

Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema(Broekstra et. al.)

Sidharth Singla



RoadMap

1. Concepts
2. Introduction
3. Need for RDFS Query language
4. Architecture(SAIL + 3 Modules)
5. Evaluation
 - a. PostgreSQL + SAIL
 - b. MySQL + SAIL
6. Future Work proposed
7. Conclusion
8. Present Sesame

Semantic Web

In addition to the classic “Web of documents” W3C is helping to build a technology stack to support a “Web of data,” the sort of data you find in databases. The term “Semantic Web” refers to W3C’s vision of the Web of linked data.

<https://www.w3.org/standards/semanticweb/>

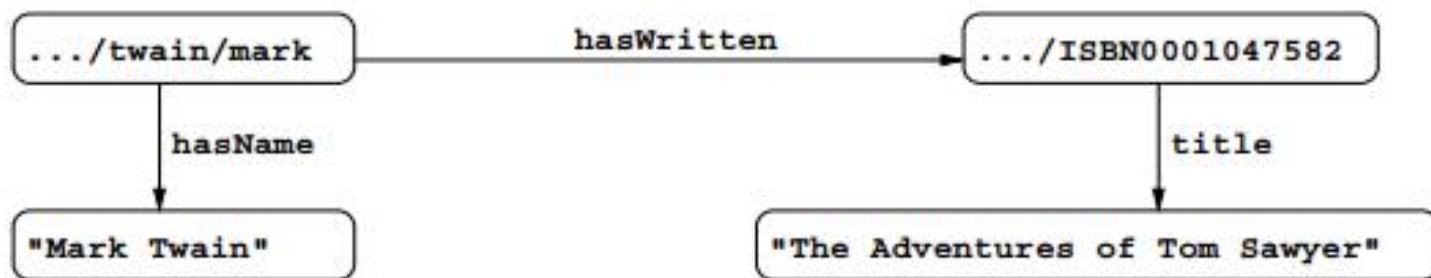
The term was coined by Tim Berners-Lee for a web of data that can be processed by machines.

RDF

The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model. RDF graphs are sets of subject-predicate-object triples, where the elements may be IRIs, blank nodes, or data-typed literals.

<https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/Overview.html>

https://en.wikipedia.org/wiki/Resource_Description_Framework

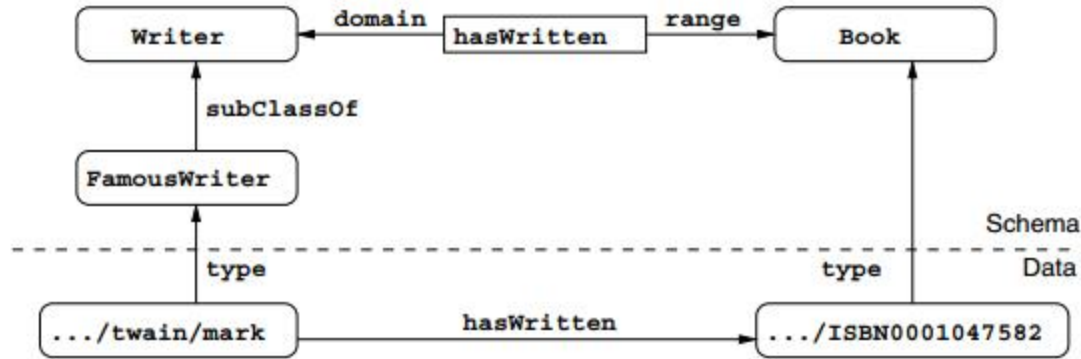


```
<rdf:Description rdf:about="http://www.famouswriters.org/twain/mark">  
  <s:hasName>Mark Twain</s:hasName>  
  <s:hasWritten rdf:resource="http://www.books.org/ISBN0001047582"/>  
</rdf:Description>
```

```
<rdf:Description rdf:about="http://www.books.org/ISBN0001047582">  
  <s:title>The Adventures of Tom Sawyer</s:title>  
</rdf:Description>
```

RDF Schema

RDF Schema (Resource Description Framework Schema) is a set of classes with certain properties using the RDF extensible knowledge representation data model, providing basic elements for the description of ontologies and is intended to structure RDF resources. https://en.wikipedia.org/wiki/RDF_Schema



Introduction

1. Sesame - An architecture for storage and querying of RDF and RDFS information.
2. First public available implementation of query language aware of RDFS semantics.
3. Developed by Aidministrator Nederland b.v.
4. Provides caching and concurrency control support.
5. Design and implementation are independent from any specific storage device.
6. Query language - RQL for RDF that offers native support for RDF Schema semantics.

Need for RDFS Query Language

3 different levels of abstraction:

1. Syntactic level - XML documents.(not necessarily true)
2. Structure level - A set of triples.
3. Semantic level - One or more graphs with partially predefined semantics.

The documents can be queried at each of these three levels.

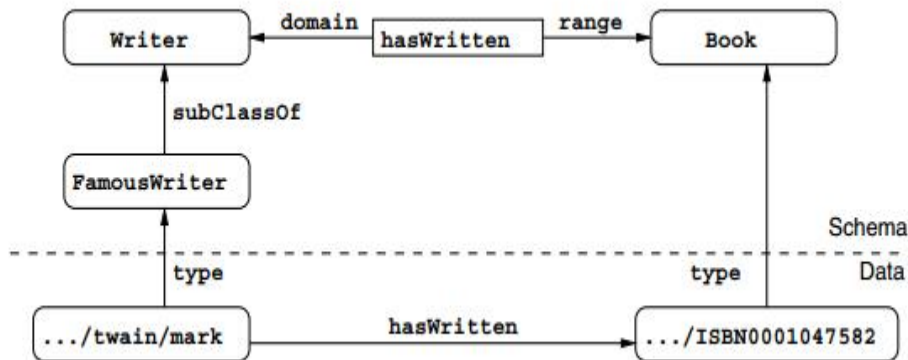
Sesame architecture queries RDF/RDFS documents at the semantic level using RQL query language.

Querying at the Syntactic Level

- ❑ Any RDF model and RDF schema can be written down in XML notation. RDF can be queried using an XML query language (for example, XQuery).
- ❑ However, RDF is not just an XML notation.
Data model very different from the XML tree structure.
Relationships not apparent from the XML tree structure become very hard to query.

Querying at the Structure Level

- ❑ Number of query languages proposed and implemented to query the triples.
- ❑ However, elements with special semantics in RDFS are interpreted as only a set of triples.



```
SELECT ?x
FROM   somesource
WHERE  (rdf::type ?x FamousWriter)
```

```
SELECT ?x
FROM   somesource
WHERE  (rdf::type ?x Writer)
```

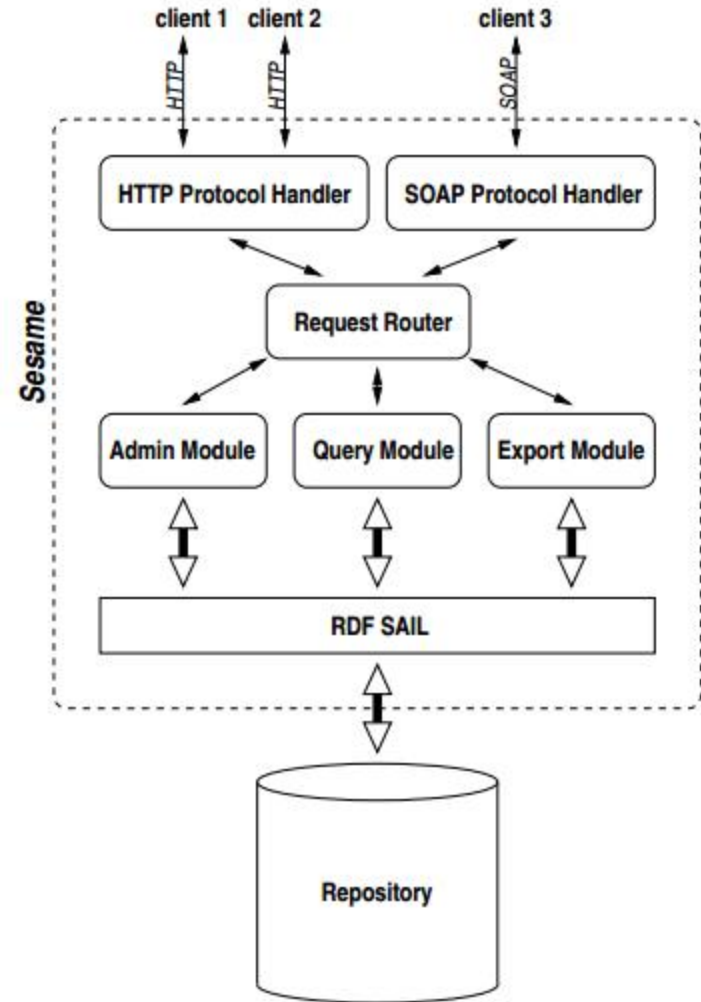
Query Fails

Querying at the Semantic Level

- ❑ RQL - A declarative query language that capture the semantics explicitly. Full knowledge entailed in RDFS description is queried.
- ❑ RQL adopts the syntax of OQL.
Language is defined by a set of core queries, a set of basic filters, and a way to build new queries through functional composition and iterators.
Core queries - Basic building blocks of RQL. Queries such as Class (retrieving all classes) and Property (retrieving all properties) are allowed.
- ❑ In example, Query Writer returns all instances of the class with name Writer. Instances of subclasses of Writer are also returned by virtue of the semantics of RDFS.

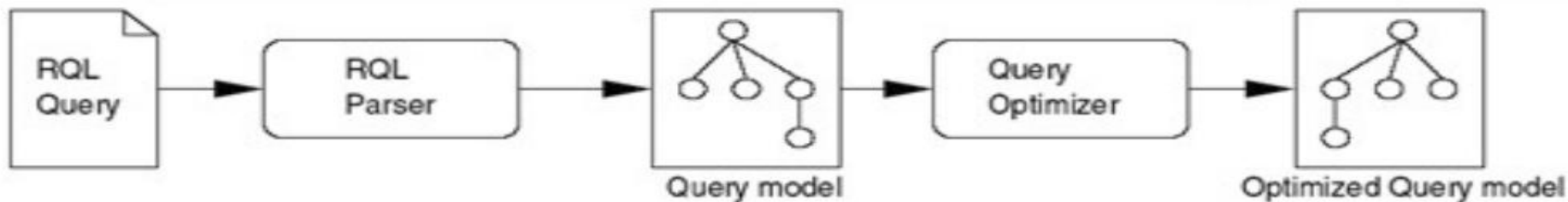
Architecture

- ❑ Sesame is DBMS-independent.
- ❑ All DBMS-specific code is concentrated in a single architectural layer of Sesame: the Storage And Inference Layer (SAIL). An API that offers RDF specific methods to its clients and translates the methods to calls to its specific DBMS.
- ❑ Sesame's functional modules are clients of the SAIL API. Three such modules: RQL query engine, RDF admin module and RDF export module.



RQL Query Module

- ❑ In Sesame, RQL implemented different from the language proposed. Domain and range restrictions support in Sesame. Support for data-typing feature is absent.
- ❑ Query parsing and optimisations.
- ❑ Queries are translated into a set of calls to the SAIL. Several optimization techniques are to be devised in the engine and the SAIL API implementation.
- ❑ Results returned in a streaming fashion. The entire result set not required to be built in memory first.



Admin Module

- ❑ Incrementally adds RDF data/schema information.
Clears a repository.
Delete on a per-statement basis functionality not yet available but is under development.
- ❑ Information is parsed using streaming ARP RDF parser(Another RDF/XML Parser). ARP is part of the Jena Toolkit. Information is delivered to the admin module on a per-statement basis: (Subject, Predicate, Object). The statement is asserted into the repository by communicating with the SAIL.

Export Module

- ❑ The module exports the contents of a repository formatted in XML-serialized RDF.
- ❑ Sesame can be used in combination with other RDF tools, as all RDF tools will at least be able to read this format.
- ❑ It is able to selectively export the schema, the data, or both.

SAIL API(1/2)

Set of Java interfaces. The API should:

- ❑ Store, retrieve and delete RDF and RDFS from repositories using interface.
- ❑ Abstract from the actual storage mechanism.
- ❑ Usable on low end hardware like PDAs and able to handle huge amounts of data efficiently on enterprise level database clusters.
- ❑ Be extendable to other RDF-based languages like DAML+OIL.

SAIL API(2/2)

- ❑ Several implementations of the SAIL API.

Most important - SQL92SAIL.

Generic implementation for SQL92.

- ❑ Aim: To be able to connect to any RDBMS .
- ❑ An inferencing module for RDFS, based on the RDFS entailment rules as specified in the RDF Model Theory. Computes the schema closure and asserts these implicates of the schema as derived statements.
Statement: (foo, rdfs:domain, bar). Implied statement asserted: (foo, rdf:type, property)

- ❑ Possible to put one on top of the other.

SAIL Caching

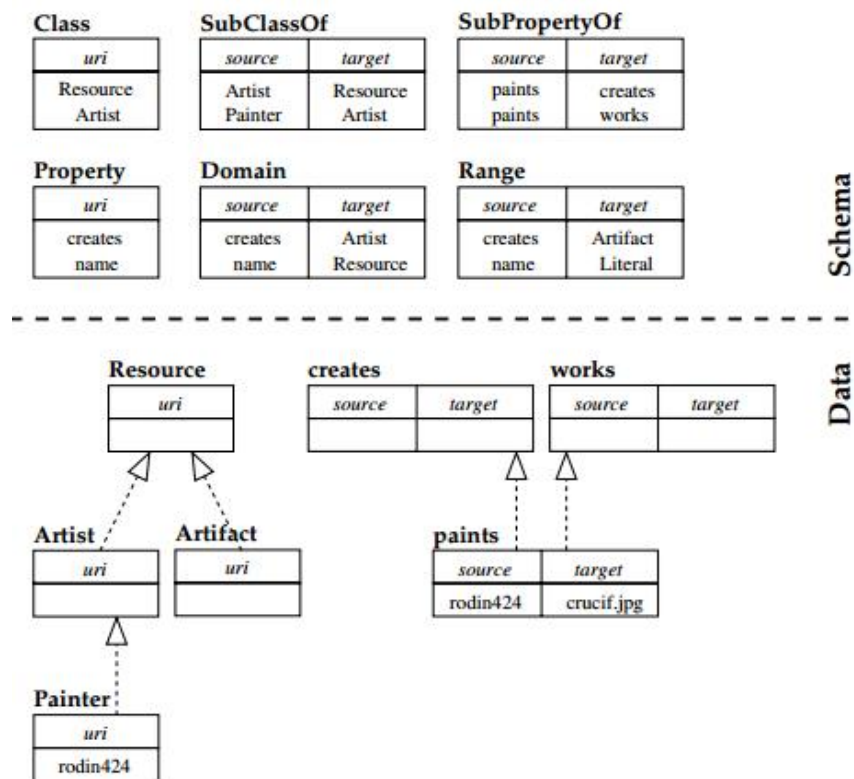
- ❑ Caches all schema data in a dedicated data structure in main memory.
- ❑ Schema data often limited in size and requested very frequently.
- ❑ Schema-caching SAIL placed on the top of arbitrary other SAILs.

SAIL Concurrency Handling

- ❑ RQL query breaks down into several operations on the SAIL level.
- ❑ Important to preserve repository consistency.
- ❑ SAIL Implementation to selectively block and release read and write access to repositories.
- ❑ Concurrency control supported for any type of repository.

PostgreSQL and SAIL

- ❑ PostgreSQL - Object-relational DBMS.
Support sub-table relations.
- ❑ Postgre SAIL - Tailored towards PostgreSQL's support for subtables
- ❑ New Table - 'resources'.
Aim: Minimize database size.
Contains all resources and literal values,
mapped to a unique ID.
- ❑ Experience - Not Satisfactory.
 - ❑ Data insertion not fast.
 - ❑ Adding a new subclass between two existing classes inefficient.

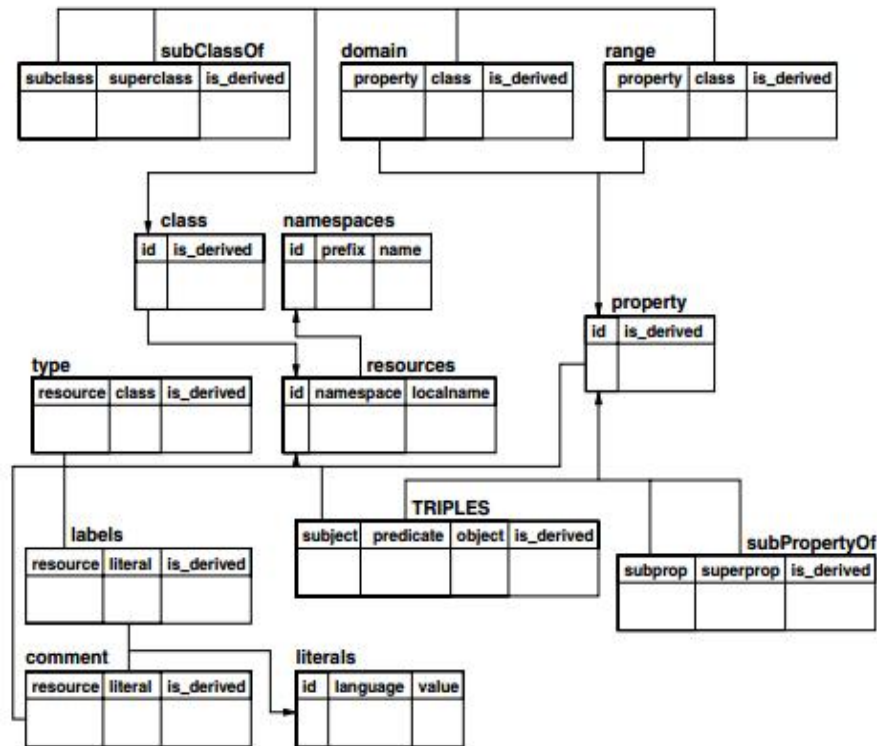


New Setup

- ❑ SQL92SAIL connect to PostgreSQL.
- ❑ RDF statements inserted into a single table with three columns: Subject, Predicate, Object.
- ❑ For querying purposes, the original PostgreSQL SAIL performed quite satisfactory.

MySQL SAIL

- ❑ SAIL implementation - strictly relational database schema.
- ❑ A number of dependencies.
- ❑ Faster lookups - Every resource and literal encoded using id field.
- ❑ is_derived column - To encode if a statement was explicitly asserted or derived from the schema information.
- ❑ Database schema does not change when the RDFS changes.
- ❑ Better Performance, especially on uploading.



Future Work

- ❑ Transaction Rollback Support. Makes Sesame - ACID compliant.
- ❑ Versioning Support.
- ❑ Advanced update support, partial deletion.
- ❑ New modules - Graphical Visualization, Query engines for different languages.

- ❑ DAML+OIL Support.
Successor language to DAML and OIL that combines features of both. Superseded by Web Ontology Language (OWL).
DAML - DARPA Agent Markup Language.
OIL - Ontology Inference Layer or Ontology Interchange Language.
The DAML program ended in early 2006.
<https://en.wikipedia.org/wiki/DAML%2BOIL>

Key Takeaways

- ❑ Sesame - An architecture for storing and querying RDF and RDF Schema.
- ❑ Important feature - Abstraction from the details of any particular repository used for the actual storage. Variety of repositories possible - relational databases, RDF triple stores, remote storage services on the Web.
- ❑ Can be used as a remote service for manipulating RDF data on the Semantic Web. Abstraction from any particular communication protocol.
- ❑ Most important SAIL Implementation- SQL92SAIL. Generic implementation for SQL92.

Sesame

Sesame

Name of the tool:	Sesame
Home page:	http://www.openrdf.org/
Date of latest release:	
Programming language(s) that can be used with the tool:	Java, Python, PHP
Relevant semantic web technologies:	RDF, RDFS, SPARQL
Categories:	Triple Store, Programming Environment, Reasoner, RDFS Reasoner, Parser
See also:	http://semanticweb.org/wiki/Sesame
Public mailing list:	http://www.openrdf.org/community.jsp
Preferred project URI:	
DOAP reference:	
Company or institution:	Aduna

(Tool description last modified on 2013-11-5.)

Description

[Sesame](#) is an open source [RDF](#) database with support for [RDFS](#) inferencing and querying. It offers a large scale of tools to developers to leverage the power of RDF and RDF Schema. It is a Java framework, there is also a [python wrapper](#).

In addition to offering ready to use native and database-backed quadstores, it also offers an extensible API to build other stores upon. There are many third-party stores that can be used through the Sesame Sail API. Examples are [4store](#), [Bigdata](#), and [OWLIM](#) (which offers optional support for OWL reasoning).

The Sesame SAIL API can also be used to extend the functionality of such stores. Geospatial and free-text indexing and search can be added to Sesame compatible quadstores with the [IndexingSail](#) extension.

<https://www.w3.org/2001/sw/wiki/Sesame>

Sesame is now rdf4j

- ❑ RDF4J offers two out-of-the-box RDF databases (the in-memory store and the native store), and in addition many third party storage solutions are available.
- ❑ RDF4J fully supports the SPARQL 1.1 query and update language.
- ❑ RDF4J supports all mainstream RDF file formats, including RDF/XML, Turtle, N-Triples, N-Quads, JSON-LD, TriG and TriX.

Welcome to RDF4J



Eclipse RDF4J (formerly known as Sesame) is a powerful Java framework for processing and handling RDF data. This includes creating, parsing, scalable storage, reasoning and querying with RDF and Linked Data. It offers an easy-to-use API that can be connected to all leading RDF database solutions.

<http://rdf4j.org/about/>



UNIVERSITY OF
WATERLOO

Discussion

- ❑ RQL is more expressive than SPARQL(Wikipedia).
Why there is no more support for RQL, and SPARQL is the standard for rdf4j ?
- ❑ What is the advantage of putting SAIL on top of each other ?
- ❑ Why is there nothing mentioned for querying a data-store with its own query language ? Eg. Querying MySQL data store with MySQL query language.
Will there be any differences in querying MySQL store with RQL and MySQL query languages ?