Status of Semantic Technologies

What is semantic technology useful for today?
And what can we expect within the coming years?

Ralf Moeller
Hamburg University of Technology

Semantic Technologies Seen as Data Generators

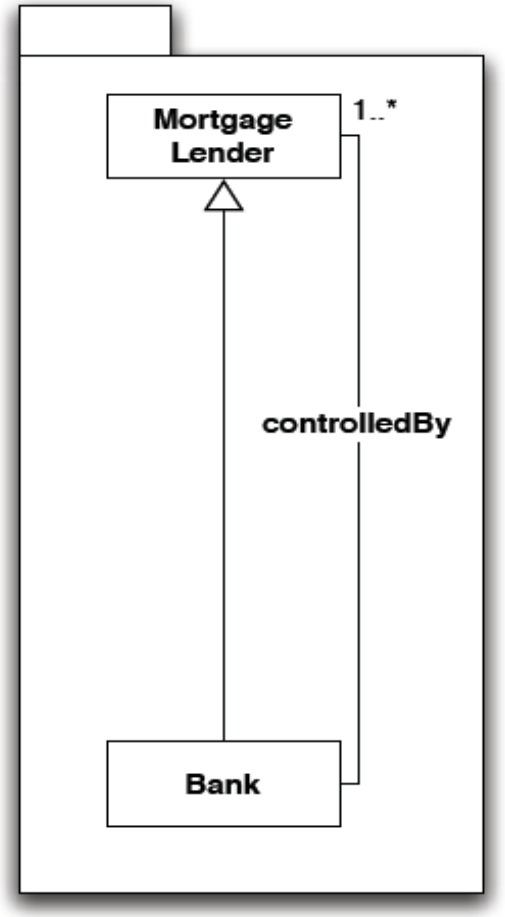
- Ontology (in the narrow sense = KB):

```
(define-role controlledBy
    :domain MortgageLender :range Bank)

(implies Bank MortgageLender)

(implies MortgageLender
   (at-least 1 controlledBy Bank))

(related Halifax LeedsBS controlledBy)

(instance RBS Bank)
```

- Query:

?- { (X) | controlledBy(X, Y) }

X=Halifax, X=RBS, X=LeedsBS



Bank
RBS

controlledBy	
Halifax	Leeds BS

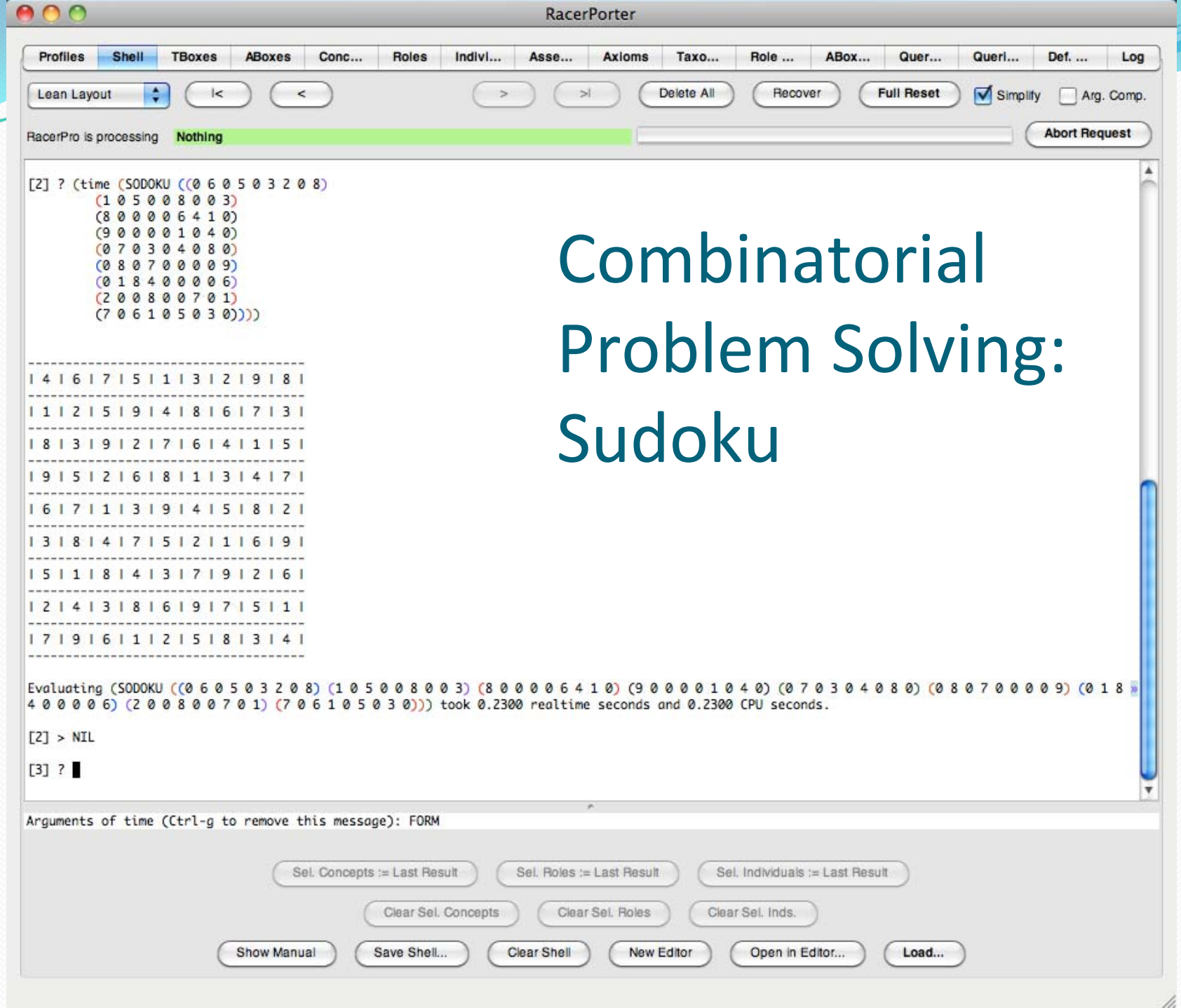
Comparison to DBs

- **Standard databases**: Use schemata for being able to (effectively) ***retrieve only stored data***
  (or aggregations thereof)
- **Semantic technologies**: Use schemata (knowledge bases, ontologies) to ***retrieve more than stored data***
  - No schemata (axioms) needed to just store data
  - The less axioms,
    - the more data can be stored
      (no inconsistencies), but…
    - the less data is implicitly added
  - Deal with ***data descriptions*** rather than data

Semantic Technologies Now: Definition

- All **representation formalisms** (and corresponding reasoning systems)
- whose representation language has a **formal semantics**,
- and whose basic decision/inference problems are **decidable**,
- and, based on a **formal query language**, potentially **data is augmented**

What can you do with semantic technologies now?



Combinatorial Problem Solving: Sudoku

Development of Domain Models

- Expressive domain modeling (EXPTIME-hard w.r.t. kb size)
  - SROIQ (+ some additions proposed)
  - SHIQ-(D) (e.g., for UML diagrams)
  - Horn-SHIQ
- Restricted expressivity such that QA scales
  - DL-Lite (for large parts of UML, subsumes RDFS)
    - Query answering can be directly done by relational engines after query is transformed (QA for conjunctive queries in AC_0)
  - Horn-SHIQ
    - Polynomial data complexity for QA (conjunctive queries)
- EL expressivity (or EL++ expressivity)
  - Basic ontology inference problems (e.g., subsumption) are tractable
  - QA can be done by rel. engines if data descriptions are extended

Ontology Development: Some Benchmark Ontologies

- Galen (A: 2700 classes, B: >20000 classes, versions without functionalities and inverses)
- Gene Ontology (GO, > 20000 classes)
- National Cancer Institute Ontology (NCI, > 20000 cl.)
- Snomed-CT (> 380,000 classes, EL)

Many others:

- Foundational Model of Anatomy (FMA)
- UMLS
- ….

SOTA

- Classifying ontologies (in secs)

	FaCT++	Pellet	HermiT	CEL	CB
GO	15.24	72.02	199.52	1.84	1.17
NCI	6.05	26.47	169.47	5.76	3.57
GalenA$^-$	3166.74	133.25	91.98	3.27	0.26
GalenA	465.35	—	45.72	*n/a*	0.32
GalenB$^-$	—	—	—	189.12	4.07
GalenB	—	—	—	*n/a*	9.58
Snomed	650.37	—	—	1185.70	49.44

Table by Yevgeny Kazakov 09

- Snomed with some axioms involving negation
  - Classify (1 hour, RacerPro)
  - Find all unsatisfiable concept names (9 mins, RacerPro)

New Reasoning Services

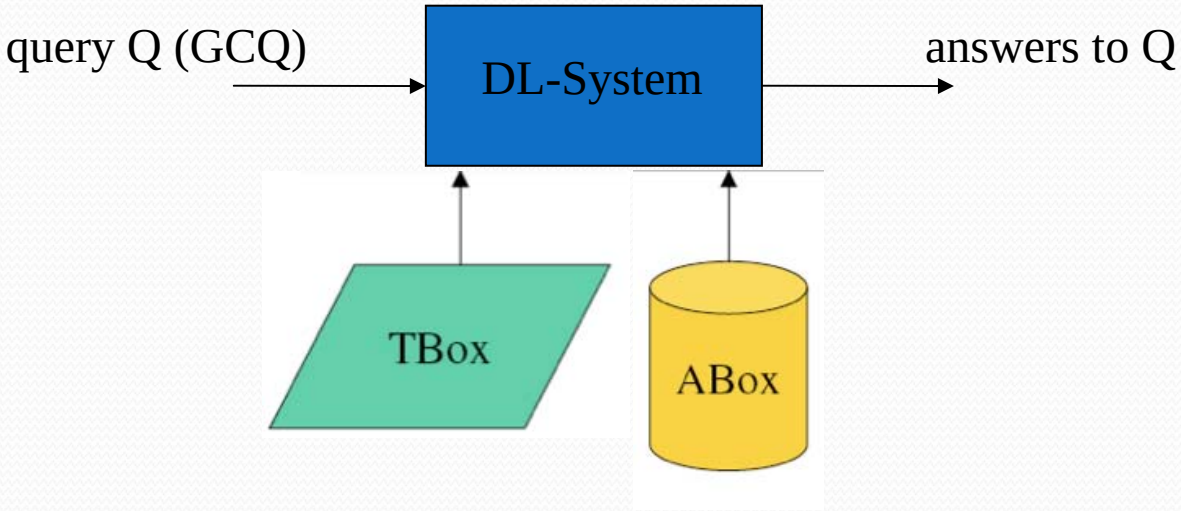
- Ontology development support
  - Explanation
    - White-box approaches
    - Black-box approaches
  - Suggestions for concepts which might be worth a name in order to better structure the ontology
- Ontology reuse support
  - Modularization

Querying w.r.t. Ontologies

- Ontologies produce (more) data
- But there might be multiple variants to produce data depending on the expressivity of the ontology language
- If some data is considered for an answer it must be available in all variants (-> certain answers)

Ontology-based Query Answering: Main-memory

- ALNHf: CLASSIC (90-96)
- SHIQ: RacerPro (99-), Pellet (04-), KAON (06-)
  - Sytems (constantly improved)
    - Pellet 2.0 (SROIQ(D)$^-$/OWL 2)
    - RacerPro 2.0 (SHIQ(D)$^-$/OWL-Lite+Q)

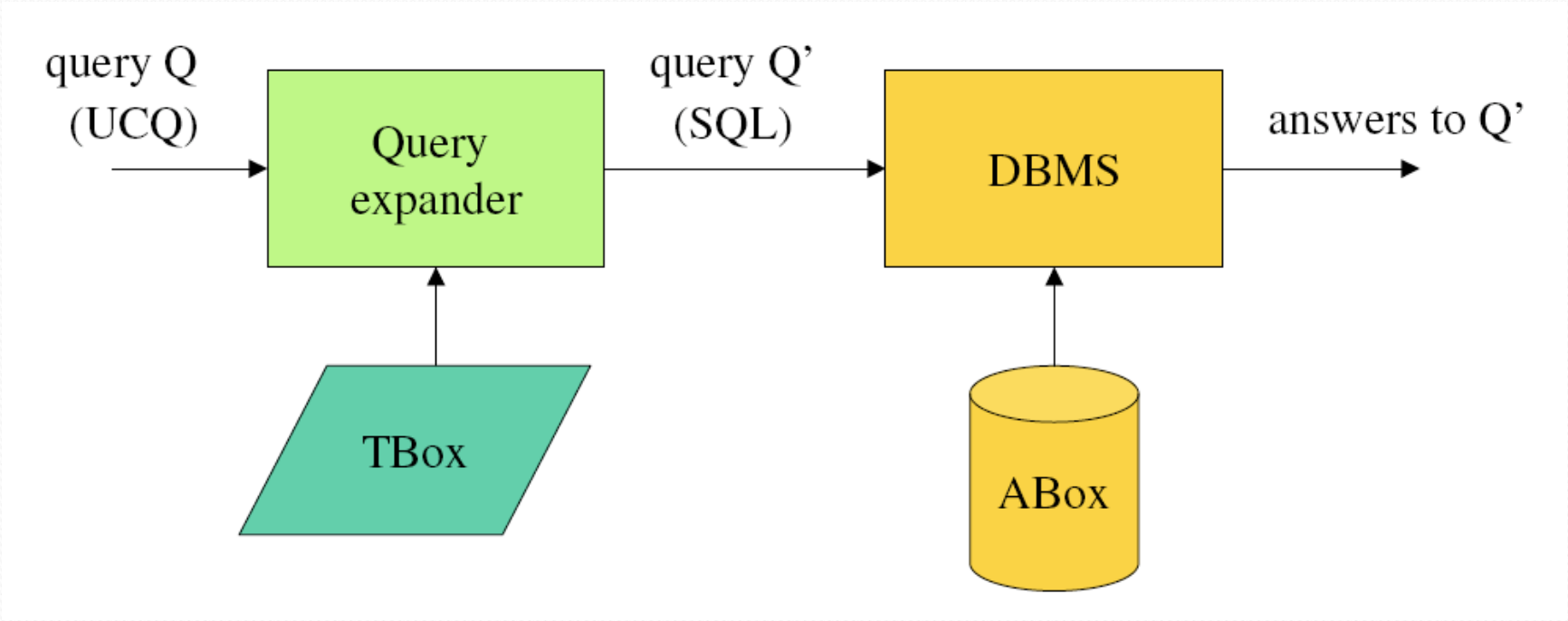


Expressivity of the Query Language

- **Conjunctive queries**
  - Quonto (DIG, SPARQL)
- **Grounded conjunctive queries**
  - Pellet (SPARQL)
  - RacerPro (DIG, OWLlink, nRQL, SPARQL)
  - SHER (SPARQL)
- **Epistemic first-order queries**
  - Expressive enough for invariant checking
  - RacerPro (nRQL)
- **Server-side functional programming**
  - RacerPro

Ontology-based Query Answering: Secondary Memory

DL-Lite: Quonto (06-)



Taken from a presentation by Riccardo Rosati

Example

TBox:

$\text{MALE} \sqsubseteq \text{PERSON}$ $\quad\quad$ $\text{FEMALE} \sqsubseteq \text{PERSON}$

$\text{MALE} \sqsubseteq \neg\text{FEMALE}$ $\quad\quad$ $\text{PERSON} \sqsubseteq \exists\text{hasFather}$

$\exists\text{hasFather}^{-} \sqsubseteq \text{MALE}$ $\quad\quad$ $\text{PERSON} \sqsubseteq \exists\text{hasMother}$

$\exists\text{hasMother}^{-} \sqsubseteq \text{FEMALE}$

input query:

$q(x) \leftarrow \text{PERSON}(x)$

rewritten query:

$$q'(x) \leftarrow \text{PERSON}(x) \vee \text{FEMALE}(x) \vee \text{MALE}(x) \vee \text{hasFather}(y,x) \vee \text{hasMother}(y,x)$$

Taken from a presentation by Riccardo Rosati

Example (contnd)

rewritten query:

$q'(x) \leftarrow \text{PERSON}(x) \vee$
$\text{FEMALE}(x) \vee$
$\text{MALE}(x) \vee$
$\text{hasFather}(y,x) \vee$
$\text{hasMother}(y,x)$

ABox:

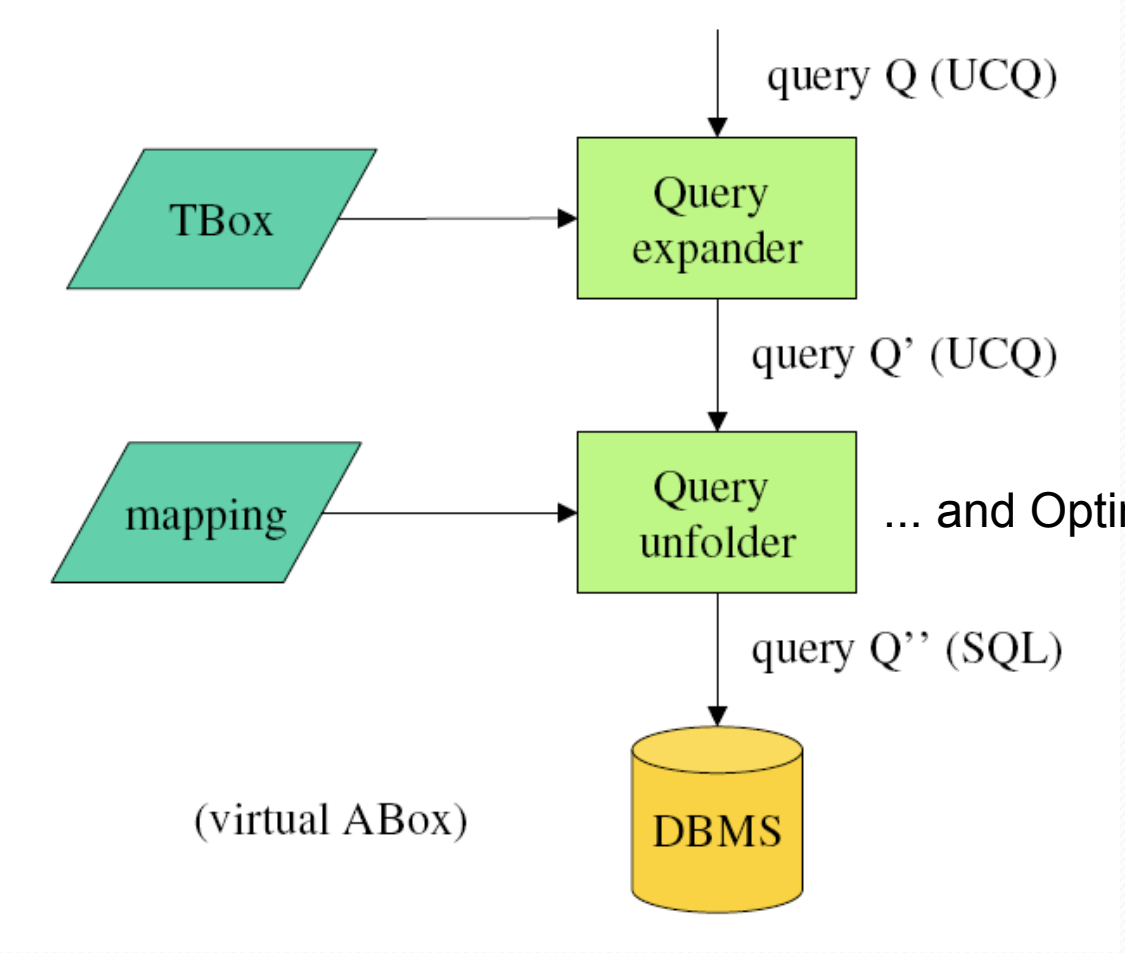
MALE(Bob)
MALE(Paul)
FEMALE(Ann)
hasFather(Paul,Ann)
hasMother(Mary,Paul)

answers to query:

{ Bob, Paul, Ann, Mary }

Taken from a presentation by Riccardo Rosati

DB Mappings (MASTRO)

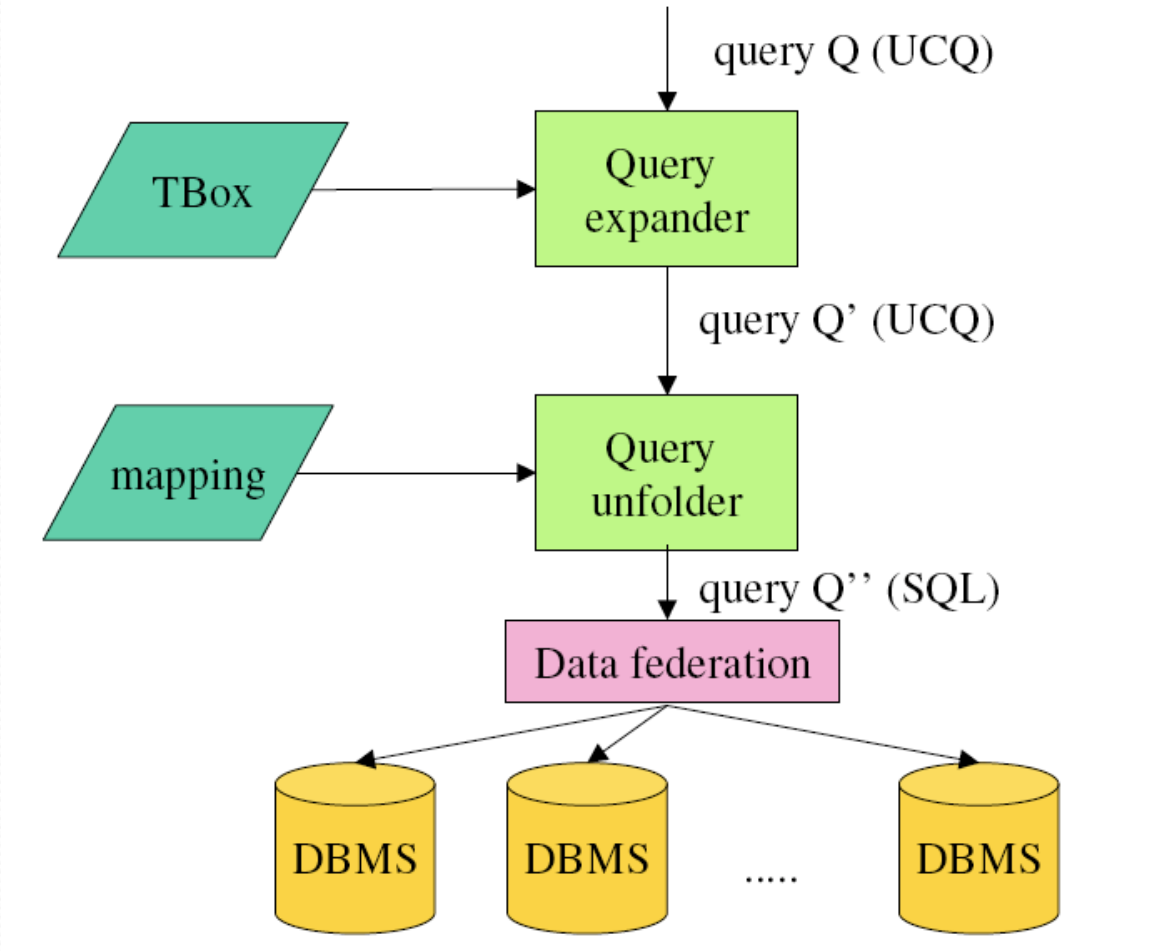


Taken from a presentation by Riccardo Rosati

Ontology-Based Data Access

- Relational data model
  - Databases (OBDA with Quonto for DL-Lite)
- RDF data model
  - Triple stores (OBDA with AllegroGraph for RDFS++)
    - Besides query languages, often additional analysis routines are provided (e.g., social network analysis in AllegroGraph)
- Object-oriented data model
  - OO databases
- XML
  - XML databases

Data Integration (MASTRO-I)



Taken from a presentation by Riccardo Rosati

Summary: What can semantic technologies be used for today?

- Combinatorial **problem solving** (e.g., Sudoku)
- Systematic **construction of domain models**
  - Consistency checks, subsumption checks
  - Explanation, modularization, concept suggestions
- Instance retrieval / **query answering**
  - Main-memory-scalable for expressive ontology languages *(if expressivity is not excessively used)*
  - Secondary-memory-scalable *(almost)* for data-oriented ontology languages
- **Data integration**

This list is definitely incomplete

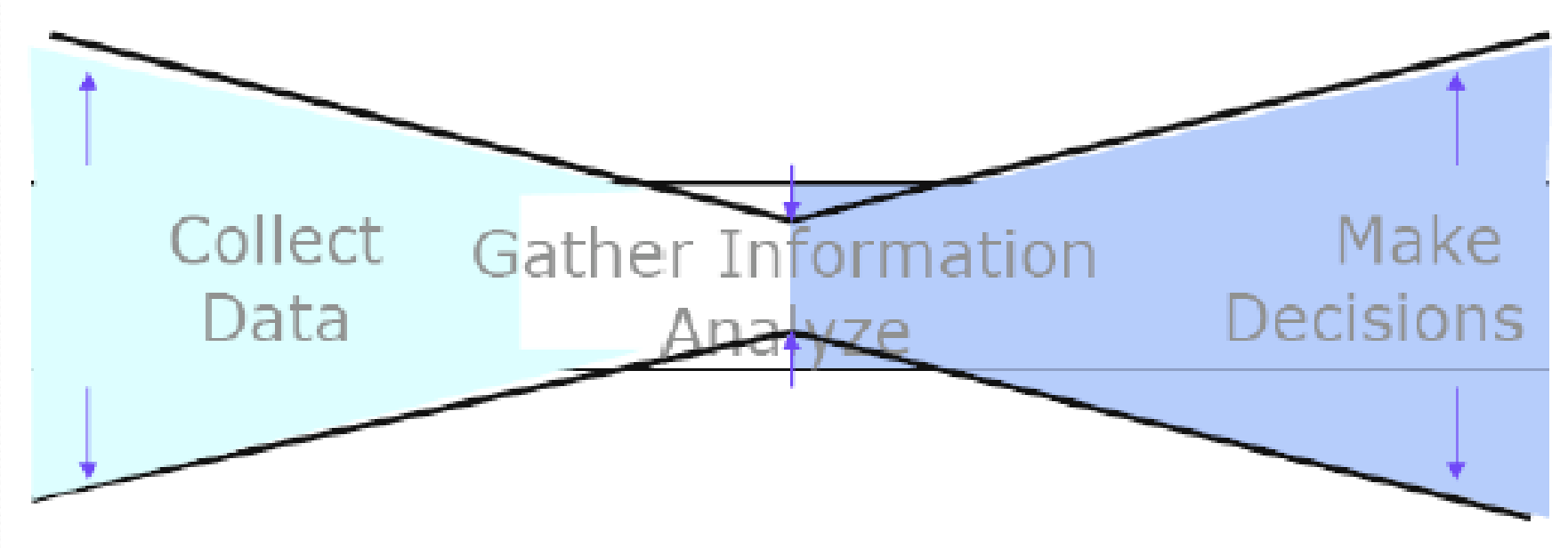
There's More to Semantic Technologies in the Future

- In some cases, only content-oriented retrieval is possible (e.g., for text, image, video, or audio data)
- Symbolic content descriptions are not directly available but have to be provided/derived(**interpretation**, see my presentation tomorrow ☺)

What techniques will be available in the future?

- Fast QA for more expressive ontology languages (based on complete approximation and sound partitioning)
- Stream-oriented data processing
  - Situation awareness / event recognition
- Systems supporting combinations of representation languages (e.g., OWL+ASP)
- Scalable reasoning with probabilistic and fuzzy representations considered
- Semantically well-founded programming with indefinite information (null values avoided)

Integrate, yes…

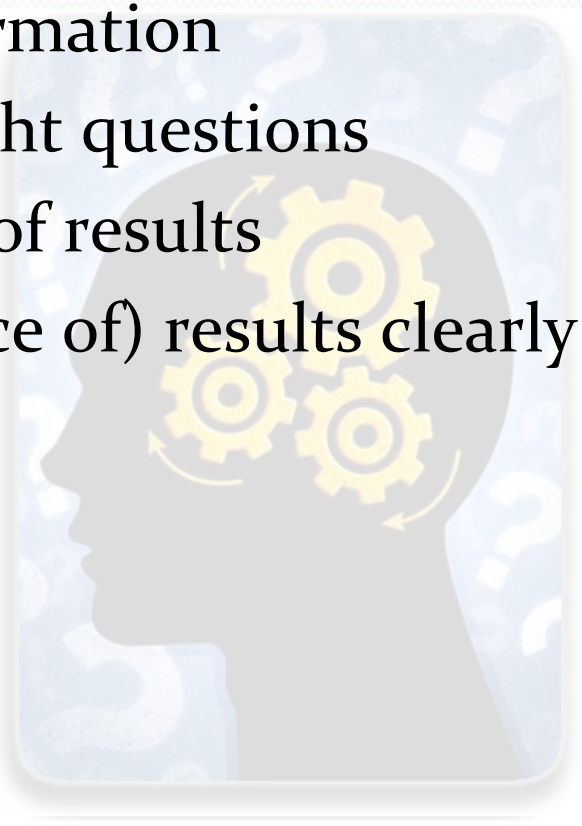


- Decisions based on information
- Analysis of data is the bottleneck

…, and understand?

Need to help decision-makers **understand** integrated information

- Find the right information
- Let user ask the right questions
- Create summaries of results
- Explain (provenance of) results clearly



Thank you

- This work has been supported by
  - the European Commission
    (Projects TONES and BOEMIE)
  - the Deutsche Forschungsgemeinschaft (DFG)
- Thanks also to all members of my group at TUHH,
  - in particular Sebastian Wandelt and Michael Wessel