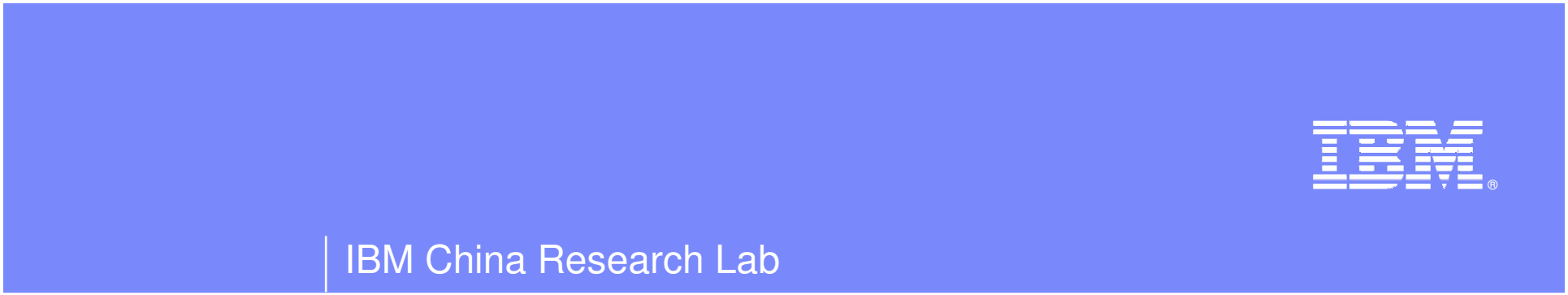


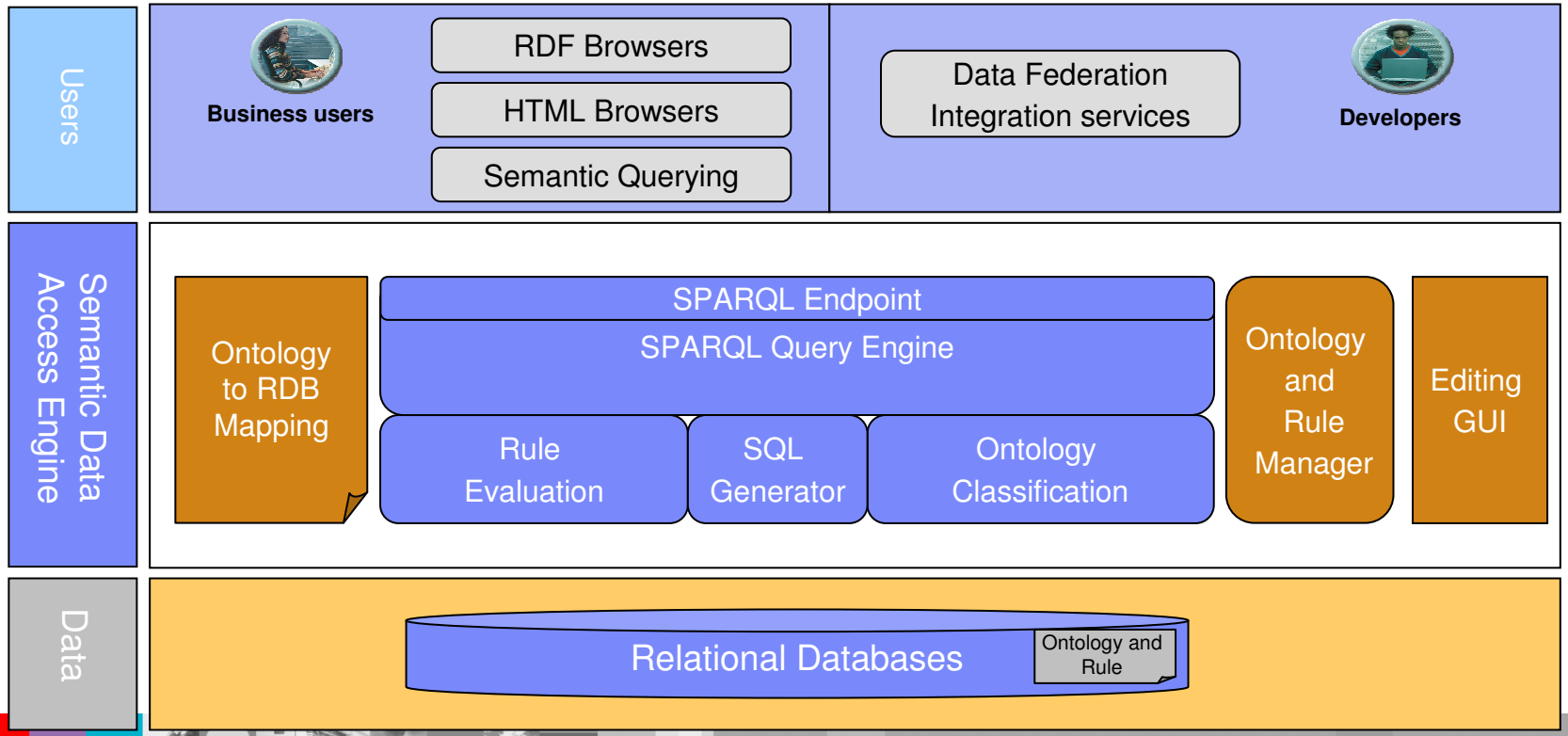
IBM China Research Lab

Semantic Data Access to Relational Databases

Presenter: Li Ma (malli@cn.ibm.com)

Semantic Data Access (SeDA) Engine

- SeDA provides enabling technologies for exposing relational data as virtual RDF graphs, drives semantic federation and integration and realizes the linked data vision of semantic web.
- Applications and Status
  - Status: SeDA v0.2 is available at IBM community sources and will be released at IBM alphaworks around Oct.
  - SeDA is a core component for Metadata Web and Semantic Master Data Management projects.
- Technical Objectives
  - Develop high-performance and scalable RDF query rewriting techniques for the generation of virtual RDF views over existing relational data
  - Develop mapping based ontology and rule reasoning for semantic querying over existing relational data

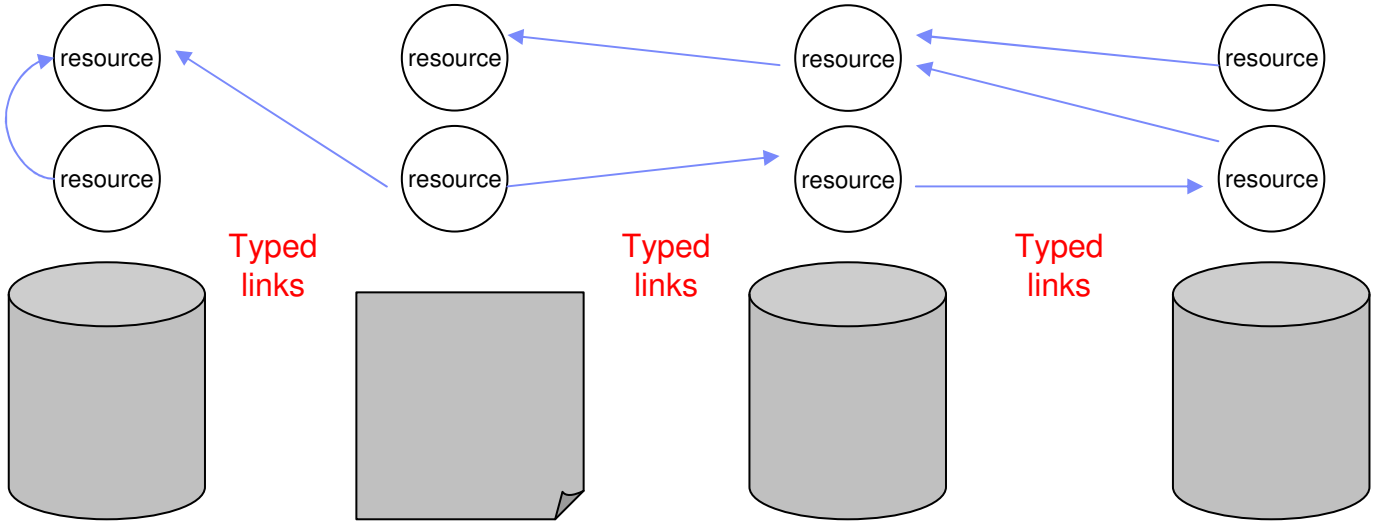


Outline

- Value Proposition of SeDA
  - Driving information federation and integration
  - Semantic querying
- Technical Solutions
- Mapping Generation from Object Models
- Semantic Web Tools

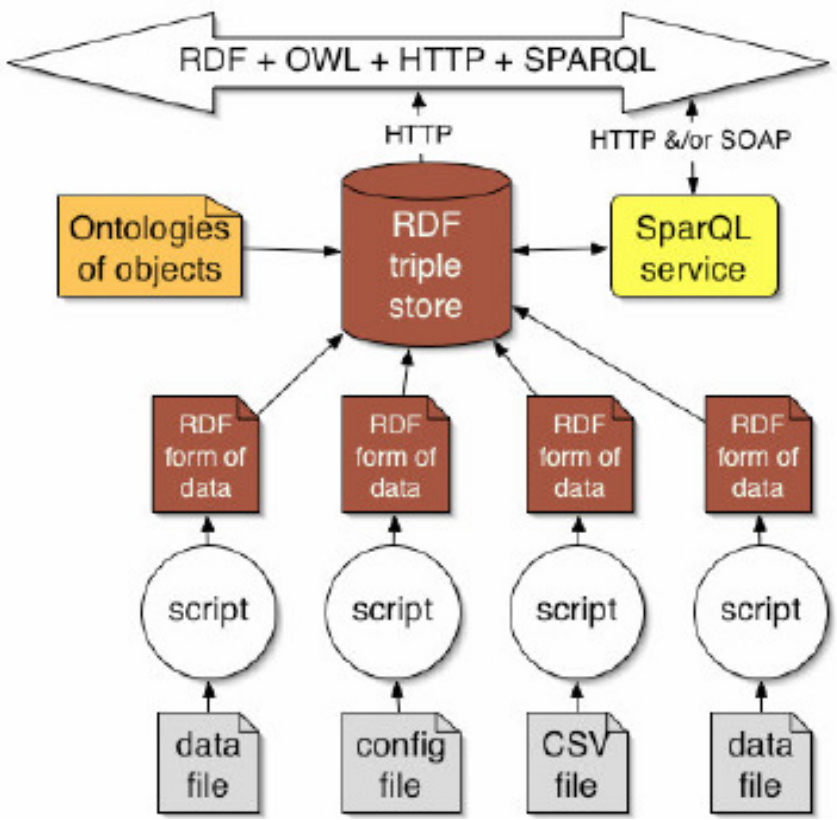
Linked Data: A Global Database at Semantic Web

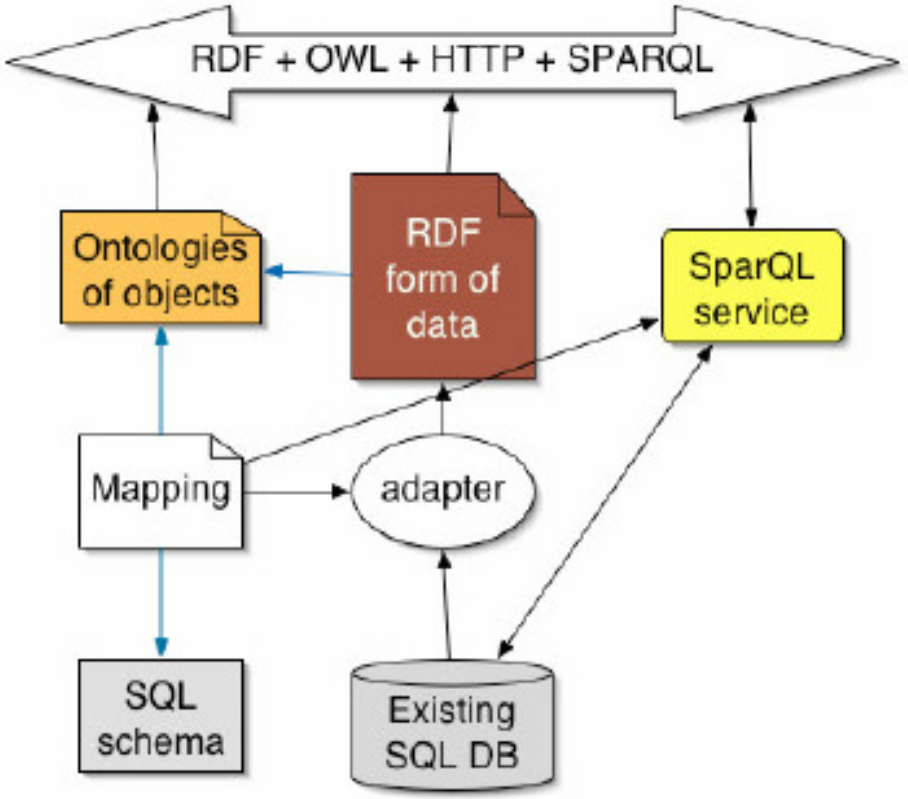
- A global database
  - Various resources (including documents) are linked together
  - Degree of structure in resources is high
  - Semantics of content and links are explicit
  - Designed for machines first, humans later
- Use URIs as names for resources
- Use HTTP URIs so that people can look up those names
- When someone looks up a URI, provide useful RDF information
- Include RDF statements that link to other URIs so that they can discover related things



Reference: Christian Bizer et al., Linked data presentation

The Data on Semantic Web





Sources: T.B. Lee's presentation

Semantic Querying over Relational Databases

Company info.

ID	Name	Location	Business1	Business2
1	BAR	Bei Jing	Memory	Wireless software
2	FOO	Paris	Optical comm.	Wireless comm.
3	ROL	New York	Banking Solut.	NULL
4	EDOX	New York	Memory	Main Board
5	GUC	Vancouver	NULL	NULL

Shareholding

ID	Name	Shareholders1	Shareholders2
1	BAR	FOO	TIT
2	FOO	GUC	Null
3	ROL	BAR	TIT
4	EDOX	ROL	Null

Semantic query

1. Find Company EDOX's all direct and indirect shareholders who are from Europe and are IT company.

2. A software company that has products about wireless telecom and is held by a Canada company

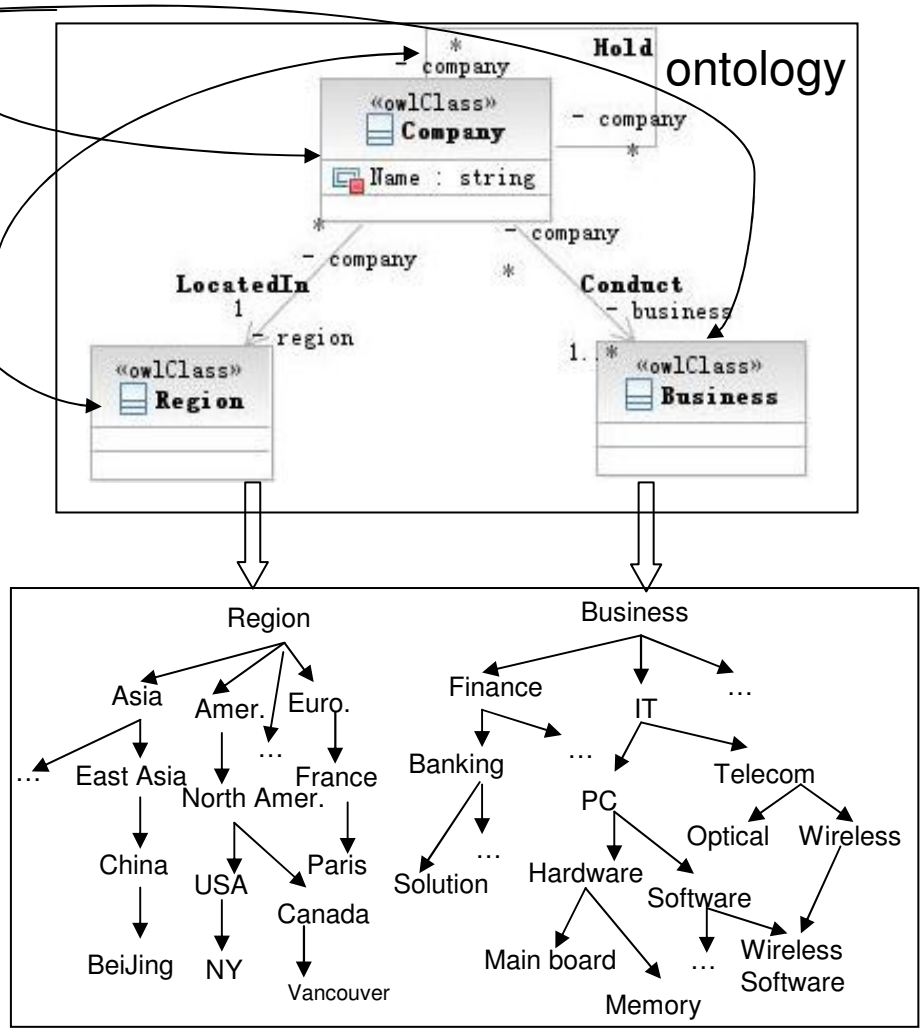
Semantic Querying over Relational Databases

Company info.

ID	Name	Location	Business1	Business2
1	BAR	Bei Jing	Memory	Wireless software
2	FOO	Paris	Optical comm.	Wireless comm.
3	ROL	New York	Banking Solut.	NULL
4	EDOX	New York	Memory	Main Board
5	GUC	Vancouver	NULL	NULL

Shareholding

ID	Name	Shareholders1	Shareholders2
1	BAR	FOO	TIT
2	FOO	GUC	Null
3	ROL	BAR	TIT
4	EDOX	ROL	Null



Semantic query

1. Find Company EDOX's all direct and indirect shareholders who are from Europe and are IT company.

2. A software company that has products about wireless telecom and is held by a Canada company

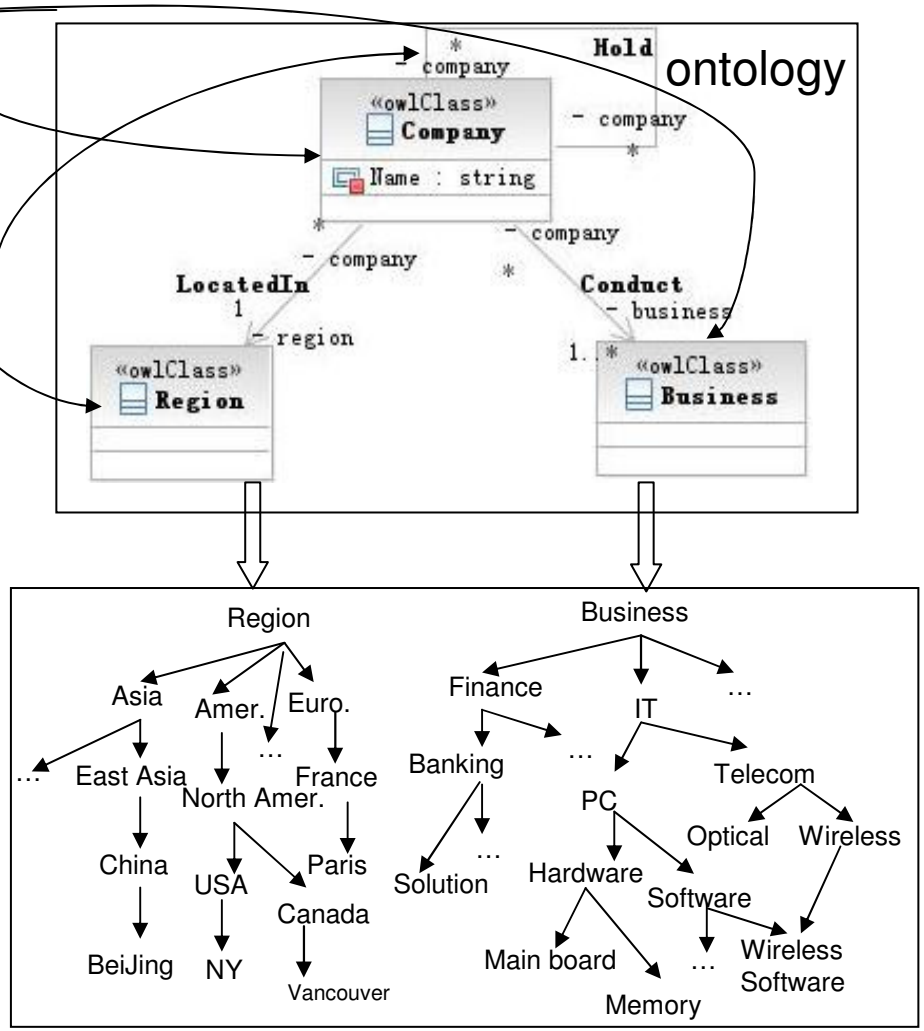
Semantic Querying over Relational Databases

Company info.

ID	Name	Location	Business1	Business2
1	BAR	Bei Jing	Memory	Wireless software
2	FOO	Paris	Optical comm.	Wireless comm.
3	ROL	New York	Banking Solut.	NULL
4	EDOX	New York	Memory	Main Board
5	GUC	Vancouver	NULL	NULL

Shareholding

ID	Name	Shareholders1	Shareholders2
1	BAR	FOO	TIT
2	FOO	GUC	Null
3	ROL	BAR	TIT
4	EDOX	ROL	Null



Semantic query

1. Find Company EDOX's all direct and indirect shareholders who are from Europe and are IT company.
FOO is retrieved using transitive closure and subsumption inference.

2. A software company that has products about wireless telecom and is held by a Canada company
BAR is retrieved using classification and subsumption inference

Data Management and Semantic Web

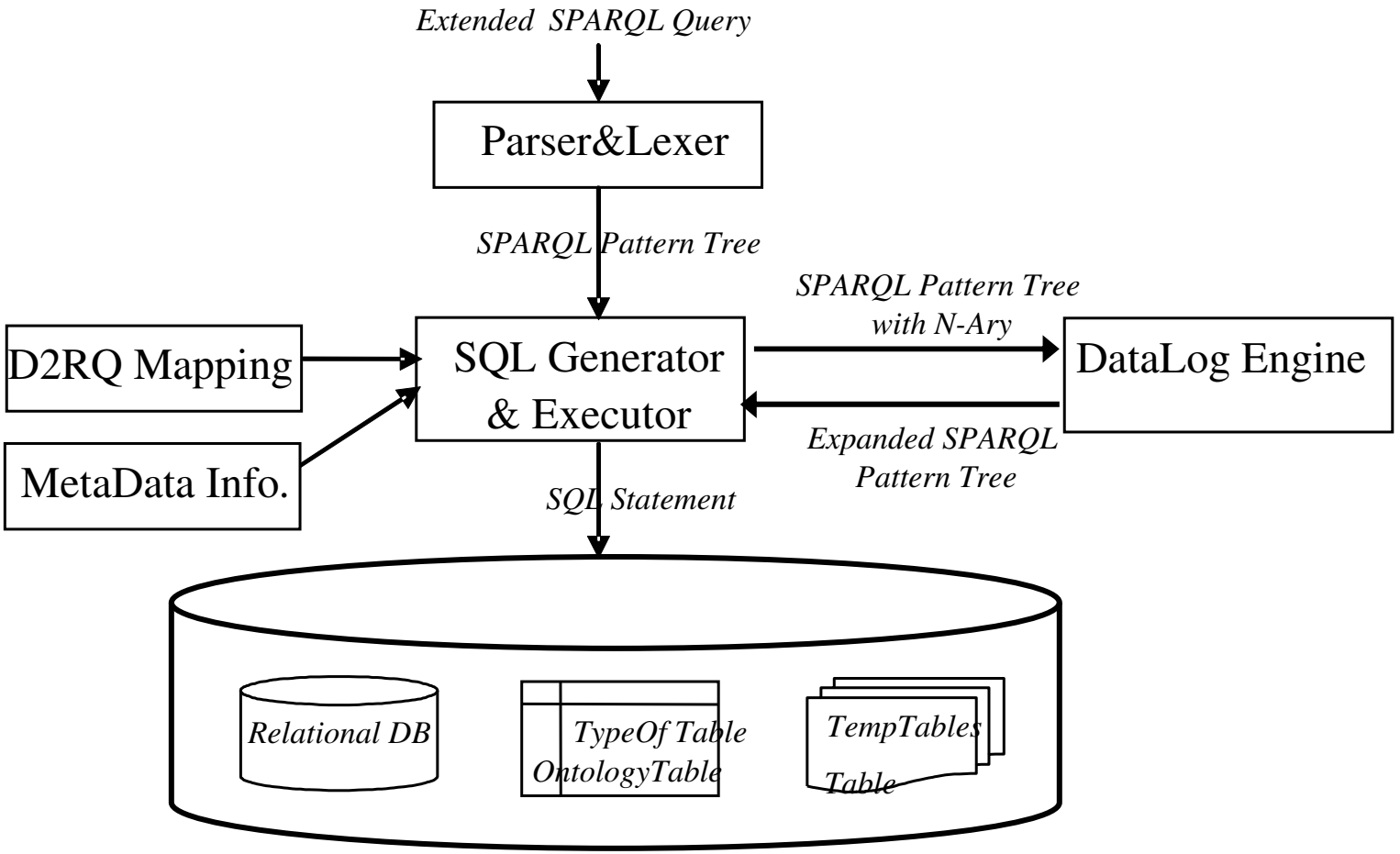
Main Motivations are in capturing Data Semantics, achieving Data Integration and Reasoning

- RDF and OWL ontologies are good in capturing data semantics
  - Can be used to define a “semantic” model of the underlying relational data that can be tailored to different domains or applications, and that hides the actual layout of data across different tables
- Allow use of additional domain knowledge in OWL ontologies while answering queries to the relational DB
- Allow use of DL reasoning while answering queries to the relational DB to improve recall
- Allow Semantic Web applications (that use an RDF/OWL data model) to have access to relational data, without having to deal with a different data model
- Virtual RDF views facilitates data federation and integration

Major Challenges

- Ontology, rule and mapping
  - Expressive mapping: semantically complete
  - Easy to use semantic modeling: more choices to users
- Query processing and reasoning
  - Effective integration of reasoning with query processing. It is impractical to materialize all inferred results in advance, which causes the data synchronization problem in an operational store.
  - Efficient SPARQL-to-SQL query rewriting based on the ontology mapping
  - Scalable query evaluation by leveraging well-developed database optimizations for query processing

Query Evaluation with Reasoning



SPARQL Query Expansion

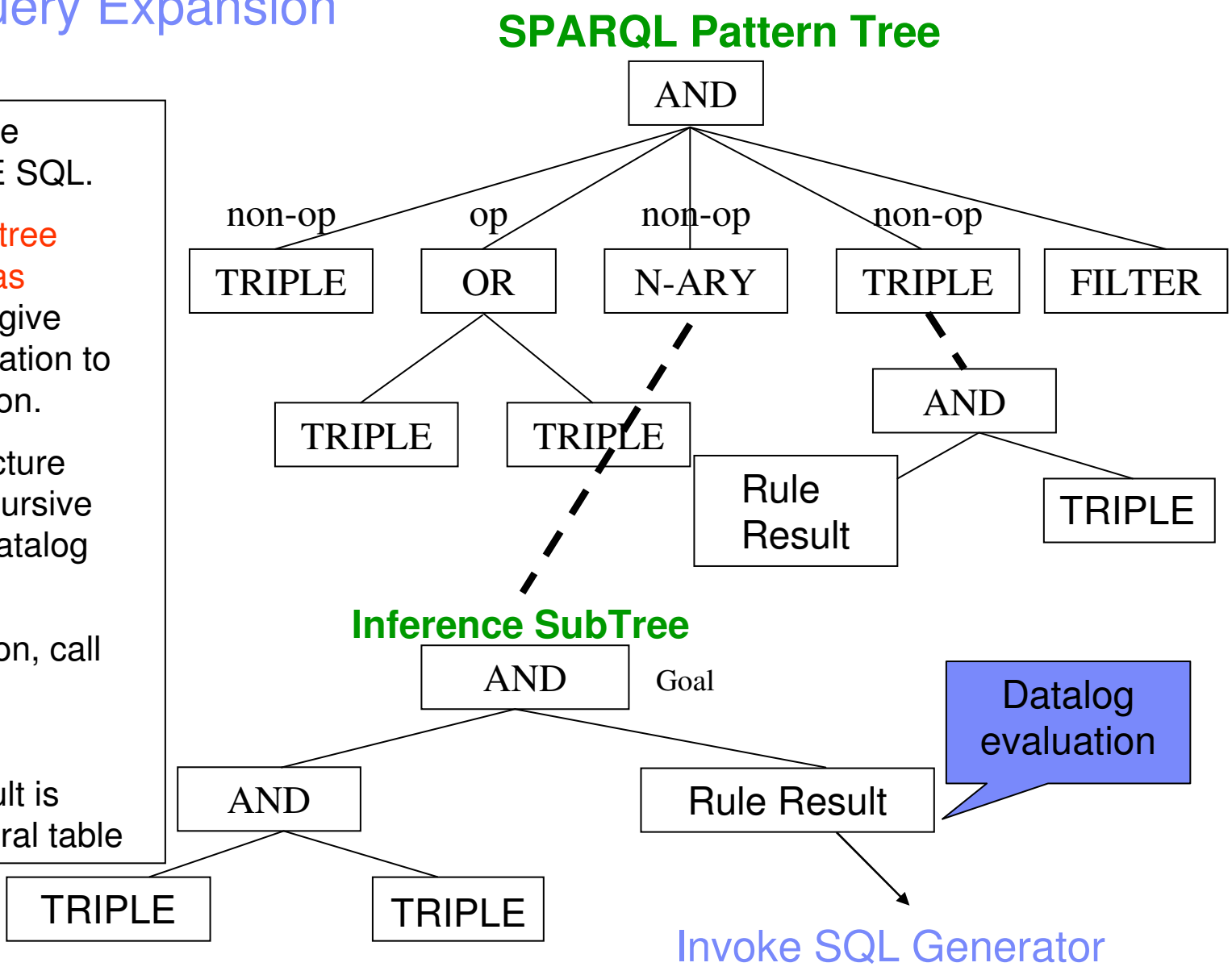
SPARQL tree will be translated into ONE SQL.

Key idea: Maintain tree structure as much as possible! This is to give DBMS more information to do query optimization.

When the tree structure cannot be kept (recursive rules), trigger the datalog evaluation.

During the evaluation, call the SQL generator repeatedly.

The evaluation result is stored in the temporal table



An Example of Rules and Query

- /* Use rules to define that sub-contract property is transitive */
- Sub_Contract(?x,?y):-
  WCC:CONTRACT_Relationship_FROM(?u, ?x),
  WCC:CONTRACT_Relationship_TO(?u, ?y),
  WCC:hasContract_Relationship_Type(?u, ?z),
  WCC:Contract_Relationship_FROM_TO_NAME(?z, 'Sub Agreement');;.

- Sub_Contract(?x,?y):-Sub_Contract(?x,?z),Sub_Contract(?z,?y);;.

Query:

Find the party that is the owner of the "costly" contract

```
Select ?pty
Where {
    Contract_NumOfLClms(?contract ?numofClaims).
    Contract_Owner(?pty ?contract).
    FILTER (?numofClaims >2) }
```

- /* The relationship Large_Claim_Agreement indicates that a contract has a claim with the amount more than $5000 */
- Large_Claim_Agreement(?clm, ?contract, ?amount):-
  WCC:CLAIM_PAID_Amount(?clm,?amount),
  WCC:CLAIM_CONTRACT_Relationship_To(?u, ?clm),
  WCC:CLAIM_CONTRACT_Relationship_From(?u, ?contract);
  ?amount > "5000"^^xsd:double;.

- /* A contract is involved in the large_claim_Agreement relationship if its sub-contract has large claims */
- Large_Claim_Agreement(?clm, ?contract, ?amount):-
  Large_Claim_Agreement(?clm, ?subcontract, ?amount),Sub_Contract(?subcontract,?contract);;.

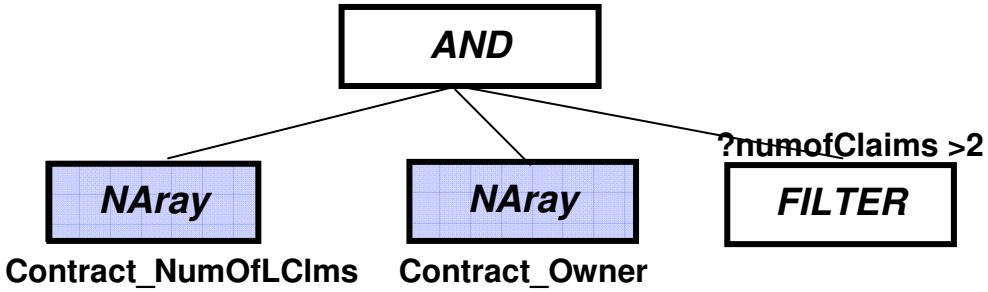
- /* Compute the contract and the total number of large claims of the contract */
- Contract_NumOfLClms(?contract, ?Agg_numclm):-
  Large_Claim_Agreement(?clm, ?contract, ?amount);; GROUP BY<?contract>COUNT(?clm).

- /* A contract owner is the party that plays the role of "owner primary" in the component of a contract */
- Contract_Owner(?pty, ?contract):-
  WCC:Assemble(?component ?contract),
  WCC:CONTRACT_ROLE_Relationship_To(?u ?component),
  WCC:CONTRACT_ROLE_Relationship_From(?u, ?pty),
  WCC: hasContract_ROLE(?u,?typeCD),
  WCC: CONTRACT_ROLE_NAME(?typeCD 'Owner Primary');;.

Evaluation Process

```
Select ?pty
Where {
    Contract_NumOfLClms(?contract ?numofClaims).
    Contract_Owner(?pty ?contract).
    FILTER (?numofClaims >2) }
```

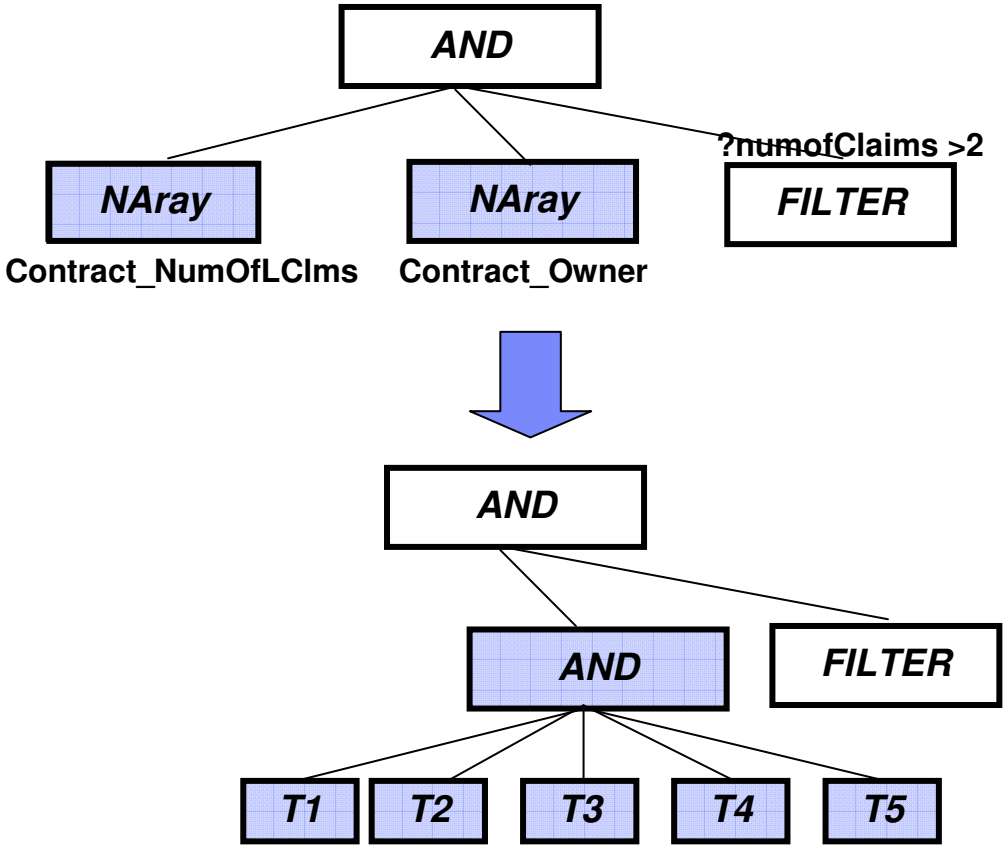
SPARQL Query Tree



Evaluation Process

```
Select ?pty
Where {
    Contract_NumOfLClms(?contract ?numofClaims).
    Contract_Owner(?pty ?contract).
    FILTER (?numofClaims >2) }
```

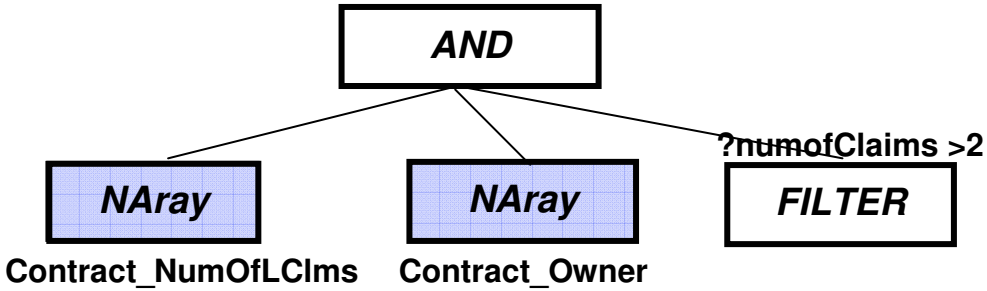
SPARQL Query Tree

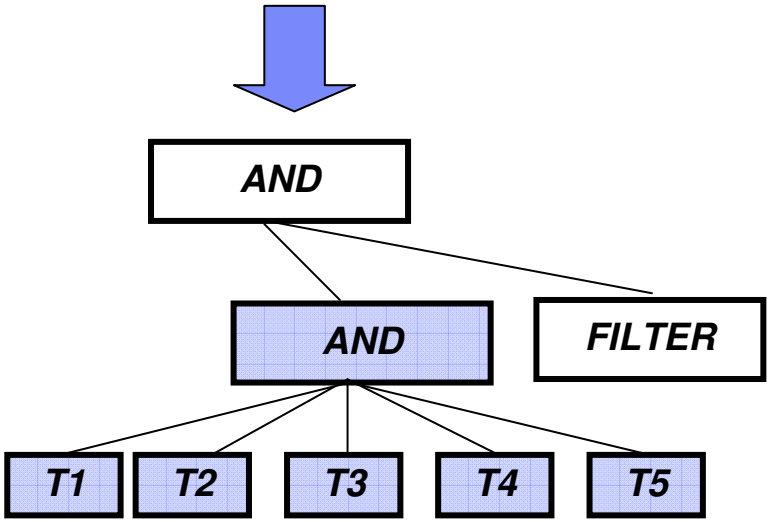


Evaluation Process

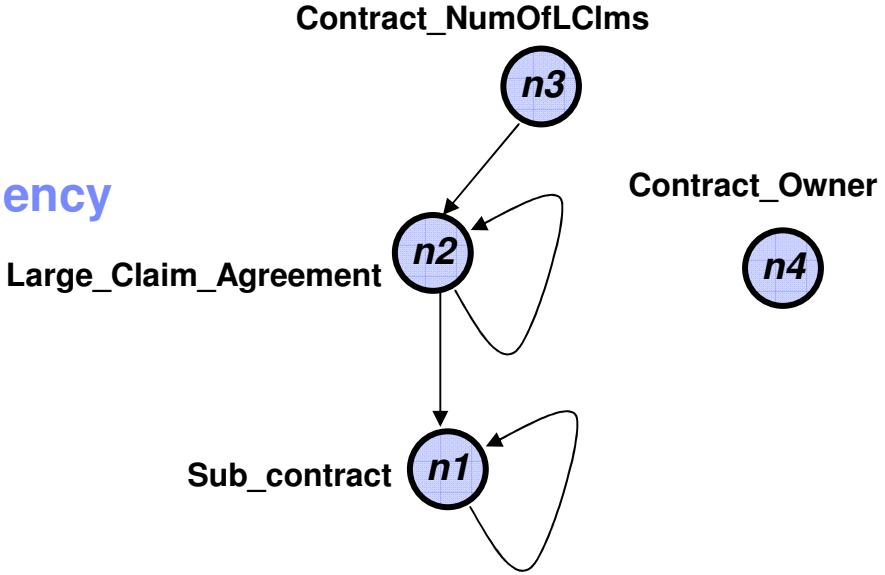
```
Select ?pty
Where {
    Contract_NumOfLClms(?contract ?numofClaims).
    Contract_Owner(?pty ?contract).
    FILTER (?numofClaims >2) }
```

SPARQL Query Tree





Dependency graph

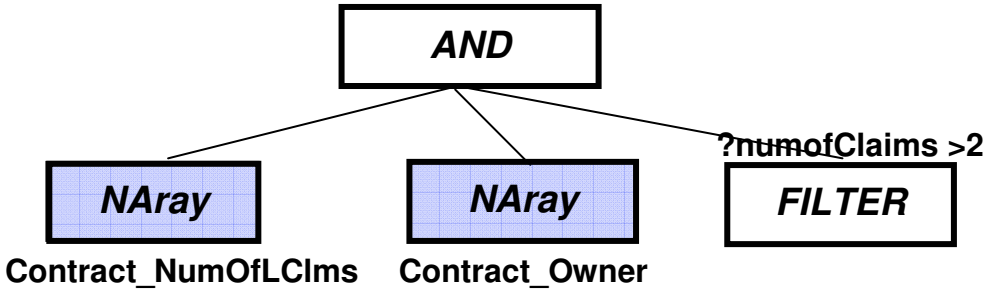


Key	Temporal table object
n1	Sub_contract

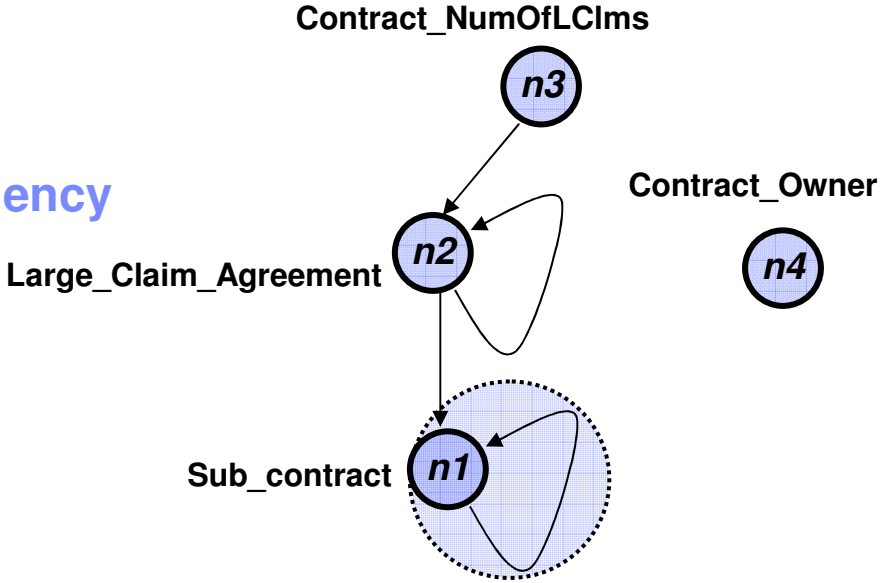
Evaluation Process

```
Select ?pty
Where {
    Contract_NumOfLClms(?contract ?numofClaims).
    Contract_Owner(?pty ?contract).
    FILTER (?numofClaims >2) }
```

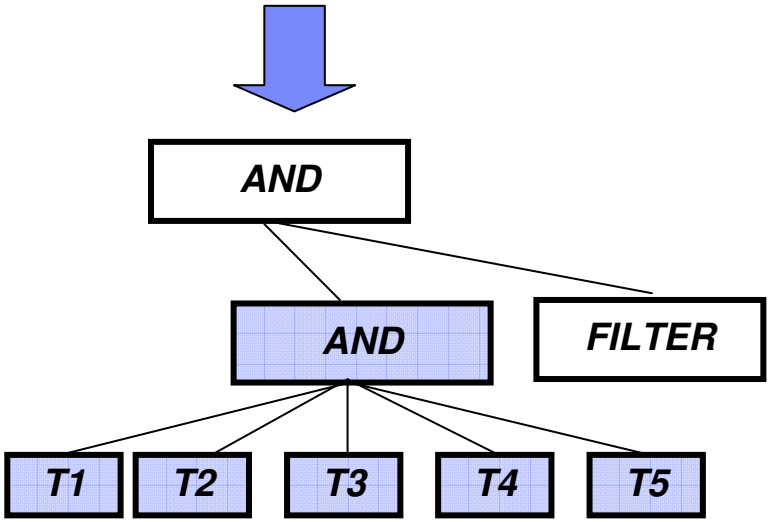
SPARQL Query Tree



Dependency graph



Evaluate: R1, R2

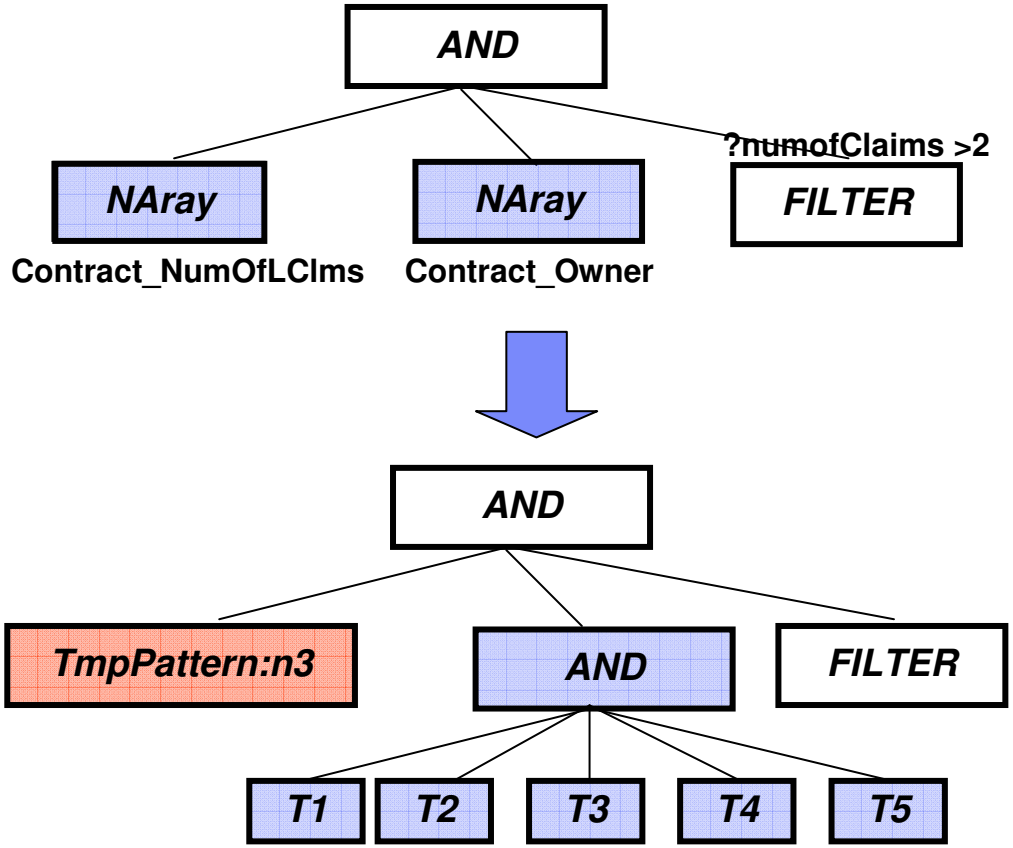


Key	Temporal table object
n1	Sub_contract

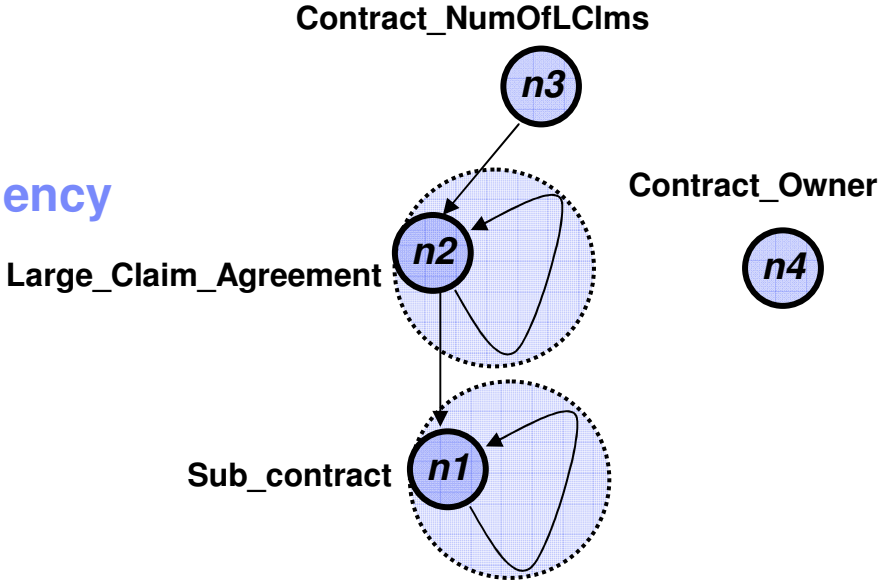
Evaluation Process

```
Select ?pty
Where {
    Contract_NumOfLClms(?contract ?numofClaims).
    Contract_Owner(?pty ?contract).
    FILTER (?numofClaims >2) }
```

SPARQL Query Tree



Dependency graph



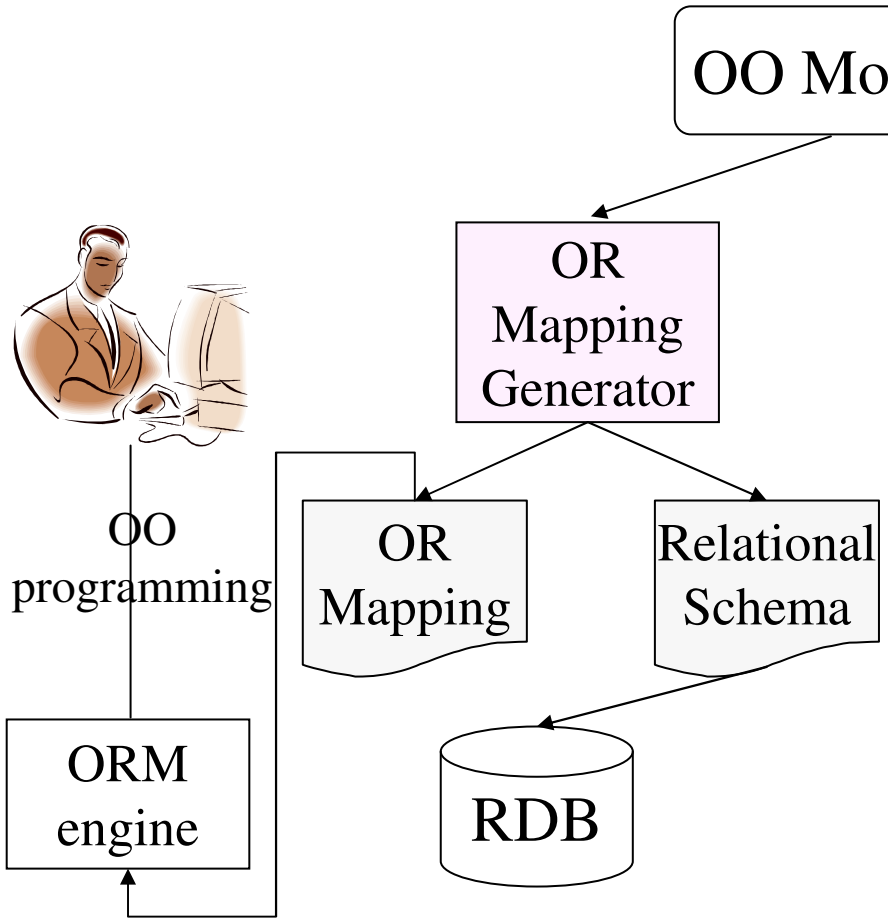
Evaluate: R1, R2

Evaluate: R3, R4

Evaluate: R5

Key	Temporal table object
n1	Sub_contract
n2	Large_Claim_Agreement
n3	Contract_NumOfLClms

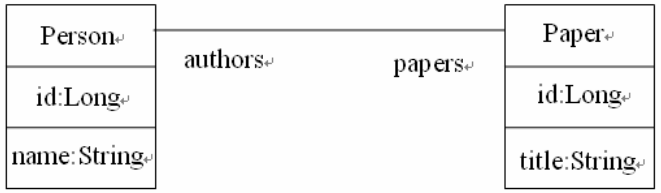
Mapping Generation from Object Models



Object Relational Mapping (ORM) has been a more and more popular technique for objects persistence

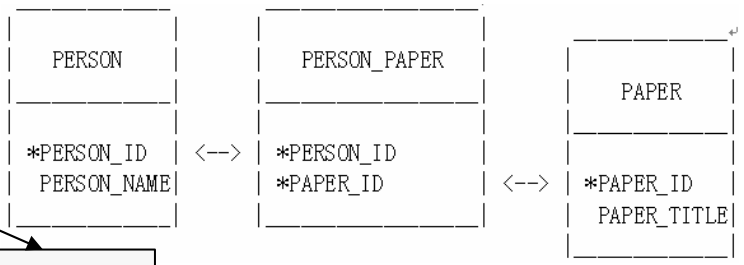
Sources: Jing Mei's presentation

JSON Object: MJ
{"id" : "123",
"name" : "Jing Mei",
"papers" : [
"UMRR: Towards an Enterprise-Wide Web of Models",
"Ontology Query Answering on Databases",] }
JSON Object: UMRR
{"id" : "321",
"title" : "UMRR: Towards an Enterprise-Wide of Models",
"authors" :["Lei Zhang", "Shengping Liu", "Jing Mei",
"Guotong Xie", "Bob Schloss", "Yue Pan",] }



OO Models

OR Mapping Generator



OO programming

OR Mapping

Relational Schema

ORM engine

RDB

PERSON

PERSON_ID	PERSON_NAME
123	Jing Mei

PAPER

PAPER_ID	PAPER_TITLE
321	UMRR: Towards an Enterprise-Wide of Models

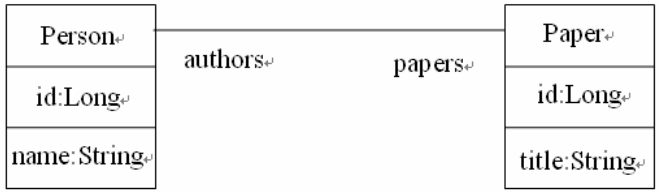
PERSON_PAPER

PERSON_ID	PAPER_ID
123	321

Object Relational Mapping (ORM) has been a more and more popular technique for objects persistence

Sources: Jing Mei's presentation

JSON Object: MJ
{"id" : "123",
"name" : "Jing Mei",
"papers" : [
"UMRR: Towards an Enterprise-Wide Web of Models",
"Ontology Query Answering on Databases",] }
JSON Object: UMRR
{"id" : "321",
"title" : "UMRR: Towards an Enterprise-Wide of Models",
"authors" :["Lei Zhang", "Shengping Liu", "Jing Mei",
"Guotong Xie", "Bob Schloss", "Yue Pan",] }




```
d:MJ a t:Person;
      t:PersonId "123";
      t:PersonName "Jing Mei";
      t:PersonPapers d:UMRR;
      t:PersonPapers d:OntoDB;
d:UMRR a t:Paper;
      t:PaperId "321";
      t:PaperTitle "UMRR…";
      t:PaperAuthors d:ZL;
      t:PaperAuthors d:LSP;
      t:PaperAuthors d:MJ;
      t:PaperAuthors d:XGT;
      t:PaperAuthors d:BS;
      t:PaperAuthors d:PY;
```

OO Models

OR Mapping Generator

PERSON (*PERSON_ID, PERSON_NAME) <--> PERSON_PAPER (*PERSON_ID, *PAPER_ID) <--> PAPER (*PAPER_ID, PAPER_TITLE)

OO programming

OR Mapping

Relational Schema

ORM engine

RDB

?

SW style access

PERSON

PERSON_ID	PERSON_NAME
123	Jing Mei

PAPER

PAPER_ID	PAPER_TITLE
321	UMRR: Towards an Enterprise-Wide of Models

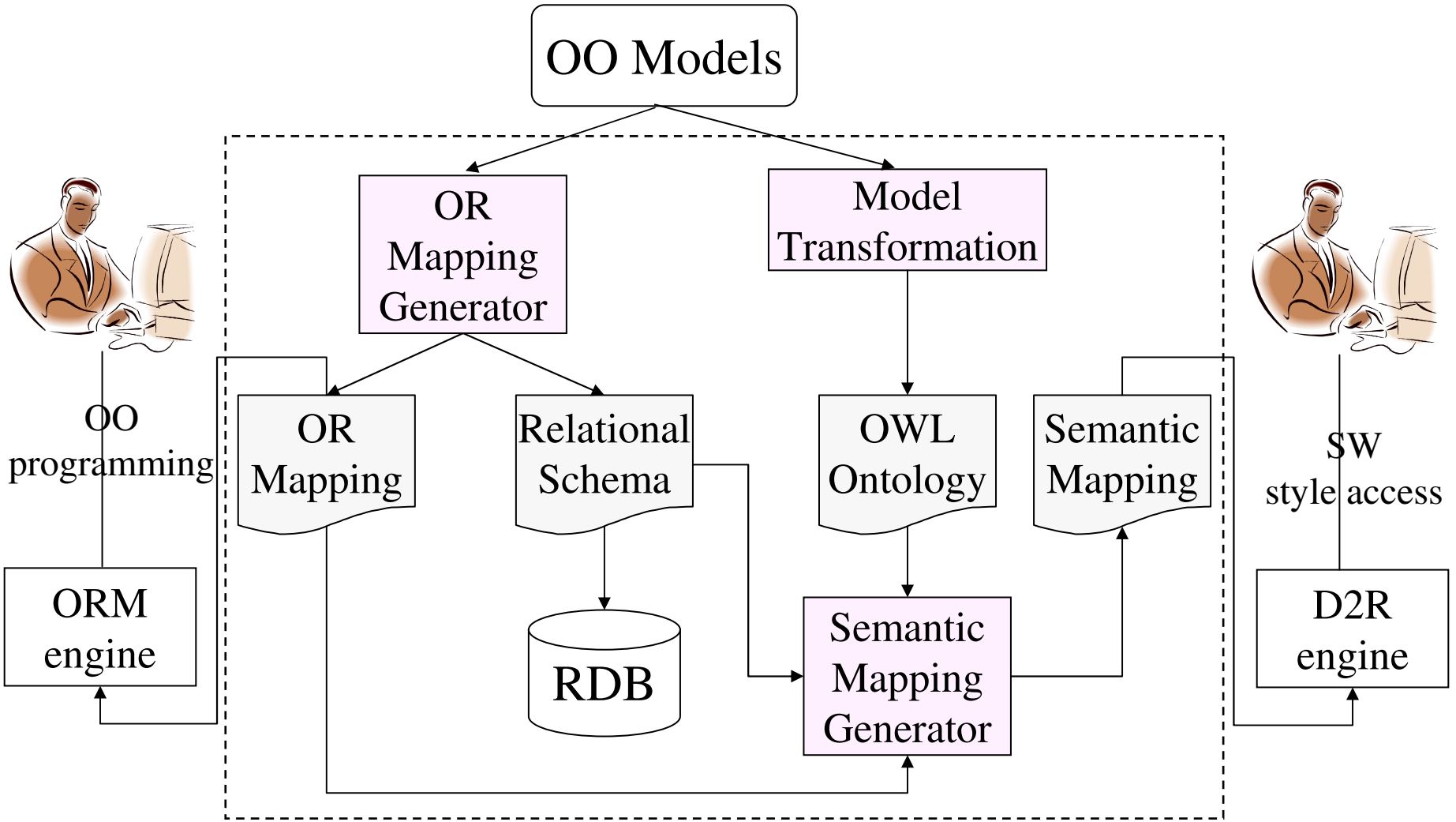
PERSON_PAPER

PERSON_ID	PAPER_ID
123	321

Object Relational Mapping (ORM) has been a more and more popular technique for objects persistence

Sources: Jing Mei's presentation

Mapping Generation from Object Models



Sources: Jing Mei's presentation

Semantic Web Tools

- **SOR@IBM Alphaworks** (http://www.alphaworks.ibm.com/tech/semanticstk)

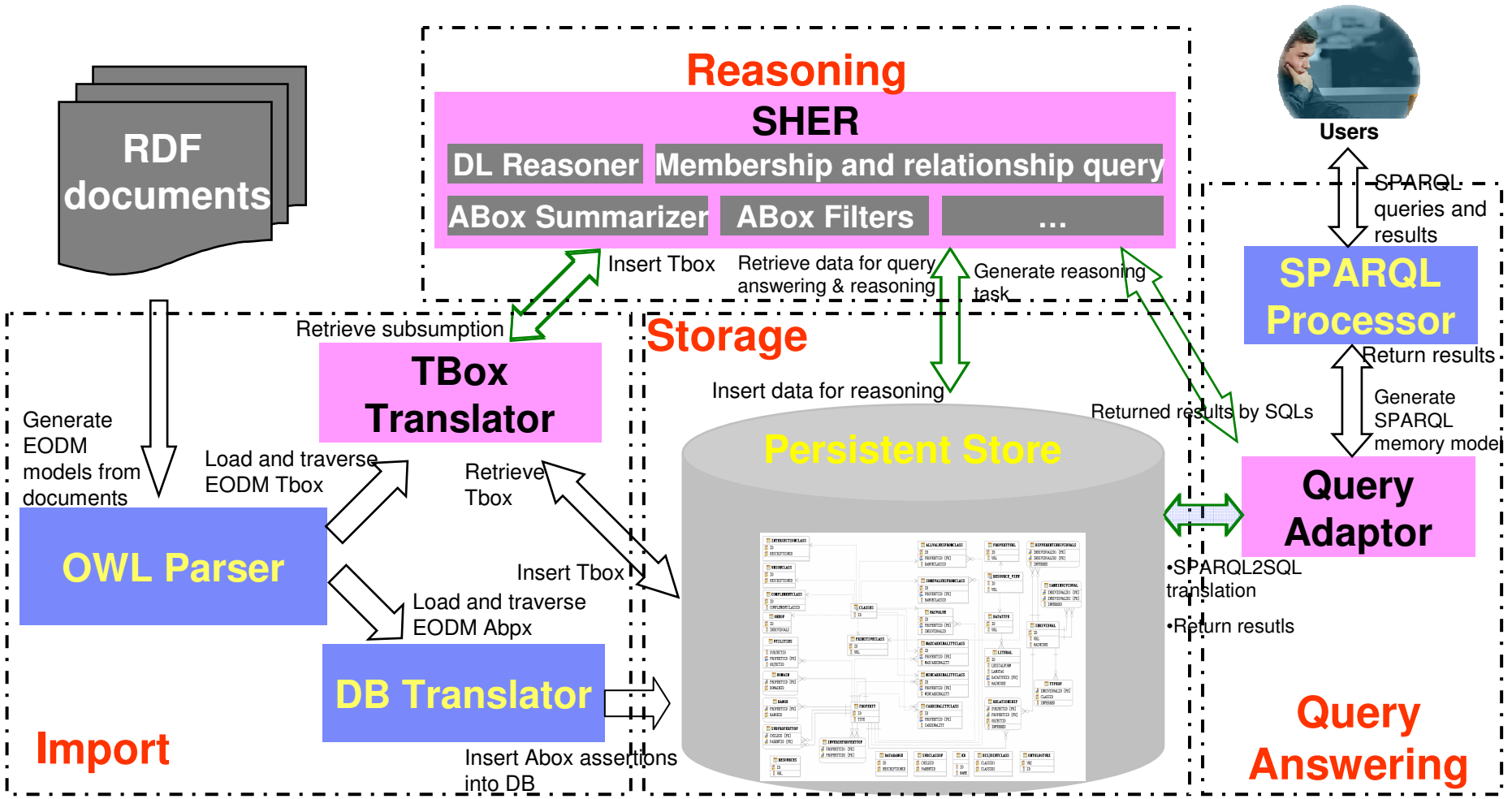
  SOR is a high performance Semantic Web data management system on top of DB2, including efficient schema and indexes design for storage, practical ontology reasoning support, and an effective SPARQL-to-SQL translation method for RDF query.

- **SHER@IBM Alphaworks** (http://www.alphaworks.ibm.com/tech/sher)

  Scalable Highly Expressive Reasoner (SHER) is a breakthrough technology that provides ontology analytics over highly expressive ontologies (OWL-DL without nominals). SHER does not do any inferencing on load; hence it deals better with quickly changing data (the downside is, of course, that reasoning is performed at query time). The tool can reason on approximately seven million triples in seconds, and it scales to data sets with 60 million triples, responding to queries in minutes. It has been used to semantically index 300 million triples from medical literature.

  SHER tolerates logical inconsistencies in the data, and it can quickly point you to these inconsistencies in the data and help you clean up inconsistencies before issuing semantic queries. The tool explains (or justifies) why a particular result set is an answer to the query; this explanation is useful for validation by domain experts.

A Pluggable Semantic Object Repository




감사합니다
Korean
多謝
Traditional Chinese
ขอบคุณ
Thai
Спасибо
Russian
Gracias
Spanish
Thank You
English
شكرا
Arabic
Obrigado
Brazilian Portuguese
多谢
Simplified Chinese
Danke
German
Grazie
Italian
Merci
French
நன்றி
Tamil
Tamil
ありがとうございました
Japanese
धन्यवाद
Hindi
Hindi

Recent Publications

- Julian Dolby, Achille Fokoue, Aditya Kalyanpur, Li Ma, Edith Schonberg, Kavitha Srinivas and Xingzhi Sun, “Scalable Conjunctive Query Evaluation Over Large and Expressive Knowledge Bases”, Accepted by ISWC2008.
- Li Ma, Chen Wang, Jing Lu, Feng Cao, Yue Pan, Yong Yu, "Effective and Efficient Semantic Web Data Management over DB2", SIGMOD 2008, pp. 1183-1194.
- Ying Yan, Chen Wang, Aoying Zhou, Weining Qian, Li Ma, Yue Pan, "Efficiently querying rdf data in triple stores", poster, WWW 2008, pp. 1053-1054.
- Robert Lu, Feng Cao, Li Ma, Yong Yu and Yue Pan, "An Effective SPARQL Support over Relational Databases", VLDB2007 Joint ODBIS & SWDB workshop on Semantic Web, Ontologies, Databases.
- Julian Dolby, Achille Fokoue, Aditya Kalyanpur, Aaron Kershenbaum, Li Ma, Edith Schonberg, Kavitha Srinivas, "Scalable Semantic Retrieval Through Summarization and Refinement", the 21st Conference on Artificial Intelligence (AAAI 2007), pp.299-304, 2007.