LDBC

Collaborative Project

FP7 – 317548

D6.6.4 Standardisation Report

Coordinator: Vladimir Alexiev (ONTO)
With contributions from: Atanas Kiryakov (ONTO), Orri Erling (OpenLink)

1st Quality reviewer: Hugh Wiliams (OpenLink)
2nd Quality reviewer: Venelin Kotsev (ONTO)

Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	M30
Actual delivery date:	M30
Version:	1.0
Total number of pages:	14
Keywords:	Standardization, benchmarking, linked data management

Abstract

This report describes the standardization activities of LDBC and the relevant situation regarding standardization in the different technological areas of LDBC. The mission of the LDBC project is to create a benchmarking authority for RDF databases, graph databases, linked data management and graph analytics. It aims to set up the relevant prerequisites: benchmarking methodology, set of concrete benchmarks and benchmarking community. The main interaction with existing standardization organizations is related to adopting relevant standards and specifications for the benchmarks developed in LDBC as well as for related tooling. The standardization activities can be summarized as follows: setting up an industry driven benchmarking organization; ensuring compliance with W3C Semantic Web standards, though the usage of and the support for the Sesame framework; developing benchmarking ontology that allows for structuring a repository of benchmark results.

Executive summary

This report describes the standardization activities of LDBC and the relevant situation regarding standardization in the different technological areas of LDBC.

The mission of the LDBC project is to create a benchmarking authority for RDF databases, graph databases, linked data management and graph analytics. It aims to set up the relevant prerequisites: benchmarking methodology, set of concrete benchmarks and benchmarking community. This is the reason to have limited resources associated with formal standardization activities in the project.

As far as existing standardization organizations are concerned, the main interaction with them is related to adopting relevant standards and specifications for the benchmarks developed in LDBC as well as for related tooling.

The standardization activities can be summarized as follows:

- Setting up an industry driven benchmarking organization. By the end of March 2015 members of LDBC include the necessary critical mass of vendors in both technology areas: Neo Technology, Ontotext AD, OpenLink Software, Sparsity Technologies, SYSTAP, IBM, Oracle, SPARQL City.
- Compliance with W3C Semantic Web standards for data representation, schema and ontology definition, and query languages. Such compliance is guaranteed through usage of the Sesame APIs; one of the members of the consortium is also major supporter of this open-source project.
- Benchmarking ontology that allows for structuring a repository of benchmark results. Such was designed and implemented in OWL.

Document Information

IST Project Number	FP7 - 317548	**Acronym**	LDBC
Full Title	LDBC		
Project URL	http://www.ldbc.eu/		
Document URL	http://www.ldbc.eu:8090/display/PROJECT/Deliverables/		
EU Project Officer	Carola Carstens		

Deliverable	**Number**	D6.6.4	**Title**	
Work Package	**Number**	WP6	**Title**	

Date of Delivery	**Contractual**	M30	**Actual**	M30
Status	version 1.0		final □	
Nature	prototype □ report ☑ dissemination □			
Dissemination level	public ☑ consortium □			

Authors (Partner)				
Responsible Author	**Name**	Vladimir Alexiev	**E-mail**	vladimir.alexiev@ontotext.com
	Partner	ONTO	**Phone**	

Abstract (for dissemination)	This report describes the standardization activities of LDBC and the relevant situation regarding standardization in the different technological areas of LDBC. The mission of the LDBC project is to create a benchmarking authority for RDF databases, graph databases, linked data management and graph analytics. It aims to set up the relevant prerequisites: benchmarking methodology, set of concrete benchmarks and benchmarking community. The standardization activities can be summarized as follows: setting up an industry driven benchmarking organization; ensuring compliance with W3C Semantic Web standards, though the usage of and the support for the Sesame framework; developing benchmarking ontology that allows for structuring a repository of benchmark results.
Keywords	Standardization, benchmarking, linked data management

Version Log			
Issue Date	**Rev. No.**	**Author**	**Change**
02.02.2015	0.1	Vladimir Alexiev	
12.03.2015	0.2	Vladimir Alexiev	Adding section on benchmark results ontology
09.04.2015	0.3	Vladimir Alexiev	Final version for internal review
09.04.2015	0.4	Atanas Kiryakov	Additions related to Sesame
10.04.2015	1.0	Atanas Kiryakov	Corrections based on internal review

Table of Contents

Abbreviations

Blueprints - TinkerPop's property graph model interface

CYPHER - Neo4J query language, a declarative language for describing patterns in graphs

Gremlin - TinkerPop's domain specific language for traversing property graphs

EARL - Evaluation and Report Language

IRI - International Resource Identifier

JSON - JavaScript Object Notation

JSON-LD - JSON for Linked Data

NLP - Natural Language Processing

NT - NTriples syntax

OWL - Web Ontology Language

RDF - Resource Description Framework

RDFa - RDF in (HTML) Attributes

RDFS - RDF Schema

RIF - Rule Interchange Format

SPARQL - SPARQL Protocol and RDF Query Language

SPIN - SPARQL Inferencing Notation

SWRL - Semantic Web Rule Language

Turtle - Terse RDF Triple Language

W3C - World Wide Web Consortium

XML - Extensible Markup Language

XSD - XML Data Types

1 Introduction

This report describes the standardization activities of LDBC and the relevant situation regarding standardization in the different technological areas of LDBC.

The mission of the LDBC project is to create a benchmarking authority for RDF databases, graph databases, linked data management and graph analytics. The EC funded project LDBC is aimed bootstrapping an industry-driven organization for benchmarking. For this purposes, it aims to set up the relevant prerequisites: benchmarking methodology, set of concrete benchmarks and benchmarking community. The overall project objective is to develop an authority that standardizes benchmarks for RDF triplestores and graph databases. This is the reason to have limited resources associated with formal standardization activities in the project.

As far as existing standardization organizations are concerned, the main interaction with them is related to adopting relevant standards and specifications for the benchmarks developed in LDBC as well as for related tooling. Note that there is difference between the two main technology areas covered by the project: triplestores and ThinkerPop-based graph databases. While the former are based on a set of W3C standards for representation of data in the Semantic Web, the later are built around set of proprietary specifications, which overtime became defacto standards.

The standardization activities are presented in this report as follows:

- setting up an industry driven benchmarking organization (section 2)
- ensuring compliance with W3C Semantic Web standards for data representation, schema and ontology definition, and query languages (sections 3 and 4)
- benchmarking ontology that allows for structuring a repository of benchmark results (section 5)

2 Benchmark Institutionalization

The main area of standardization addressed by LDBC are the benchmarks themselves, and the established process for benchmark institutionalization, which includes:

- Established interest and input by an industry-relevant user group
- A large amount of technical work related to benchmarking infrastructure and concrete benchmarks
- Definition of benchmark execution and reporting procedures, including Full Disclosure Report template
- Building consensus amongst vendors that the benchmarks are fair and significant

Managing the standardization aspects of LDBC's benchmarking mission is a continuous task. At the end of the project we can say that we have succeeded in establishing the LDB Council as a viable organization. The following industry members have joined:

- Neo Technology, http://neo4j.com/ ;
- Ontotext AD, http://www.ontotext.com ;
- OpenLink Software, http://www.openlinksw.com/ ;
- Sparsity Technologies, http://www.sparsity-technologies.com/ ;
- SYSTAP (the developers of BlazeGraph, previously known as BigData), http://www.blazegraph.com/ ;
- IBM, http://www.ibm.com ;
- Oracle, http://www.oracle.com ;
- SPARQLcity, http://www.sparqlcity.com/ .

Significantly, the two large IT companies (IBM and Oracle) participate with both their RDF and their Graph database divisions. This way, more than half of the leading triplestore vendors joined LDBC. Few others have expressed interest to join soon, such is the case with Clark & Parsia (the developer of StarDog), MarkLogic and Franz Inc. (the developer of AllegroGraph). On the graph database side, LDBC includes a critical mass of the leading vendors, namely Neo Technology, Sparsity, SYSTAP, ORACLE and IBM.

The two established benchmarks (SPB and SNB) provide research and development targets to vendors and produce results that are naturally useful to users. This will lead to their de-facto adoption as "standards" by relevant user communities. For example, Ontotext prospective clients in the publishing industry are running SPB, and a reduced version of the benchmark is used in Ontotext GraphDB's regular performance testing.

The LDB Council is well poised to "adopt" and institutionalize other interesting benchmarks, since it has the "weight" and an established process for turning benchmarks from research artifacts to industry "go to" sources of information regarding the performance and functionality of RDF and graph databases

Another aspect is potential cooperation of the LDB Council with other benchmarking bodies such as the TPC. Peter Boncz (the scientific director of the project) was in contact with the TPC organization about this. We believe that with the expanding importance of NoSQL databases, the mutual interests of the two organizations will increase.

3 Compliance With Semantic Web Standards

The World Wide Web Consortium (W3C) provides an extensive set of Semantic Web-related standards for data representation, schema and ontology specification, query language and more.

3.1 Overview of Semantic Web Standards

Below we provide a summary of the relevant standards:

- RDF: Resource Description Framework [1]: an abstract graph data model consisting of triples "subject predicate object". It depends on web architecture standards such as International Resource Identifiers (IRIs) for expressing resources and properties, XML Data Types (XSD) for typing literals, BCP 47 and IANA language registry for languages of literals. It can be expressed in a number of concrete syntaxes, some of which were standardized only recently with RDF 1.1.
  - RDF/XML: the original RDF syntax, mandatory but most verbose
  - NT: NTriples – simple line-oriented syntax
  - Turtle: Terse RDF Triple Language: allows a number of abbreviations including prefixes (namespaces) and blank nodes
  - RDF/JSON: an older way to express RDF in JSON, superseded by JSON-LD
  - JSON-LD: JSON for Linked Data: a way to express RDF in JSON, which is easier for web applications to consume
  - RDFa: RDF in Attributes: a way to express RDF directly in HTML, which together with Microdata and Microformats is popular with public-facing web sites for Search Engine Optimization (especially when used in conjunction with the schema.org ontology)
- RDF extension for named graphs. Turns the triples into quads by adding a "graph" IRI (also called "context"). A de-facto standard for perhaps 8 years and implemented in most RDF input/output libraries and triple stores, this was standardized only recently with RDF 1.1. Similar to RDF, it offers a number of concrete syntaxes:
  - TriG: an extension of the Turtle syntax
  - NQuads: an extension of the NTriples syntax
  - TriX: representing quads in XML
- RDFS: RDF Schema [2]: adds to RDF simple class-based and sub-property reasoning
- OWL: Web Ontology Language [3]: adds a lot more powerful ontology constructs and reasoning, based on Description Logics. This has been refined over two major iterations (OWL and OWL2) that together define the following profiles or sub-languages of increasing reasoning power and difficulty (from polynomial to multi-exponential and intractable):
  - OWL Lite: slightly more than RDFS
  - OWL 2 RL: rule-based reasoning, appropriate for reasoning with web data
  - OWL 2 QL: tailored for very large amounts of data (A-Box reasoning). Can provide an ontological access layer over relational databases
  - OWL 2 EL: tailored for very large ontologies (T-Box reasoning)
  - OWL 2 DL: includes the full power of Description Logics
  - OWL Full: intractable
- Like RDF, OWL supports several transmission formats (concrete syntaxes):
  - OWL is fully expressible in RDF, so any RDF format can be used. RDF/XML is mandatory though very verbose. For simple ontologies, most often Turtle is used.

  - Manchester Notation is derived from formal notations in Description Logics and is a lot more concise for complex axioms, e.g. OWL Restrictions.
  - OWL Functional Syntax is used in the standard
  - OWL XML is a dedicated XML format
- SPARQL: SPARQL Protocol and RDF Query Language [4]. Currently at version 1.1 this is a large family of standards that includes:
  - Query Language: patterned after SQL, but includes 4 query forms instead of 1 (ASK, SELECT, CONSTRUCT, DESCRIBE) and capabilities for graph pattern matching. SPARQL 1.1 includes many enhancements over 1.0, including aggregates, sub-queries, negation, etc
  - Update: update operations, including INSERT, DELETE, CLEAR (for triples), LOAD (for RDF files), and CREATE, DROP, COPY, MOVE, ADD (for graphs)
  - Service Description: a way for a SPARQL endpoint to advertise its capabilities and result formats in a machine readable way (the SD ontology), as well as the datasets that it serves (through binding to VOID)
  - Federated Query: a way for one SPARQL query to invoke a query on another endpoint
  - Query Result Formats: JSON, CSV, TSV, XML and content negotiation. These standards apply to the tabular query forms (ASK, SELECT) while the graph query forms (CONSTRUCT, DESCRIBE) return RDF in any supported format
  - Entailment Regimes: instruct the repository what inference to apply when answering the queries.
  - Protocol: REST access to SPARQL endpoints using HTTP
  - Graph Store HTTP Protocol: a way to access an RDF repository through simple HTTP operations (GET, PUT) rather than complex queries

The RDF benchmarks considered by LDBC (Semantic Publishing Benchmark, Instance Matching Benchmark, Reasoning Benchmark) comply with all relevant RDF standards. (SPB is institutionalized, while the others are researched or developed, but not yet institutionalized.) In particular:

- SPB uses standard-compliant SPARQL queries, which are designed to be both realistic with respect to business needs, and exercise the repository in a significant way
- The Reasoning Benchmark exercises compliance with respect to OWL semantics, both for correctness, and also for performance

3.2 Using the Sesame Framework for Compliance with W3C Standards

There are two established open-source API frameworks that support these set of standards:

- Sesame OpenRDF, http://rdf4j.org/ – started as project of Aduna Software, recently it has been transformed into Eclipse Foundation project;
- Jena, https://jena.apache.org/ – started as project of HP Labs, Bristol, today is an Apache Software Foundation project.

In LDBC, compliance with the W3C standards is ensured through usage of the Sesame framework. Sesame is used in the SPB benchmark for the following purposes:

- RDF parsers are used to generate the datasets – both Reference Data and Metadata;
- SPARQL parser is used for validation of the query syntax;
- SPARQL Protocol (popular as SPARQL end-point) specification is used for communication between the benchmark drivers and the engines (SUT, system under test).

In order to guarantee the continuing development of Sesame in compliance with the evolving W3C specifications, Ontotext AD – one of the member of LDBC, continuously supports the coordinating

contributor to this open-source project. This support is not charged to the project, partly because this contributor leaves outside the European Union.

3.3 Potential RDF- and SPARQL-Related Standardization

The W3C community process for development and adoption of specification is fairly robust, making sure that, once adopted, specifications cover broad range of requirements from various stakeholders. As a result this process is relatively slow and heavy-weight. One member of the EC project LDBC consortium (Ontotext AD) is W3C member and has priority access to the standardization process of W3C. Two members of the LDB Council (the organization established to drive the initiative after the end of the EC project) are very influential members of W3C – these are IBM and ORACLE. Based on the experience of these W3C members with standardization it was decided that the most efficient way to promote new features for extension of existing specifications is to prove that those already have broad industrial adoption.

The strategy for standardization of SPARQL extension that was selected was to design the SPB benchmark such a way that:

- it allows an engine to pass the benchmark using the existing SPARQL standard, while at the same time
- demonstrate that specific extensions of SPARQL (e.g. FTS and geo-spatial) allow for much cleaner definition of the queries and for more efficient query evaluation.

When implementing the SPB benchmark vendors have the freedom to use proprietary SPARQL extensions, given that their implementation returns correct results, according to the validation mechanisms incorporated in SPB, and they can pass an audit of the benchmark results, according the auditing rules.

This way vendors who do not support the relevant extension can implement SPB based on the general, standard-compliant set of queries, while at the same time vendors who support such extensions can demonstrate the benefit of using those. As a result, we expect that those extensions that really prove to be beneficial will be implemented by a critical mass of the vendors and this will provide arguments to approach W3C with request to standardize those extensions.

Follows an overview of several areas of potential extensions of SPARQL and other specifications, which have seen some movement towards standardization and are considered in the design of the SPB benchmark. They involve important features of industrial RDF repositories and are strong candidates for inclusion in enterprise benchmarks. They are covered to some extent in D4.4.2 Benchmark Design for Reasoning. Follows a quick summary of such candidates for extension.

Full-Text Search and faceting. Every repository implements this in a custom way, with widely varying levels of elaboration. Interesting features include:

- Configuring what to index: which nodes to include, and how to collect text for each node to index (FTS molecule);
- Optimized handling of owl:sameAs synonyms (i.e. set of URIs which are declared to be equivalent connecting them through owl:sameAs mapping);
- FTS query language features, such as Boolean, fuzzy, etc.
- Plugging NLP resources such as specific language analyzers, stemmers etc.
- Faceting, i.e. dynamic recalculation of number of matching nodes that relate to some selected values;
- Ranking based on relevance scoring;
- Hit highlighting and returning matching excerpts.

Geo-spatial querying. The GeoSPARQL [5] standard integrates relevant standards by the Open Geospatial Consortium (OGC), including geometries and features, region algebras, querying functions and predicates. OpenLink Software, as part of its involvement in the GeoKnow EU project (http://geoknow.eu) has performed major updates in its Geo-spatial support in Virtuoso to be OGC and GeoSPARQL compliant. With these standards being use in the Geo-spatial benchmarking tasks in GeoKnow some of which has been adopted by the SNB benchmark.

Rule languages. There are many options, but relatively few examples of deployment of powerful rule languages over large amounts of data. The newly formed RDF Shapes working group is expected to do a lot of work that has a close relation to rules

- W3C has standardized RIF
- Another viable alternative is SWRL
- Jena Rules offer several interesting options, including combining backward and forward reasoning
- SPIN provides a way to implement rules and constraints using the full expressive power of SPARQL and includes an extensive library of functions
- Ontotext GraphDB rules have simple features, but their effect can be "reversed" to implement incremental retract (triple deletion)

4 Graph Query Languages

The Semantic Network Benchmark is defined in an implementation-independent way, i.e. an abstract data model and queries defined in English. The current SNB implementations include:

- Virtuoso SPARQL 1.1
- Virtuoso SQL
- Neo4J API
- Neo4J CYPHER - a declarative query language for describing patterns in graphs
- Sparksee API

In addition, two other popular ways for accessing graph data are:

- Blueprints API - TinkerPop's property graph model interface
- Gremlin - TinkerPop's domain specific language for traversing property graphs

As you can see, there are many different ways for accessing graph data, and no clear candidate for standardization. There are many emerging graph database technologies, and SNB is helping to shape the research agenda by pushing the envelope on important graph database choke points. Thus the LDBC workloads are helping to clarify the required graph engine features, and provide valuable experience for a future standardization process

LDBC may play a future role as host to a graph query language standardization body. Such an activity has also been contemplated under the auspices of the W3C but since this is not a web-specific activity and does not necessarily involve URIs, the W3C is not necessarily interested. Hence it has been proposed (notably by LDB Council member Oracle) that the LDB Council should host an interest group around this. Even though this is not a benchmarking activity, most of the stakeholders are already involved with LDBC, hence the query language activity would be a natural avenue of expansion in the scope of the LDB Council. At present, Peter Boncz is having discussions with Claudio Gutieras, who among other things has defined the formal semantics of SPARQL, as the head of an eventual graph query language task force in the LDB Council.

Partly due to this, the LDB Council board has discussed the possibility of renaming the organization to reflect a broader charter, for example Linked Data Technologies Council instead of Linked Data Benchmarking Council. The advent of a task force for graph query languages would go towards establishing the LDB Council as a household name on the frontiers of database technology.

To appreciate the scope of work for such standardization, we can contemplate the time it took for RDF standards to reach maturity:

- 10 years from RDF 1.0 (2004) to RDF 1.1 (2014)
- 8 years from OWL1 (2004) to OWL2 (2012)
- 5 years from SPARQL 1.0 (2008) to SPARQL 1.1 (2013)

While RDF 1.1 includes only incremental updates over RDF 1.0 (with the exception of JSON-LD and RDFa that are brand new concrete syntaxes), OWL and SPARQL have changed dramatically from their initial versions. In particular, SPARQL 1.0 was barely usable because of important missing features.

5 Benchmarking Ontology

In addition to the benchmarks developed by LDBC, there are a number of other benchmarks that are popular in the industry. As an example, we can mention:

- Lehigh University Benchmark (LUBM) [6]
- Berlin SPARQL Benchmark (BSBM) [7]
- SPARQL Performance Benchmark (SP2Bench) [8]
- DBpedia SPARQL Benchmark [9]

Some of them have been justly criticized on various grounds, such as:

- Test data that does not reflect well enough typical graph structures
- Unbalanced query set that allows some queries to dominate the results
- Not addressing important performance mechanisms and choke points

Nevertheless, many of these benchmarks are exercised by various semantic technology users. Each of the benchmarks has its own dataset parameters, scale factors, running configurations, query definitions and mixes, benchmark drivers, output data (e.g. the sort of statistical information being collected), output formats, etc. This makes it hard for vendors or clients to execute the benchmarks, collect and publish results, compare and collate the results of several vendors.

Therefore it is beneficial to standardize the representation of benchmark inputs, outputs and runs. The Benchmarking Ontology whose development has started during LDBC and will continue has such a mission.

- We have completed the description of output results (using the W3C CUBE ontology, and SDMX and custom vocabularies for measures and dimensions)
- We still need to describe the execution environment (software and hardware), dataset scale factors, execution parameters, query definitions, benchmark and driver versions

We obtained a lot of the inspiration for this work from W3C conformance test suites and results expressed in RDF (e.g. the Evaluation and Report Language, EARL). W3C working groups routinely use RDF result capturing to produce their Implementation Reports that communicate conformance testing results.

After the Benchmarking Ontology is completed, it can be adopted by the LDB Council to produce parts of the Full Disclosure Reports, and then submitted to W3C for standardization.

References

[1] : RDF: Resource Description Framework: http://www.w3.org/RDF/

[2] : RDF Schema: http://www.w3.org/TR/rdf-schema/

[3] : OWL: Web Ontology Language: http://www.w3.org/2001/sw/wiki/OWL

[4] : SPARQL Protocol and RDF Query Language: http://www.w3.org/TR/rdf-sparql-protocol/

[5] : GeoSPARQL : http://www.opengeospatial.org/standards/geosparql

[6] : Leigh University Benchmark : http://swat.cse.lehigh.edu/projects/lubm/

[7] : Berlin SPARQL Benchmark :
http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/

[8] : SPARQL Performance Benchmark :
http://dbis.informatik.uni-freiburg.de/forschung/projekte/SP2B/

[9] : DBpedia SPARQL Benchmark : http://aksw.org/Projects/DBPSB.html