FAKULTÄT FÜR MATHEMATIK UND INFORMATIK
Institut für Informatik
Prof. Dr. Heiner Stuckenschmidt

What‘s Wrong with the Semantic Web

XML-Days

Berlin, 25. September 2006

This talk is NOT about

- General OWL-bashing
- Why databases can solve everything anyways
- Web 2.0, Tagging, Social-semantic software, …
- Semantic Desktop, semantic Grid, semantic web services, semantic …..

This Talk IS about

- The Vision of the Semantic Web
- Semantic Web Technologies in 10 Minutes
- The typical Semantic Web Application in 2006
- Problems: Where the assumptions break
- A (personal) perspective on solutions
- The typical semantic web application in 2010 (hopefully)

The Original Vision

- Berners-Lee, Hendler, Lassila: The Semantic Web, Scientific American, May 17th 2001
- Cited (and abused) extensively in literature and science marketing

Key Ideas

- Smart Devices
- Personal Information Agents
- Knowledge about objects, time and space
- Trusted Information

Observations

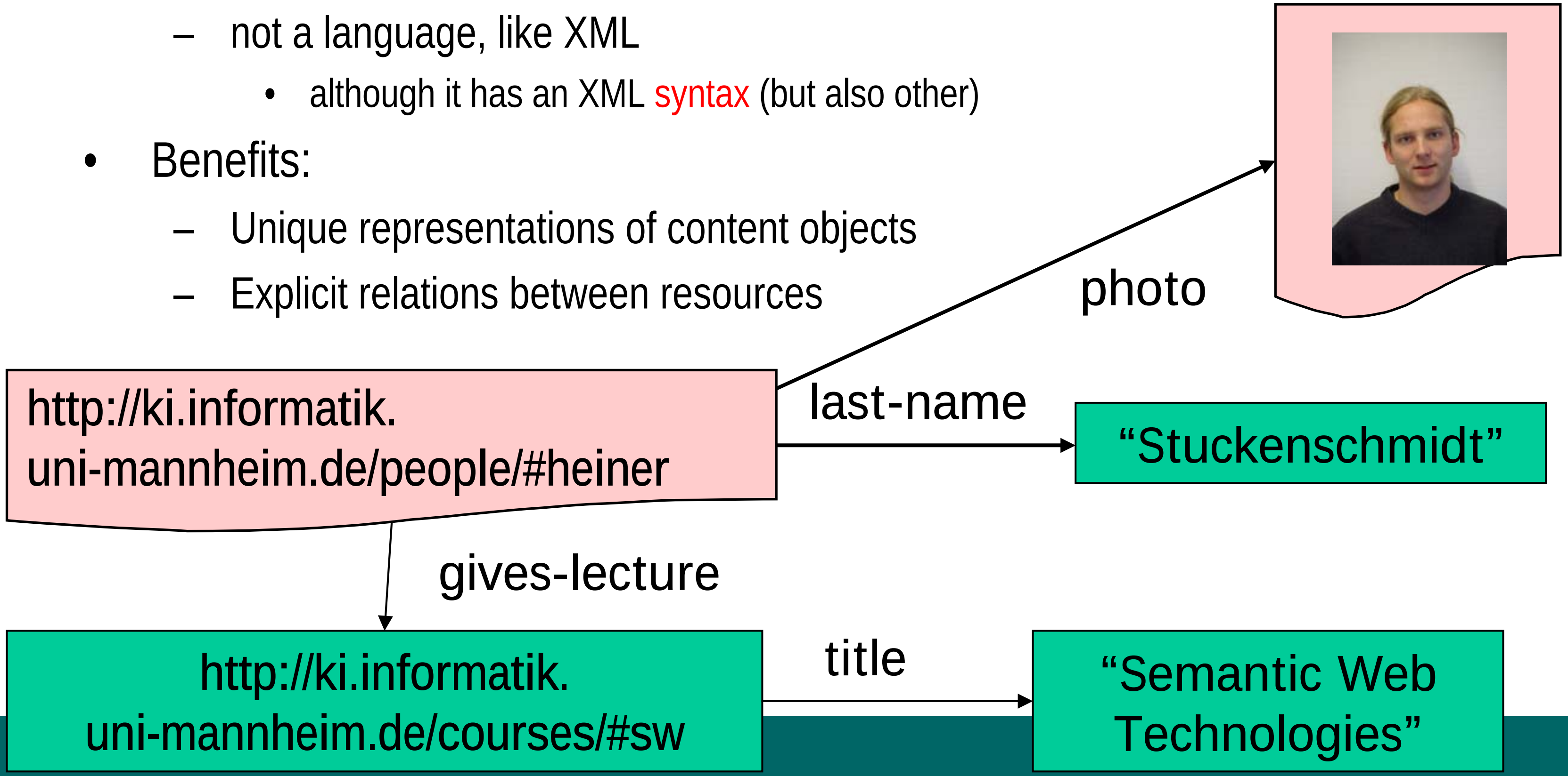
- Part of the traditional Web
- Paradoxes, incompleteness and other logical fallacies
- Ontology Integration and Ontology Mappings
- Evolution of Knowledge and Meaning

Key Technologies

- Machine-Readable Metadata
  - Based on XML and RDF
- Logic, Inference Rules and Proofs
- Ontologies
- Agent Technologies
  (nowadays read: „web services“)

RDF: Resource Description Framework

- RDF is a data model
  - used to describe meta-data of a piece of data
  - not a language, like XML
    - although it has an XML syntax (but also other)
- Benefits:
  - Unique representations of content objects
  - Explicit relations between resources



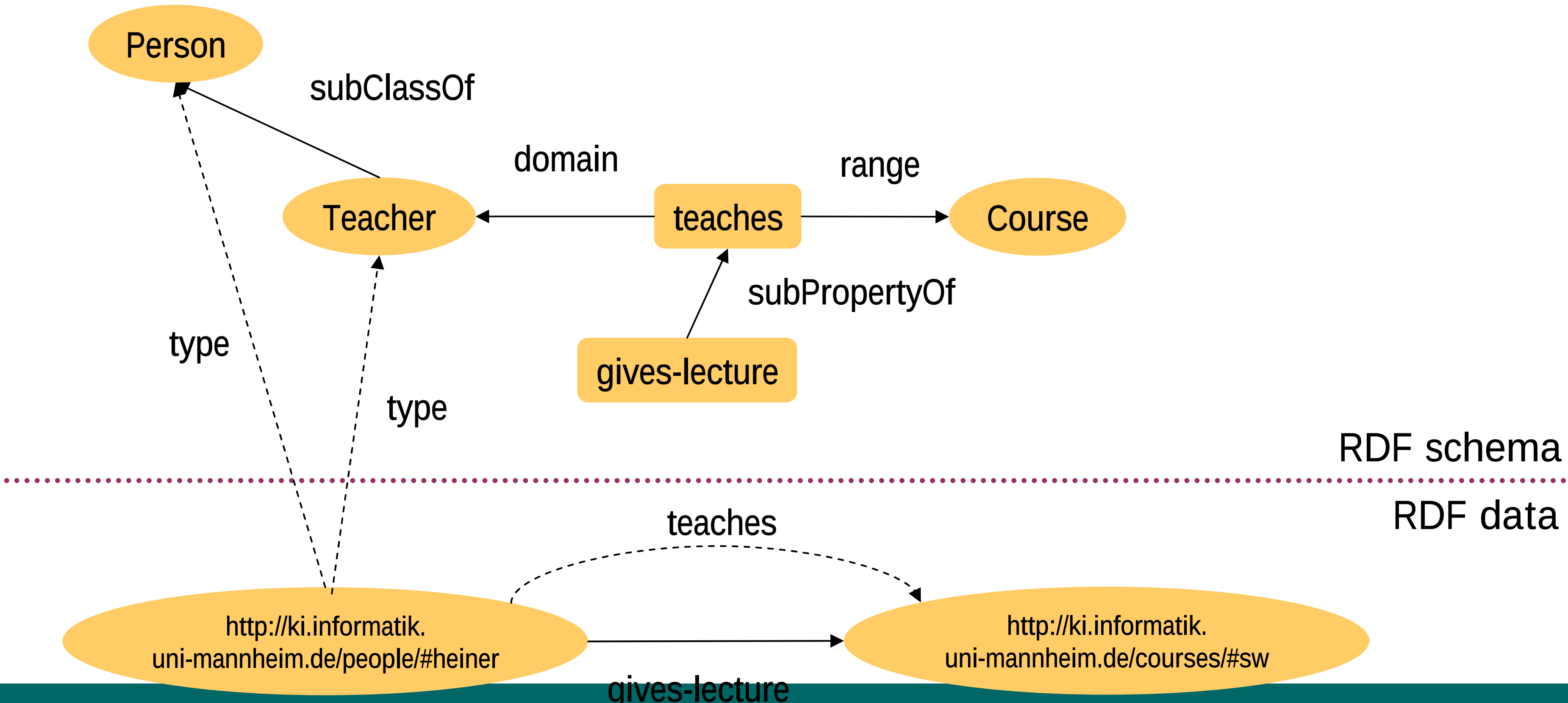
UNIVERSITÄT MANNHEIM

RDF Schema

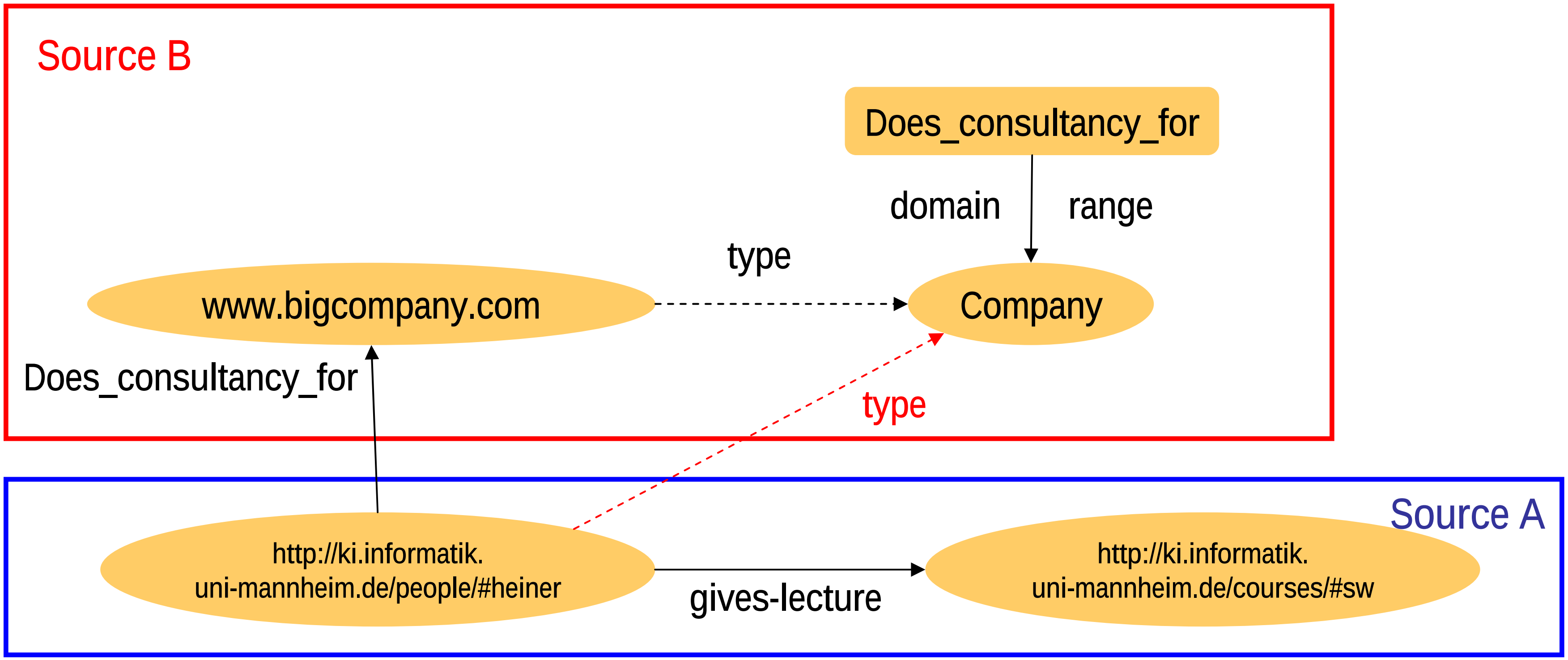
(X R Y), (R subPropertyOf Q) → (X Q Y)

(X R Y), (R domain C) → (X type C)

(X type C), (C subClassOf D) → (X type D)

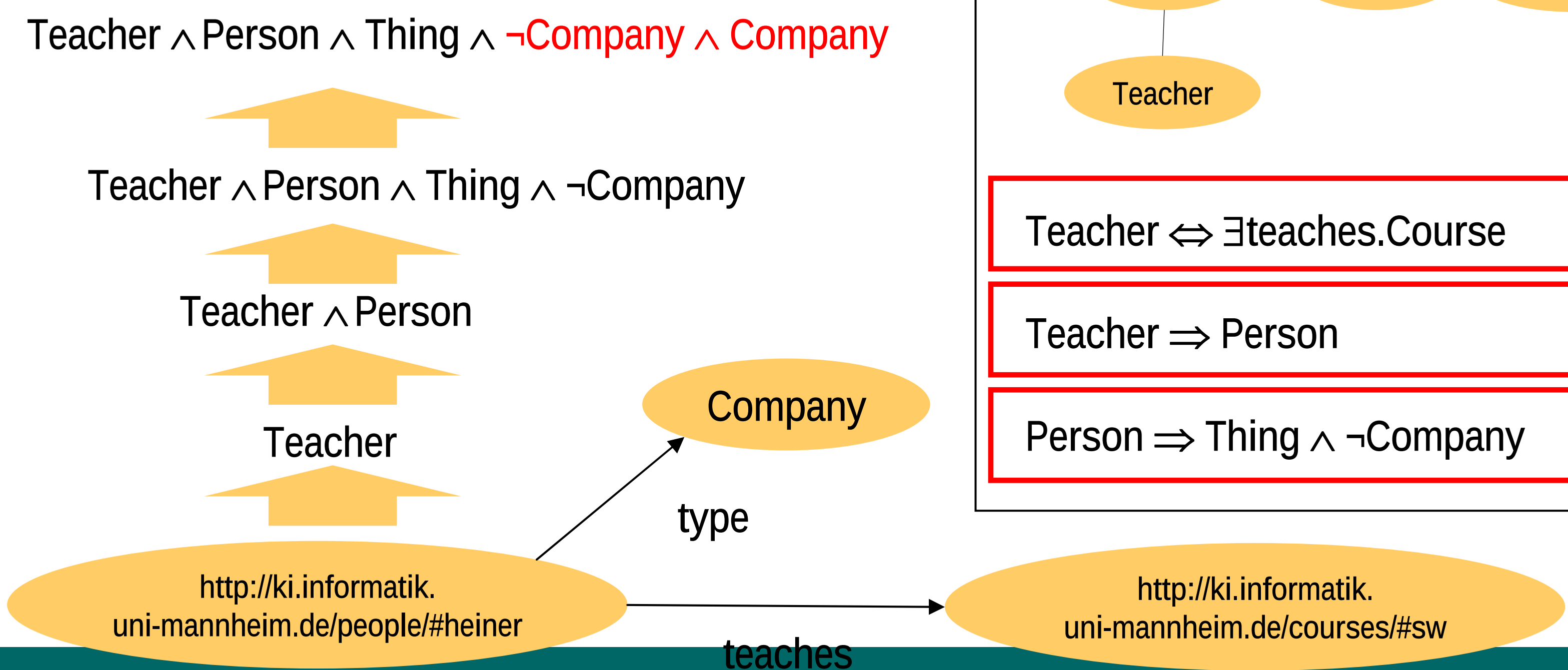


Problem: no semantic guarantees

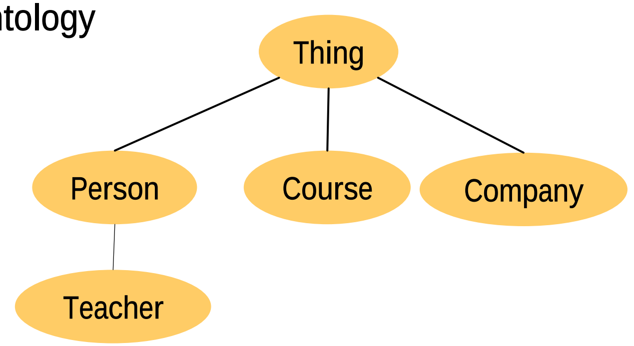


Logical Reasoning about Resources

- Logical Axioms limit allowed interpretations:



Ontology



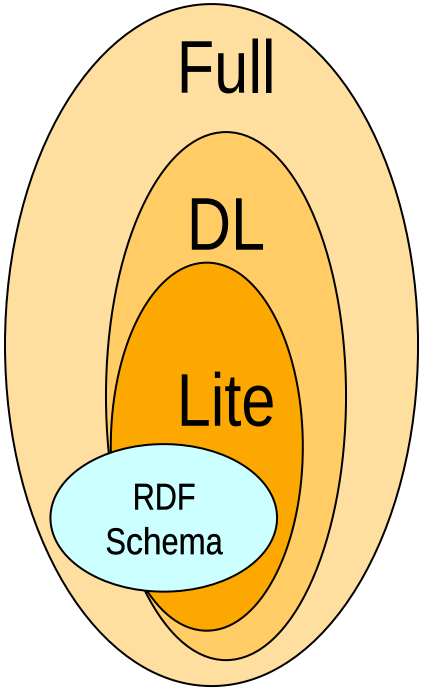
Teacher ⇔ ∃teaches.Course

Teacher ⇒ Person

Person ⇒ Thing ∧ ¬Company

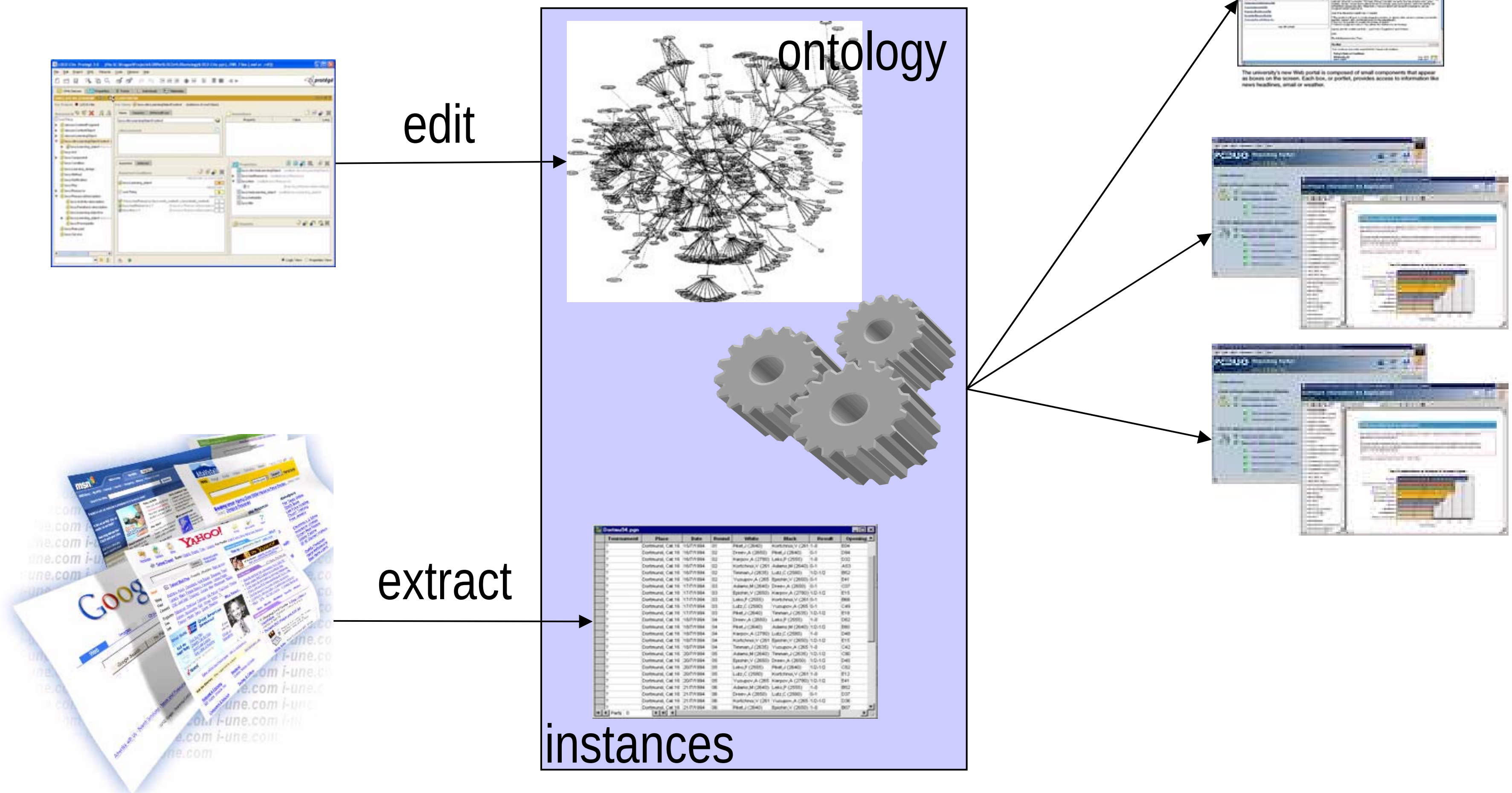
The Web Ontology Language OWL

- OWL Lite:
  - Classification hierarchy
  - Simple constraints
- OWL DL:
  - Maximal expressiveness
  - While maintaining tractability
  - Standard formalisation
- OWL Full:
  - Very high expressiveness
  - Loosing tractability
  - Non-standard formalisation
  - All syntactic freedom of RDF (self-modifying)



Syntactic layering
Semantic layering

Where are we now

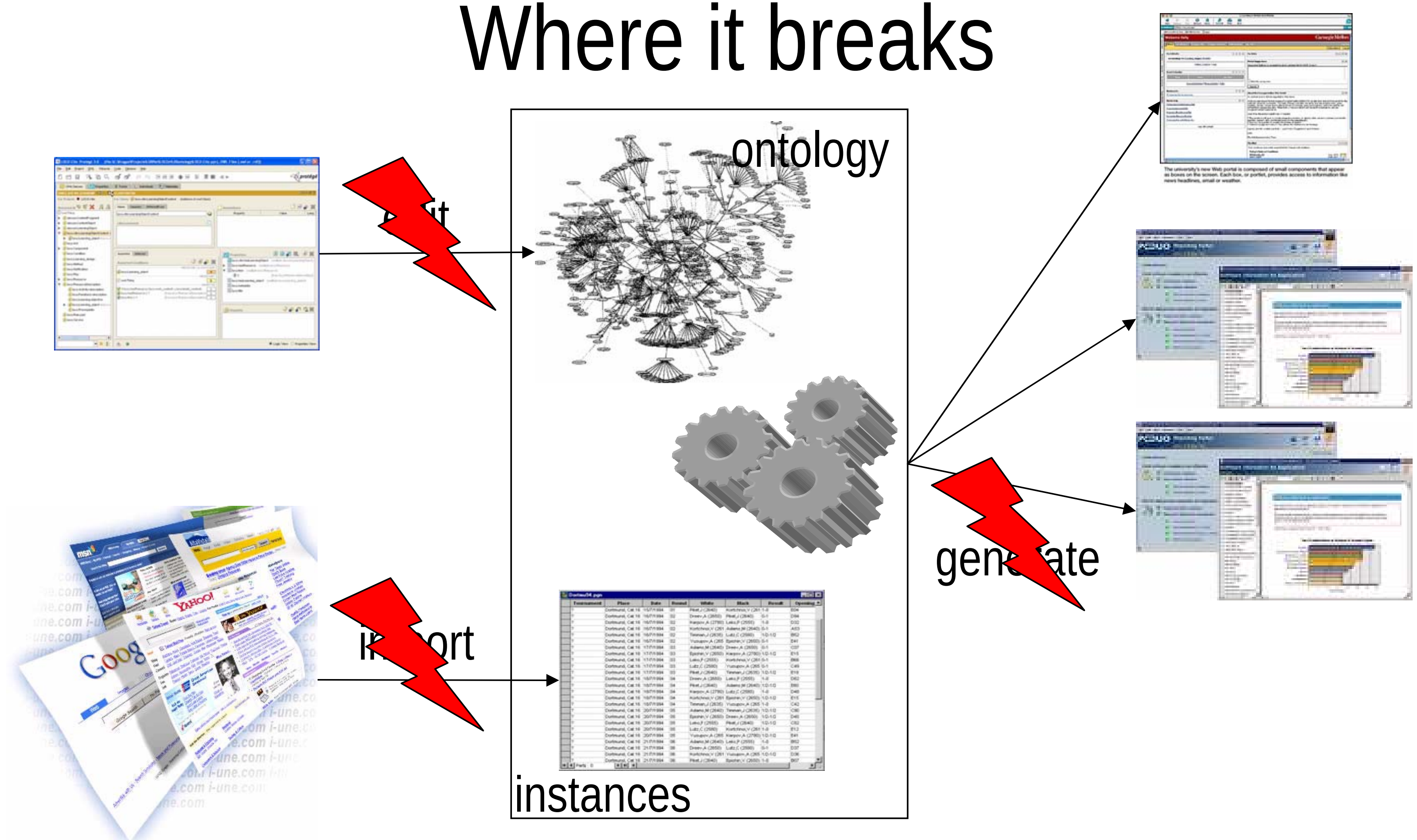


Living in the Real Web

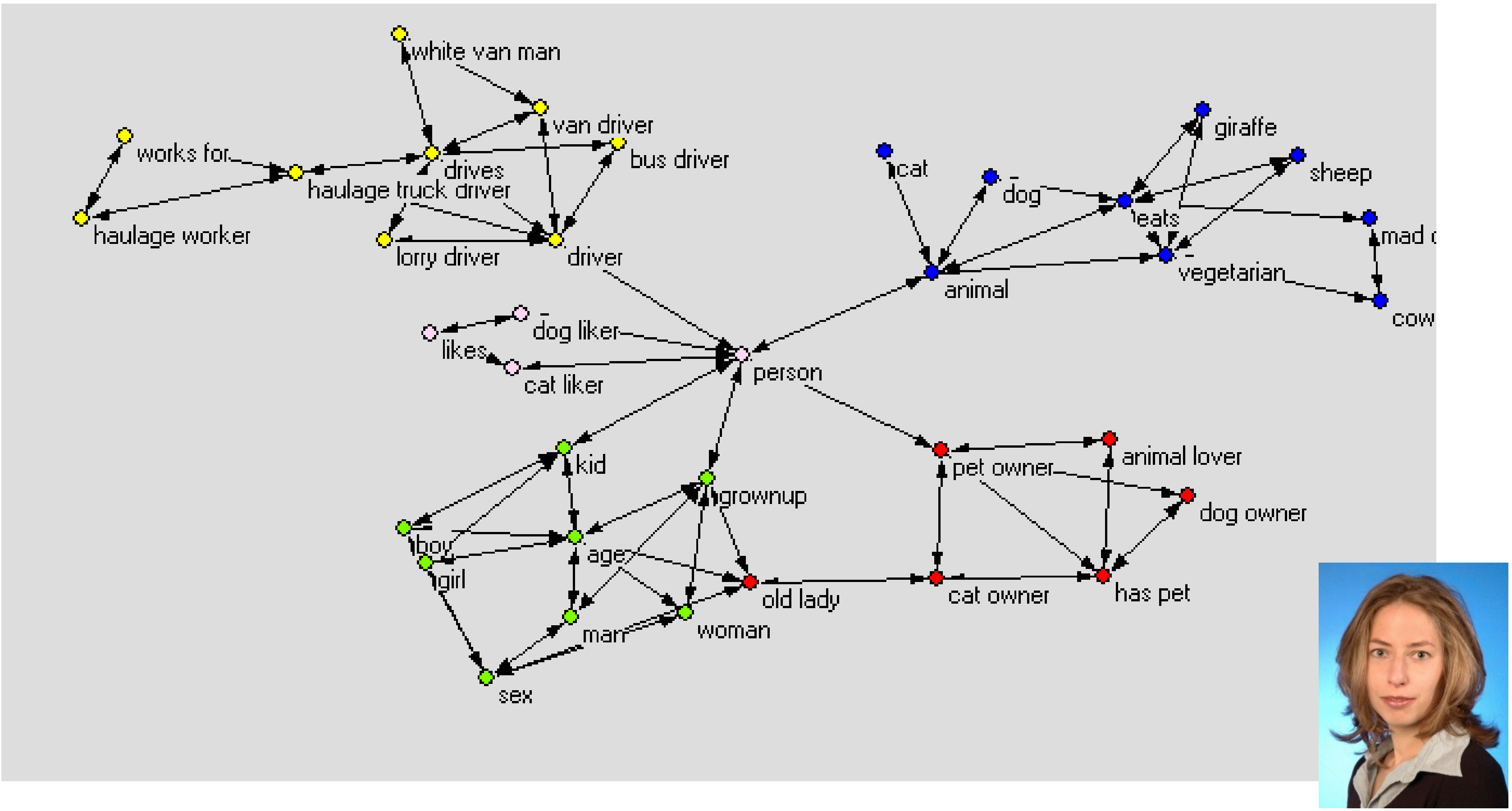
1. The Technical Level
   – Distributed Information
   – P2P-like Architecture
   – Possible Failures

2. The Content Level
   – Inconsistency and Incompleteness
   – Heterogeneity and Ontology Alignment
   – Multimedia Information Extraction

Where it breaks



Modular Ontologies

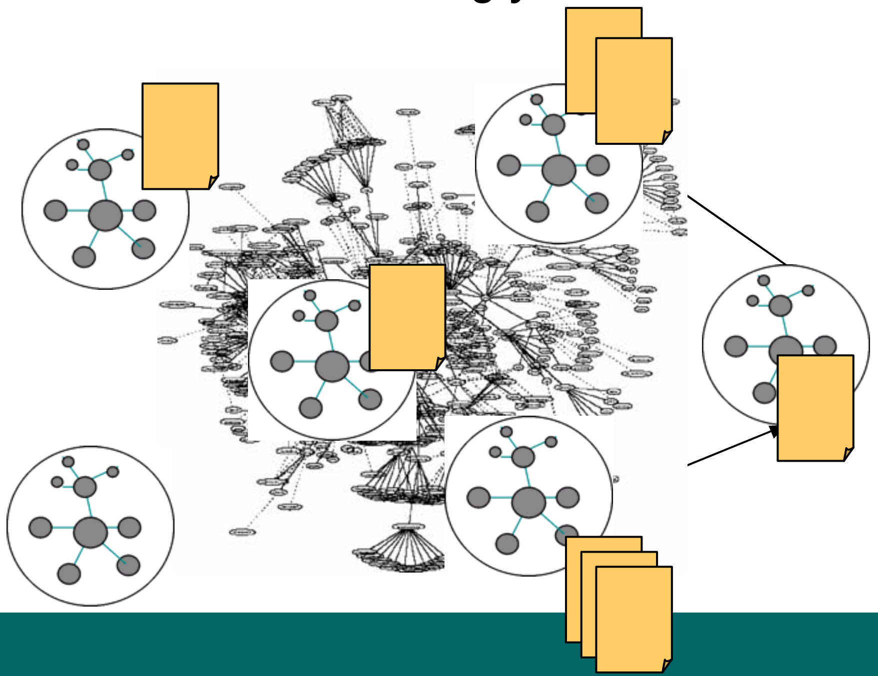


The Case for Modularization

- Distributed Development, Maintenance and use
  - Experts can update their portion independently of other parts
- Selective Publication and Use of Terminologies
  - Stable subsets can be published in the development phase
  - Users can chose relevant subset of an ontology
- Manual Inspection and Validation
  - Small, coherent modules are easier to understand
- Editing, Visualization and Reasoning
  - Available tools do not scale to very large ontologies

Partitioning Ontologies

- Scenario 1: Structured System
- Idea: create a network where each node is in charge of one partition and distribute data and metadata accordingly



Advantages:

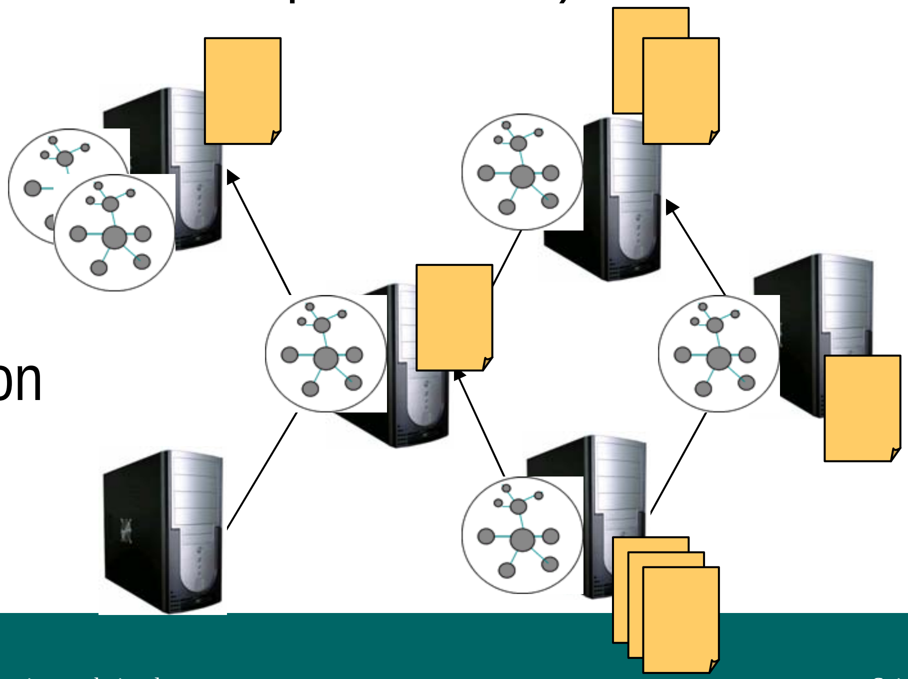
- Reduction of bottlenecks
- No single point of failure
- Possibiliy for semanic routing based on ontology structure

Partitioning Ontologies

- Scenario 2: Unstructured System
- Idea: Distribute Partitions in an optimal way in an existing network (ontology should be close to the point of use)

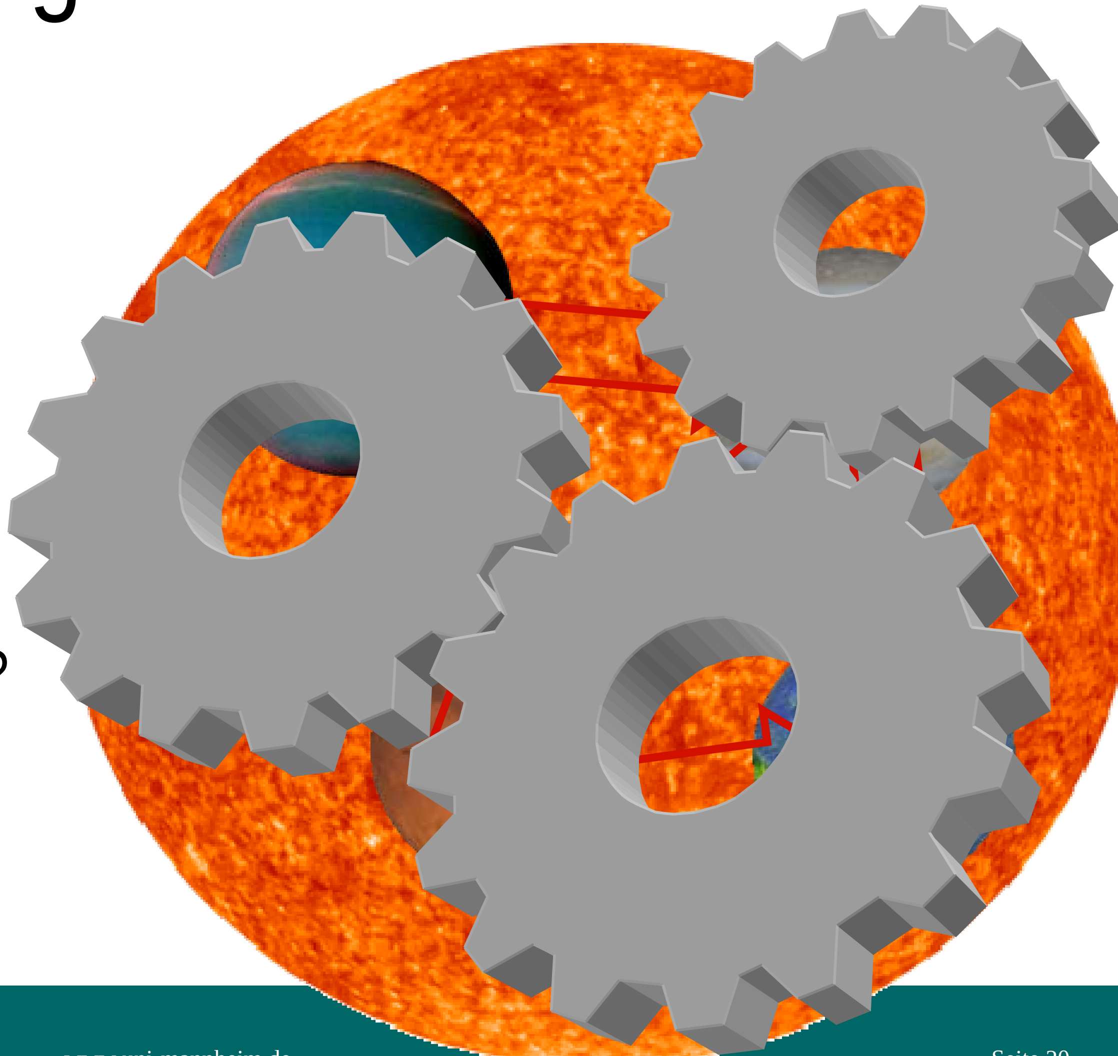
Advantages:

- local processing
- Use-based partitioning
- Improvement of communication based on concept similarity



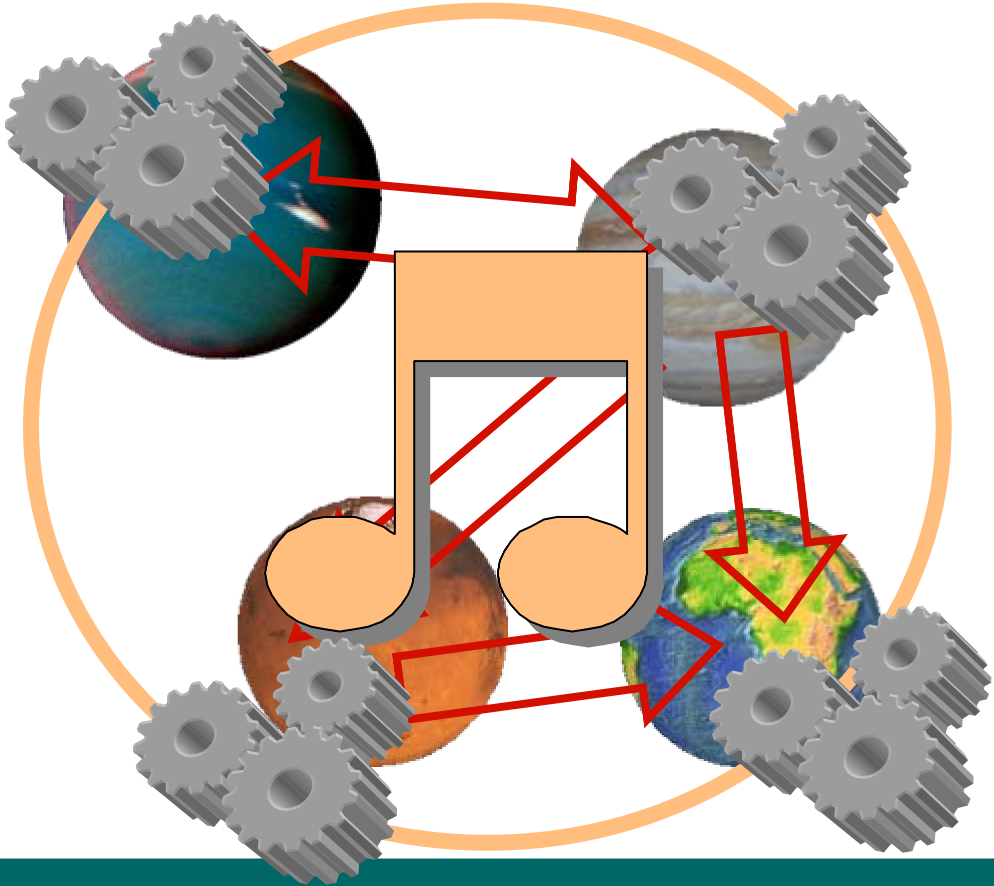
So your ontologies are distributed…

- current reasoners:
  - Global ontology
  - Reason in global ontology
- Problems:
  - Scalability?
  - Reasoning specificity?
  - Privacy? Autonomy?
  - Robustnes?



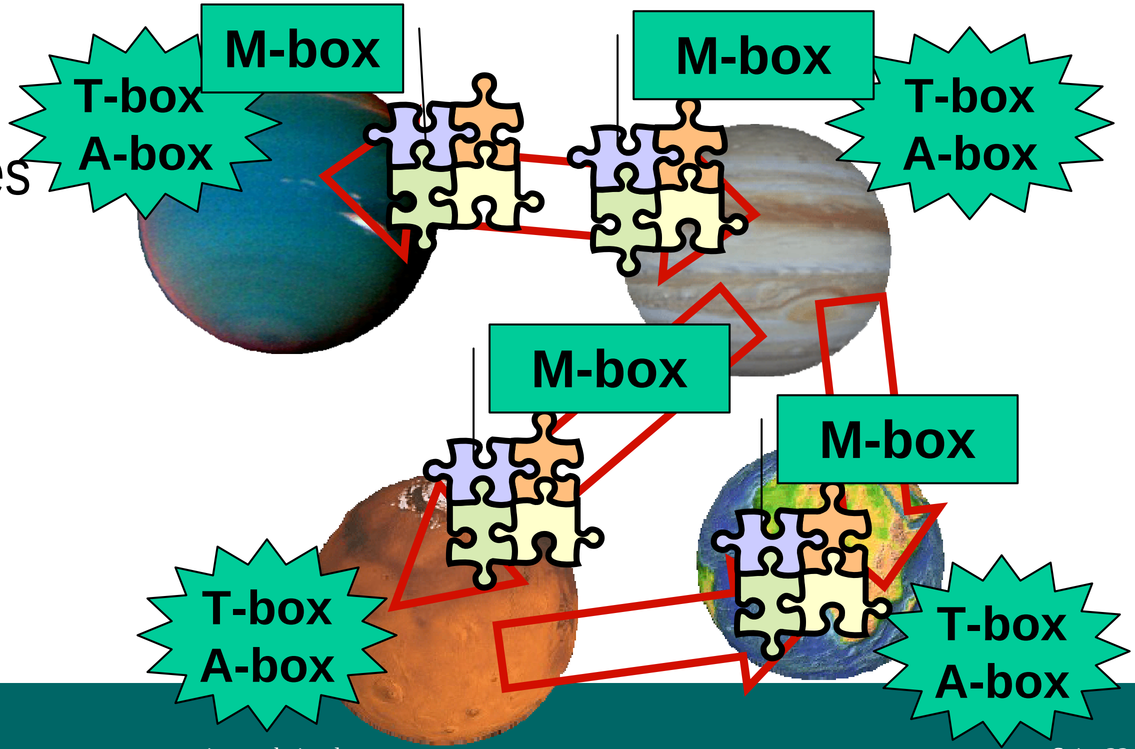
…your Reasoning should be as well

- Alternative approach:
  - Local reasoning
  - Suitable combination
- Requirements:
  - Formal framework
  - Reasoning algorithm
  - The reason-able system implementation



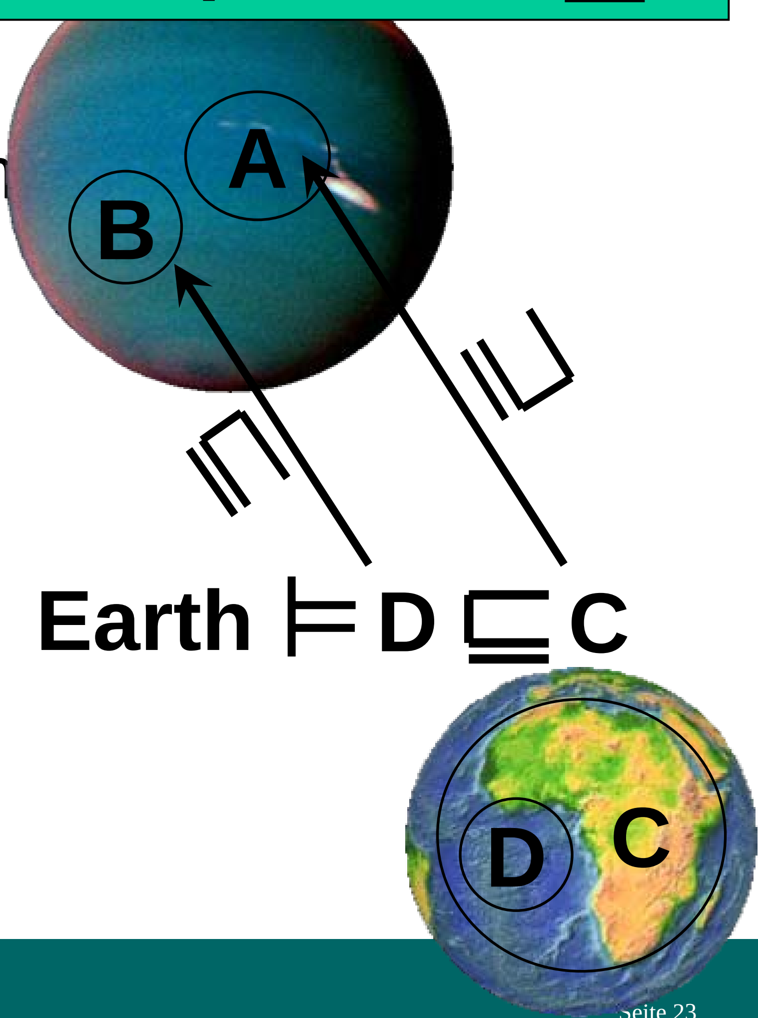
A Formal framework: C-OWL in a nutshell

- Distributed Description Logics (DDL)
  - Captures the case of multiple ontologies pairwise linked by semantic mappings
- Ontologies:
  - DL knowledge bases (T-box and A-box)
- Mappings:
  - Bridge rules (M-box)

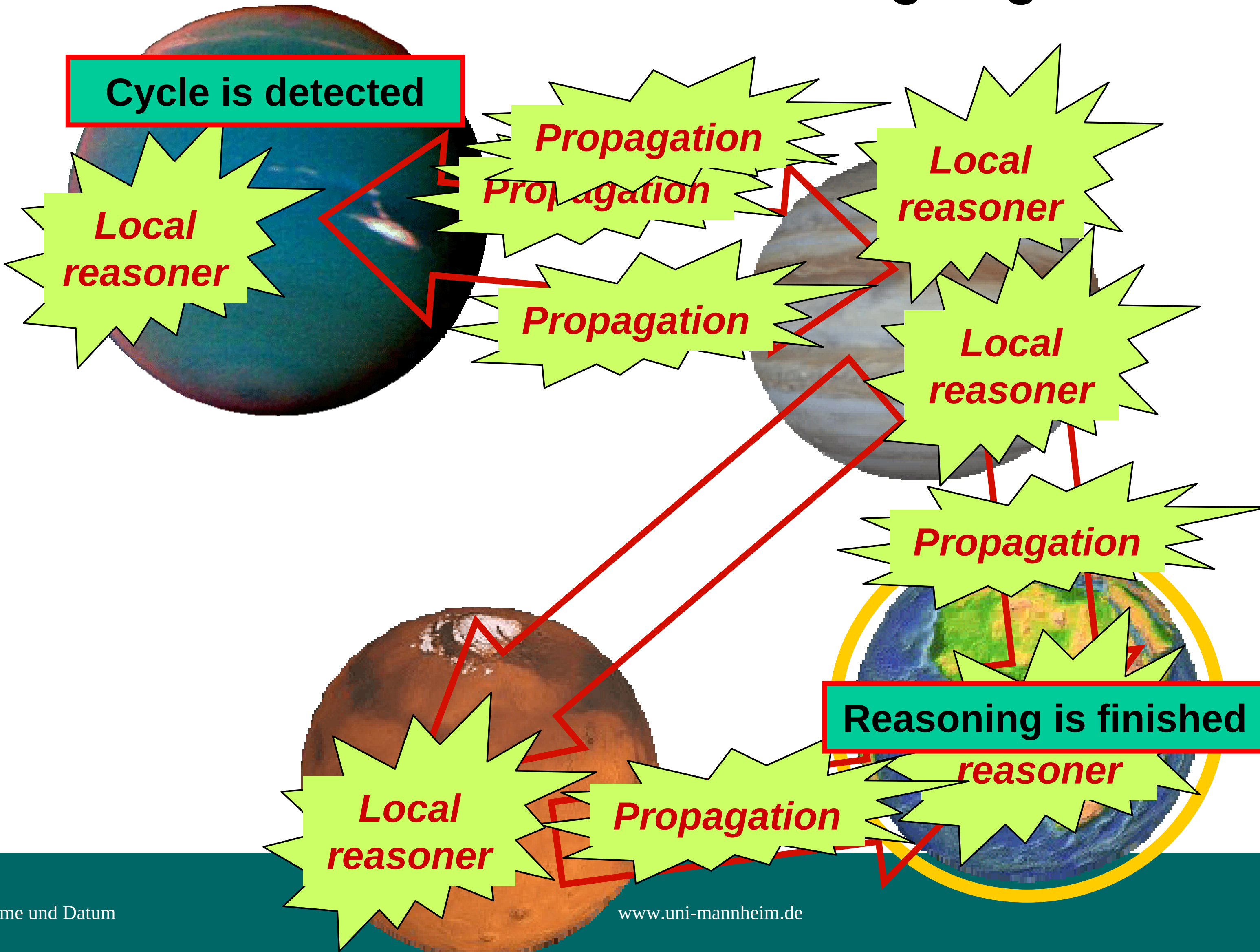


Reasoni

- Similarly to OWL-DL:
  - Core reasoning task in DDL –con
    subsumption
- Difference from OWL-DL:
  - Scope – Galaxy
  - Mappings matter
- Subsumption in DDL –
  a global subsumption

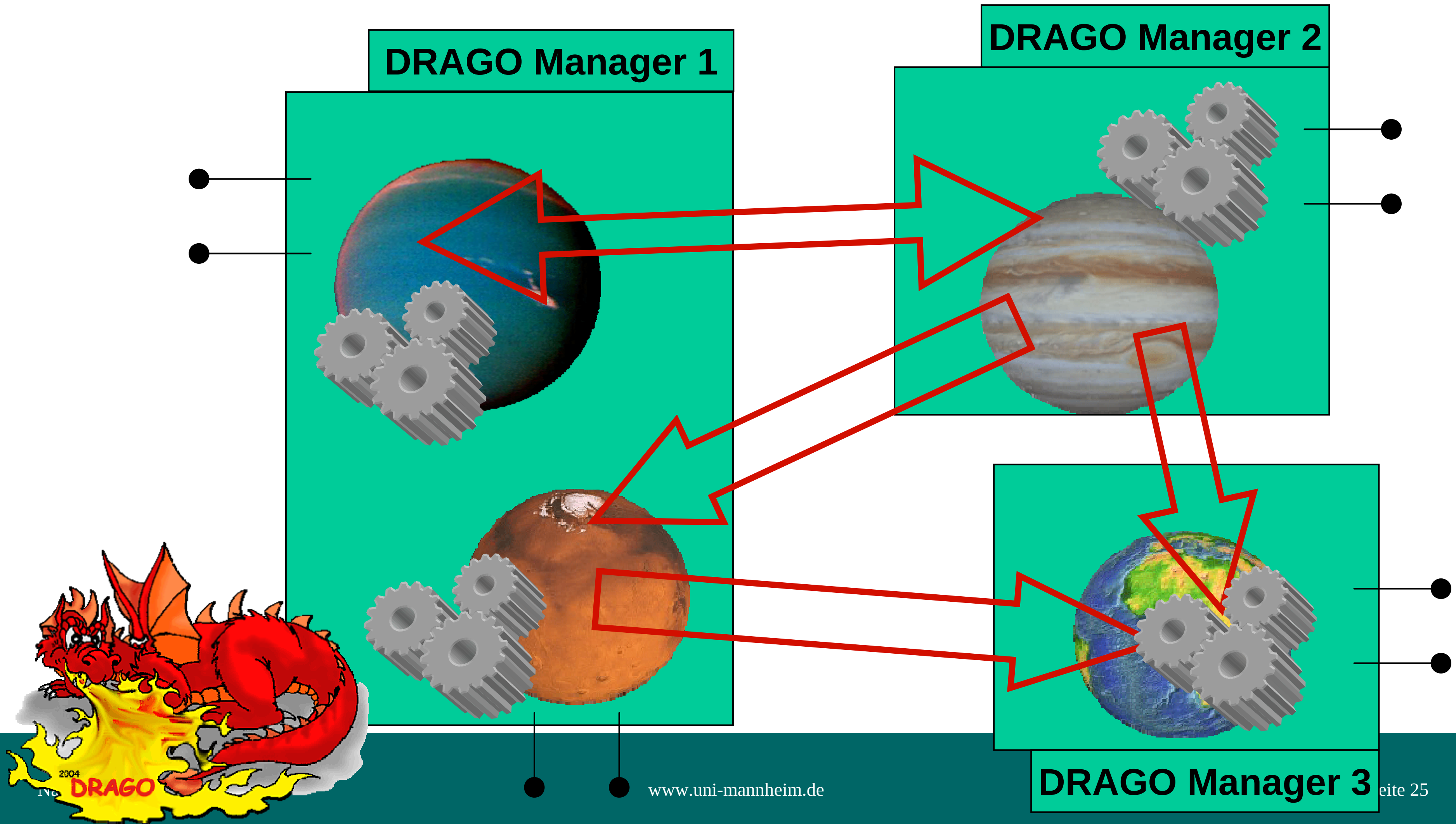


Distributed reasoning algorithm



DRAGO

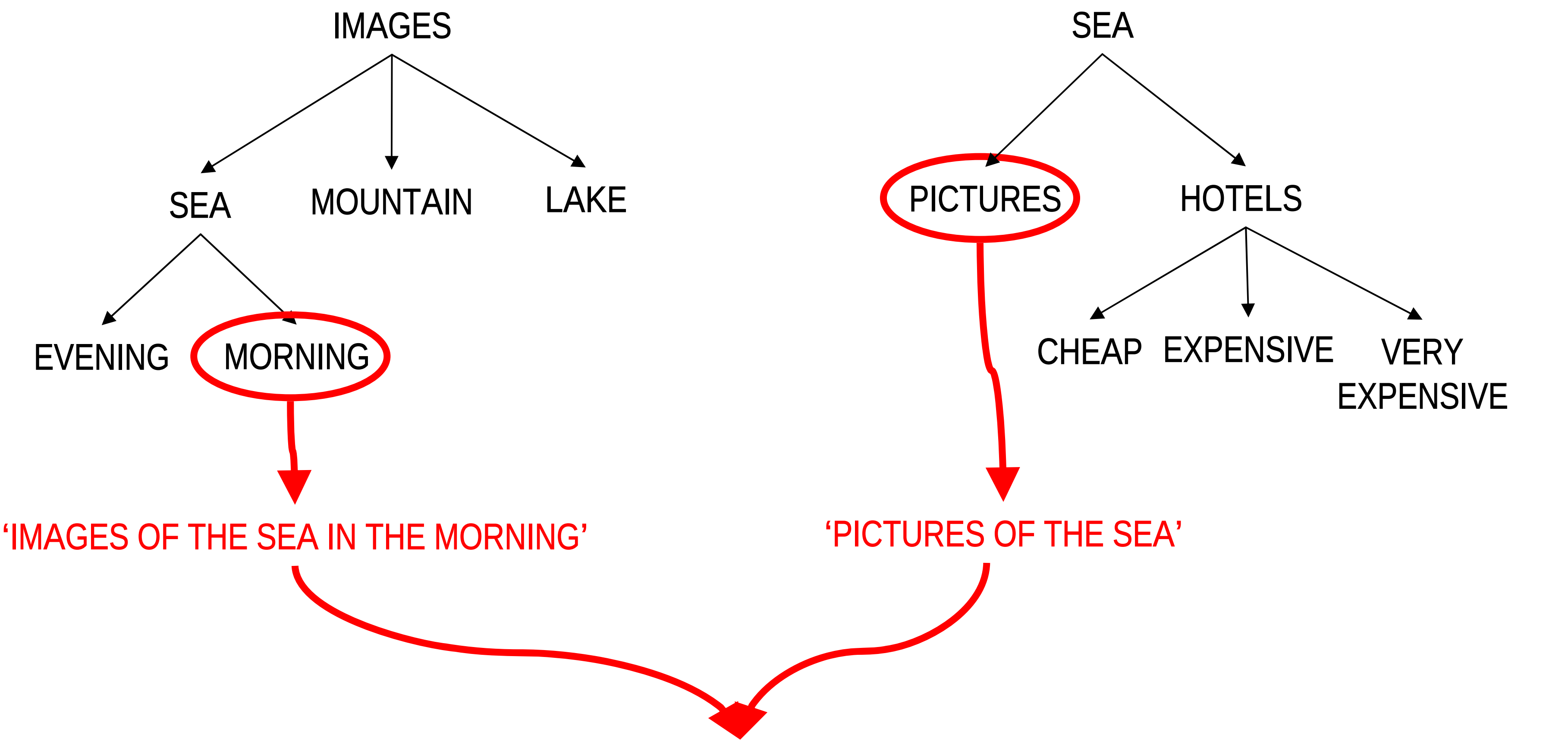
Distributed Reasoning Architecture for Galaxy of Ontologies



Where do the mappings come from ?

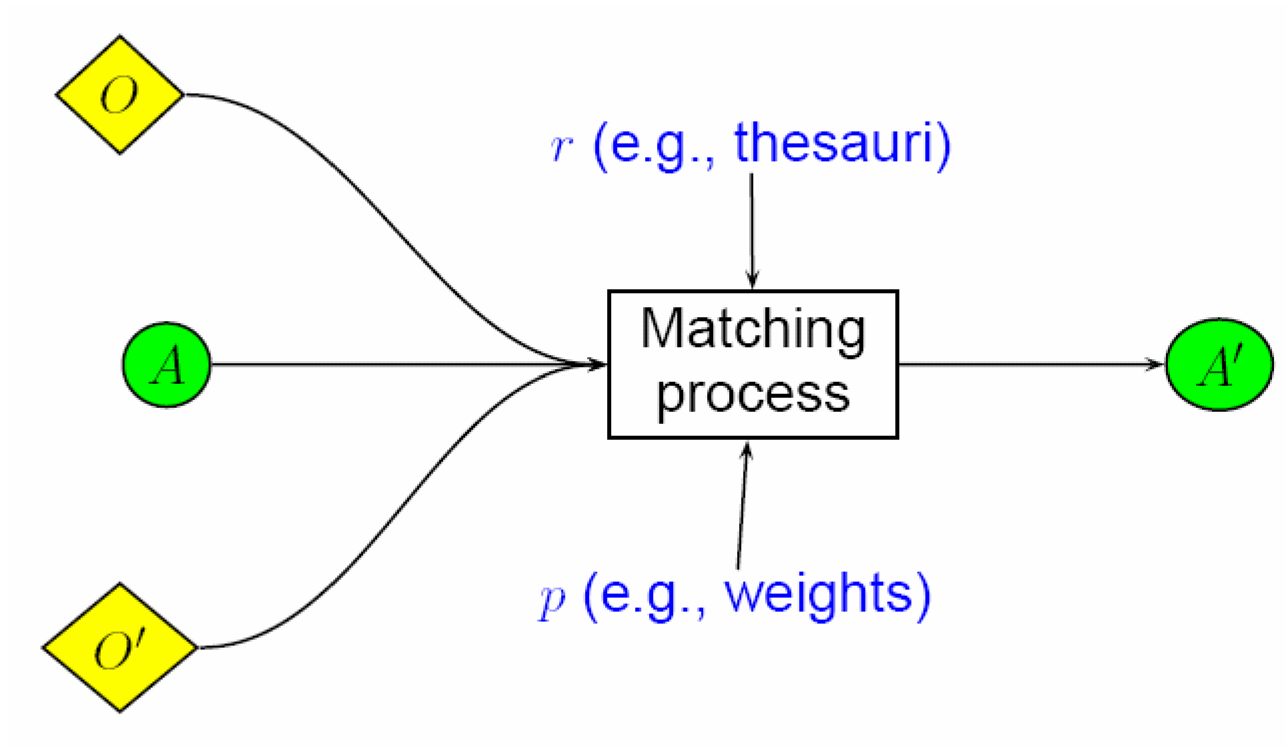
- Decompositional Approach:
  - One existing Ontology is split into different ones
  - Mappings arise naturally from decomposition
- Compositional Approach:
  - Different Ontologies exist
  - Mappings have to be found based on semantic correspondences

Heterogeneous Classifications



'IMAGES OF THE SEA IN THE MORNING' $\subseteq$ 'PICTURES OF THE SEA'

Ontology Matching



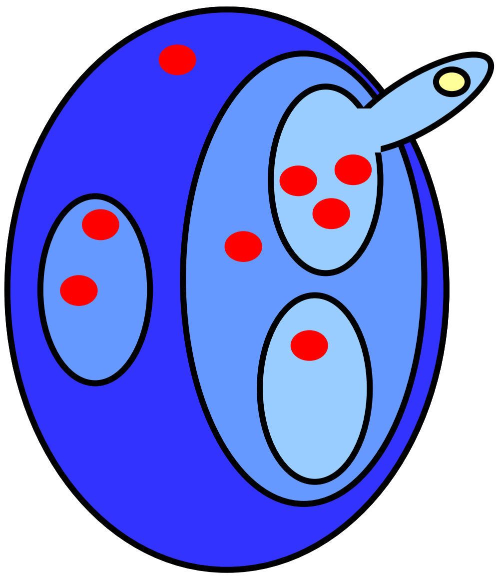
Result: Mappings

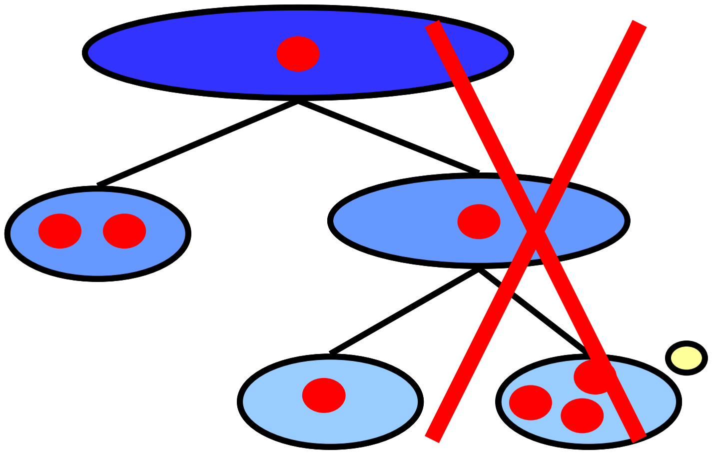
- Mapping elements are 5-Tuples: (id,e,e‘,R,n)
  - *id* is a unique identifier for a given mapping element
  - e and e‘ are entities in the mapped ontologies
  - *R* is a relation that holds between the elements
  - *n* is a confidence measure for the mapping

- Two possible readings the measure
  - Degree to which the entities relate , but also
  - Confidence that the result of the matching is correct

Problem: Partial Matching

- Sloppy terminologies need robust inference

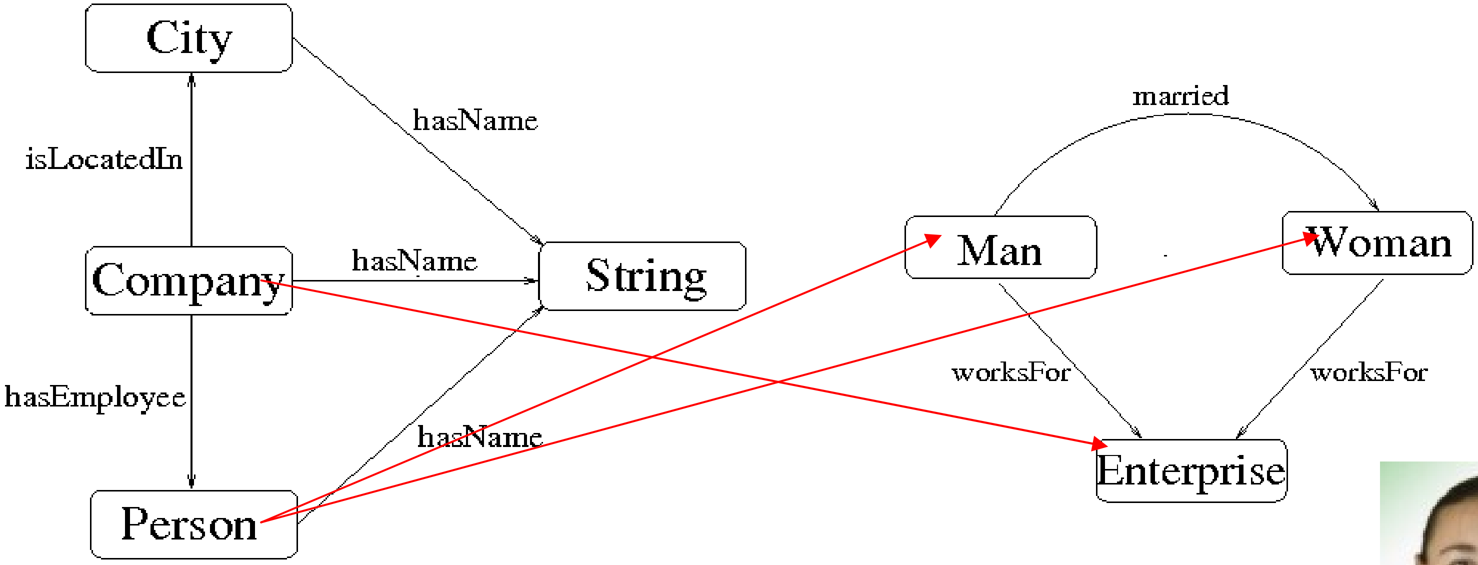


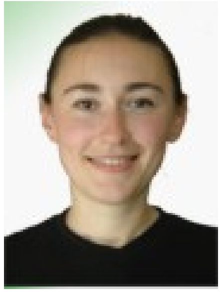


almost subClassOf

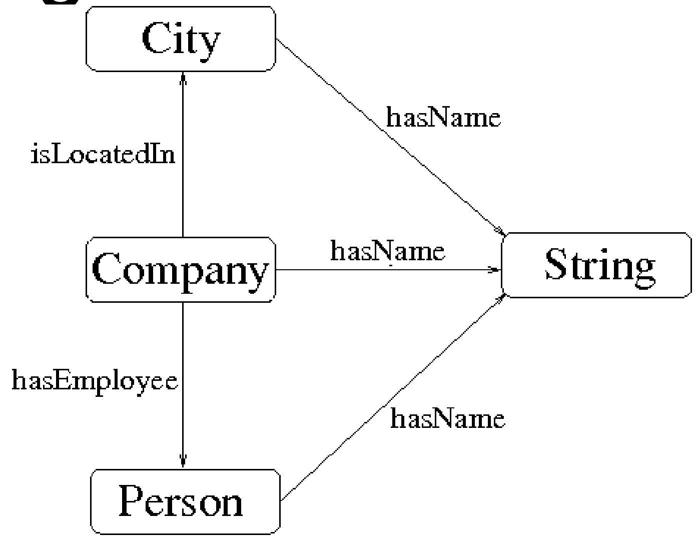
Reasoning with probabilistic Mappings

- Ontology on Peer 1: (O1)
- Ontology on Peer 2: (O2)





Normalization: Ontologies as Rules



Company v ∀isLocatedIn.City

Company v ∀hasEmployee.Person

Company v ∀hasName.String

Person v ∀hasName.String

City v ∀hasName.String

Company(X), isLocatedIn(X,Y) → City(Y).

Company(X), hasEmployee(X,Y) → Person(Y).

Company(X), hasName(X, Y) → String(Y).

Person(X), hasName(X,Y) → String(Y).

City(X), hasName(X,Y) → String(Y).

Bayesian Logic Programs

man(Bob).

married(Bob, Alice).

worksFor(Bob, Microsoft).

worksFor(Alice, Microsoft).

X	P(X)
true	1.0
false	0.0

X ∈ {man(Bob),
married(Bob,Alice)
worksFor(Bob,Microsoft)
worksFor(Alice,Microsoft)}

woman(Y) | man(X), married(X, Y).

P(woman(Y) = true | man(X) = true, married(X, Y) = true) = 0.8
P(woman(Y) = true | man(X) = true, married(X, Y) = false) = 0.1
P(woman(Y) = true | man(X) = false, married(X, Y) = true) = 0.45
P(woman(Y) = true | man(X) = false, married(X, Y) = false) = 0.1

Bayesian Logic Programs (example cont'd)

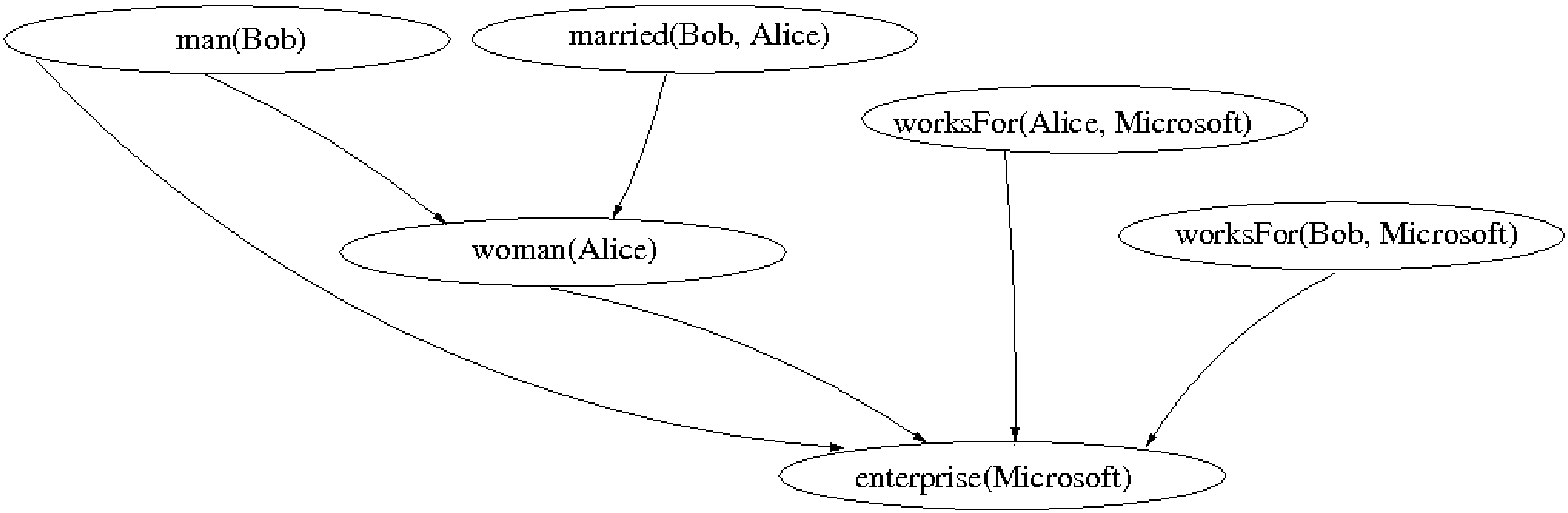
enterprise(Y) | man(X), worksFor(X, Y).

P(enterprise(Y) = true | man(X) = true, worksFor(X, Y) = true) = 0.9
P(enterprise(Y) = true | man(X) = true, worksFor(X, Y) = false) = 0.02
P(enterprise(Y) = true | man(X) = false, worksFor(X, Y) = true) = 0.7
P(enterprise(Y) = true | man(X) = false, worksFor(X, Y) = false) = 0.001

enterprise(Y) | woman(X), worksFor(X, Y).

P(enterprise(Y) = true | woman(X) = true, worksFor(X, Y) = true) = 0.8
P(enterprise(Y) = true | woman(X) = true, worksFor(X, Y) = false) = 0.01
P(enterprise(Y) = true | woman(X) = false, worksFor(X, Y) = true) = 0.6
P(enterprise(Y) = true | woman(X) = false, worksFor(X, Y) = false) = 0.0001

Translation to Bayesian Network



(CPTs omitted, see previous slide)

Problem: Wrong Mappings

- Example 1: Matching Fallacies
  - i:Author $\subseteq$ Person
  - j:Authorization $\subseteq \neg$Person
  - i:Person $\xrightarrow{\equiv}$ j:Person
  - i:Author $\xrightarrow{\equiv}$ j:Authorization

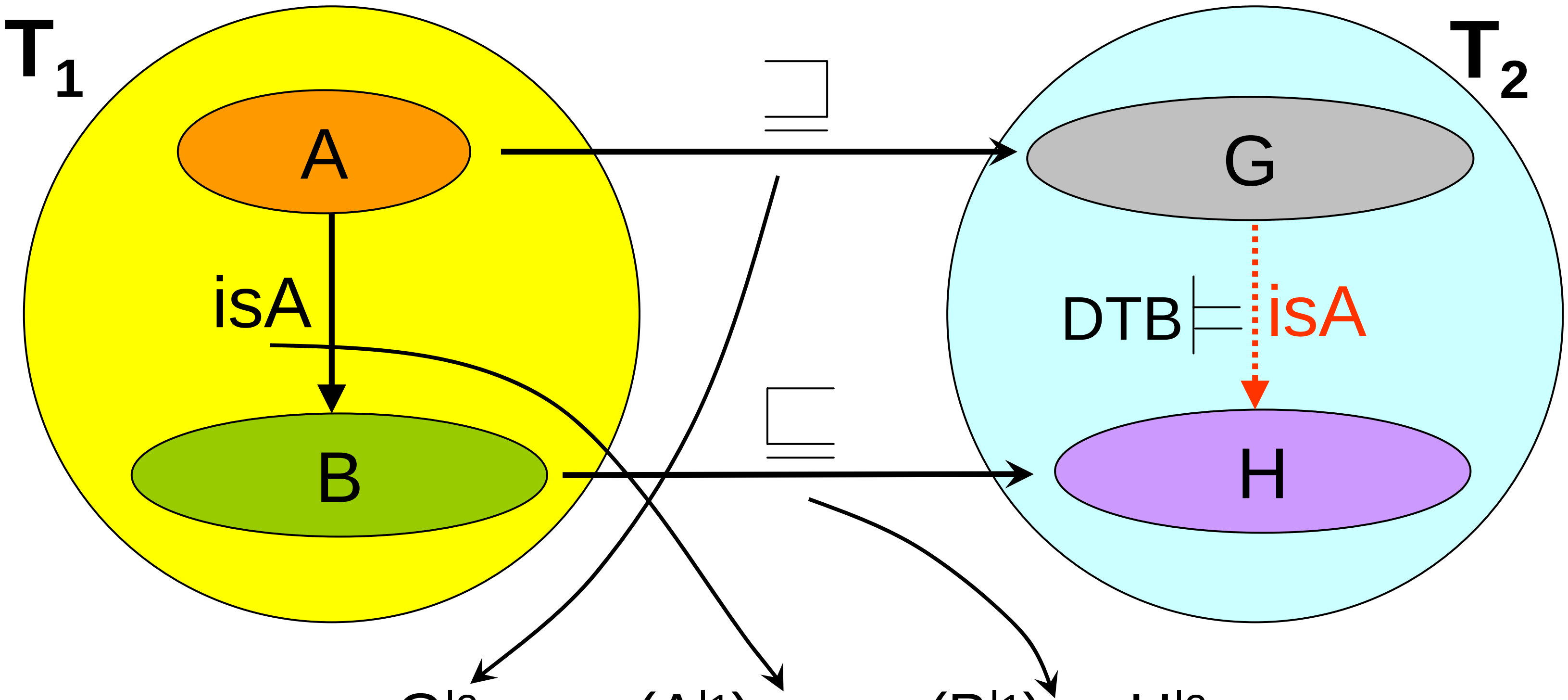
- Example 2: Modelling Fallacies
  - i: SportsCar $\subseteq$ Car
  - j:UselessThings $\equiv \neg$UselessThings
  - i:Car $\xrightarrow{\subseteq}$ j: UsefulThing
  - i:SportsCar $\xrightarrow{\subseteq}$ j: $\neg$UsefulThing

Reasoning ABOUT Mappings

- Check formal properties of mappings:
  - Are there inconsistent mappings
  - Are there redundant mappings
  - Are there implied mappings
- Use Cases:
  - Support for manual mapping creation
  - Validation of automatically created mappings

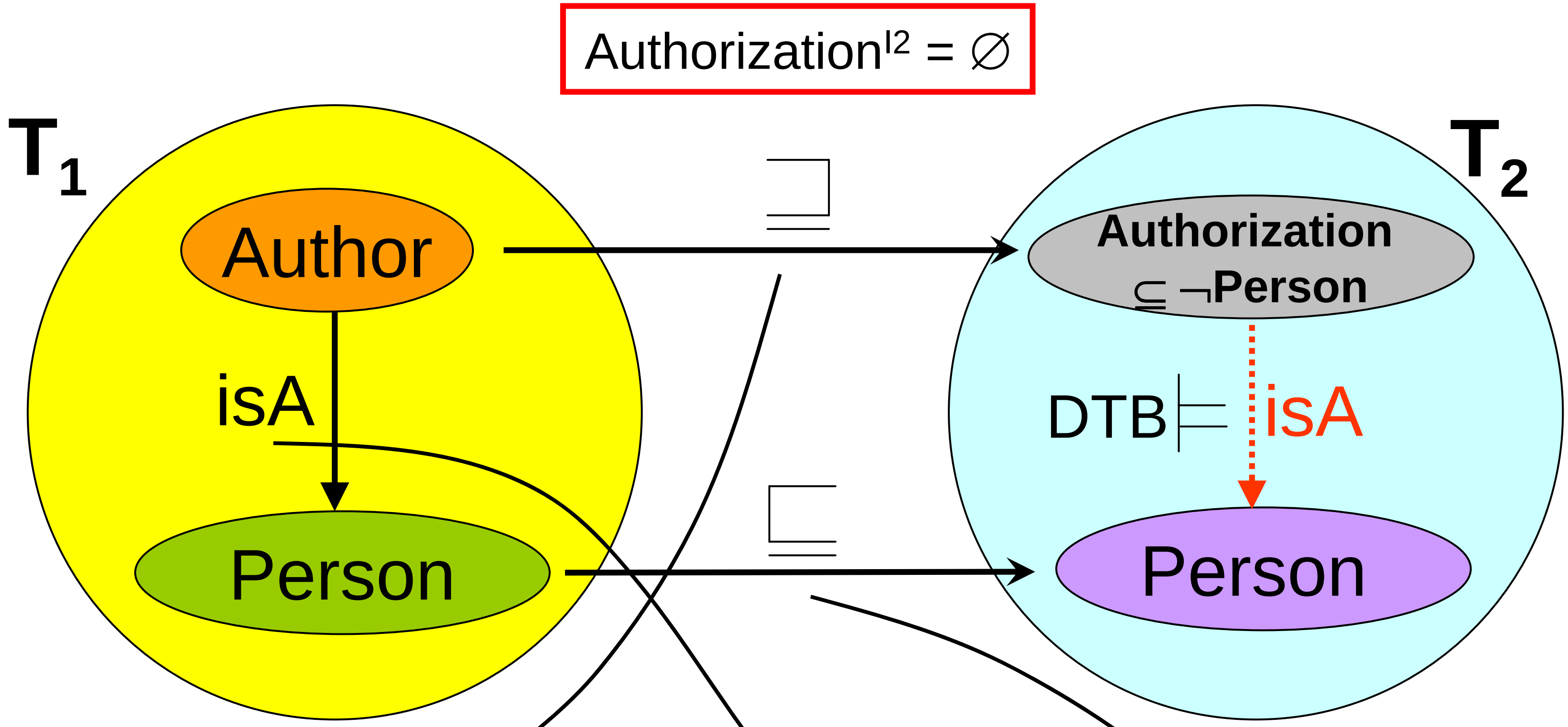
Subsumption propagation in DDL

$$DTB = \langle T_1, T_2, B_{12} \rangle$$



$$G^{I_2} \subseteq r_{12}(A^{I_1}) \subseteq r_{12}(B^{I_1}) \subseteq H^{I_2}$$

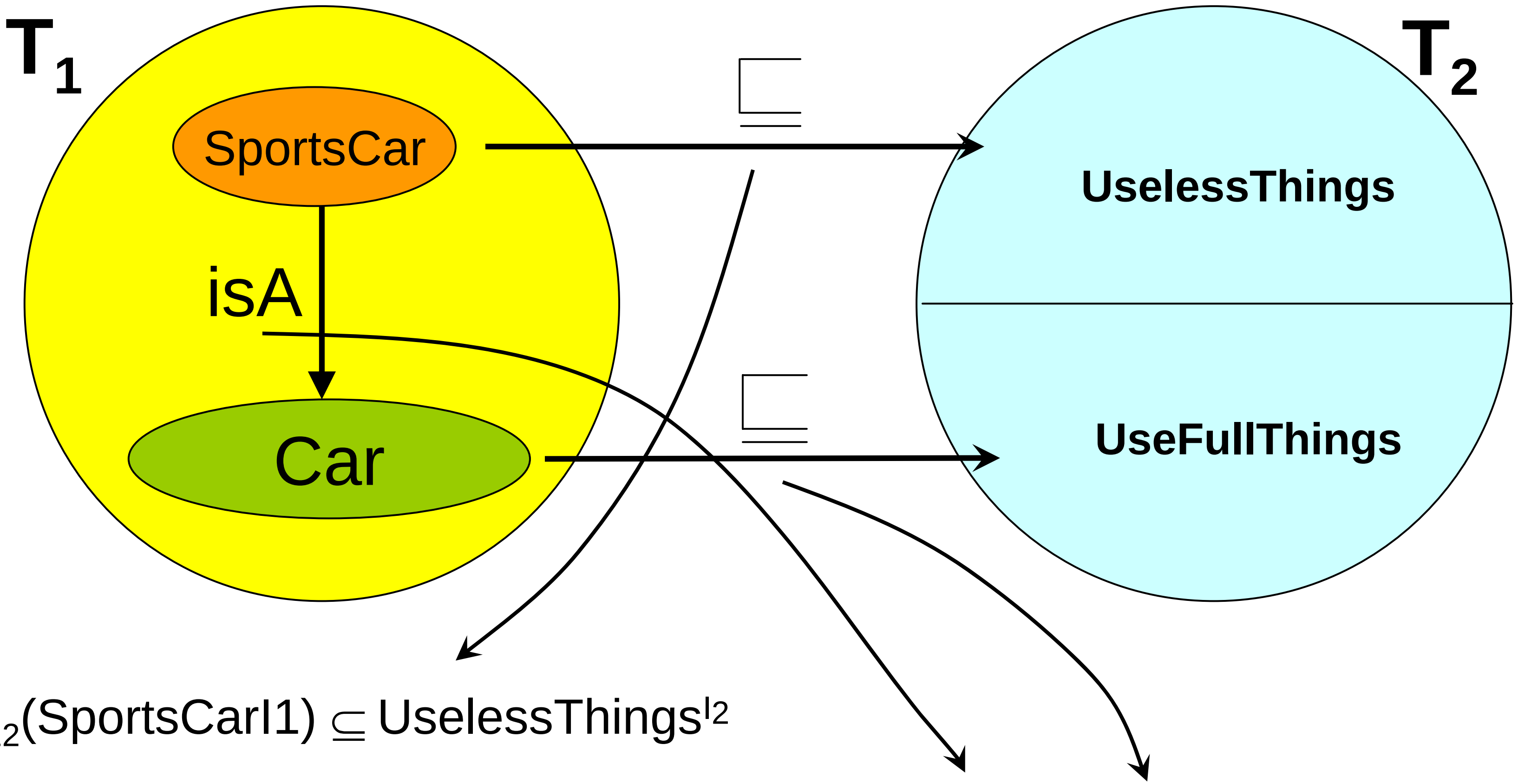
Example 1: Inconsistency



$$\text{Authorization}^{I2} \subseteq r_{12}(\text{Authorization}^{I1}) \subseteq r_{12}(\text{Person}^{I1}) \subseteq \text{Person}^{I2}$$

Example 2: Embedding

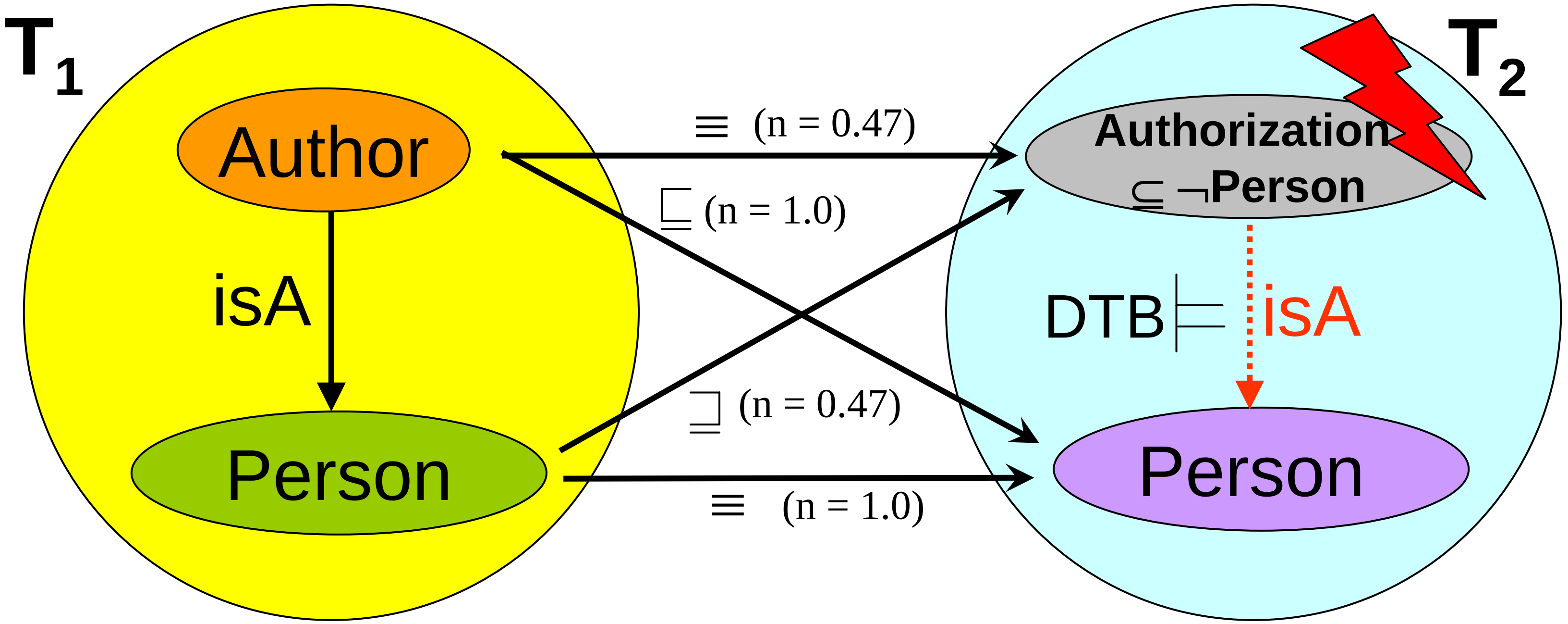
$r_{12}(\text{SportsCar}) = \emptyset$



$r_{12}(\text{SportsCarI1}) \subseteq \text{UselessThings}^{I2}$

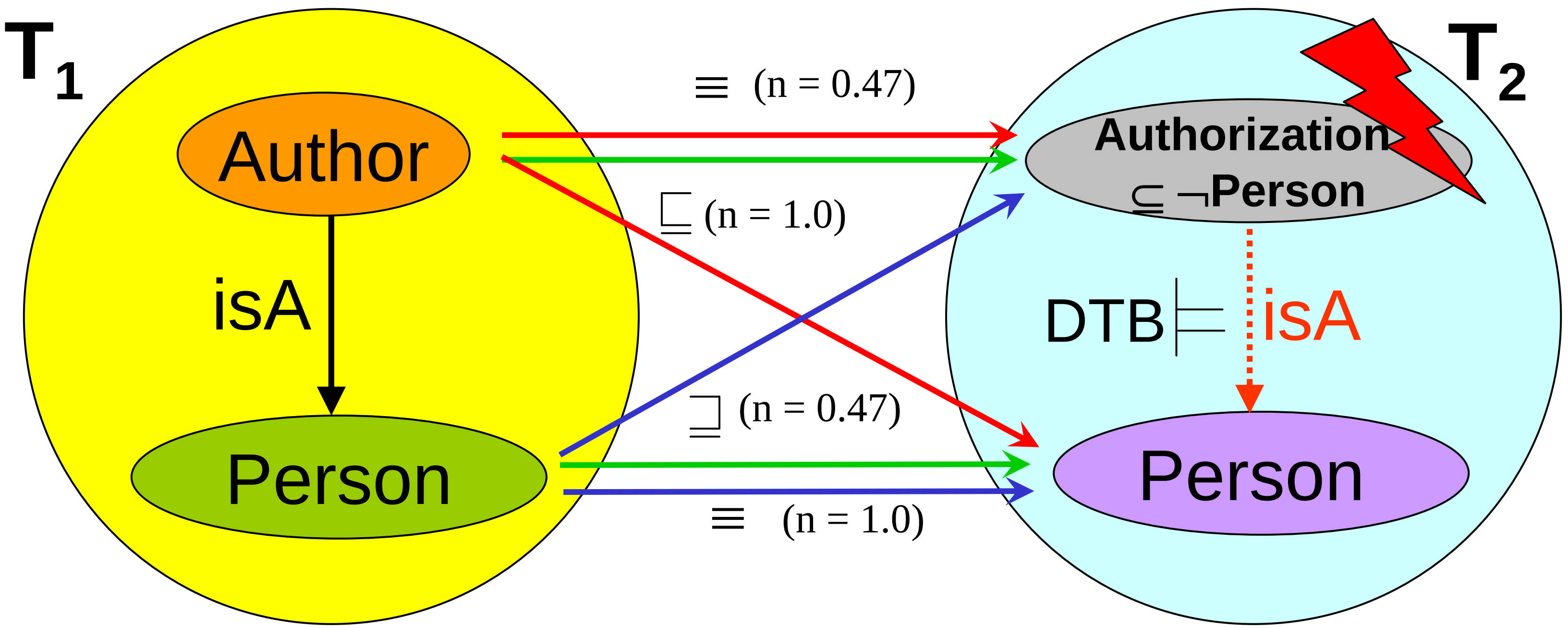
$r_{12}(\text{SportsCarI1}) \subseteq r_{12}(\text{Car}^{I1}) \subseteq \text{UseFullThings}^{I2}$

Repairing Mappings



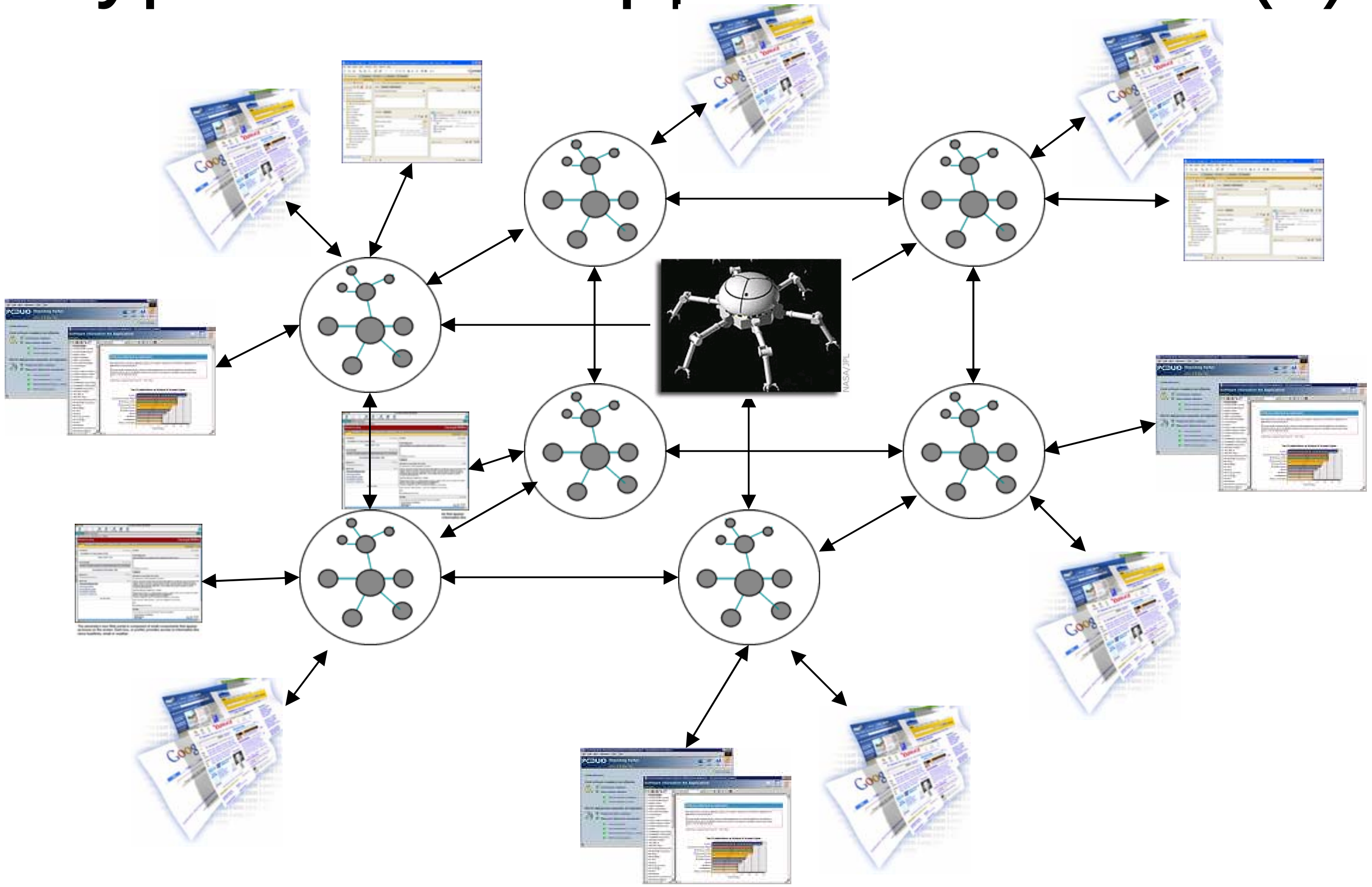
1. Syntactic Matching
2. Structural Matching
3. Analysis

Repairing Mappings



1. Syntactic Matching
2. Structural Matching
3. Analysis
4. Compute Conflict Sets
5. Select Problematic Rules and repair mapping

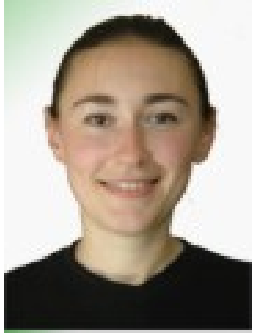
The typical Web Application 2010 (?)



Conclusions

- So what is wrong with the semantic web so far ?
  - A lot of work was done on language for describing rich information semantics (which is good!)
  - Too little attention has been paid to the specific needs of a distributed environment such as the Web (this is not so good)
- Is it any good then ?
  - YES. There are many useful applications with a rather centralized nature (community portals, company web sites)
  - YES. People start recognizing the need for distributed abd robust approaches.

The KR & KM Research Group



- Christian Meilicke
- Livia Predoiu
- Anne Schlicht
- Heiner Stuckenschmidt

Acknowledgement

- Thanks to Rainer Eckstein for the invitation

- During the Presentation, I used material originally created by:
  - Luciano Serafini, Andrei Tamilin, ITC-IRST Trento
  - Frank van Harmelen, Vrije Universiteit Amsterdam
  - Jerome Euzenat, INRIA Rhones-Alpes
  - Pavel Shvaiko, Fausto Giunchiglia, University of Trento

- The DRAGO System for distributed reasoning with ontologies can be downloaded at: http://sra.itc.it/projects/drago/

- The work is partially funded by the German Science Foundation in the Emmy-Noether Programme