

A Database Backend for OWL

October 24, 2009

Jörg Henß
Joachim Kleb
Stephan Grimm
Jürgen Bock

FZI Research Center for Information Technology
at the University of Karlsruhe
Germany

- Why do we need a Database Backend for OWL?
 - Scalable Storage
 - Fast Access
 - Transactions, Versioning, Permissions ...

- Why yet another Storage Layer?
 - Mostly triple-based Systems today
 - Cumbersome
 - **Native** support for OWL
 - Direct Manipulation

- Not our Focus: Reasoning

- By **native** we understand:
 - Mapping OWL language constructs one-to-one to storage layer

- Triple Structure
 - RDF-Store

- Axiomatic view
 - Restrictions, cardinalities
 - OWL Axioms act on **objects** not on **nodes**
 - An object model for OWL is required

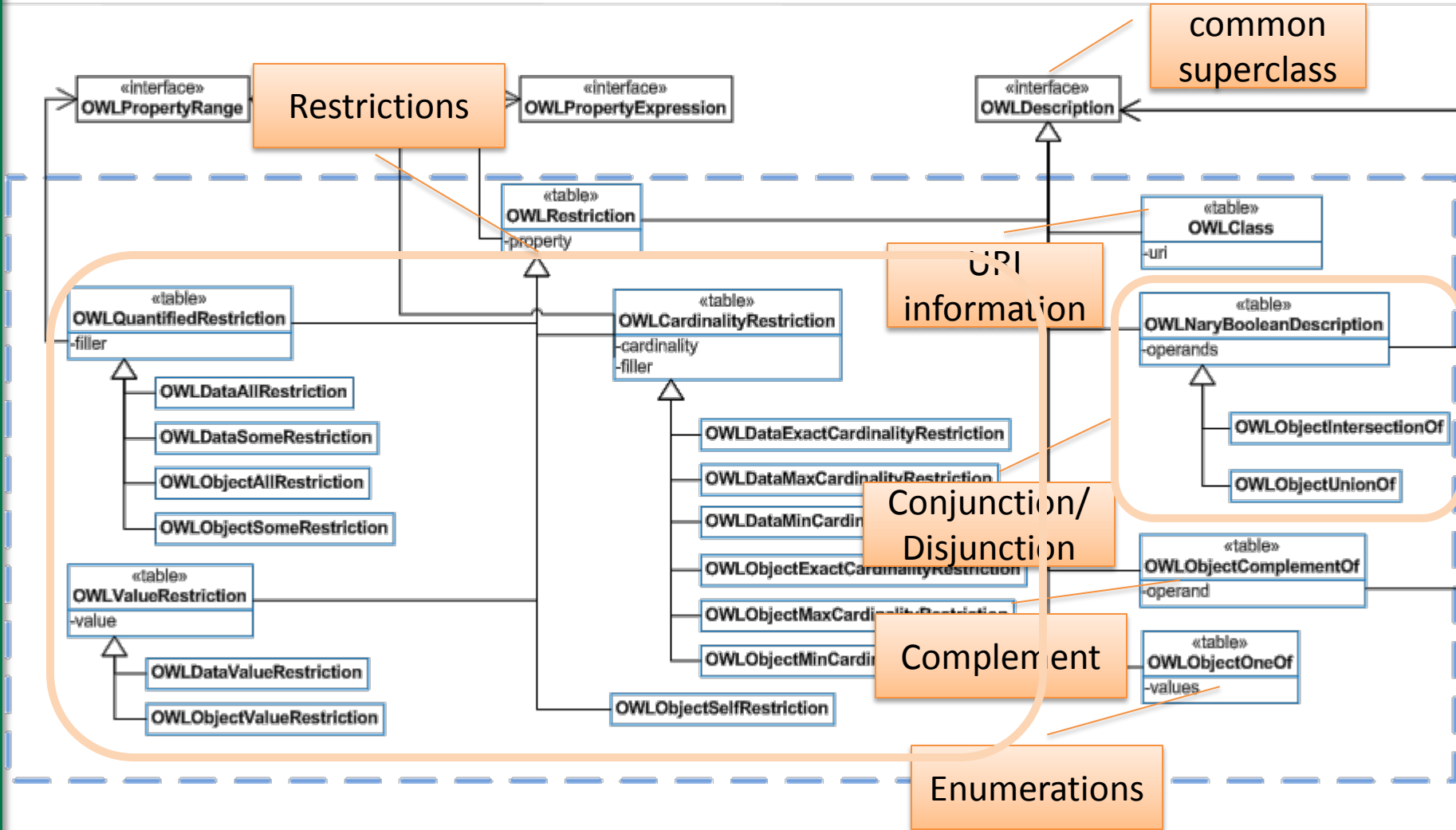
Schema Representation

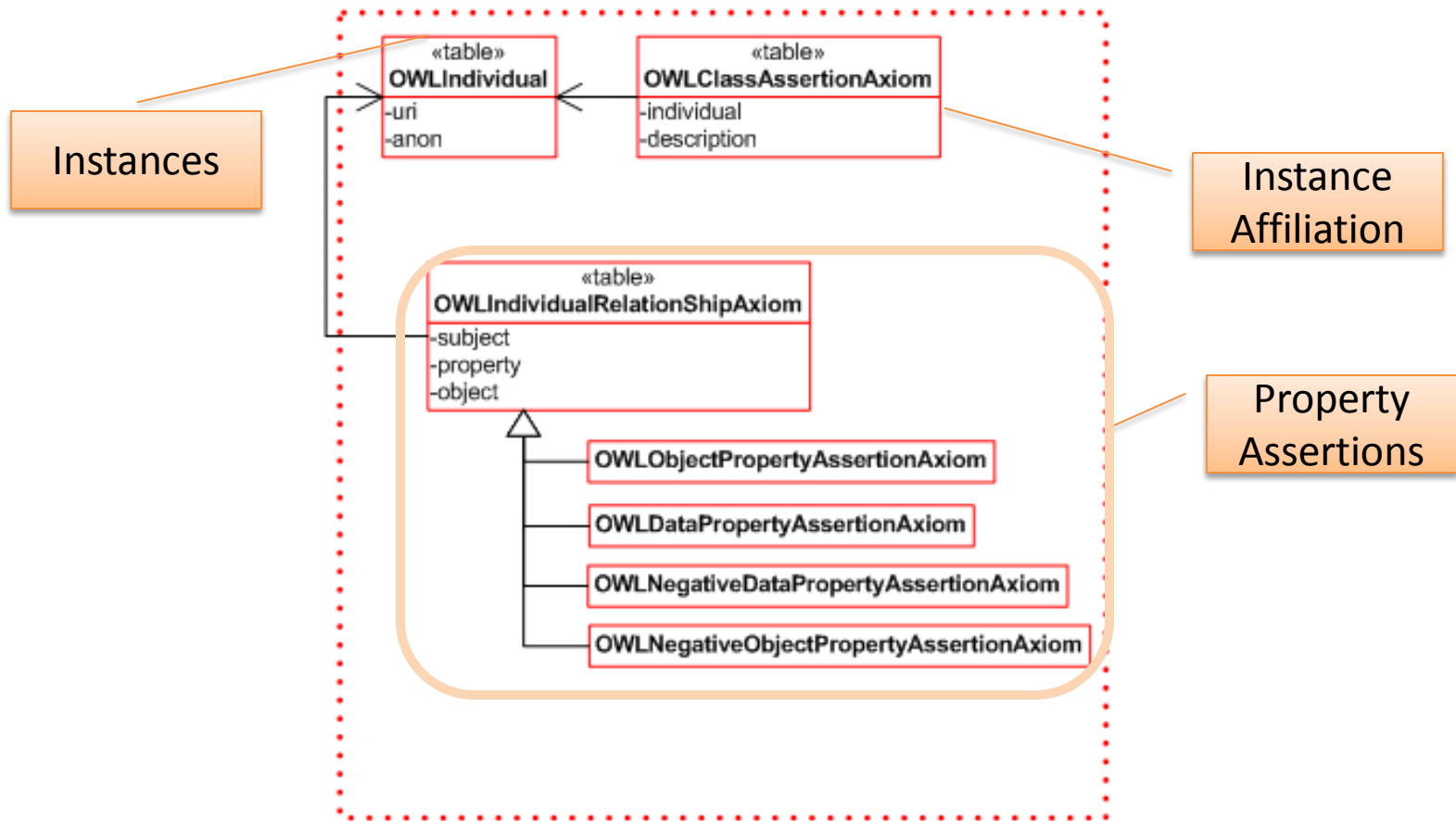
- OWL as Objects and Axioms
 - Classes, Individuals, etc.
 - Class Assertions, etc.

- OWL-API as Model for OWL
 - Java based API for OWL
 - Maintained by University of Manchester
 - OWL 2 ready
 - Protégé 4 is based upon

- Use of Object-Relational mapping for persistence
 - Stores object and axiom information in database
 - Several Mapping Strategies

Complex Classes





- Comparison with other database-based Systems:
 - IBM SOR
 - Jena SDB
 - Owlgres

- Ontologies with varying Complexity
 - FMA, wine9, LUBM

- Mixture of common tasks tested
 - Import and Load Time
 - Retrieval Queries
 - Assertion Queries
 - Axiom and Statistical Queries

Evaluation - Results

wine9:

	OWLDB	SDB	Owlgres	SOR
Import	314368	61484	123718	352323
Load	1579	953	47	51200

Table 1. Import and load time (ms).

	OWLDB	SDB	Owlgres	SOR
Annotation	14172	234	10	390
Class	3610	3828	125	360

Table 2. Retrieval Queries (ms).

	OWLDB	SDB	Owlgres	SOR
Individual	63	63	4468	125
Object Property	250	79	47	263
Data Property	31	15	3	47

Table 3. Individual assertion retrieval time (ms).

	OWLDB	SDB	Owlgres	SOR
Sub Classes	52	1359	3	93
Inverse Properties	47	110	2	63
Transitive Properties	3	3	-	15
All Classes	31	329	3	30
All Individuals	1265	507524	203	188

Table 4. Statistical query retrieval time (ms).

- Mixed Results
 - Batch-Loading speeds up Import
 - Strong Influence of Tbox Caching

- Memory Consumption much lower

Conclusion

- Novel Approach to Ontology Persistence
 - Object Relational Mapping
- Still Prototype
 - Further Optimisations required
- Next Steps:
 - Support OWL-evolution
 - Usage of full-text indices
- Download address
 - <http://owldb.sourceforge.net/>
- Part of the German Theseus Research Project