

adesso

business.
people.
technology.



STI · INNSBRUCK



RUBEN

A Rule Engine Benchmarking Framework

RuleML Webinar – 30.11.2022

Kevin Angele, Jürgen Angele, Umutcan Simsek, and Dieter Fensel

Outline

1. Motivation
2. RUBEN
3. Test Cases
4. Evaluation
5. Conclusion and Future Work

1. Motivation

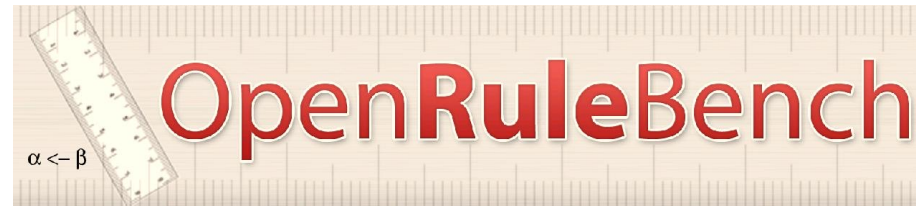
1. Motivation

Why a rule benchmarking framework?

Open source

not maintained
anymore

last report
2011



set of
test cases

integration of new
rule engines

set of
bash scripts

1. Motivation

Why a rule benchmarking framework?

relatively new
2017

meta-model
translated for
rule-engines

translation from
meta-model required

Bobek, S., & Misiak, P. (2017, June). **Framework for benchmarking rule-based inference engines.**
In *International Conference on Artificial Intelligence and Soft Computing* (pp. 399-410). Springer, Cham.

fully automated
evaluation

only generated
test cases
supported

bash script for
rule-engine

2. RUBEN

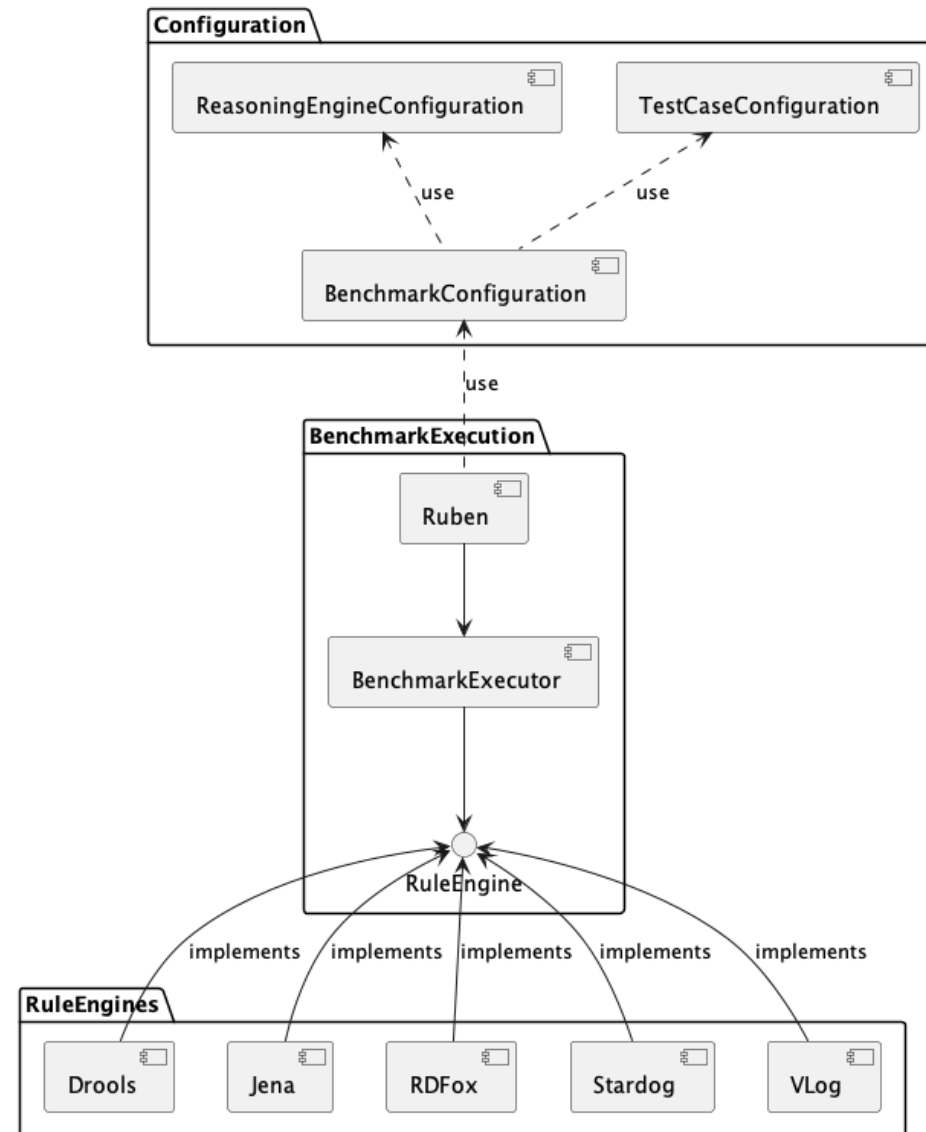
2. RUBEN

General

- **Rule Engine Benchmarking Framework**
- Simple interface for rule engines
- Benchmark execution without scripts
- Results in various formats
 - CSV
 - JSON
- Test cases from OpenRuleBench
- Yearly evaluation
- Aim to encourage the community for contribution

2. RUBEN

Components



2. RUBEN

Configuration

Property	Description
<i>name</i>	Configuration name
<i>engines</i>	Engines to be used for the evaluation.
<i>testCases</i>	Test cases to be included in the evaluation.
<i>testDataPath</i>	Path to the folder containing the required test data.

General Configuration

2. RUBEN

Configuration

Property	Description
<i>name</i>	Configuration name
<i>engines</i>	Engines to be used for the evaluation.
<i>testCases</i>	Test cases to be included in the evaluation.
<i>testDataPath</i>	Path to the folder containing the required test data.

General Configuration

Property	Description
<i>name</i>	Name of the reasoning engine.
<i>classpath</i>	Refers to the implementation of the rule engine.
<i>settings (optional)</i>	Additional settings (key-values) for the rule engine.

Rule Engine Configuration



2. RUBEN

Configuration

Property	Description
<i>name</i>	Configuration name
<i>engines</i>	Engines to be used for the evaluation.
<i>testCases</i>	Test cases to be included in the evaluation.
<i>testDataPath</i>	Path to the folder containing the required test data.

General Configuration

Property	Description
<i>name</i>	Name of the reasoning engine.
<i>classpath</i>	Refers to the implementation of the rule engine.
<i>settings (optional)</i>	Additional settings (key-values) for the rule engine.

Rule Engine Configuration

Property	Description
<i>testCategory</i>	Category the test belongs to.
<i>testCaseIdentifier</i>	Unique identifier for the given test case.
<i>testName</i>	Name for tests within a given test category.

Test Case Configuration

2. RUBEN

Rule Engine Interface

Method	Description
<i>cleanUp()</i>	Clean up after the evaluation of a test case.
<i>executeQuery()</i>	Executes a single query.
<i>materialize()</i>	Starts the materialization or reruns it.
<i>prepare(testDataPath, testCase)</i>	Load relevant data and rules.
<i>shutdown()</i>	Stop all processes initiated by the rule engine. Should delete all temporary data.

3. Test Cases

3. Test Cases

Liang, S., Fodor, P., Wan, H., & Kifer, M. (2009, April). OpenRuleBench: An analysis of the performance of rule engines. In *Proceedings of the 18th international conference on World wide web* (pp. 601-610).

- Large Join Tests
 - Join1
 - Join2
 - LUBM-derived tests
 - Mondial
 - DBLP
- Datalog Recursion
 - Transitive closure
 - Same-generation siblings problem
 - WordNet
 - Wine Ontology
- Default negation
 - Predicate-stratified negation

4. Evaluation

4. Evaluation

Engines

Apache Jena

a Java-based framework including two rule engines (bottom-up and top-down)

(Drools)

a bottom-up engine based on the Rete algorithm

RDFOx

is implemented in C++ providing modules for Java and Python and supports materialisation-based parallel datalog reasoning

Stardog

is implemented in Java providing extensive reasoning capabilities, including datalog evaluation

VLog

is implemented in C++ and provides an efficient Datalog engine for large knowledge graphs supporting RDF, OWL, and SPARQL

4. Evaluation

Results – Without Materialization

Table 5

Large joins, join1, no query bindings (time in seconds)

query	a(X,Y)	b1(X,Y)	b2(X,Y)
Size	50000	50000	50000
Drools	error	error	error
Jena	timeout	104.9	2.7
Stardog	exception	37.2	1.4

Table 6

Datalog recursion, same generation, no query bindings (time in seconds)

size	6000	6000	24000	24000
Cyclic data	no	yes	no	yes
Drools	error	error	error	error
Jena	53.8	61	189.1	239.1

4. Evaluation

Results - Without Materialization

Table 7

Datalog recursion, transitive closure, no query bindings (time in seconds)

size	50000	50000	500000	500000
Cyclic data	no	yes	no	yes
Drools	error	error	error	error
Jena	8.9	28.6	76.7	346.5
Stardog	10	27.8	66.6	262.5

4. Evaluation

Results – With Materialization

preliminary

	Large Joins	Datalog Recursion	Datalog Recursion	Datalog Recursion	Datalog Recursion
	Join 1	Same Generation	Same Generation	Transitive Closure	Transitive Closure
	50000	24000	24000	500000	500000
		No cyc	Cyc	No cyc	Cyc
RDFOx	140	7.3	8.6	6.7	29
VLog	394.7	25.5	28.8	16.8	62.1

Materialization time in seconds

4. Evaluation

Drools



Help us fixing the Drools test cases:

<https://github.com/kev-ang/RUBEN>

5. Conclusion and Future Work

5. Conclusion and Future Work

- **Rule Engine Benchmarking Framework (RUBEN)**
 - Java
 - Simple interface
 - OpenRuleBench test cases
 - Subset of the OpenRuleBench rule engines
- Finalize materialization support
- Common rule format in addition

5. Conclusion and Future Work

- We plan a yearly evaluation!
- We encourage you to help us in extending the set of
 - rule engines
 - test cases

Thank you!



Kevin Angele
kevin.angele@sti2.at
<https://sti2.at>

kevin.angele@onlim.com
<https://onlim.com>



Jürgen Angele
juergen.angele@adesso.de
<https://adesso.de>



Umutcan Simsek
umutcan.simsek@sti2.at
<https://sti2.at>



Dieter Fensel
dieter.fensel@sti2.at
<https://sti2.at>



www.sti2.at

www.uibk.ac.at