International Workshop on Semantic Big Data (SBD 2016)
in conjunction with the 2016 ACM SIGMOD Conference in San Francisco, USA

Querying and reasoning over large scale building datasets: an outline of a performance benchmark

Pieter Pauwels, Tarcisio Mendes de Farias, Chi Zhang,
Ana Roxin, Jakob Beetz, Jos De Roo, Christophe Nicolle

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

Agenda

Introduction
- Context description
- Problem identified

Testing environment
- ifcOWL and building models
- Rules and queries
- Triple stores

Results
- Query performance
- Additional findings

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

Context description

- The architectural design and construction domains work on a daily basis with massive amounts of data.
- In the context of BIM, a neutral, interoperable representation of information consists in the Industry Foundation Classes (IFC) standard
  - Difficult to handle the EXPRESS format
- Semantic Web technologies have been identified as a possible solution
  - Semantic data enrichment
  - Schema and data transformations
- A semantic approach involves 3 main components:

Schema (Tbox)	Instances (ABox)	Rules (RBox)
• OWL ontology • Information structure	• Assertions • Respects schema definition	• If-Then statements • Involving elements from the ABox and theTBox

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

Problem identified

- Different implementations exist for the components (TBox, ABox, RBox) of such Semantic approach
  - Diverse reasoning engines
  - Diverse query processing techniques
  - Diverse query handling
  - Diverse dataset size
  - Diverse dataset complexity

Expressiveness vs. performance

- Missing an appropriate rule and query execution performance benchmark

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

Performance benchmark variables

- Main components

Schema (TBox)	Instances (ABox)	Rules (RBox)
• ifcOWL	• 369 ifcOWL-compliant building models	• 68 data transformation rules

- These elements are implemented into 3 different systems
  - SPIN (SPARQL Inference Notation) and Jena
  - EYE
  - Stardog

- An ensemble of queries is addressed to the so-created systems

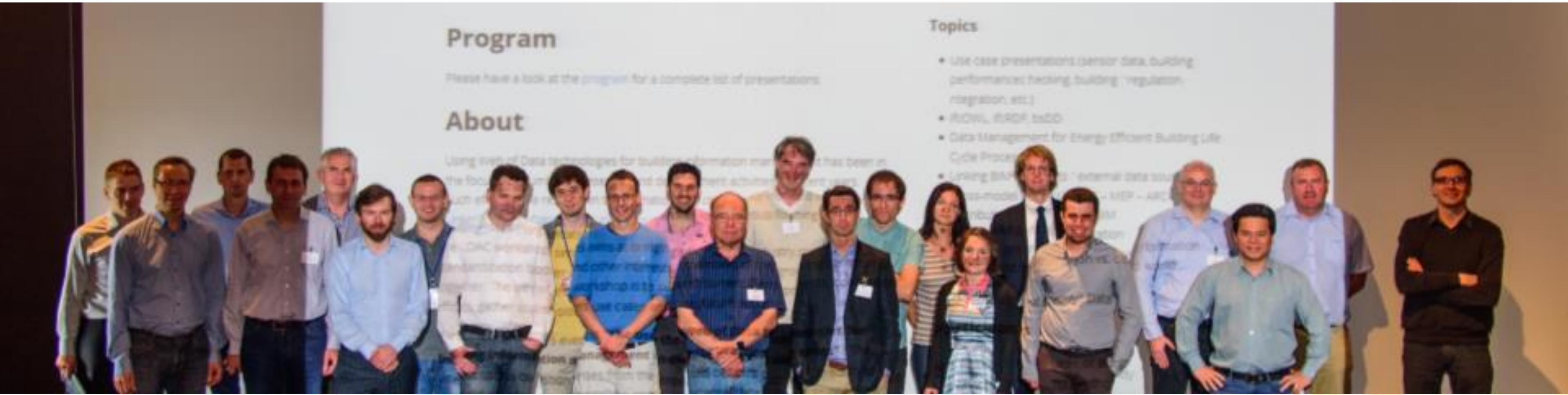
Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

TBox – the ifcOWL ontology

- All building models are encoded using the ifcOWL ontology
  - Built up under the impulse of numerous initiatives during the last 10 years

- The ontology used is the one that is made publicly available by the **buildingSMART Linked Data Working Group** (LDWG)
  - http://ifcowl.openbimstandards.org/IFC4#
  - http://ifcowl.openbimstandards.org/IFC4_ADD1#
  - http://ifcowl.openbimstandards.org/IFC2X3_TC1#
  - http://ifcowl.openbimstandards.org/IFC2X3_Final#

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be



Call for papers – special issue in SWJ

- Semantic Web Journal – Interoperability, Usability, Applicability
  - http://www.semantic-web-journal.net
- Special issue on **"Semantic Technologies and Interoperability in the Built Environment"**

Ontologies for AEC/FM	Linking BIM models to external data sources	Multiple scale integration through semanitc interoperability	Multilingual data access and annotation	Query processing, query performance
Semantic-based building monitoring systems	Reasoning with building data	Building data publication strategies	Big Linked Data for building information	

- Important dates
  - March, 1st 2017 – paper submission deadline
  - May 1st 2017 – notification of acceptance

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

ifcOWL Stats

Axioms		21306
Logical Axioms		13649
Classes		1230
Object properties		1578
Data properties		5
Individuals	1627	
DL expressivity		SROIQ(D)
SubClassOf axioms		4622
EquivalentClasses axioms		266
DisjointClasses axioms	2429	
SubObjectPropertyOf axioms		1
InverseObjectProperties axioms	94	
FunctionalObjectProperty axioms	1441	
TransitiveObjectProperty axioms	1	
ObjectPropertyDomain axioms	1577	
ObjectPropertyRange axioms		1576
FunctionalDataProperty axioms	5	
DataPropertyDomain axioms		5
DataPropertyRange axioms		5

Pieter Pauwels and Walter Terkaj, EXPRESS to OWL for construction industry: towards a recommendable and usable ifcOWL ontology. Automation in Construction 63: 100-133 (2016).

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

ABox – Building sets

- Some BIM models are publicly available (364), whereas other are undisclosed (5)

Building information models created with different BIM modelling environments → Exported to IFC2x3 → Transformed into ifcOWL-compliant RDF graphs using a publicly available converter

BIM environment	Number of files
Tekla Structures	227 (61,5%)
unknown or manual	38 (10,3%)
Autodesk Revit	27 (7,3%)
Xella BIM	15
Autodesk AutoCAD	12
iTConcrete	9
SDS	8
Nemetschek AllPlan	7
GraphiSoft ArchiCAD	5
Various others	21

IFC instances	Average file size	Number of files
0 – 500,000	0 – 30 MB	321
500,000 – 2,000,000	30 – 100 MB	37
> 2,000,000	> 100 MB	11

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

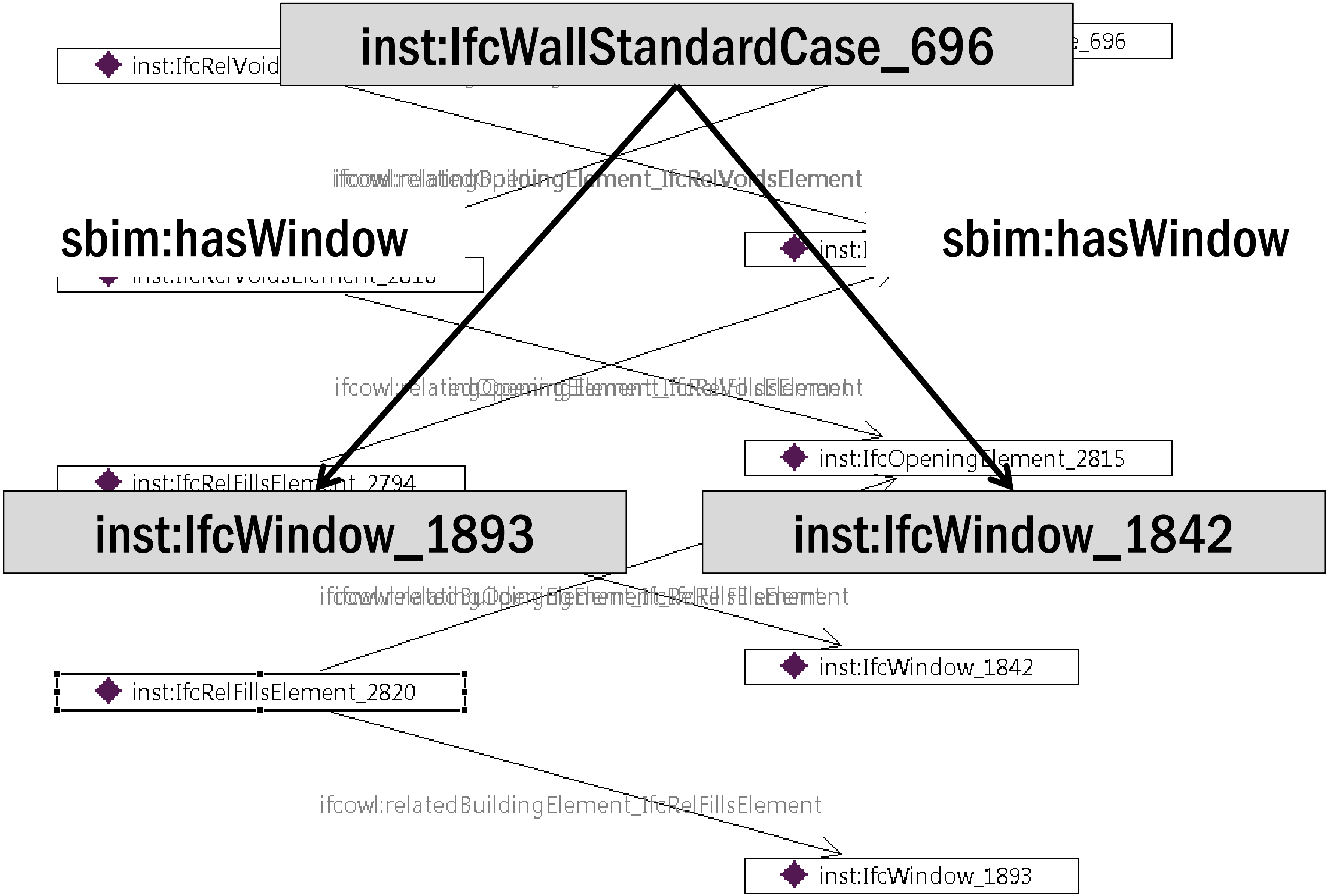
RBox – Data transformation rules

- Need for a representative set of rewrite rules
- 68 manually built rules
- Classified in several rule sets according to their content

Rule Set (RS)	Description
RS1	Contains 2 rules for rewriting property set references into additional property statements **`sbd:hasPropertySet`** and **`sbd:hasProperty`**. This is a small, yet often used rule set that can be used in many contexts to simplify querying and data publication of common simple properties attached to IFC entity instances.
RS2	Includes 31 rules, all involving subtypes of the **`IfcRelationship`** class (e.g. **`ifcowl:IfcRelAssigns`**, **`ifcowl:IfcRelDecomposes`**, **`ifcowl:IfcRelAssociates`**, **`ifcowl:IfcRelDefines`**, **`ifcowl:IfcRelConnects`**)
RS3	Contains 3 rules related to handling lists in IFC.
RS4	Contains one rule that allows wrapping simple data types.
RS5	Consists of 20 rules for inferring single property statements **`sbd:hasPropertySet`** and **`sbd:hasProperty`**.
RS6	Extends RS5 and RS1 with 6 additional rules for inferring whether an objet is internal or external to a building.
RS7	Contains 7 rules dealing with the (de)composition of building spaces and spatial elements.

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

ifcOWL Example Transformation



Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

Implementation

SPIN + Jena TDB

- Implemented based on the open source APIs of Topbraid SPIN (SPIN API 1.4.0) and Apache Jena (Jena Core 2.11.0, Jena ARQ 2.11.0, Jena TDB 1.0.0)
- Rules are written with Topbraid Composer Free version, and they are exported as RDF Turtle files.
- A small Java program is implemented to read RDF models, schema, rules from the TDB store and query data.
- All the SPARQL queries are configured using the Jena org.apache.jena.sparql.algebra package
- To avoid unnecessary reasoning processes, in this test environment only the RDFS vocabulary is supported.

EYE

- Version 'EYE-Winter16.0302.1557' ('SWI-Prolog 7.2.3 (amd64): Aug 25 2015, 12:24:59').
- EYE is a semi-backward reasoner enhanced with Euler path detection.
- As our rule set currently contains only rules using =>, forward reasoning will take place.
- Each command is executed 5 times
- Each command includes the full ontology, the full set of rules and the RDFS vocabulary, as well as one of the 369 building model files and one of the 3 query files.
- No triple store is used: triples are processed directly from the considered files.

Stardog

- 4.0.2 Stardog semantic graph database (Java 8, RDF 1.1 graph data model, OWL2 profiles, SPARQL 1.1)
- OWL reasoner + rule engine.
- Support of SWRL rules, backward-chaining reasoning
- Reasoning is performed by applying a query rewriting approach (SWRL rules are taken into account during the query rewriting process).
- Stardog allows attaining a DL-expressivity level of SROIQ(D).
- In this approach, SWRL rules are taken into account during the query rewriting process.

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

Queries

- We have built a limited list of 60 queries, each of which triggers at least one of the available rules.
- As we focus here on query execution performance, the considered queries are entirely based on the right-hand sides of the considered rules.
- 3 queries:
  - Q1 a simple query with little results,
  - Q2 a simple query with many results,
  - and Q3 a complex query that triggers a considerable number of rules

Query	Query Contents
Q1	`?obj sbd:hasProperty ?p`

Q2	`?point sbd:hasCoordinateX ?x .`
	`?point sbd:hasCoordinateY ?y .`
	`?point sbd:hasCoordinateZ ?z`

Q3	`?d rdf:type sbd:ExternalWall`

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

Test environment

- In one central server
  - Supplied by the University of Burgundy, research group CheckSem,
  - Following specifications: Ubuntu OS, Intel Xeon CPU E5-2430 at 2.2GHz, 6 cores and 16GB of DDR3 RAM memory

- 3 Virtual Machines (VMs) were set up in this central server
  - SPIN VM (Jena TDB), EYE VM (EYE inference engine), Stardog VM (Stardog triplestore)

- The VMs were managed as separate test environments and
  - Each of these VMs had 2 cores out of 6 allocated
  - Each contained the above resources (ontologies, data, rules, queries).

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

Results

- Queries applied on 6 hand-picked building models of varying size
- In the SPIN approach
  - For Q1 and Q2, the execution time = backward-chaining inference process + actual query execution time
  - For Q3, execution time = query execution time itself
- In the EYE approach
  - Networking time is ignored
- In the Stardog approach
  - Execution time = backward-chaining inference + actual query execution time

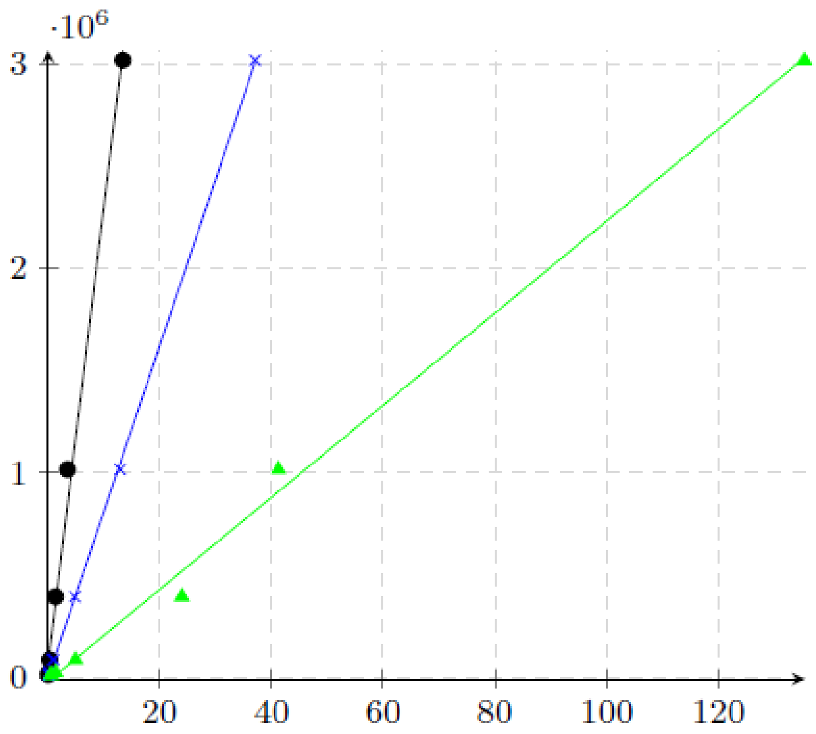
Query	Building Model	SPIN (s)	EYE (s)	Stardog (s)
Q1 (simple, little results)	BM1	135,36	37,11	**13,44**
	BM2	1,47	0,29	**0,17**
	BM3	24,01	4,87	**1,4**
	BM4	41,28	12,95	**3,55**
	BM5	4,99	1,05	**0,33**
	BM6	0,55	0,16	**0,08**
Q2 (simple, many results)	BM1	46,17	**2,10**	6,82
	BM2	92,03	**4,20**	15,83
	BM3	82,68	**4,12**	15,28
	BM4	19,93	**1,04**	2,81
	BM5	3,69	**0,21**	1,36
	BM6	0,74	**0,045**	1,00
Q3 (complex)	BM1	0,001	**0,001**	0,07
	BM2	0,006	**0,003**	0,12
	BM3	**0,002**	0,003	0,31
	BM4	0,005	**0,001**	0,20
	BM5	0,006	**0,013**	0,20
	BM6	0,001	**0,001**	0,13

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

Query time related to result count

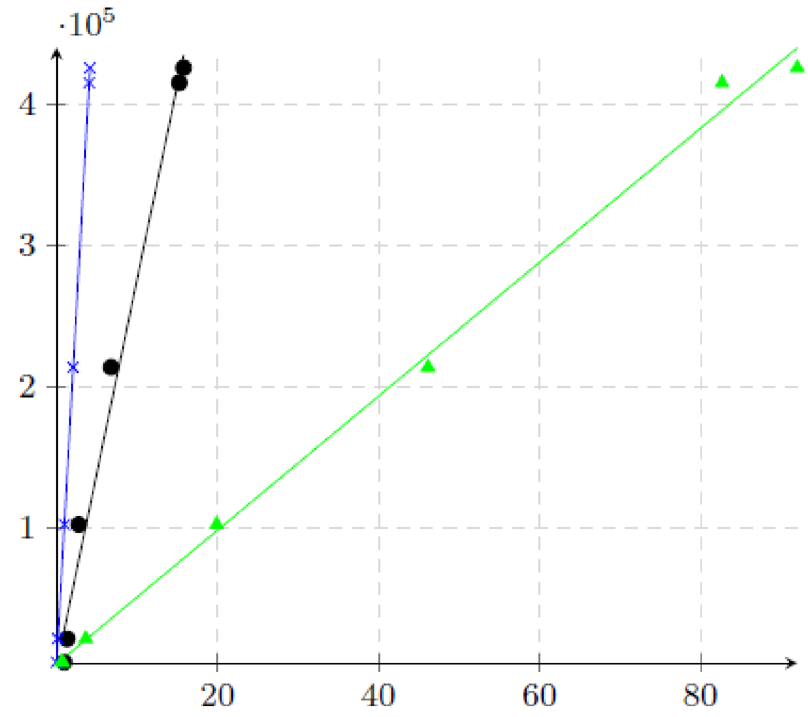
For Q1 for each of the considered approaches

(green = SPIN; blue = EYE; black = Stardog)



For Q2 for each of the considered approaches

(green = SPIN; blue = EYE; black = Stardog)



Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

Additional findings

Indexing algorithms, query rewriting techniques, and rule handling strategies

- The three considered procedures are quite far apart from each other, explaining the considerable performance differences, not only between the procedures, but also between diverse usages within one and the same system.
- Algorithms and optimization techniques used for each approach aren't entirely used: differences in indexation algorithms, query rewriting techniques and rule handling strategies used.

Forward- versus backward-chaining

- The disadvantage of forward-chaining reasoning process is that millions of triples can be materialized (EYE, SPIN for Q1 and Q2)
- Using backward-chaining reasoning allows avoiding triple materialization, thus saving query execution time (Stardog, SPIN for Q3).

Type of data in the building model

- Query Q3 triggers a rule that in turn triggers several other rules in the rule set. If the first rule does not fire, however, the process stops early.
- Query Q2, however, fires relatively long rules. It takes more time to make these matches in all three approaches.

Impact of the triple store

- Loading files in memory at query execution time leads to considerable delays.

Impact of the number of output results

- Linear relation: the more results are available, the more triples need to be matched, leading to more assertions.

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

Conclusion and future work

- Comparison of 3 different approaches
  - SPIN, EYE and Stardog
- 3 queries applied over 6 different building models

- Future work consists in
  - Specifying more this initial performance benchmark with additional data and rules
  - Executing additional queries on the rest of the set of building models
  - Comparing results on a wider scale:
    - — for the individual approaches separately,
    - — as well as with other approaches not considered here.

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

International Workshop on Semantic Big Data (SBD 2016)
in conjunction with the 2016 ACM SIGMOD Conference in San Francisco, USA

Thank you for your attention.

Pieter Pauwels, Tarcisio Mendes de Farias, Chi Zhang,
Ana Roxin, Jakob Beetz, Jos De Roo, Christophe Nicolle

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be

Query	Building Model	SPIN	EYE	Stardog
Q1	BM1	135,36	37,11	13,44
	BM2	1,47	0,29	0,17
	BM3	24,01	4,87	1,4
	BM4	41,28	12,95	3,55
	BM5	4,99	1,05	0,33
	BM6	0,55	0,16	0,08
Q2	BM1	46,17	2,10	6,82
	BM2	92,03	4,20	15,83
	BM3	82,68	4,12	15,28
	BM4	19,93	1,04	2,81
	BM5	3,69	0,21	1,36
	BM6	0,74	0,045	1,00
Q3	BM1	0,001	0,001	0,07
	BM2	0,006	0,003	0,12
	BM3	0,002	0,003	0,31
	BM4	0,005	0,001	0,20
	BM5	0,006	0,013	0,20
	BM6	0,001	0,001	0,13

Ana ROXIN – ana-maria.roxin@u-bourgogne.fr
Pieter PAUWELS – Pieter.pauwels@ugent.be