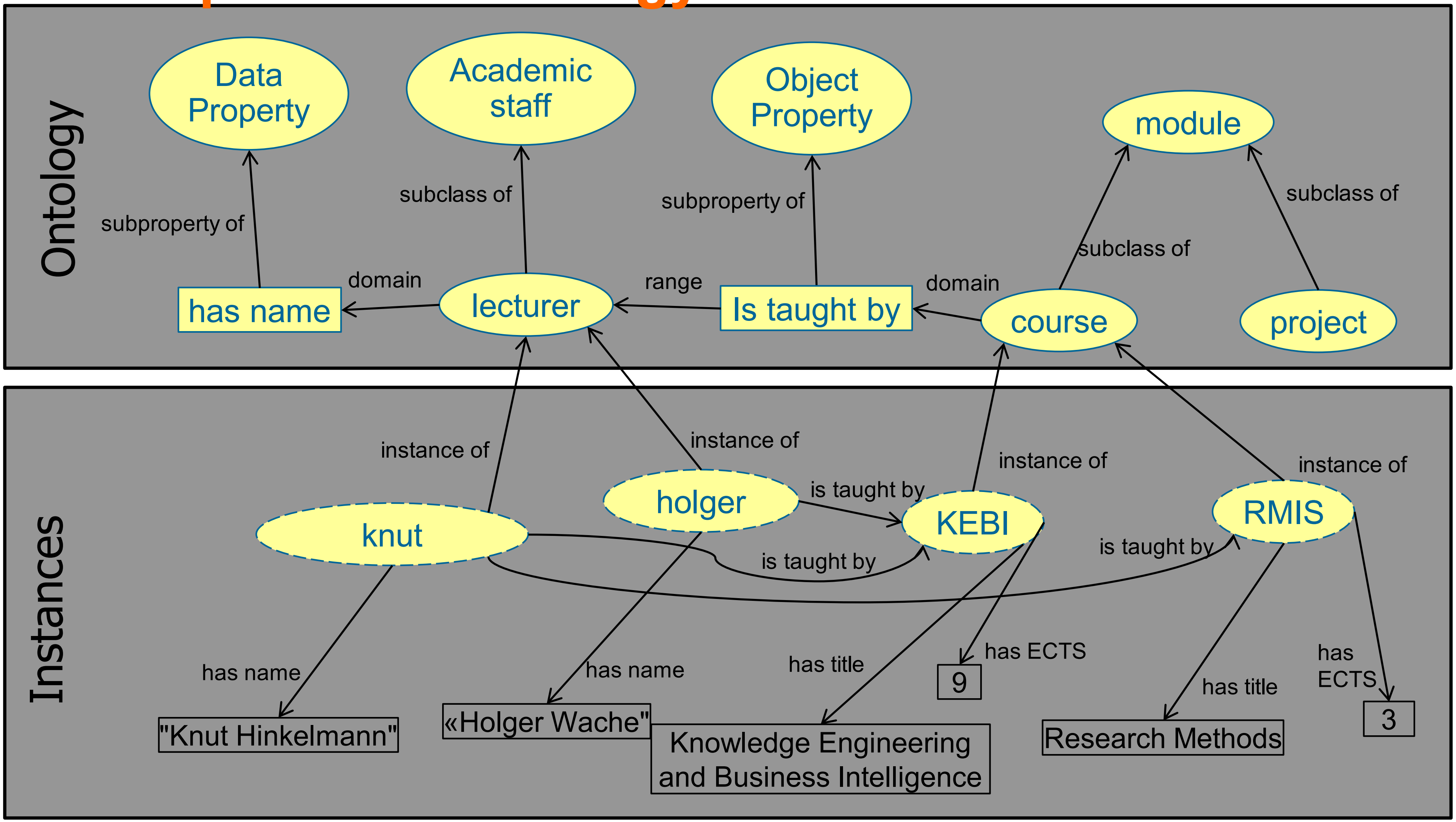
IF Temperature = *normal* AND humidity = *high*
THEN heating power is *high*



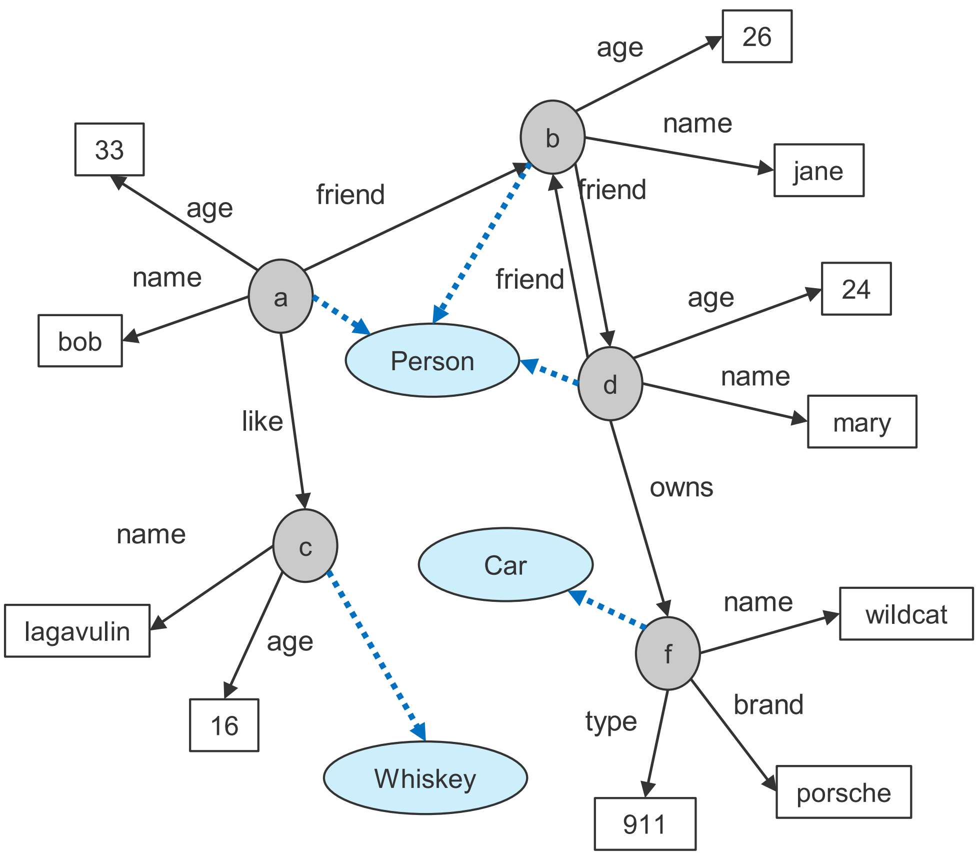
Knowledge Graphs - Ontologies



Example of an Ontology



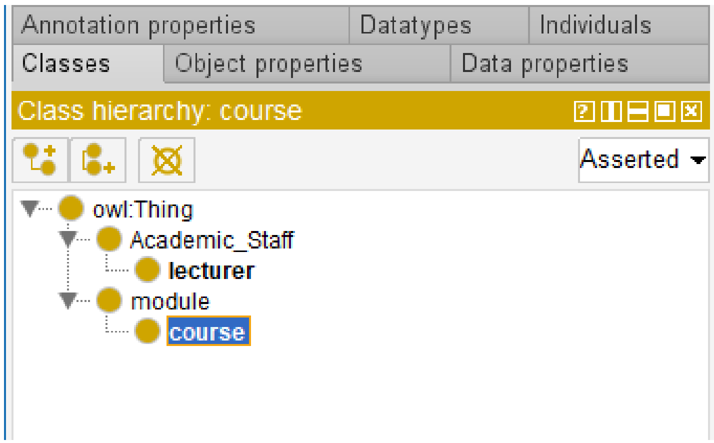
KnowledgeNets, Triples and Types



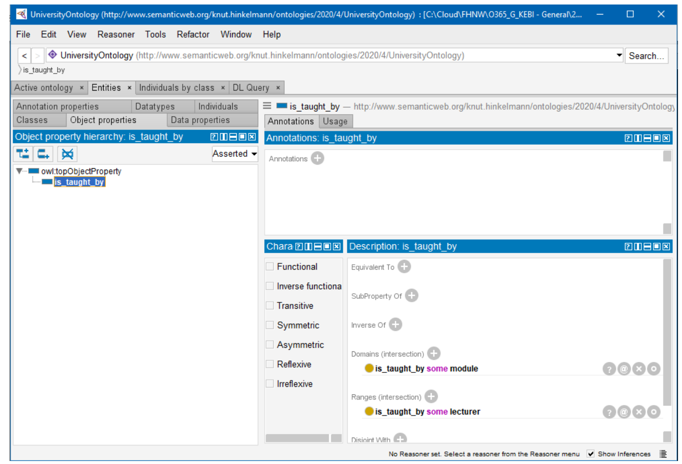

```
triple(a,rdf:type,Person)
triple(a,name,bob).
triple(a,age,33).
triple(a,friend,b).
triple(a,like,c).

triple(b,rdf:type,Person)
triple(b,name,jane).
triple(b,age,26).
triple(b,friend,d).

triple(c,rdf:type,Whiskey)
triple(c,name,lagavulin).
triple(c,age,16).

triple(d,rdf:type,Person)
triple(d,name,tom).
triple(d,age,24).
triple(d,friend,b).
triple(d,owns,f).

triple(f,rdf:type,Car)
triple(f,name,wildcat).
triple(f,brand,porsche).
triple(f,type,911).
```

Ontology Engineering

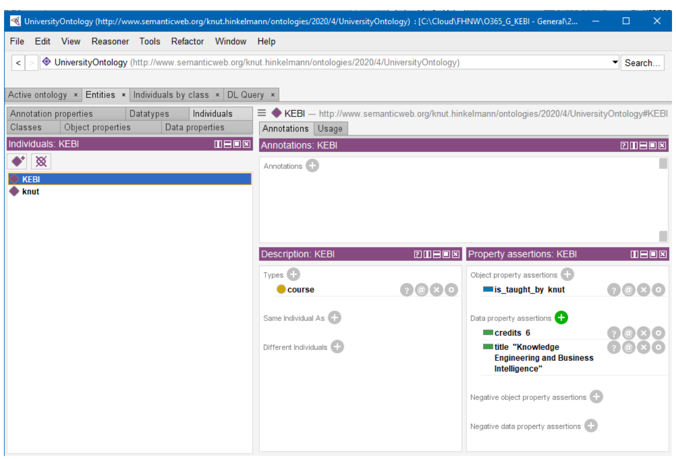
Class Hierachy



Properties



Instances

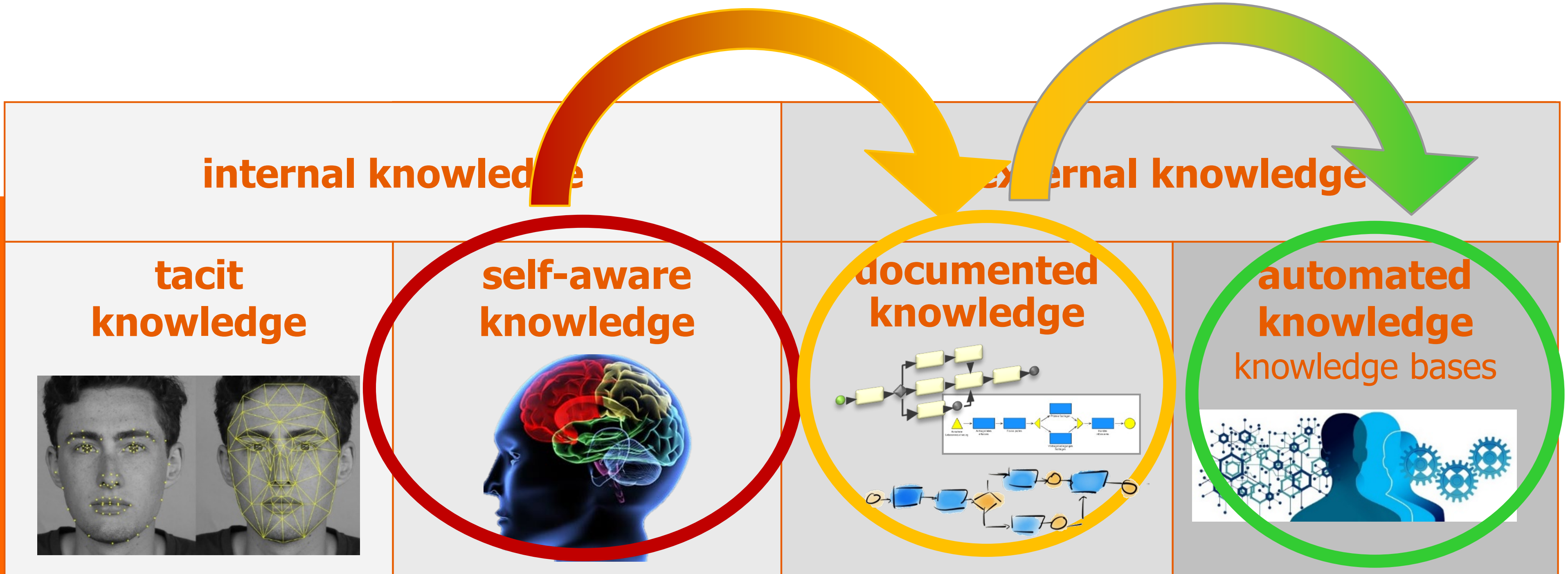


Queries

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?instance
        WHERE { ?instance rdf:type lecturer}
```

A Two-step Approach for Building a Knowledge Base

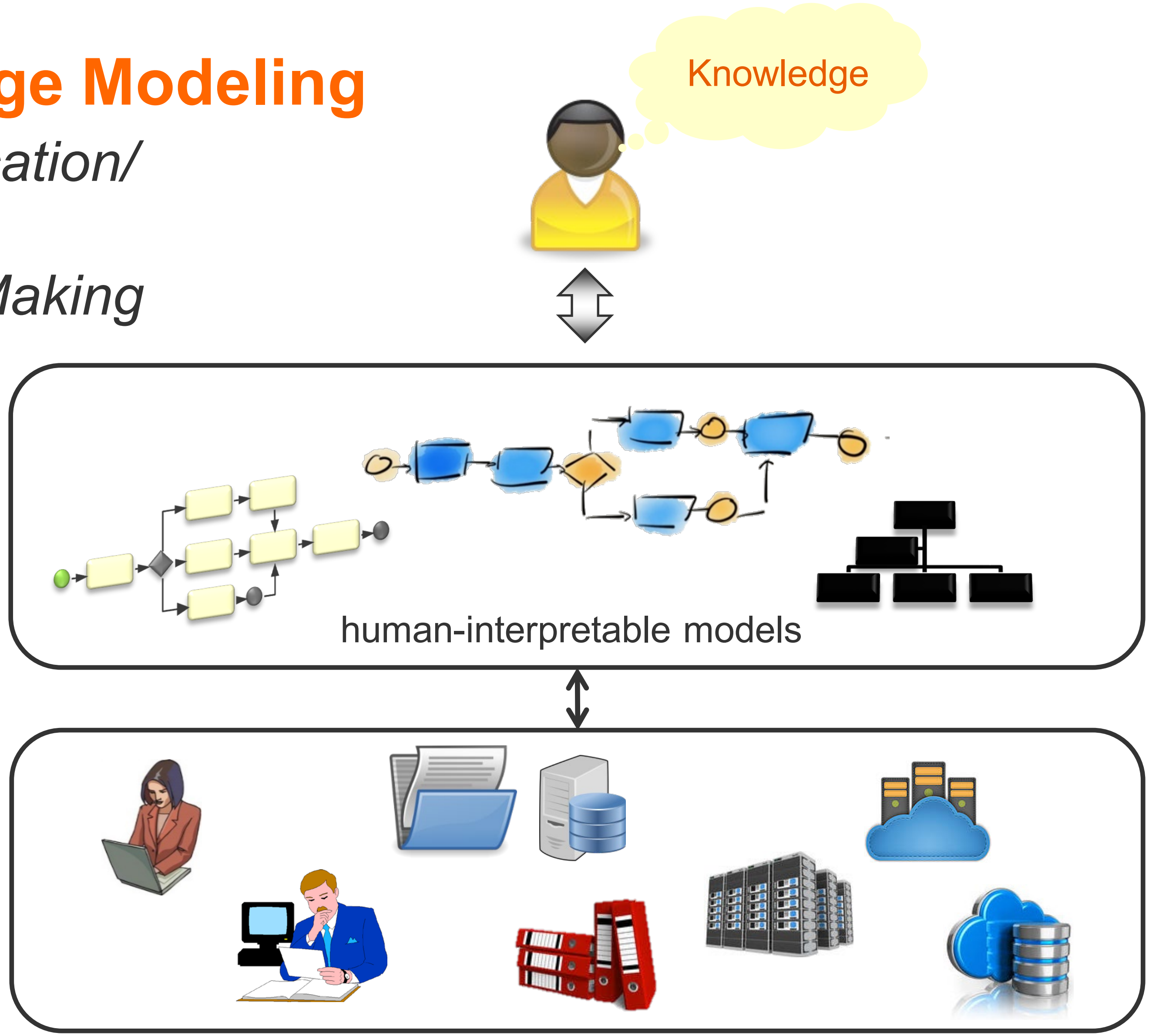


Knowledge Modeling

Communication/
Analysis/
Decision Making

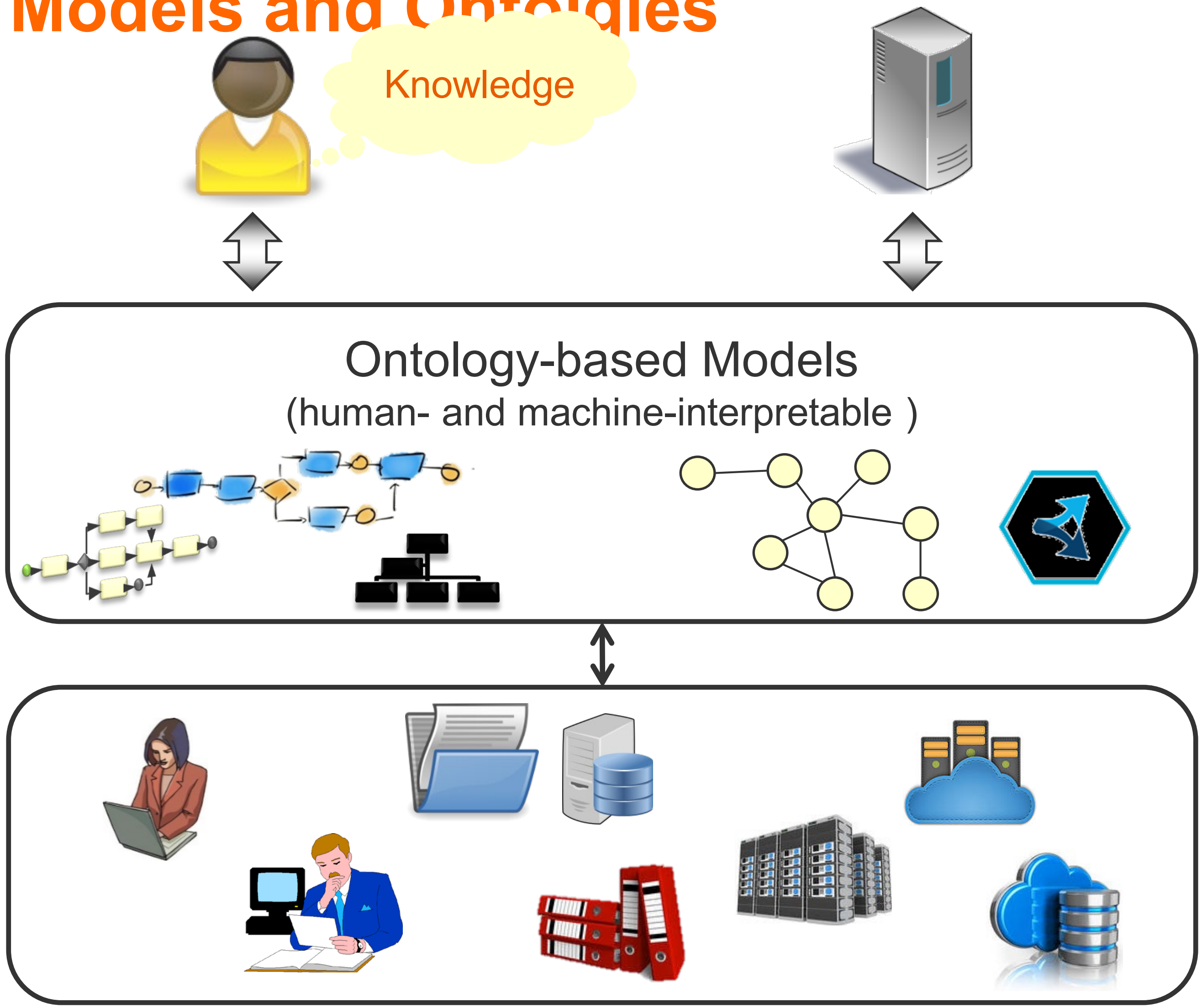
Models

Reality



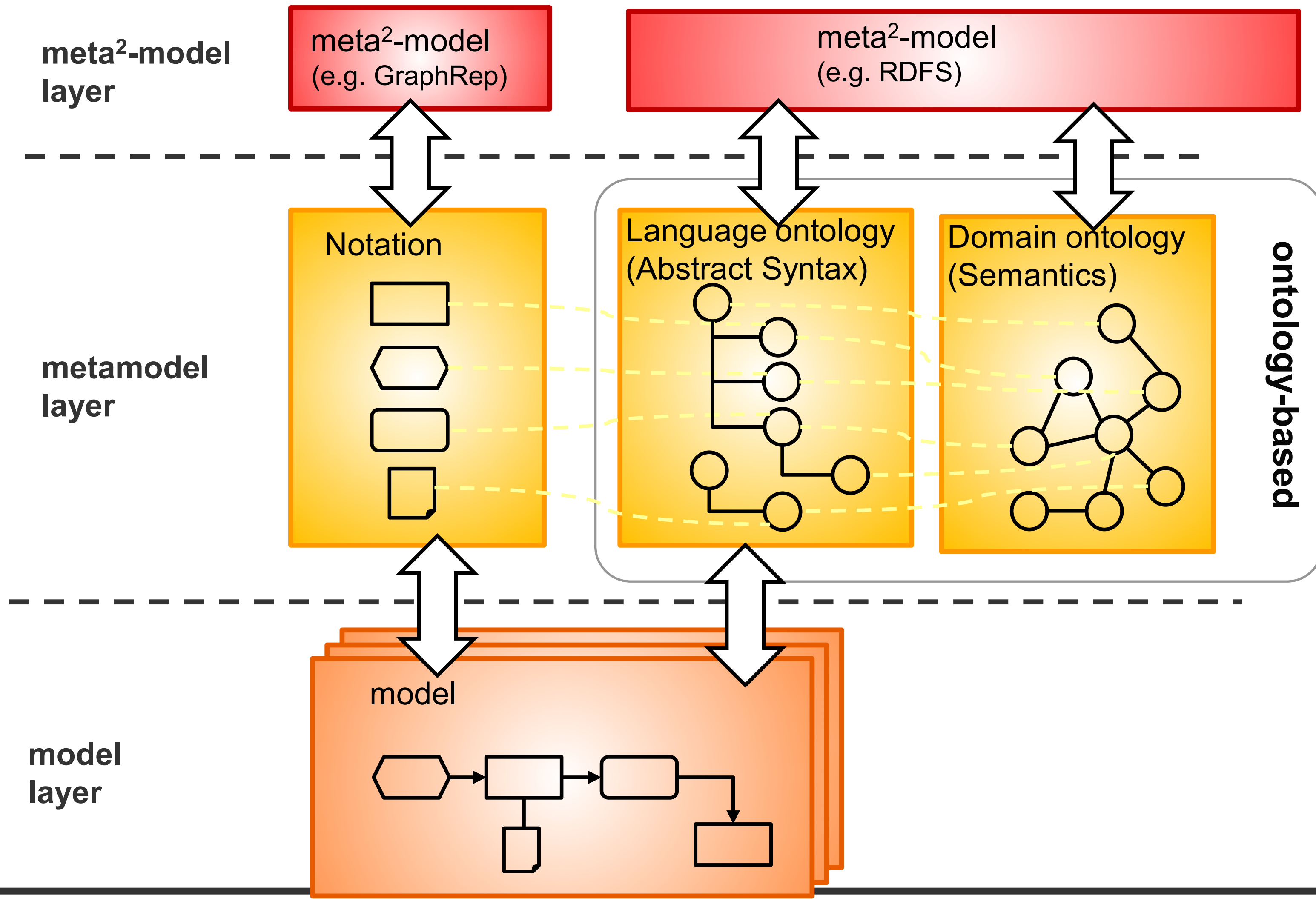
Combine Models and Ontolgies

Models + Knowledge



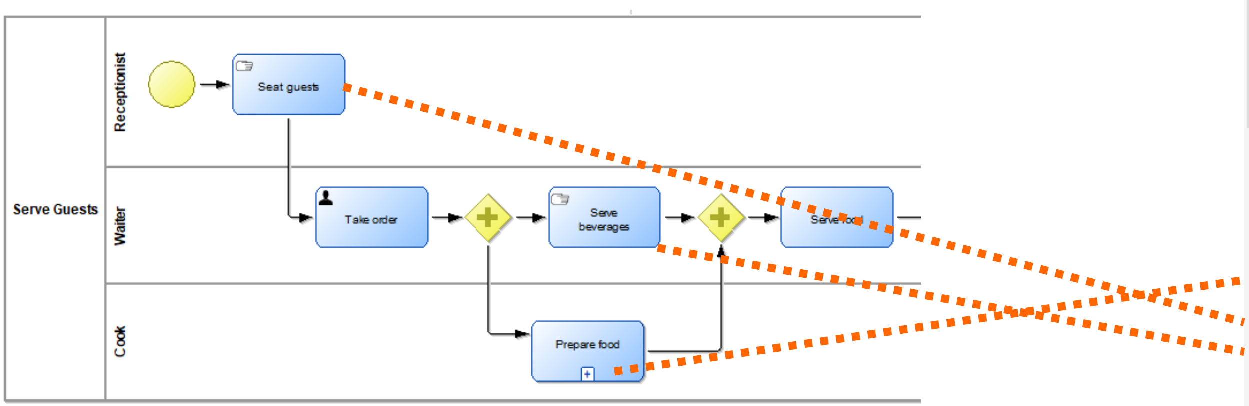
Reality

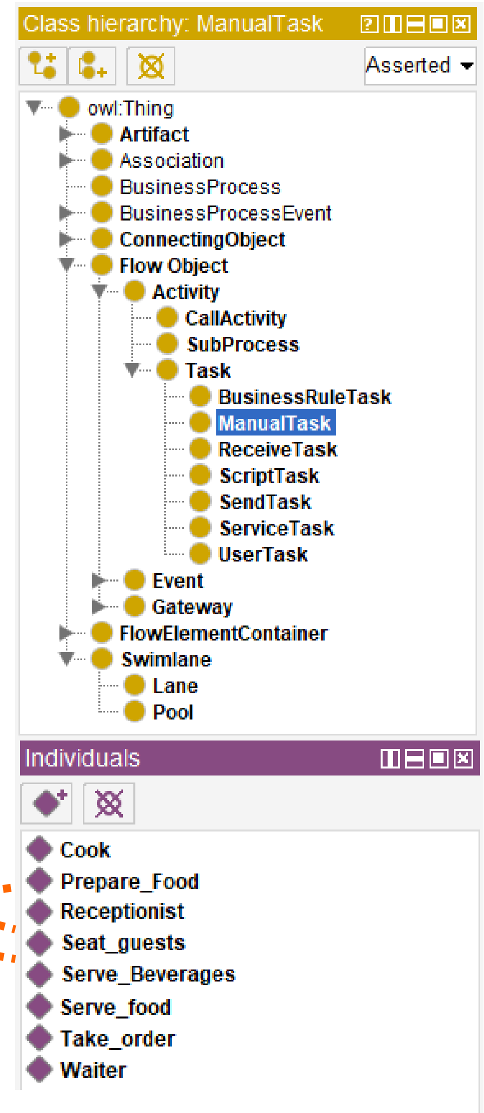
Ontology-based Metamodeling



Ontology-Based Metamodel

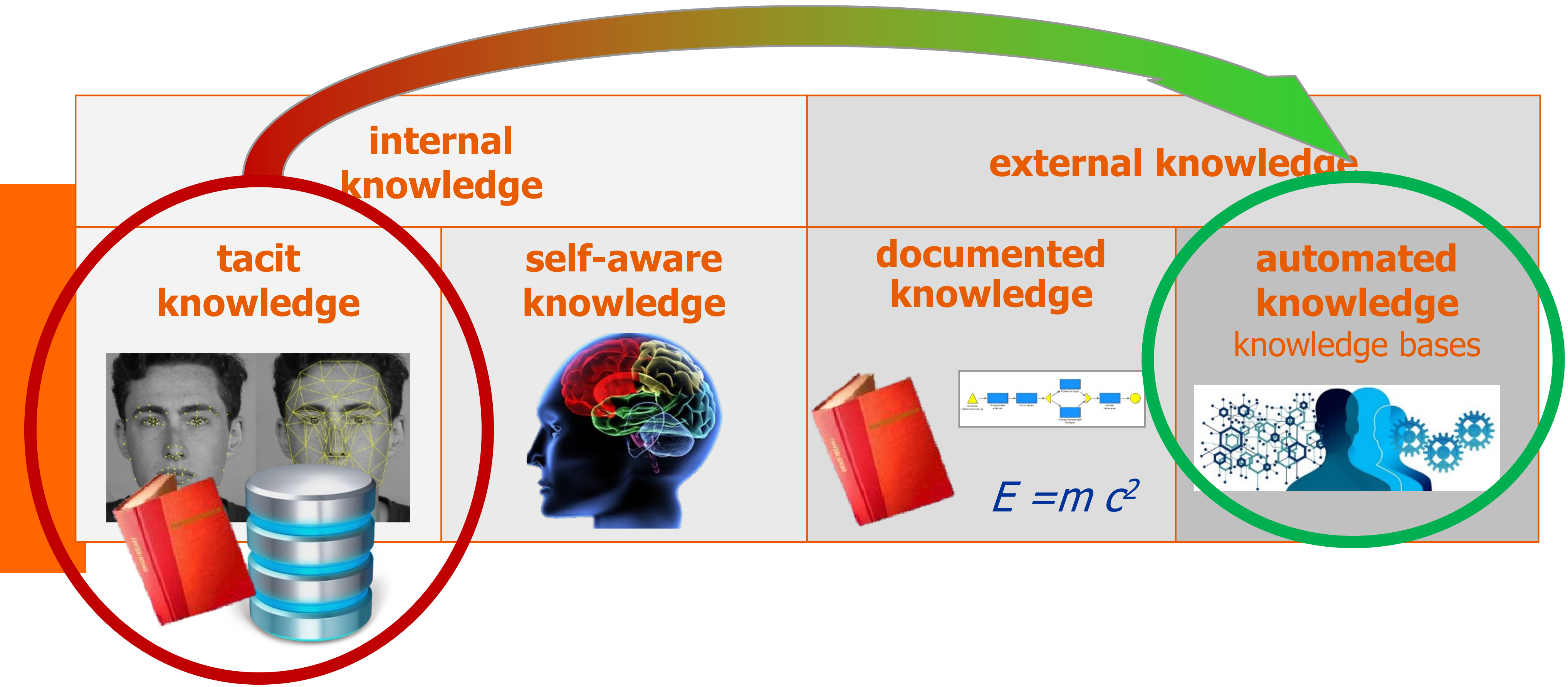
- Single environment for modelling and ontology
- Model elements are directly created as instances in the ontology



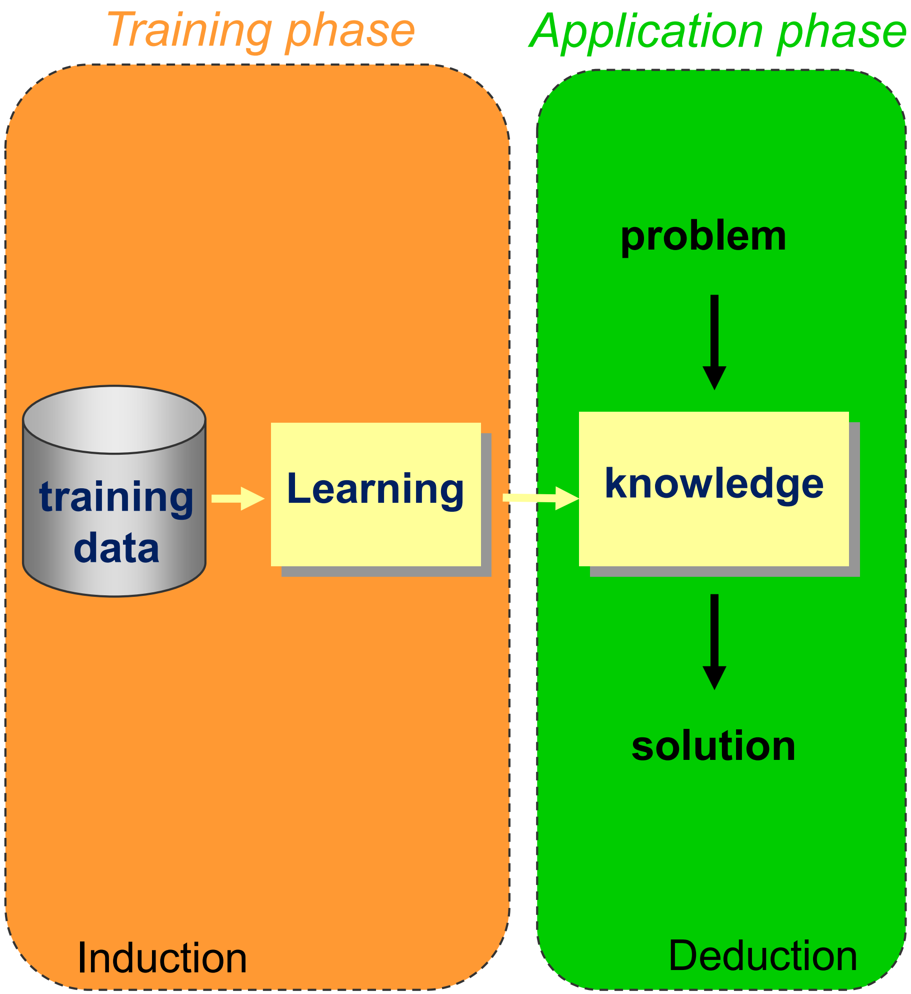


Machine Learning

Machine Learning:
Learning (Tacit) Knowledge from Data



Machine Learning: General Idea



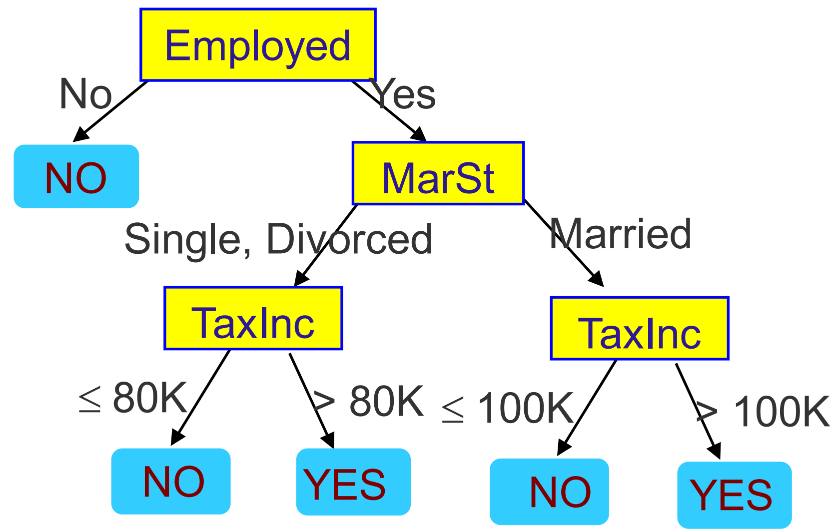
- **Training**
  - Collect data for the problem
  - Use the data to learn how to solve the type of problem
  - Result: Knowledge base
- **Application**
  - Use the learned knowledge for new problems

Learning Decision Trees: Generalisation of Data

categorical categorical continuous class

Tid	Employed	Marital Status	Taxable Income	accept
1	No	Single	125K	**No**
2	Yes	Married	160K	**Yes**
3	Yes	Single	70K	**No**
4	No	Married	120K	**No**
5	Yes	Divorced	95K	**Yes**
6	Yes	Married	60K	**No**
7	No	Divorced	220K	**No**
8	Yes	Single	85K	**Yes**
9	Yes	Married	95K	**No**
10	Yes	Single	90K	**Yes**

Training Data



Model: Decision Tree

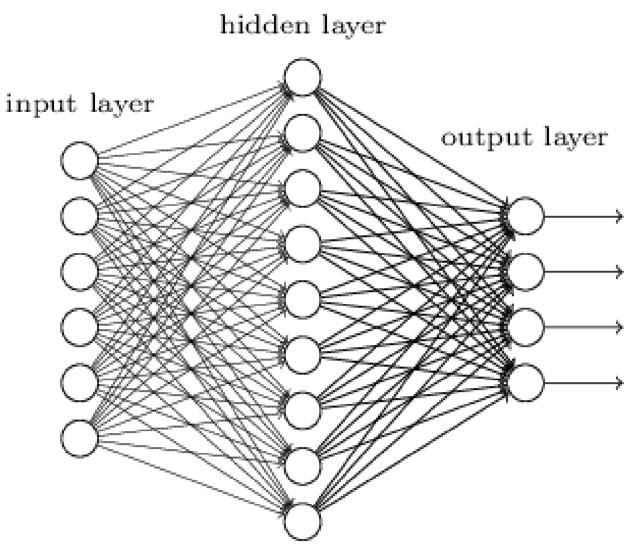
Credit Worthiness

	Employed	Marital Status	Taxable Income	Accept
	Yes, No	*Single, Divorced, Married*	*Integer*	*Yes, No*
1	No			No
2	Yes	Single	> 80K	Yes
3	Yes	Divorced	> 80K	Yes
4	Yes	Single	≤ 80K	No
5	Yes	Divorced	≤ 80K	No
6	Yes	Married	> 100K	Yes
7	Yes	Married	≤ 100K	No

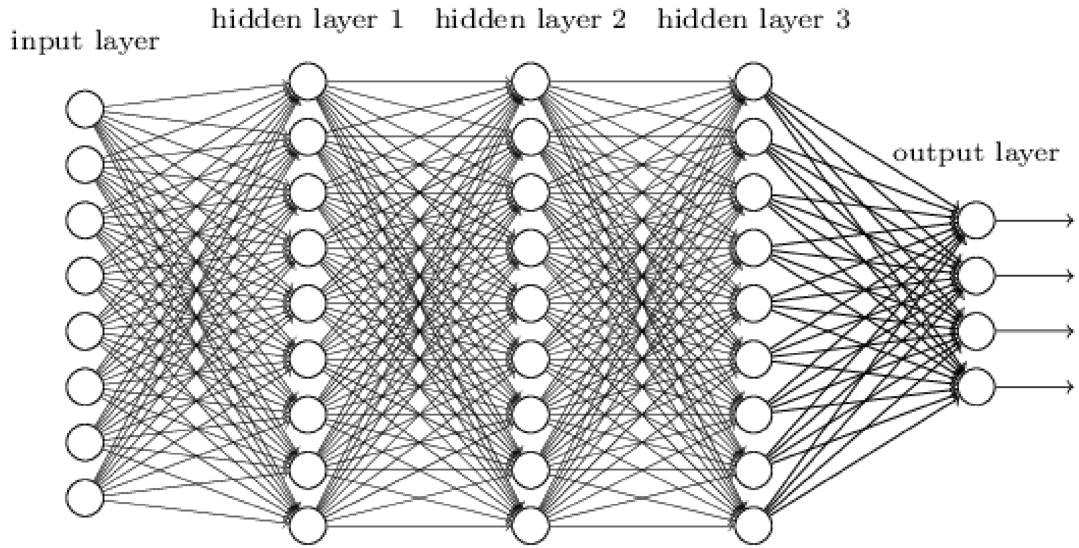
Model: Decision Table

Neural Networks

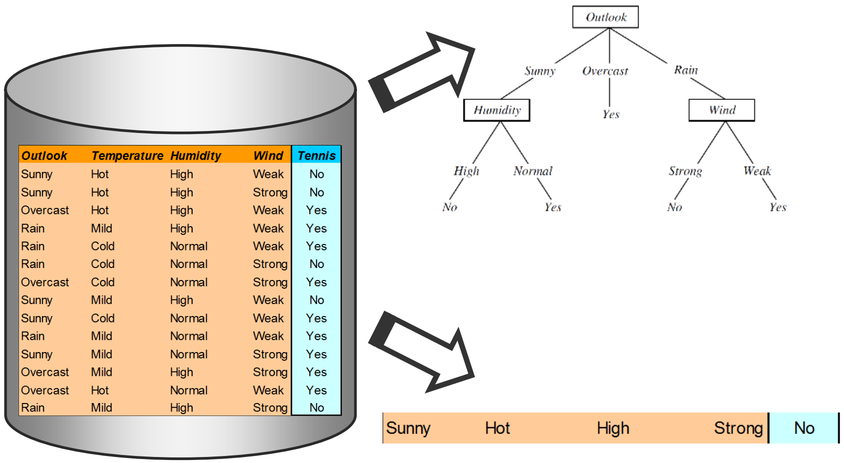
"Non-deep" feedforward neural network



Deep neural network



Two ways of Learning from Experience

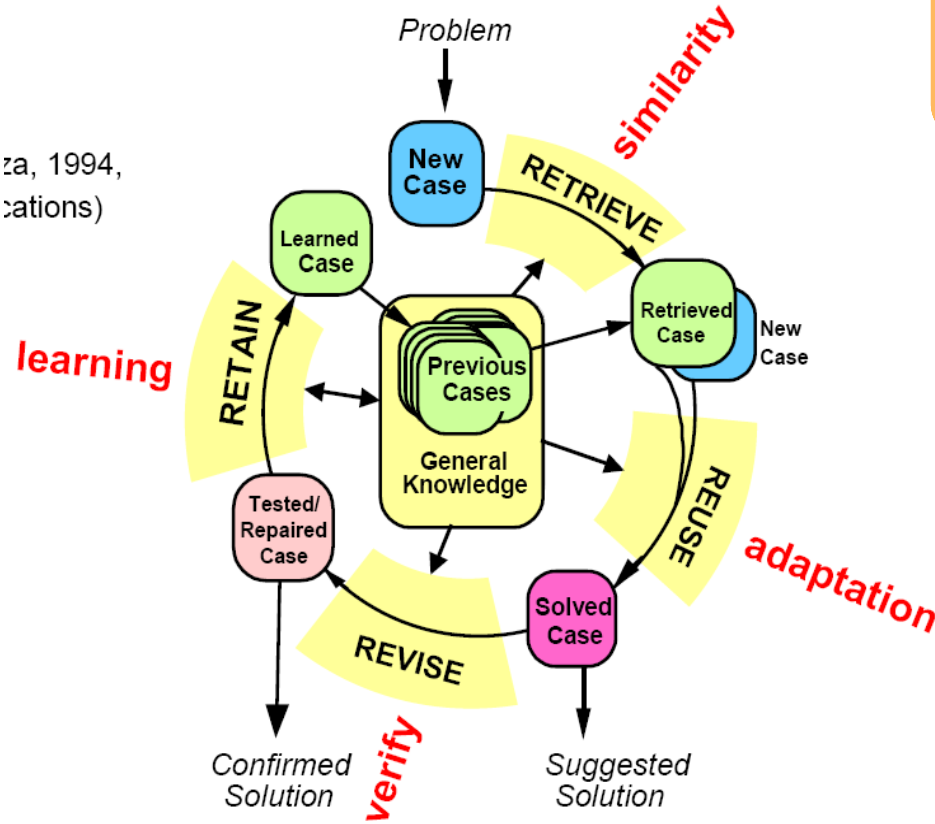


Outlook	Temperature	Humidity	Wind	Tennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cold	Normal	Weak	Yes
Rain	Cold	Normal	Strong	No
Overcast	Cold	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cold	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Sunny	Hot	High	Strong	No

- There are two ways of learning from data
  - **Machine Learning:**
    - Learn a set of rules from data
    - Apply this model for any new case
  - **Case-Based Reasoning (CBR):**
    - For a new situation find the most similar data set and take the conclusion
    - If no appropriate data set is found, solve the new case ad hoc and store it

CBR Cycle



Retrieve …

most similar case or cases

Reuse …

the information and knowledge in that case to solve the problem

Revise …

the proposed solution if necessary

Retain …

the parts of this experience likely to be useful for future problem solving

Source: K.-D. Althoff & A. Aamodt: Relating case-based problem solving and learning methods to task and domain characteristics. AI Communications 1996

Similarity Calculation for Attribute-Value Pairs

Cases resp. meta-data are represented by *n* attributes A_1 ,...,A_n

- each attribute A_i has type T_i

case A:
- A_1: x_1
- A_2: x_2
- …
- A_n: x_n

solution/information:
- ...

sim_{A1}, sim_{A2}, sim_{An}

query Q:
- A_1: y_1
- A_2: y_2
- …
- A_n: y_n

Local similarity: for each attribute a similarity function is defined

- $sim_{Ai}(x_i, y_i): T_i \times T_i \rightarrow [0..1]$
- local similarity measures depend on the type of the attribute

Global similarity: combining values for local similarity

- $sim(A,Q) = F(sim_{A1}(x_1,y_1), sim_{A2}(x_1,y_1), \ldots, sim_{An}(x_n,y_n))$
- $F:[0..1]n \rightarrow [0..1]$ is called an **aggregation function**

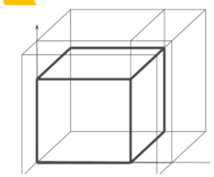
Business Intelligence

BI overview

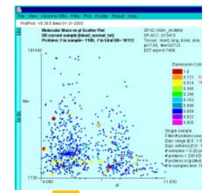
	strategic	*operative*
Questions	• What are our goals? • Are we reaching our goals? • If not, where is the problem?	• Which credit applications should be accepted? • Who are potential csutomers for the new product?

Analyses

measure, aggregate, visualise

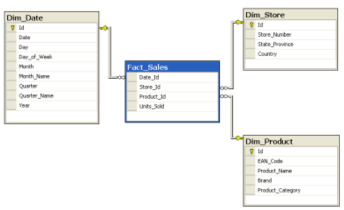


Ad hoc queries, OLAP



find patterns (data mining)

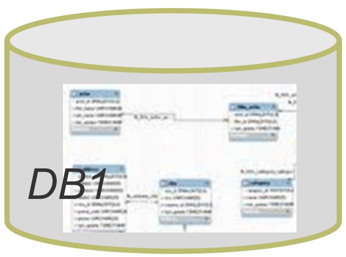
dimensional modelling

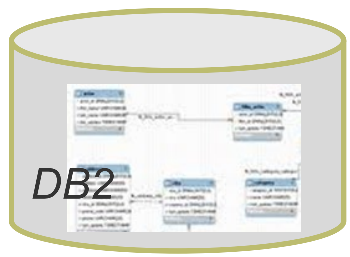


Data Warehouse

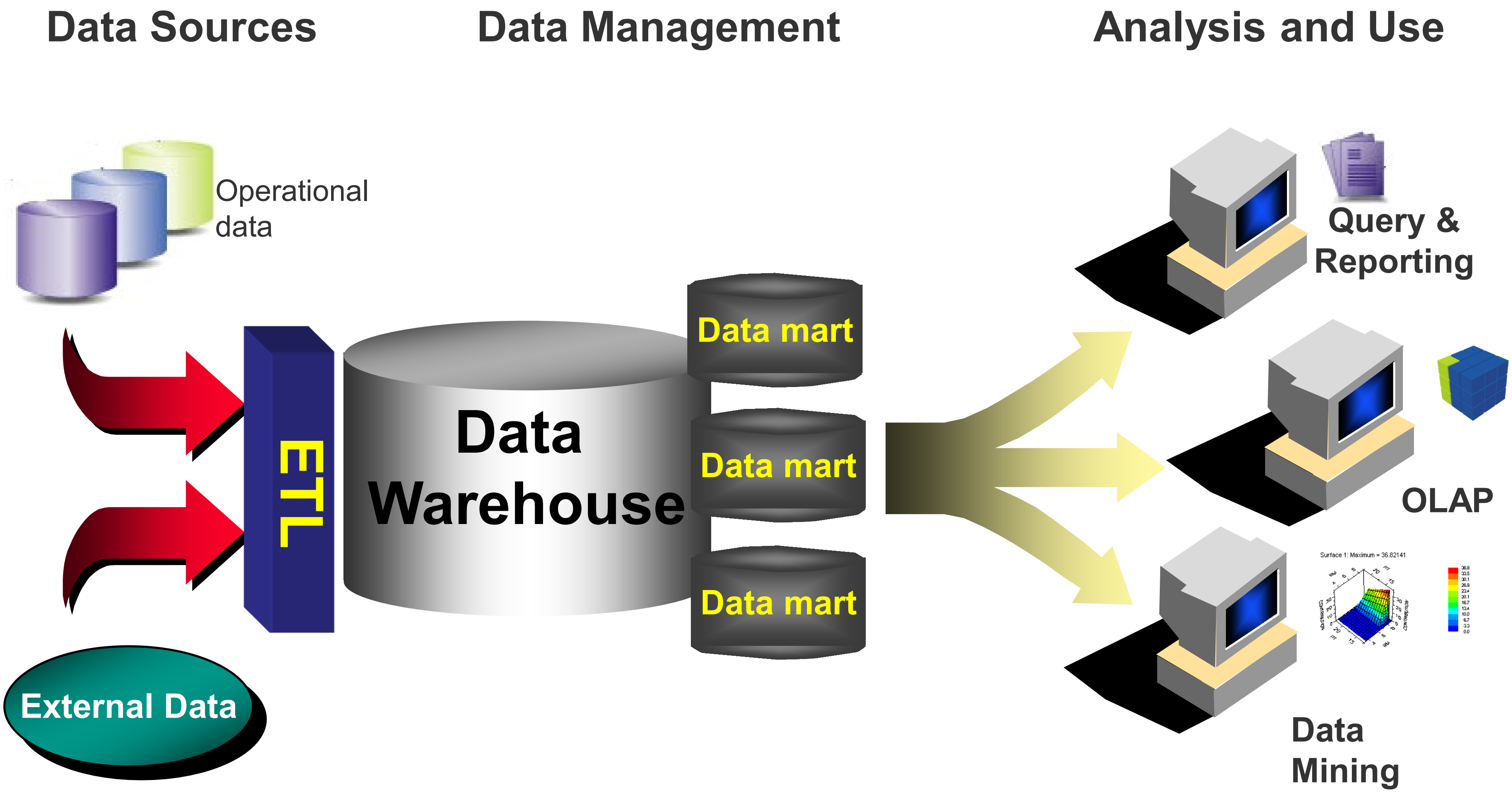
ETL ***ETL*** ***IE***

raw data

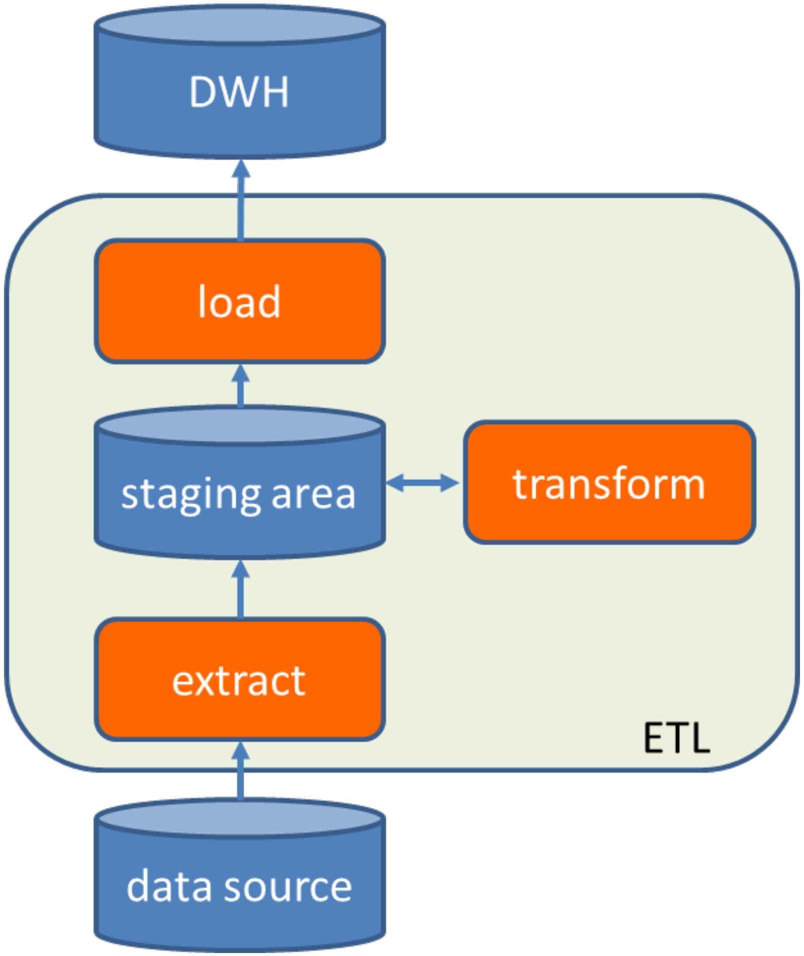




Business Intelligence



ETL process



- The process of
  - **extracting** relevant data from source systems
  - **transforming** the data into the target format defined for the DWH or data mart
  - **loading** the data into the DWH

Business Performance Management

Strategy/Goals:
Where do we want to go?

Objectives:
How can we achieve the strategy?

Measurement:
Did we reach our objectives - KPIs?

Finance

Customers

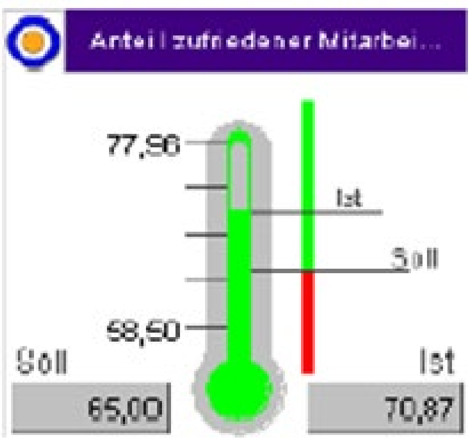
Processes

Innovation

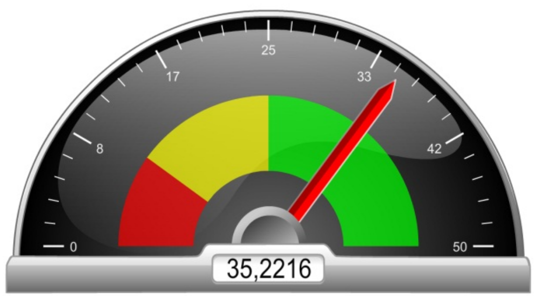
Quelle: (Niven 2003)

KPI Visualisation

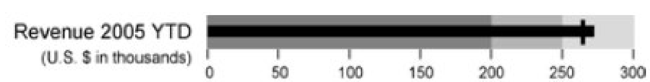
- (usually) needs to highlight
  - the target value
  - the actual value
  - the ranges of «red (poor), yellow (satisfactory), green (good)», if defined



thermometer

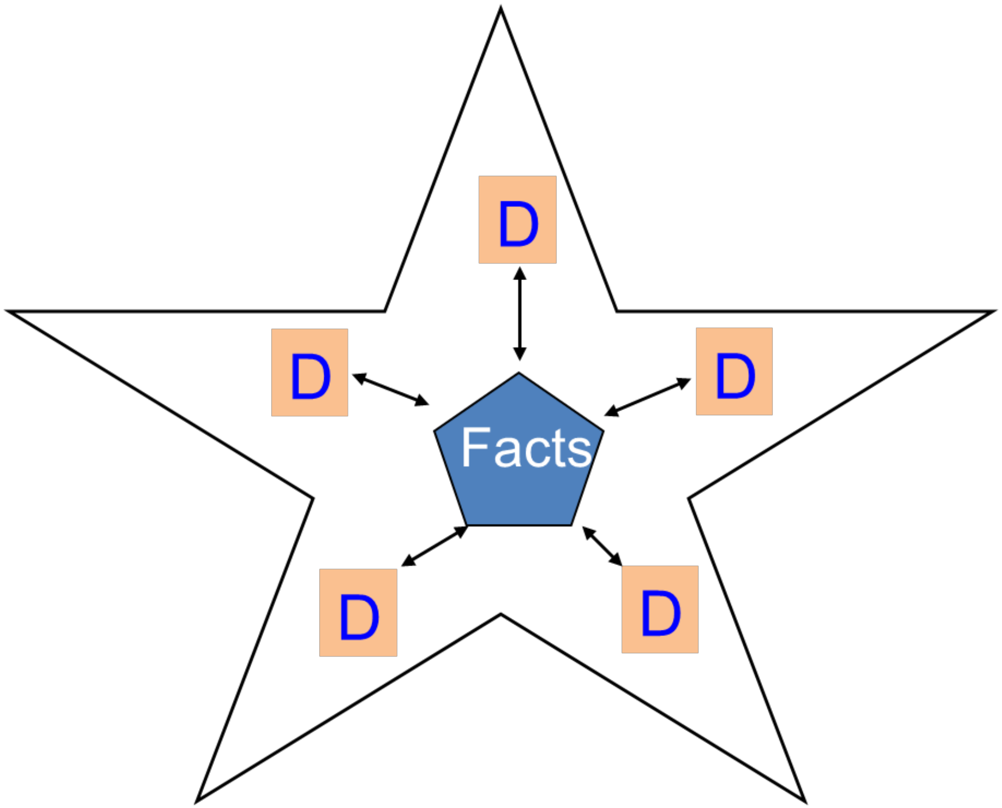


dial chart



bullet graph
(for color-blind!)

Star Schema for Relational Data Warehouses /Marts to support OLAP

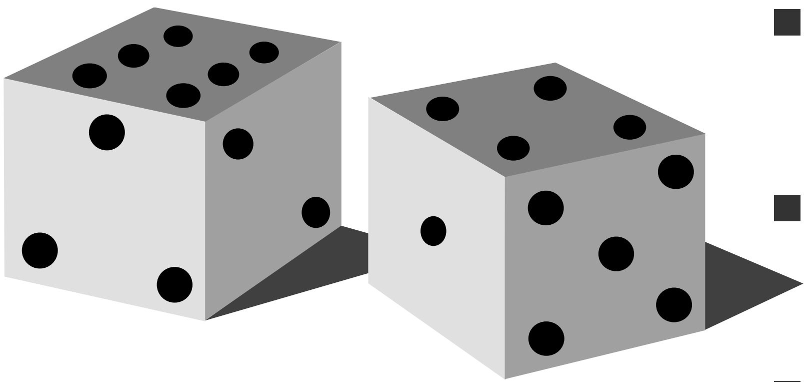


Star Schema:

logical database schema, which places dimension tables of a relational database aroung a fact table for easy querying

Maping of multidimensional data to two-dimensional tables.

Dicing and Slicing



- An OLAP cube can be regarded as a multidimensional cube
- From a cube only two dimensions are visible on a two-dimensional interfact (e.g. as a table)
- Slicing
  - Contraining one dimension
- Dicing
  - Constraining several dimensions
- Pivoting
  - "turning" the cube to show other dimension
- Roll-up/Drill-down – Split/Merge
  - Aggregate or detailing views

OLAP Operation – Slicing and Dicing