Stream Reasoning For Linked Data

M. Balduini, J-P Calbimonte, O. Corcho, D. Dell'Aglio, E. Della Valle, and J.Z. Pan

http://streamreasoning.org/sr4ld2013

ISWC 2013
Sydney, Australia

OWL Reasoning and Stream Reasoning with LOD

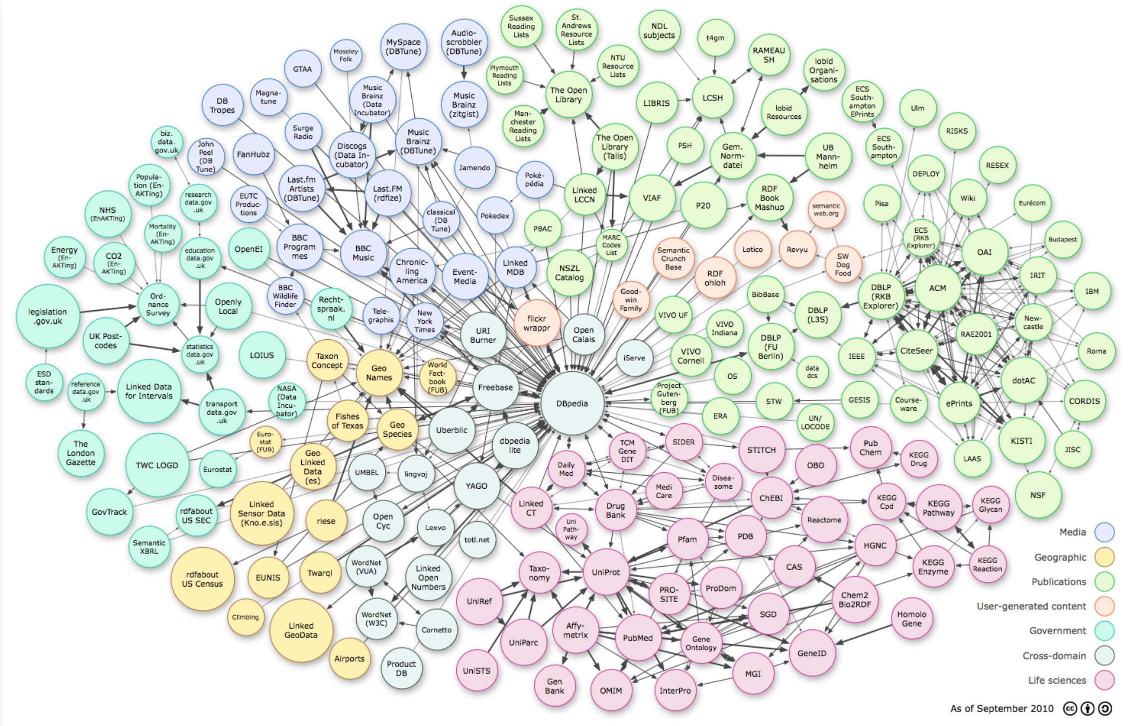
Jeff Z. Pan
University of Aberdeen
http://homepages.abdn.ac.uk/jeff.z.pan/pages/

Share, Remix, Reuse — Legally

ISWC 2013 Sydney, Australia

- This work is licensed under the Creative Commons Attribution 3.0 Unported License.
- **Your are free:**
  - **to Share** — to copy, distribute and transmit the work
  - **to Remix** — to adapt the work
- **Under the following conditions**
  - **Attribution** — You must attribute the work by inserting
    - "[source http://streamreasoning.org/sr4ld2013]" at the end of each reused slide
    - a credits slide stating
      - These slides are partially based on "Streaming Reasoning for Linked Data 2013" by M. Balduini, J-P Calbimonte, O. Corcho, D. Dell'Aglio, E. Della Valle, and J.Z. Pan http://streamreasoning.org/sr4ld2013
- To view a copy of this license, visit http://creativecommons.org/licenses/by/3.0/

LOD = Linked Open Data / Linked Ontological Data



<rdf:Description rdf:about="http://www.w3.org/
People/Berners-Lee/card#i">
<foaf:knows rdf:resource="http://dblp.l3s.de/
d2r/…/Dan_Brickley ">
</rdf:Description>

Agenda

1. OWL Reasoning with LOD (30m)
2. OWL Stream Reasoning with LOD (40m)
3. Hands-on session (20m)

What is an Ontology

ISWC 2013 Sydney, Australia

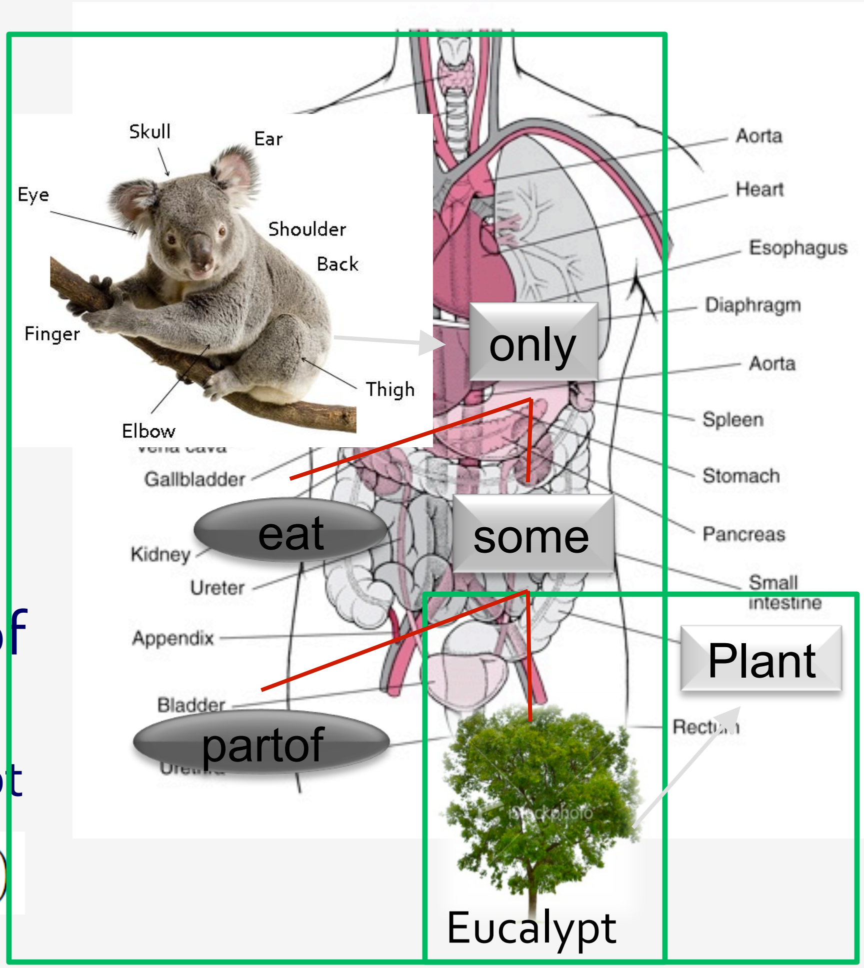
A formal dictionary of domain vocabulary

- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy
  - Koala
- Specifies meaning (semantics) of terms
  - Koala eat only some part of Eucalypt

$$Koala \sqsubseteq \forall eat.(\exists partof.Eucalypt)$$

  - Eucalypt is Plant

$$Eucalypt \sqsubseteq Plant$$



Components of Ontology

- A TBox (Terminonagy Box) is a set of "schema" axioms (sentences), e.g.:

$$Koala \sqsubseteq \forall eat.(\exists partof.Eucalypt)$$
$$Eucalypt \sqsubseteq Plant$$

  - i.e., a **background theory for the vocabulary**

- An ABox (Assertion Box) is a set of "data" axioms (ground facts), e.g.:

gummy: Koala

Ontology Reasoning

- Infer **implicit** knowledge from **explicit** knowledge

$$Koala \sqsubseteq \forall eat.(\exists partof.Eucalypt)$$
$$Eucalypt \sqsubseteq Plant$$

$$Koala \sqsubseteq \forall eat.\exists partof.Plant$$
$$Plant \sqcup \exists partof.Plant \sqsubseteq VegeFood$$

$$Koala \sqsubseteq \forall eat.VegeFood$$
$$\forall eat.VegeFood \sqsubseteq Herbivore$$

$$Koala \sqsubseteq Herbivore$$

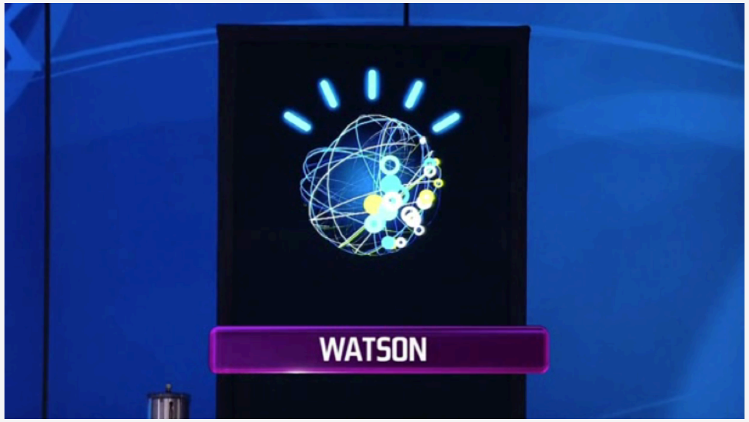
Query Answering

Example:

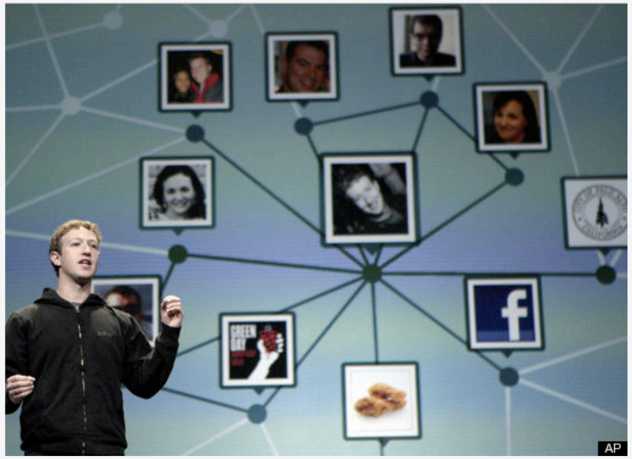
```
SELECT ?X,?Y
FROM <http://example.org/animal.owl>
WHERE {?X eat ?Y .}

SELECT ?X
FROM <http://example.org/animal.owl>
WHERE {?X rdf:type Herbivore .}
```

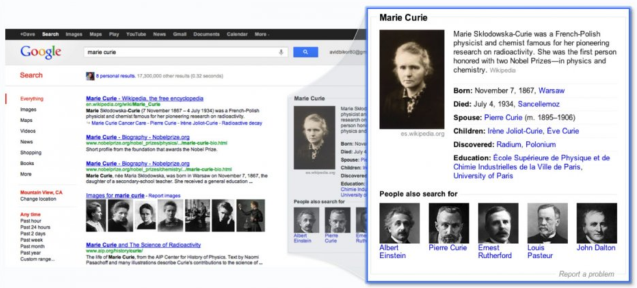
The need for bridging the gap between data and queries



Question Answering



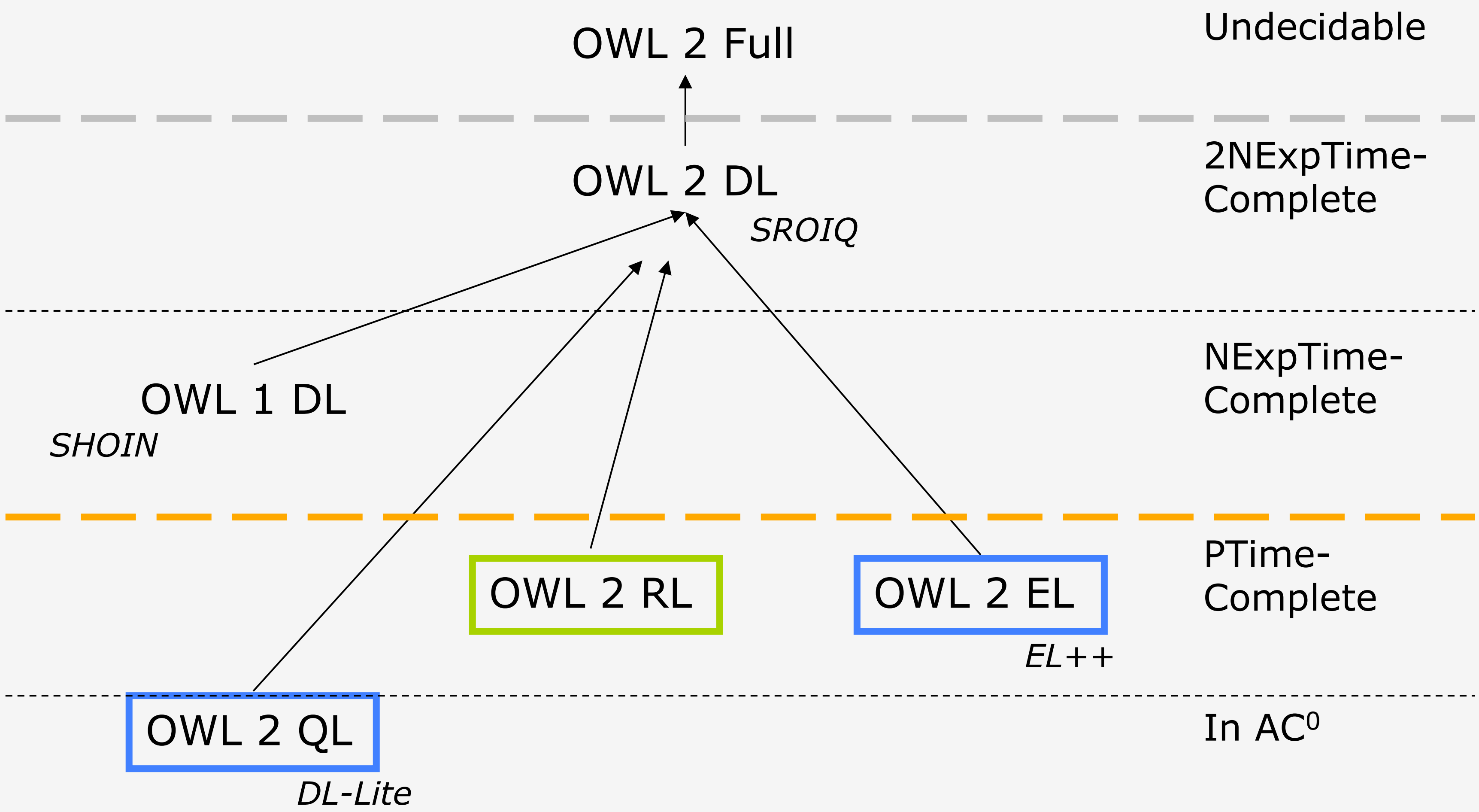
Social Discovery



Semantic Search

Digital Economy

The OWL family tree



OWL 2 EL

- A (near maximal) fragment of OWL 2 such that
  - Satisfiability checking is in PTime (**PTime-Complete**)
  - Data complexity of query answering also PTime-Complete
- Based on **EL** family of description logics [Baader et al. 2005]
- Can exploit **saturation** based reasoning techniques
  - Computes complete classification in "one pass"
  - Computationally optimal (PTime for EL)
  - Can be extended to Horn fragment of OWL DL [Kazakov 2009]

Saturation-based Technique (basics)

- Normalise ontology axioms to standard form:

$$A \sqsubseteq B \quad A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C$$

- Saturate using inference rules:

$$\frac{A \sqsubseteq B \quad B \sqsubseteq C}{A \sqsubseteq C} \qquad \frac{A \sqsubseteq B \quad A \sqsubseteq C \quad B \sqcap C \sqsubseteq D}{A \sqsubseteq D}$$

$$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$$

- Extension to Horn fragment requires (many) more rules

Saturation-based Technique (basics)

ISWC 2013 Sydney, Australia

Example:

ex:Performer $\equiv$ foaf:Person $\sqcap$ $\exists$mo:perform_in.mo:Performance

mo:MusicArtist $\equiv$ foaf:Person $\sqcap$ $\exists$mo:perform_in. ex:MusicPerformance

ex:MusicPerformance $\sqsubseteq$ mo:Performance

ex:Performer $\sqsubseteq$ foaf:Person

ex:Performer $\sqsubseteq$ $\exists$mo:perform_in.mo:Performance

$\exists$mo:perform_in.mo:Performance $\sqsubseteq$ PP

foaf:Person $\sqcap$ PP $\sqsubseteq$ ex:Performer

mo:MusicArtist $\sqsubseteq$ foaf:Person

mo:MusicArtist $\sqsubseteq$ $\exists$mo:perform_in.ex:MusicPerformance

$\exists$mo:perform_in.ex:MusicPerformance $\sqsubseteq$ PMP

foaf:Person $\sqcap$ PMP $\sqsubseteq$ mo:MusicArtist

ex:MusicPerformance $\sqsubseteq$ mo:Performance

Saturation-based Technique (basics)

ISWC 2013 Sydney, Australia

Example:

ex:Performer $\sqsubseteq$ foaf:Person
ex:Performer $\sqsubseteq$ $\exists$mo:perform_in.mo:Performance
$\exists$mo:perform_in.mo:Performance $\sqsubseteq$ PP
foaf:Person $\sqcap$ PP $\sqsubseteq$ ex:Performer

mo:MusicArtist $\sqsubseteq$ foaf:Person
mo:MusicArtist $\sqsubseteq$ $\exists$mo:perform_in.ex:MusicPerformance
$\exists$mo:perform_in.ex:MusicPerformance $\sqsubseteq$ PMP
foaf:Person $\sqcap$ PMP $\sqsubseteq$ mo:MusicArtist

ex:MusicPerformance $\sqsubseteq$ mo:Performance

$$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$$

mo:MusicArtist $\sqsubseteq$ PP

Saturation-based Technique (basics)

Example:

ex:Performer $\sqsubseteq$ foaf:Person
ex:Performer $\sqsubseteq$ ∃mo:perform_in.mo:Performance
∃mo:perform_in.mo:Performance $\sqsubseteq$ PP
foaf:Person $\sqcap$ PP $\sqsubseteq$ ex:Performer

mo:MusicArtist $\sqsubseteq$ foaf:Person
mo:MusicArtist $\sqsubseteq$ ∃mo:perform_in.ex:MusicPerformance
∃mo:perform_in.ex:MusicPerformance $\sqsubseteq$ PMP
foaf:Person $\sqcap$ PMP $\sqsubseteq$ mo:MusicArtist

ex:MusicPerformance $\sqsubseteq$ mo:Performance

$$\frac{A \sqsubseteq B \quad A \sqsubseteq C \quad B \sqcap C \sqsubseteq D}{A \sqsubseteq D}$$

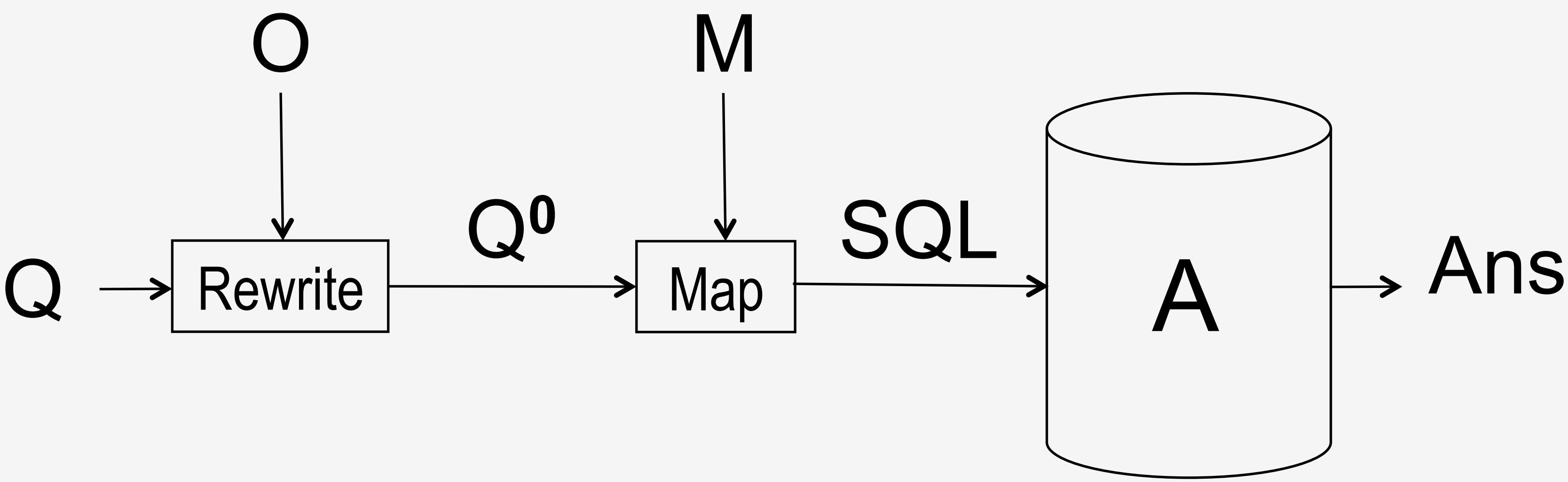
mo:MusicArtist $\sqsubseteq$ PP
mo:MusicArtist $\sqsubseteq$ ex:Performer

OWL 2 QL

- A (near maximal) fragment of OWL 2 such that
  - Data complexity of conjunctive query answering in **AC^0**
- Based on **DL-Lite** family of description logics [Calvanese et al. 2005; 2006; 2008]
- Can exploit **query rewriting** based reasoning technique
  - Data storage and query evaluation can be delegated to standard **RDBMS**
  - Novel technique to prevent exponential blowup produced by rewritings [Kontchakov et al. 2010, Rosati and Almatelli 2010]
  - Can be extended to more expressive languages (beyond AC^0) by delegating query answering to a Datalog engine [Perez-Urbina et al. 2009]

Query Rewriting Technique (basics)

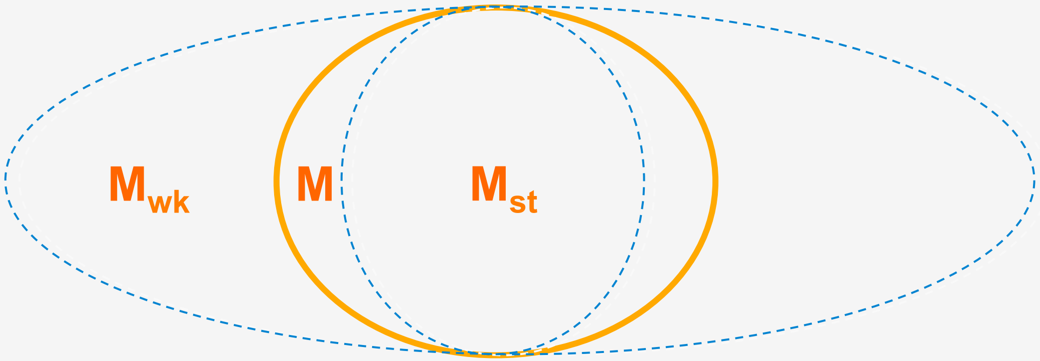
- Given ontology O and query Q, use O to rewrite Q as Q^0 s.t., for any set of ground facts A:
  - ans(Q, O, A) = ans(Q^0, ;, A)
- Use (GAV) mapping M to map Q^0 to SQL query



Approximate Reasoning in OWL 2 DL

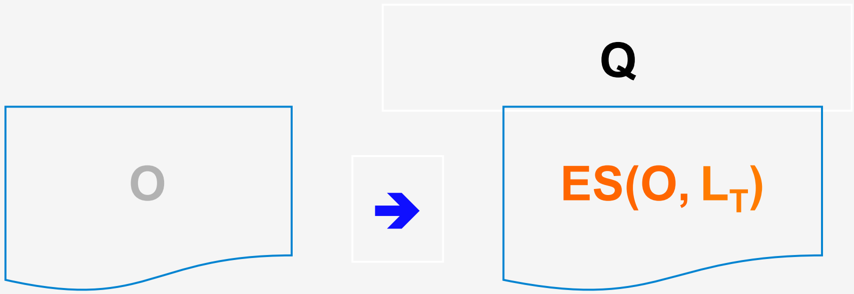
- Idea: to compile a source ontology O (in more expressive L_S) into its upper/lower bound (in less expressive L_T)

- Entailment set ES(O, L_T) of O in L_T
  - The set of **all** L_T axioms that are entailed by O under N_C, N_P and N_I



Semantic Approximation [Pan and Thomas, 2007]

- Strongest weaker approximation for QL ES(O, DL-Lite$_{core}$) of an OWL2 DL O is finite and unique.
- Theorem 1: Given an ontology O, a conjunctive query q(X) and an evaluation [X→S], if ES(O, DL-Lite$_{core}$) |= $q_{[X \to S]}$, then O |= $q_{[X \to S]}$.
- Theorem 2: Given an ontology O_S, a database-style conjunctive query q(X) without non-distinguished variables and an evaluation [X→S], ES(O_S, DL-Lite$_{core}$) |= $q_{[X \to S]}$ iff O_S |= $q_{[X \to S]}$.



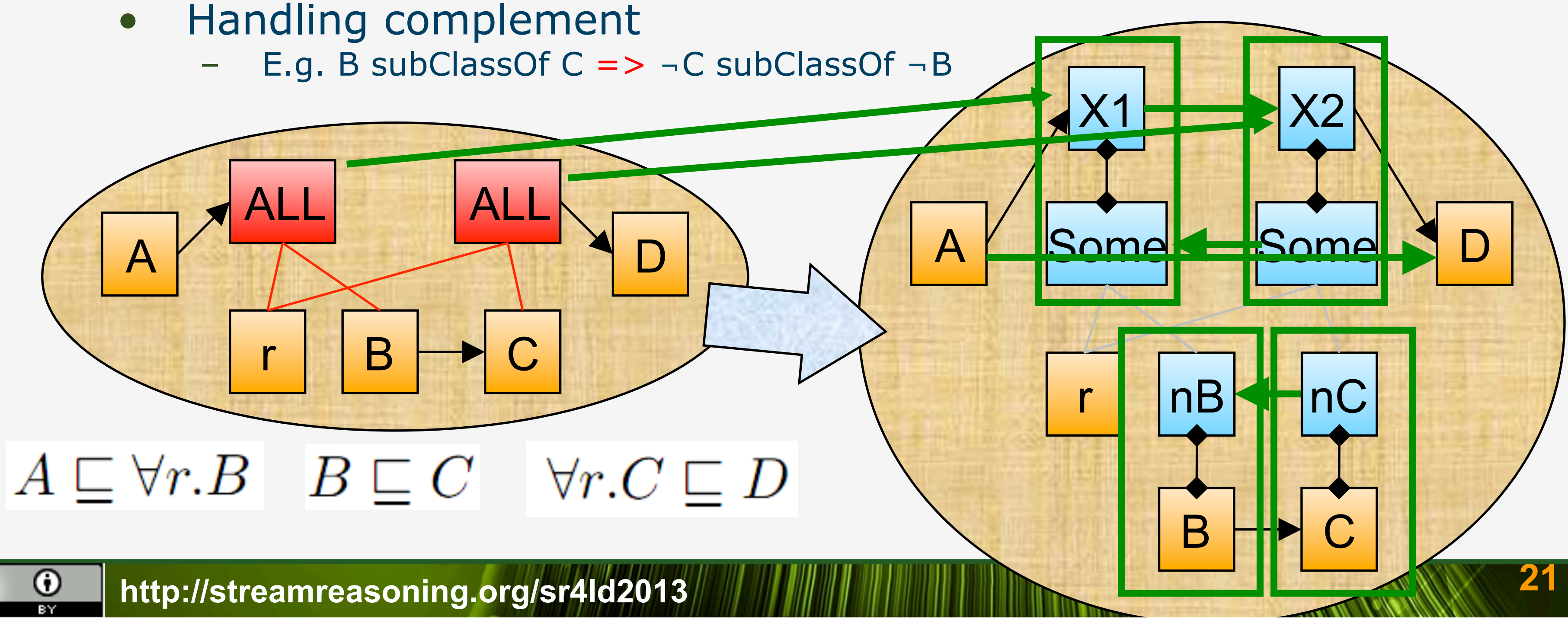
Syntactic Approximate Reasoning [Ren et al, 2010]

- Syntactic approximation from OWL2 DL to OWL2 EL
  - Minor syntactic gap results in major complexity difference
  - Using approximation to bridge the gap

DL ROQ (large subset of OWL2 DL)	DL EL++ (large subset of OWL2 EL)
$\top \mid \bot \mid A \mid C \sqcap D \mid \exists r.C \mid \{a\} \mid \boxed{\neg C \mid \geq nR.C}$	$\top \mid \bot \mid A \mid C \sqcap D \mid \exists r.C \mid \{a\}$
$C \sqsubseteq D \qquad r \sqsubseteq s, r_1 \circ \ldots \circ r_n \sqsubseteq s$	
$a : C \qquad (a, b) : r$	
N2EXPTIME-complete	**PTIME-complete**

Example: Syntactic Approximations

- Represent non-OWL2-EL concepts with fresh named concepts
  - E.g., ∀r.C subClassOf D ➔ $\mathbf{A}_{\forall r.C}$ subClassOf D
- Maintain semantic relations for these named concepts
  - *complementary relations*
  - *cardinality relations*
- Additional tractable completion Rules (on top of the EL ones), e.g.
  - Handling complement
    - E.g. B subClassOf C => ¬C subClassOf ¬B



Evaluation: Syntactic Approximations

ISWC 2013 Sydney, Australia

Table 15: Ontology Materialisation Comparison (t/o: out of time)

Ontology $\mathcal{O}$	FaCT++	HermiT	Pellet	REL	
				time	recall
Semintec	t/o	4.661	6.172	2.774	100%
VICODI	t/o	18.282	9.828	3.101	100%
Wine	364.8	47.865	200.982	0.172	100%
Sweet Numerics	2.139	32.419	2.016	0.578	100%
MGED	t/o	7.247	449.807	0.234	100%
BoC small	484.32	32.045	t/o	0.578	100%
BoC middle	t/o	426.013	t/o	3.348	100%
BoC big	t/o	t/o	t/o	22.058	N/A

Table 12: TBox Classification Comparison (t/o: out of time; e/o: terminate with error)

Ontology $\mathcal{O}$	FaCT++	HermiT	Pellet	REL	
				time	recall
Biological Process	2.792	5.796	9.383	2.499	100%
Cellular Component	4.648	9.186	12.36	4.513	100%
GO	2.729	5.904	9.461	2.499	100%
Cyc	14.36	11.152	140.15	2.514	100%
FMA Constitutional	e/o	t/o	t/o	57.589	N/A
DLP	0.188	34.409	t/o	0.156	100%
EFO	2.825	3.201	t/o	0.531	100%
OBI	93.66	172.311	t/o	1.943	100%
NCIt	44.24	810.87	360.582	34.801	100%

Agenda

1. OWL Reasoning with LOD (30m)
2. OWL Stream Reasoning with LOD (40m)
3. Hands-on session (20m)

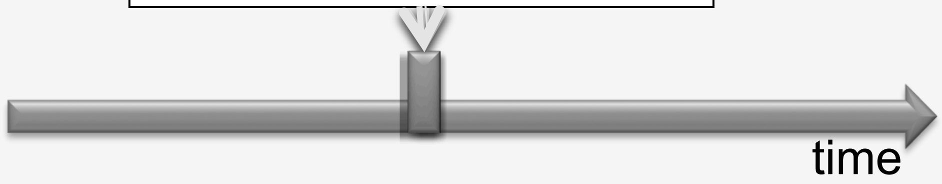
LOD Streams

- **Two streams [Ren and Pan, 2011]**
  - to-erase stream
  - to-add stream
- A LOD stream O[0,n] is a sequence of classical ontologies O(0), O(1), …, O(n):
  - O(0) is the initial ontology
  - Er(i) axioms to erase from O(i)
  - Ad(i) axioms to add into O(i)
  - O(i+1) = O(i) U Ad(i) \Er(i)
- Key task
  - Answer a set of monitoring queries at each snapshot ontology O(i)

- …

$$
\begin{array}{l}
interest \circ interest^{-} \sqsubseteq shareInterest \\
retweet \circ tweet^{-} \sqsubseteq follow \\
(David, SemanticWeb) : interest \\
(Alex, SemanticWeb) : interest \\
Alex \neq David
\end{array}
$$

$$
\begin{array}{l}
(David, Post_3) : tweet \\
(Alex, Post_3) : retweet \\
(David, Post_4) : tweet
\end{array}
$$

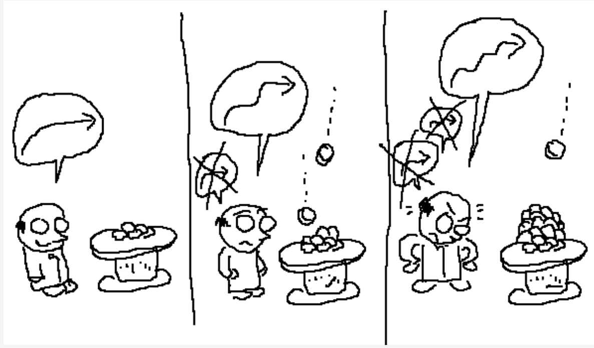


Can We Learn from Existing Work?

- The DRed (Delete and Re-derive) approach [Volz et. al. 2005]
  - Maintaining the materialisation of the knowledge base
  - Over-delete impacted entailments
  - Re-derive impacted entailments
  - Derive new entailments
- Key techniques
  - Delete: justification
  - Re-derive: incremental reasoning

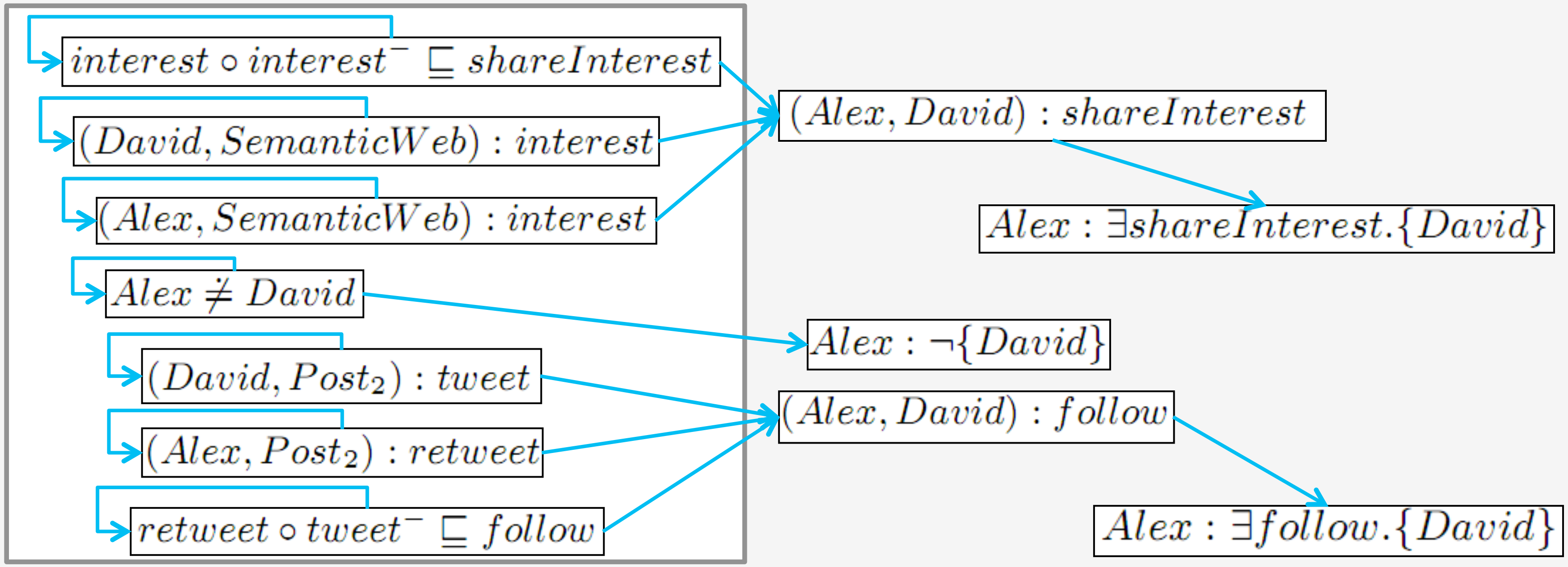
Justification: Key Enabler for Delete

- Justification
  - Given an ontology O and a reasoning result rs
  - A justification J(rs) is a minimal subset of O that imply rs
  - There could be multiple justifications
- Challenges:
  - Computing one justification for OWL2-DL is costly
  - Computing all justifications is NP-complete even for OWL2 tractable profiles
- One justification at a time is needed
  - If the current justification J(rs) and Er(i) overlap
  - then rs should be removed as well



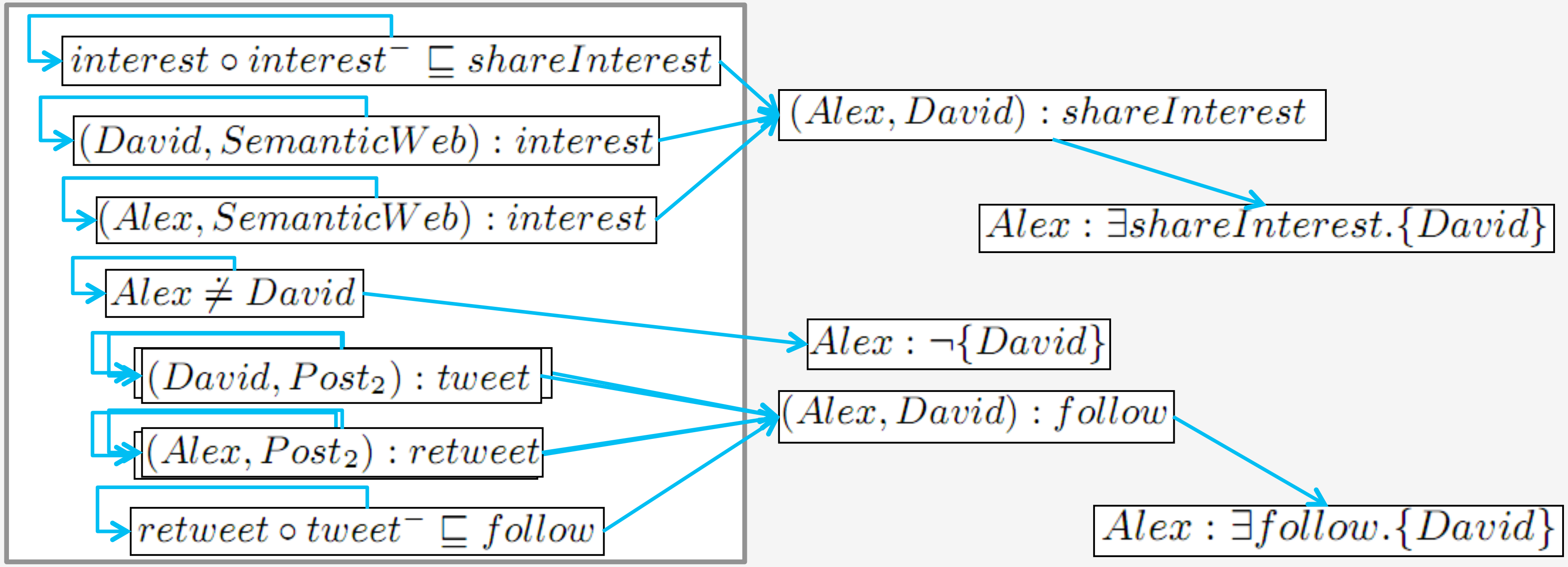
Truth Maintenance System

- A directed graph
  - Nodes: axioms / entailments
  - Edges: derivation relations among axioms / entailments
  - All entailments are reachable from their justifications
    - Easy to identify impacted entailments



Delete and Re-derive with TMS

- Erasing
  - Remove all nodes reachable from the erased axioms
  - Removing all corresponding edges
- Adding
  - Adding added axioms as new nodes into the graph
  - Inferring new results
  - Establishing new edges



Key Challenges

- Q1: How to efficiently perform reasoning and compute justifications?
- - Q1.1 tractable profiles such as OWL2 EL
- - Q1.2 OWL2 DL

- Q2: How to perform incremental reasoning
  - based on justifications

Forward chaining reasoning in OWL 2 EL

- Normalise ontology axioms to standard form:

  $A \sqsubseteq B \quad A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C$

- Saturate using inference rules:

$$\frac{A \sqsubseteq B \quad B \sqsubseteq C}{A \sqsubseteq C} \qquad \frac{A \sqsubseteq B \quad A \sqsubseteq C \quad B \sqcap C \sqsubseteq D}{A \sqsubseteq D}$$

$$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$$

- Applicable approaches
  - data driven forward chaining
  - query driven forward chaining

Stream Reasoning in OWL 2 EL [Ren and Pan, 2011]

- Optimised memory consumption
  - **Reduce** the number of maintained nodes and edges

- We entail an axiom $C \sqsubseteq \exists r.D$ if it is **classification-relevant**, i.e. contributing to the reasoning results we are looking for:
  - E.g. there is some $\exists r.A \sqsubseteq B$
  - or, $r \sqsubseteq s$ and *s* is classification-relevant
  - or, …

$$
\begin{tikzcd}
\boxed{ActiveTalk \sqsubseteq \exists in.Session} \arrow[dr] & \\
& \boxed{\{talk1\} \sqsubseteq \exists in.Session} \\
\boxed{\{talk1\} \sqsubseteq ActiveTalk} \arrow[ur] &
\end{tikzcd}
$$

in is **NOT** classification-relevant

We do **NOT** need to compute or maintain the result

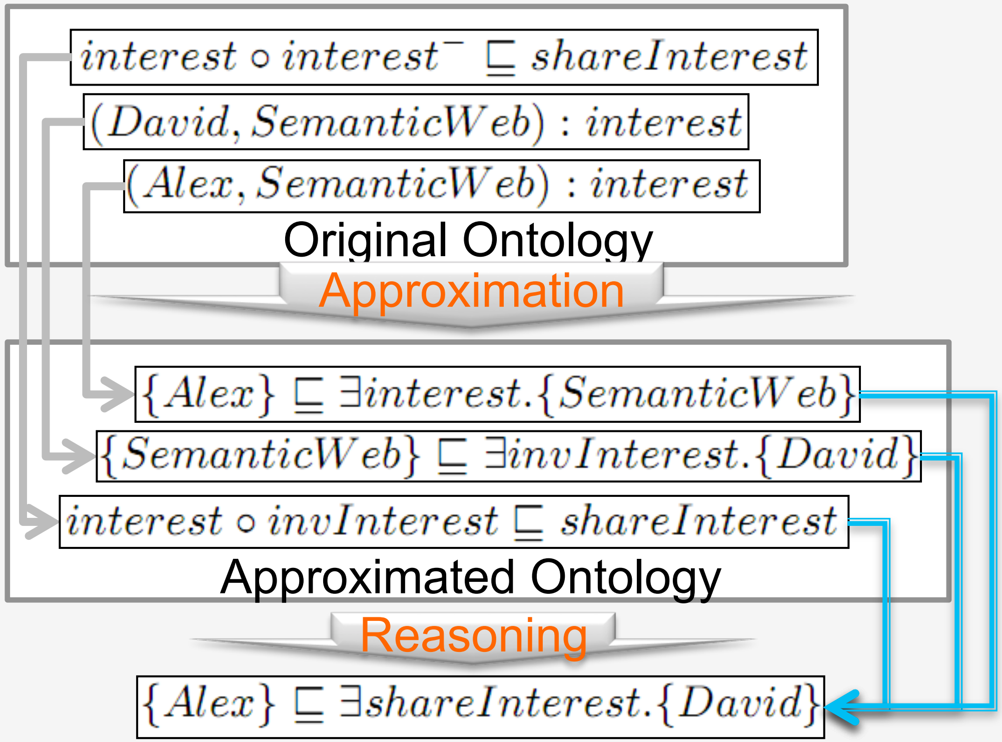
Stream Reasoning in OWL 2 QL

ISWC 2013 Sydney, Australia

- Reasoning in OWL 2 QL is based on query rewriting
- Given ontology O and query Q, use O to rewrite Q as Q^0 s.t., for any set of ground facts A:
  - ans(Q, O, A) = ans(Q^0, ;, A)
- Some related new results:
  - Incremental query rewriting [Venetis et. al. 2012]
  - Temporal DL-Lite [Borgwardt et. al., 2013]
  - DL-Lite with aggregate queries [Kostylev and Reutter, 2013]
- Still be room for further research

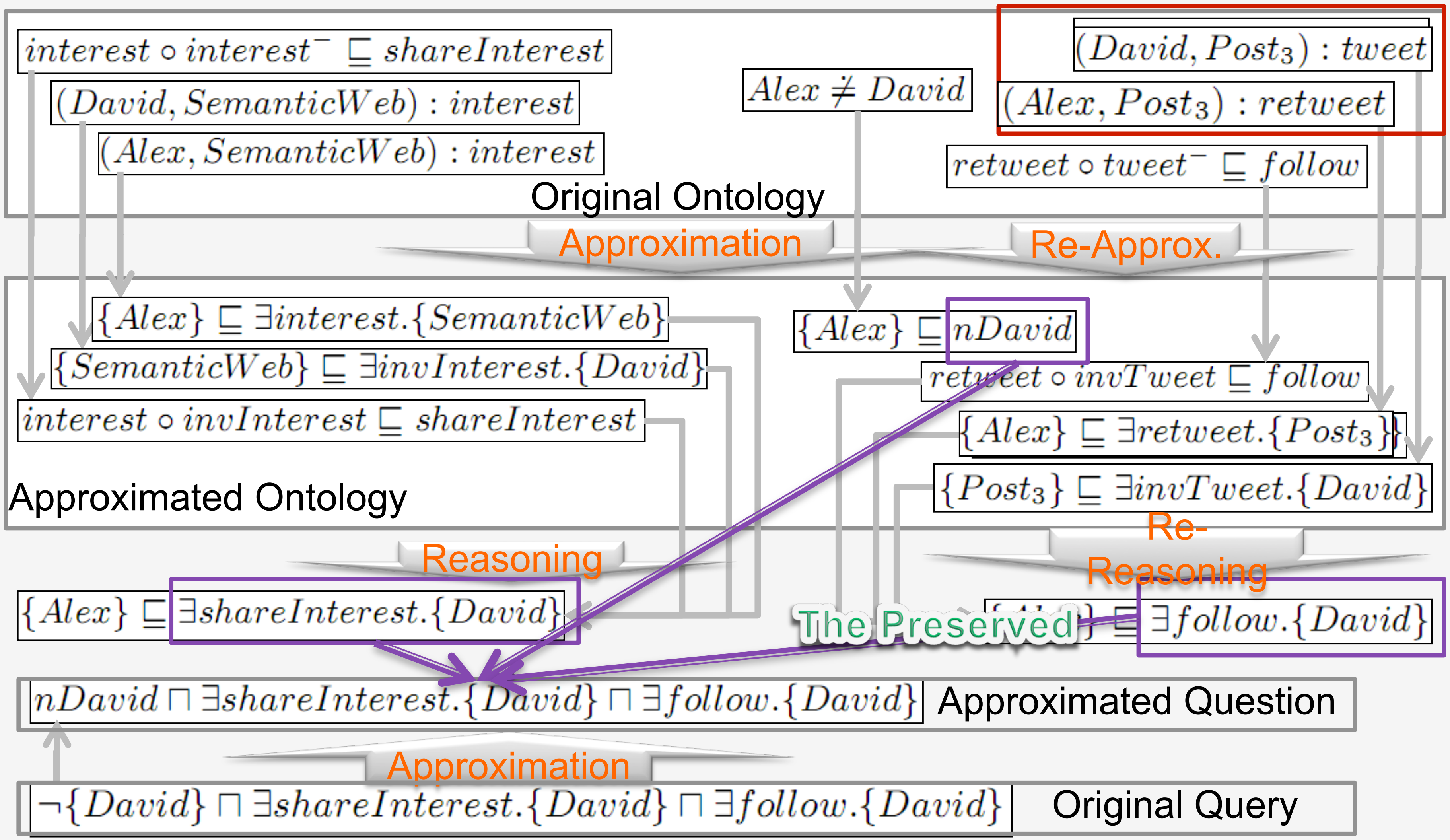
Stream Reasoning in OWL 2 DL

- Generate a TMS when doing approximation and reasoning
  - Nodes:
    - Asserted axioms;
    - Approximated axioms;
    - Entailed axioms;
  - Edges:
    - Created during approximation and reasoning



Stream Reasoning with Syntactic Approximated-TMS

ISWC 2013 Sydney, Australia



Stream Reasoning with Syntactic Approximated-TMS

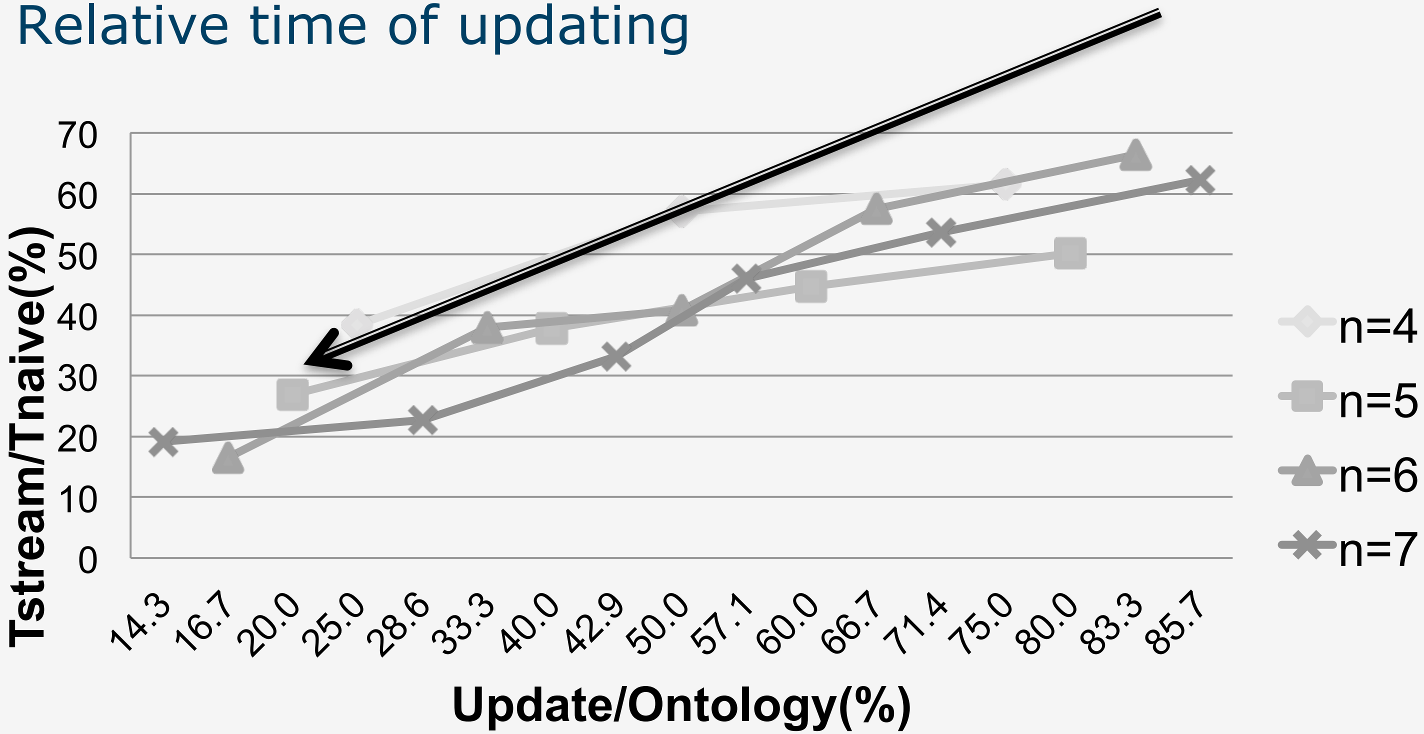
- Improvement:
  - The approximation can be reused when adding new axioms
  - Reasoning complexity is reduced
- Quality:
  - Soundness-guaranteed
    - Entailed axioms will never be over-looked in erasing

Evaluation

- LUBM
  - Arbitrarily large ABox
  - Non-trivial reasoning
    - Different possible sources of a same entailment
- Stream simulation
  - Generating a LUBM ontology
  - Preserve the TBox through the stream
    - To ensure the difficulty of reasoning
  - ABox stream
    - Partitioning the ABox
    - Swap sub-ABoxes in stream updating

Evaluation

- Criteria
  - Absolute volume of the stream
  - Relative volume of the update
  - Absolute time of updating
  - Relative time of updating



A brief summary

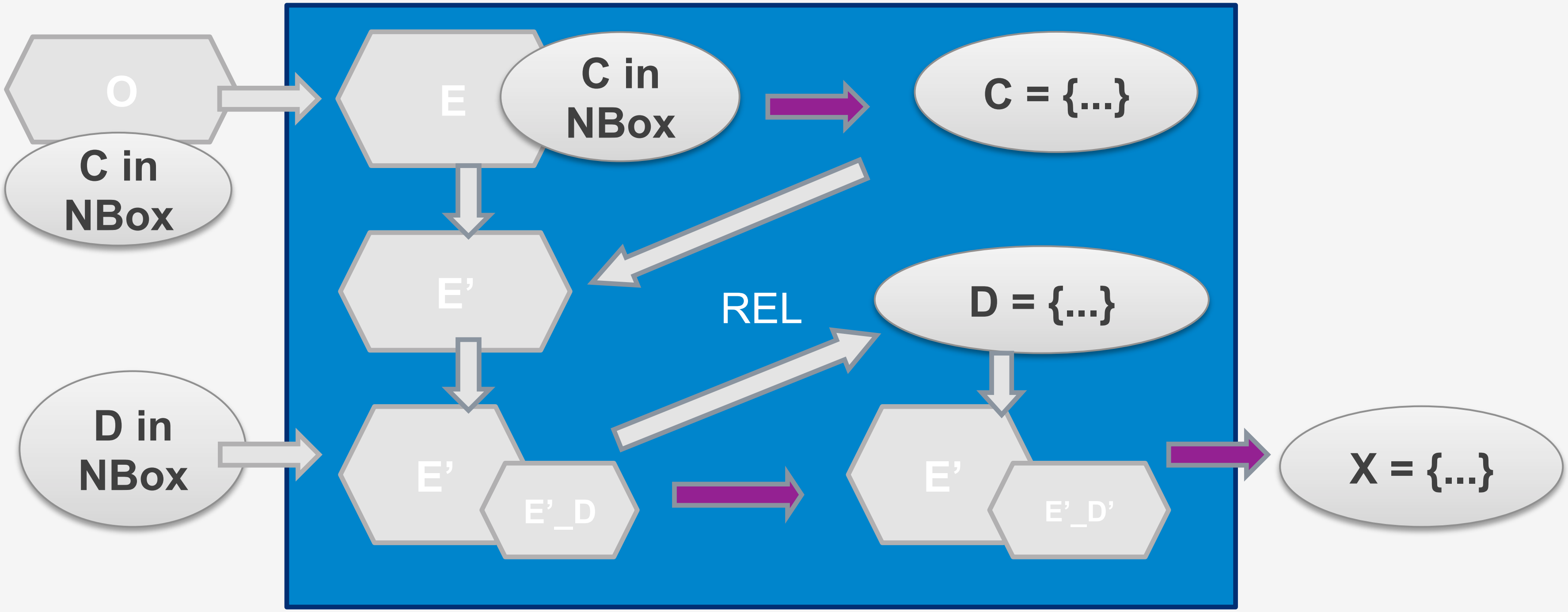
ISWC 2013 Sydney, Australia

- Advantage
  - Significantly reduce the needed number of entailment checking
- Disadvantages & optimisations to address them
  - Require additional computation to form the edges
    - Using saturation-based algorithms to generate the TMS on-the-fly
      - EL+, Horn-SHIQ vs. Tableau-based or Datalog-based algorithms
  - Significantly increase the memory consumption
    - Saturation algorithms already make intermediate results more "reusable"
    - Optimising the algorithm to further minimise intermediate results
  - High complexity in expressive languages
    - approximation

Combining stream reasoning with local closed world reasoning

ISWC 2013 Sydney, Australia

- Useful in deployment lifecycle
  - deployed components should be closed (in principle)
  - undeployed components remain open
- Non-monotonic reasoning with stream



Local closed world reasoning in NBox

- Motivation:
  - There are knowledge and data that users have complete knowledge about
  - such as spicy dishes
- Solution: NBox (Negation as failure Box) [JTST2010]
  - TBox: a set of schema axioms
  - ABox: a set of data axioms
  - NBox: a set of closed concepts and roles
    - Declarative: Permanently closed, with annotation property
    - Runtime: temporally closed, with API provided

Food	Note
Curry Chicken	Minor Spicy
Salmon Fillet	
Spicy Grilled Shrimp	Spicy
Pepper Salad	Vege

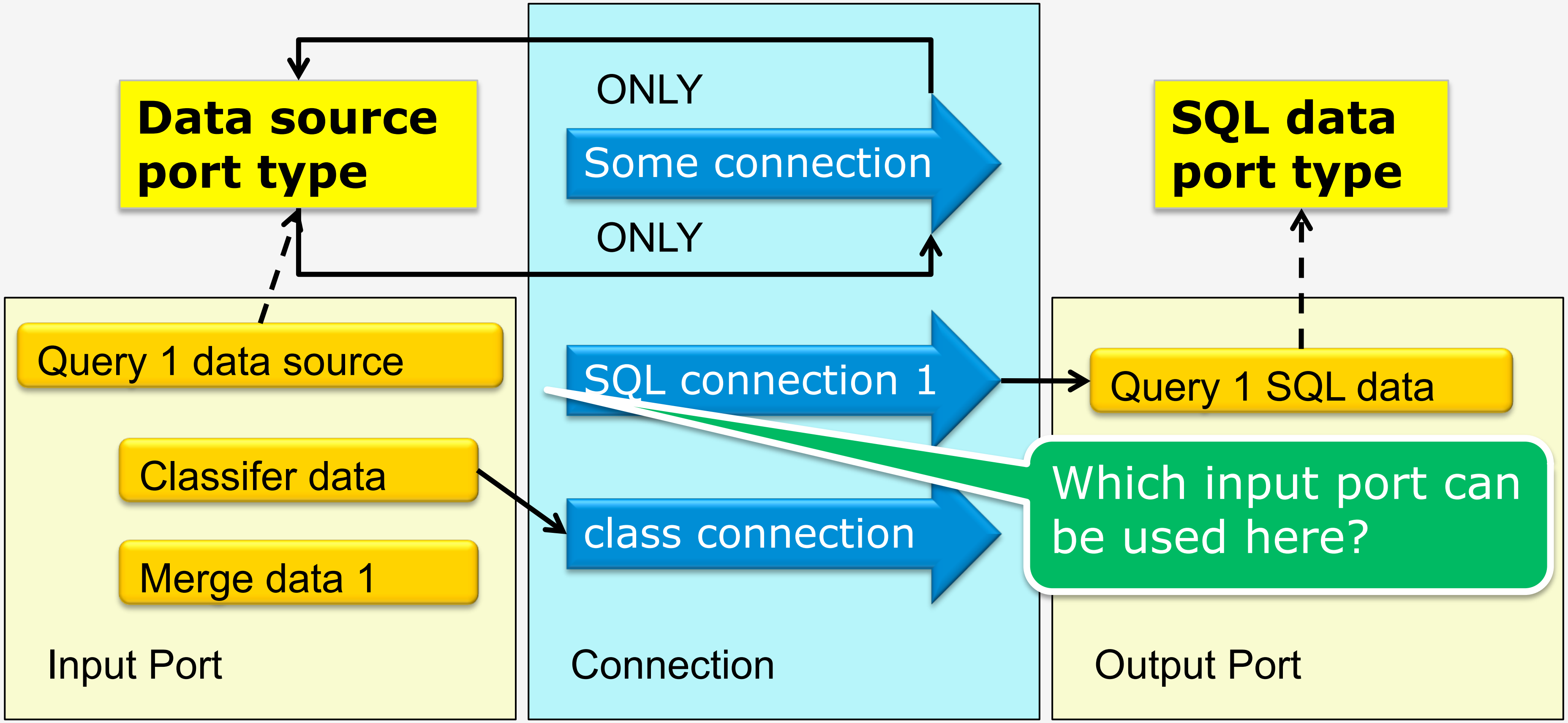
Name	Vegetarian
Jeff	No
Yuting	Yes
Jek	
Yuan	No

Example: NBox Reasoning

ISWC 2013 Sydney, Australia

MOSTPROJECT http://most-project.eu

- Local closed world reasoning in DMIL

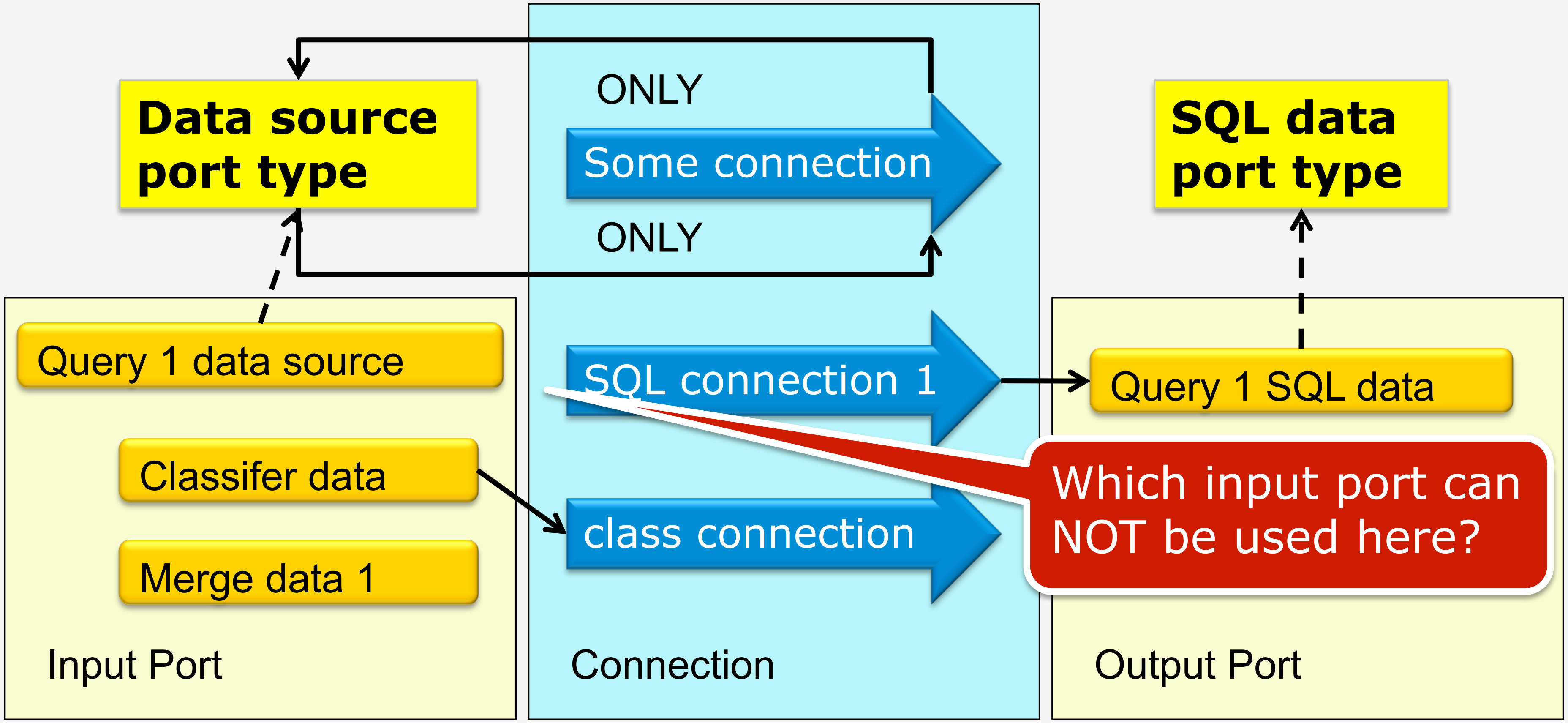


Example: NBox Reasoning

ISWC 2013 Sydney, Australia

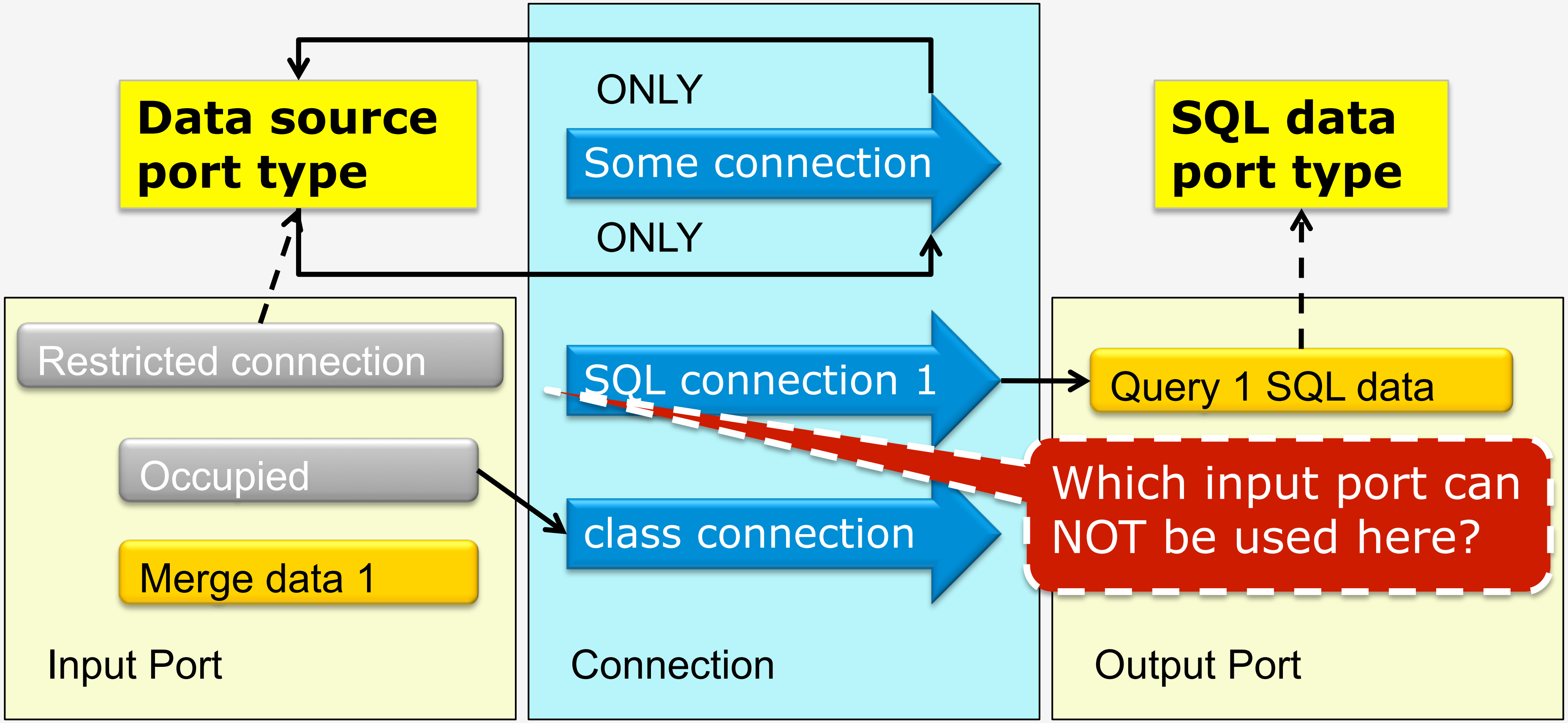
- Local closed world reasoning in DMIL

MOSTPROJECT http://most-project.eu



Example: NBox Reasoning

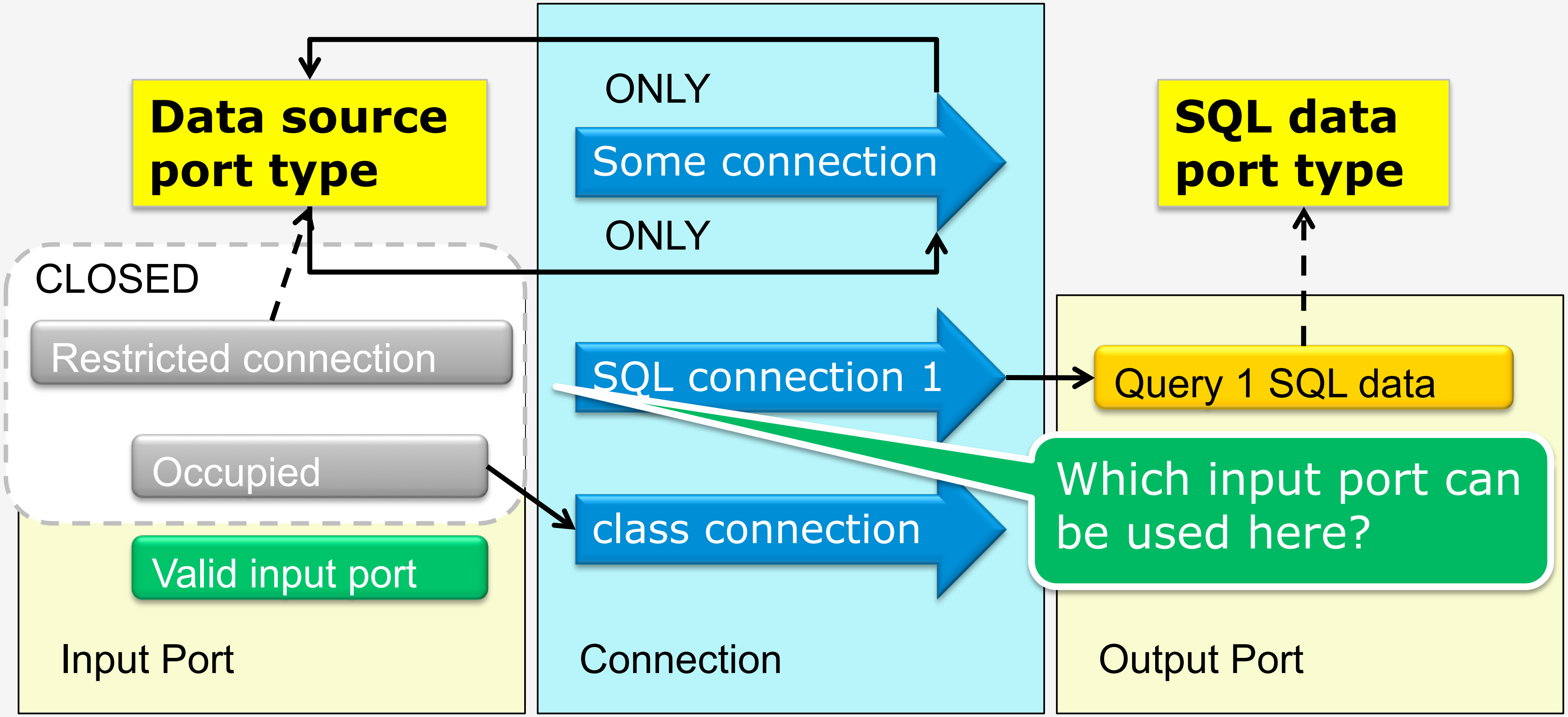
- Local closed world reasoning in DMIL



Example: NBox Reasoning

- Local closed world reasoning in DMIL

MOSTPROJECT http://most-project.eu



A brief note on NBox

- Adding NBox into OWL DL/OWL 2 DL won't increase the complexity for reasoning
  - some approximate reasoning algorithm is available in implemented in TrOWL [Ren et. al. 2010]
- However, adding NBox would make DL-Lite and EL intractable in general
  - unless some safe conditions are satisfied [Lutz et. al. 2012]

Research Challenges

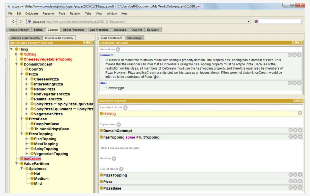
- Dealing with noisy and uncertain data
- Stream LOD reasoning in other OWL2 profiles
  - Such as QL and RL
- Further optimisations techniques
  - Such as decomposition and parallelisation
- Other notions of stream reasoning
- Relations between stream reasoning and temporal reasoning

Take Home Message

Approximate reasoning and TMS play key roles in OWL 2 DL stream reasoning

Where to find more information

TrOWL (with publication list and tutorials):
http://trowl.eu/



www.trowl.eu

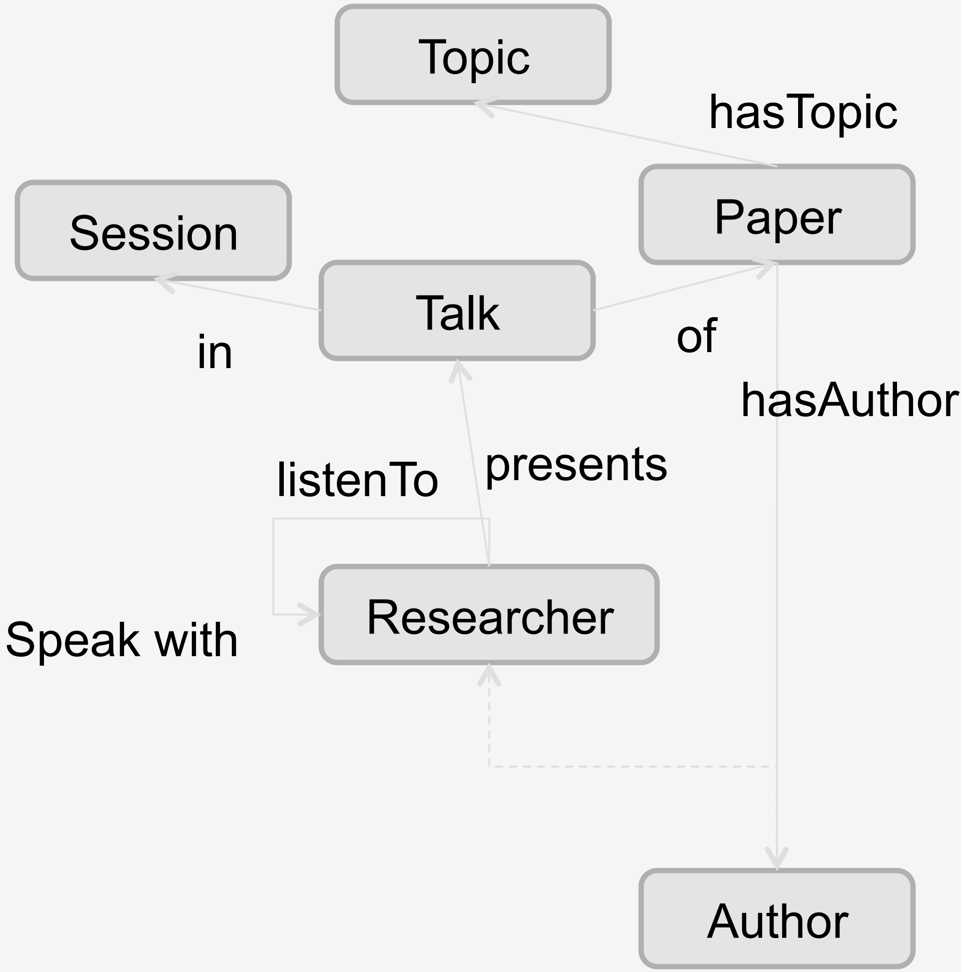
http://trowl.eu/owl-dbc/

Agenda

1. OWL Reasoning with LOD (30m)
2. OWL Stream Reasoning with LOD (40m)
3. Hands-on session (20m)

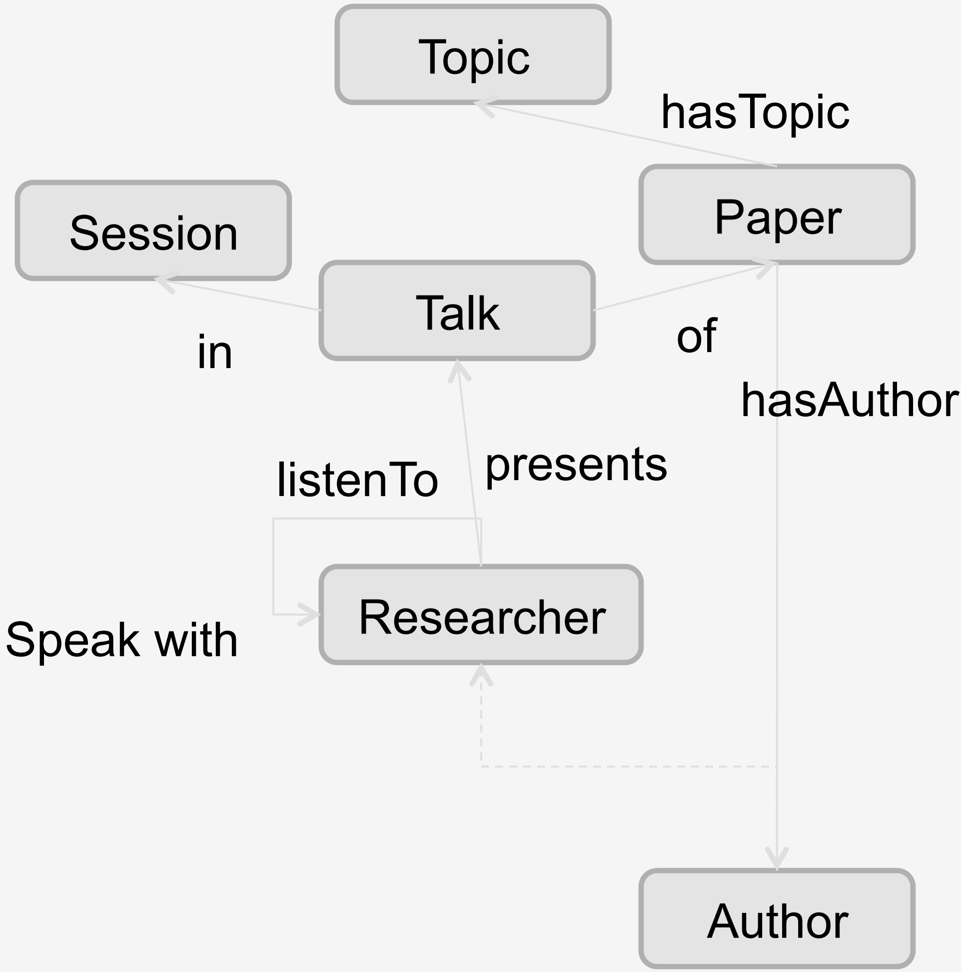
Scenario

- Researchers present their works in conference sessions.
  - "Introducing HermiT2: an NLogSpace Reasoner of OWL2" – by Ian Horrocks
  - @ Session of Reasoning
- Researchers activities can be collected from mobile sensors (e.g. RFID) or web crawlers.
  - @Jeff: "Ian just gave an very interesting presentation about his new paper"
  - @Yuan: "I just had a discussion with Jeff"



Ontology

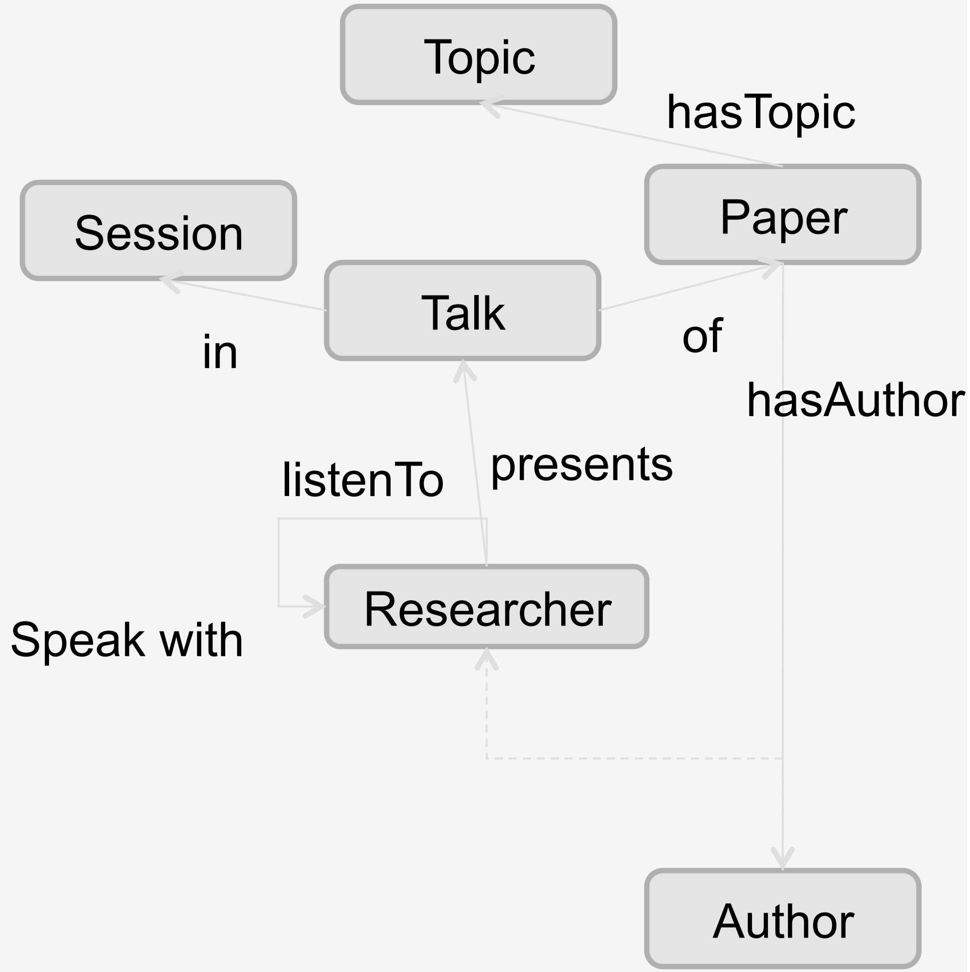
- Presenters attend sessions:
  - presents o in -> attend
- Listeners attend sessions:
  - listenTo o in -> attend
- Discussions happens in sessions:
  - speakWith o attend -> attend



Ontology (cont.)

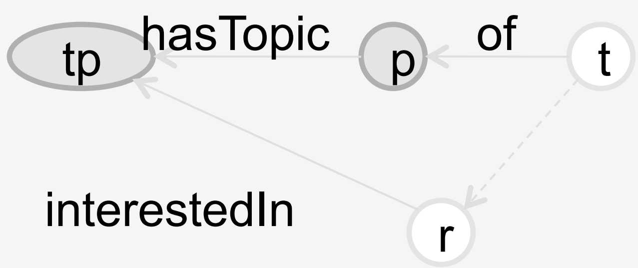
ISWC 2013 Sydney, Australia

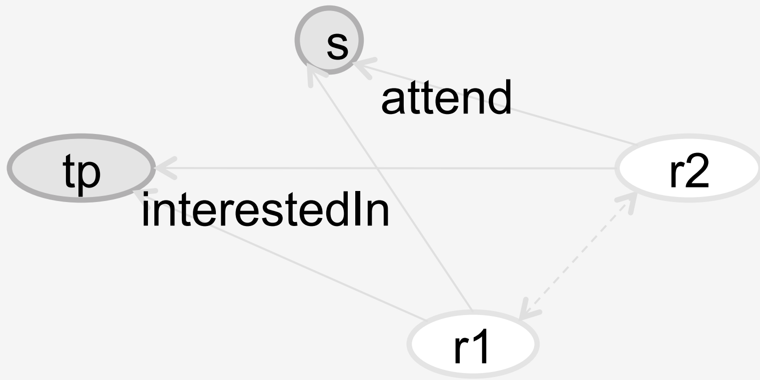
- Authors are interested in the topics
  - Inv(hasAuthor) o hasTopic -> interestedIn
- Listeners are interested in the topics
  - listenTo o of o hasTopic -> interestedIn



Query Examples

- Recommending
  - Talks of paper with interesting topic
    - t,r:- (t,p):of, (p,tp):hasTopic, (r,tp):interestedIn.
  - Researchers in the same session with same interest
    - r1,r2:- (r1,tp):interestedIn, (r2,tp):interestedIn, (r1,s):attend, (r2,s):attend.





Why Reasoning Matters

- Inference
  - Researcher activity information
    - The attend relation chain
  - Event information
    - Which ongoing talk is happening at what time in which venue given by whom?
- Dynamics
  - Researcher activities are happening in real time
  - Events are happening in real time

Setting Up (option 1)

ISWC 2013 Sydney, Australia

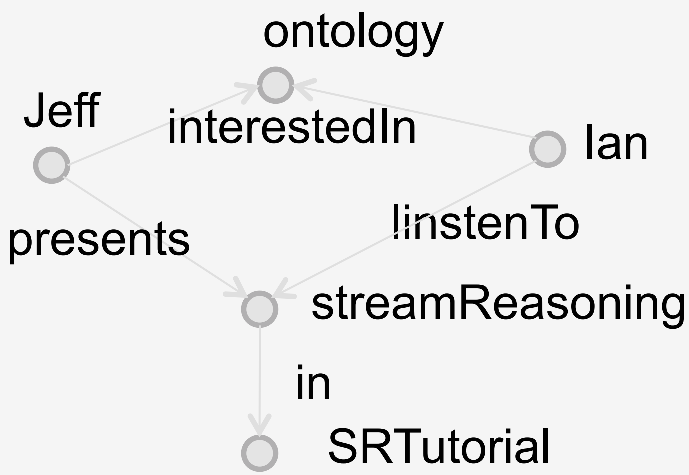
- We use Eclipse IDE.
  - Put the SRTutorial folder into your Eclipse workspace
  - Create a Java project called SRTutorial in Eclipse. Eclipse will automatically use the source files inside the folder
  - Make sure the two jar files in the \lib folder are included in the build path of the project
- TBox: SR.owl;
- ABox: SRt0-SRt2.owl;
- Run the example
  - demo/Example/SRTutorialExample.java runs the example
  - demo/Example/SRTutorialJustifiedExample.java runs the example with explanations for each answer

Setting Up: For Those Who Have No IDE

- Run the example directly with the pre-compiled jar files
  - SRTutorialExample.jar runs the example
    - Execute "java –jar SRTutorialExample.jar" in console
  - SRTutorialJustifiedExample.jar runs the example with explanations for each answer
    - Execute "java –jar SRTutorialJustifiedExample.jar" in console

Example

- t0:
  - (Jeff, streamReasoning):presents,
  - (Ian, streamReasoning):listenTo,
  - (streamReasoning, SRTutorial):in,
  - (Jeff, ontology):interestedIn,
  - (Ian, ontology):interestedIn

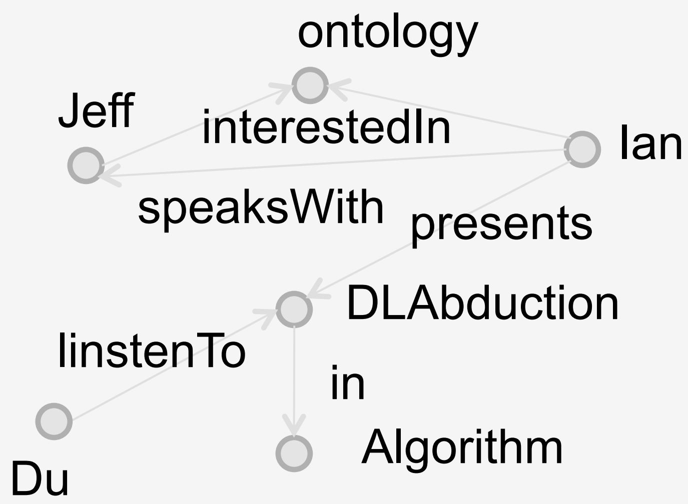


- Query: who are active participants that interested in ontology?
  - Jeff
  - Ian

- They both attend the SRTutorial session and interested in ontology

Example

- t1:
  - (Jeff, ontology):interestedIn,
  - (Ian, ontology):interestedIn
  - (Ian, DLAbduction):presents,
  - (Du, DLAbduction):listenTo,
  - (DLAbduction, Algorithm):in,
  - (Ian, Jeff):speaksWith,

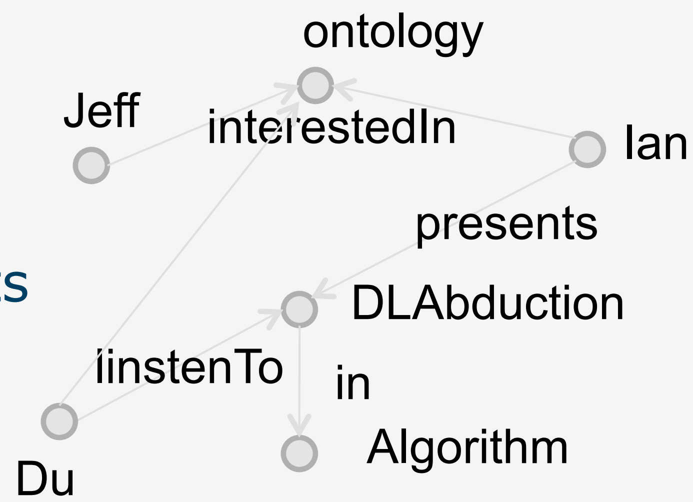


- Query: who are active participants that interested in ontology?
  - Jeff
  - Ian

- They both attend the Algorithm session and interested in ontology

Example

- t2:
  - (Jeff, ontology):interestedIn,
  - (Ian, ontology):interestedIn
  - (Ian, DLAbduction):presents,
  - (Du, DLAbduction):listenTo,
  - (DLAbduction, Algorithm):in,
  - (Du, ontology):interested



- Query: who are active participants that interested in ontology?
  - Ian
  - Du

- Du becomes interested in ontology
- Jeff needs to attend a meeting

Stream Reasoning For Linked Data

M. Balduini, J-P Calbimonte, O. Corcho, D. Dell'Aglio, E. Della Valle, and J.Z. Pan

http://streamreasoning.org/sr4ld2013

ISWC 2013
Sydney, Australia

OWL Reasoning and Stream Reasoning with LOD

Jeff Z. Pan
University of Aberdeen
http://homepages.abdn.ac.uk/jeff.z.pan/pages/