HOBBIT

Collaborative Project

Holistic Benchmarking of Big Linked Data

Project Number: 688227 **Start Date of Project:** 2015/12/01 **Duration:** 36 months

Deliverable 9.2.1
Annual Public Report of the First Year

Dissemination Level	Public
Due Date of Deliverable	Month 12, 30/11/2016
Actual Submission Date	Month 12, 30/11/2016
Work Package	WP9 - Project Management
Task	T9.2
Type	Report
Approval Status	Final
Version	1.0
Number of Pages	30
Filename	D9.2.1_Annual_Public_Report_I.pdf

Abstract: This deliverable reports on the progress of the HOBBIT project accomplished during its first year. The deliverable targets a general audience. In particular, it focuses on the achievements pertaining to the development of the platform, the benchmarks, the challenges and the HOBBIT community. We paid attention to presenting these achievements in brief and succinct terms as well as using easy-to-understand language. Technical details to the results presented herein can be found in the deliverables at the project website http://project-hobbit.eu.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688227.

History

Version	Date	Reason	Revised by
0.0	04/11/2016	First draft created	Nadine Jochimsen (InfAI)
0.1	15/11/2016	Additions by project partners	All
0.2	16/11/2016	Draft revised	Axel-Cyrille Ngonga Ngomo (InfAI)
0.3	23/11/2016	Peer reviewed	Michael Röder and Axel-Cyrille Ngonga Ngomo (InfAI)
1.0	30/11/2016	Review and Feedback added	Axel-Cyrille Ngonga Ngomo (InfAI)
1.0	30/11/2016	Review and Feedback added	Nadine Jochimsen (InfAI)
1.0	01/12/2016	Final version submitted	Nadine Jochimsen (InfAI)

Author List

Organization	Name	Contact Information
InfAI	Nadine Jochimsen	jochimsen@infai.org
InfAI	Axel-Cyrille Ngonga Ngomo	ngonga@infai.org
InfAI	Michael Röder	roeder@informatik.uni-leipzig.de
InfAI	Kleanthi Georgala	georgala@informatik.uni-leipzig.de
InfAI	René Speck	speck@informatik.uni-leipzig.de
iMEC	Frank Saillau	Frank.Salliau@ugent.be
iMEC	Ruben Taelman	ruben.taelman@ugent.be
AGT	Martin Strohbach	MStrohbach@agtinternational.com
AGT	Roman Katerinenko	RKaterinenko@agtinternational.com
IAIS	Jens Lehmann	Jens.Lehmann@iais.fraunhofer.de
IAIS	Henning Peztka	Henning.Petzka@iais.fraunhofer.de
IAIS	Bastian Haarmann	Bastian.Haarmann@iais.fraunhofer.de
USU	Roman Korf	r.korf@usu-software.de
USU	Alexa Schumacher	a.schumacher@usu-software.de
FORTH	Irini Fundulaki	fundul@ics.forth.gr
FORTH	Tzanina Saveta	jsaveta@ics.forth.gr
NCSR	Anastasia Krithara	akrithara@iit.demokritos.gr
NCSR	Vassiliki Rentoumi	vrentoumi@iit.demokritos.gr
OpenLink	Mirko Spasic	mspasic@openlinksw.com
OpenLink	Milos Jovanovik	mjovanovik@openlinksw.com
TomTom	Oliver Kannenberg	Oliver.Kannenberg@tomtom.com

Executive Summary

This deliverable gives an overview of the progress of the HOBBIT project accomplished during its first year and targets a general audience. The focus during the first year was on (1) creating the prerequisites necessary to run the project (website, dissemination channels, etc.), (2) gathering feedback from the benchmarking community and potential users, (3) specifying the technical outcomes of the project as well as (4) beginning with the implementation of these outcomes.

In the introduction (Section 1), we give an overview of the project and outline the rationale behind the project. An overview of the components that play a role in the project is given and the reader is led to an overall understanding of what HOBBIT is about. The subsequent section, Section 2, is slightly more involved. The goal of the section is to present the benchmarks that HOBBIT targets to provide in more detail. Special attention is given to providing enough details for the core ideas behind the benchmarks to be understood while not providing too many details so as to ensure that even technically less versed readers can get an idea of what the benchmarks are about. The HOBBIT benchmarking platform, one of the core results of the project, is at the center of Section 3. Here, we give an overview of the technical architecture of the platform. We also present its components, aiming at elucidating how the results generated by the platform are created. In addition, the mimicking algorithms that will underlie the platform are briefly presented.

The HOBBIT platform is the core technical asset used in the HOBBIT challenges. These challenges aim to gather comparable results on the performance of relevant state-of-the-art systems. We are currently planning to organize 5 challenges, which are described in Section 4. There, we point to the benchmarks that are to be used in the challenges and the expected target groups and events for the challenges. A brief summary of our dissemination and outreach activities is given in Section 5 and 6. In essence, these two sections discuss how HOBBIT has been received so far by the community in terms of engagement as well as the efforts carried out by the HOBBIT consortium to get the world interested in the project.

The document is concluded by a summary, which condenses some of the key achievements so far and points to future works. Throughout the document, we paid attention to presenting our current achievements in brief and succinct terms as well as using easy-to-understand language. Technical details can be found in the corresponding deliverables at the project website http://project-hobbit.eu.

Abbreviations and Acronyms

BLD Big Linked Data

KPIs Key Performance Indicators

IM Instance Matching

SD Source Dataset

TD Target Dataset

OKE Open Knowledge Extraction

Contents

List of Tables

List of Figures

1 Introduction

1.1 Rationale

> Big Data is one of the key assets of the future. However, the cost and effort required for introducing Big Data technology in a value chain is significant. Mastering the creation of value from Big Data will enhance European competitiveness, will result in economic growth and jobs, and will deliver societal benefit.[1]

It is thus of utmost importance to reduce the costs and hurdles required to introduce Big Data processing into the European industry. A key step towards abolishing the barriers to the adoption and deployment of Big Data is to provide European companies with open benchmarking reports that allow them to assess the fitness of existing solutions for their purposes as well as to evaluate their own progress w.r.t. processing Big Data. However, achieving this goal demands:

1. The **deployment of benchmarks** on data that reflects reality within realistic settings.
2. The provision of corresponding **industry-relevant Key Performance Indicators** (**KPIs**).
3. The **computation of comparable results** on standardized hardware.
4. The institution of an **independent and thus bias-free organization** to conduct regular benchmarks and provide the European industry with up-to-date performance results.

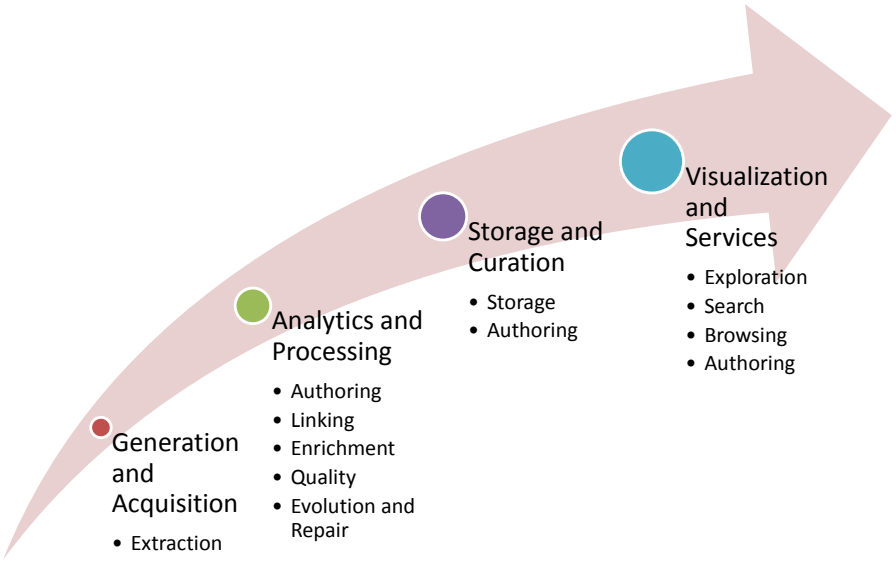

Figure 1: Mapping of Linked Data Lifecycle steps (bullet points) to the four steps of the Big Data value chain

Addressing these tasks for all possible instantiations of the Big Data Value Chain would be a Herculean task not realizable within a span of three years. In HOBBIT, we thus address all of the points above while focusing our efforts on a particular aspect of Big Data processing: Linked Data

[1] See http://bigdatavalue.eu and http://bigdataeurope.eu.

extracted, processed and generated along the Big Data Value Chain (in the following Big Linked Data (BLD), see Figure 1).

Over the last years, Linked Data has migrated from an academic concept to an industry-ready approach for data management. Content providers (e.g., Yahoo!, Wolters-Kluwers), car manufacturers (e.g., Mercedes, BMW, Renault), transport authorities (e.g., the UK, Gent and Madrid Transport Authorities), city authorities (e.g., Dublin, Madrid) and many more now use Linked Data in their everyday business. Moreover, a large number of endeavors are developing means to make use of the advantages of Linked Data in domains with Big Data requirements as diverse as the manufacturing industry (e.g., Heidelberg Druckmaschinen), media (BBC, Press Association, Google, Yahoo!), the health sector (MayoClinic, Siemens), chemical plants (Bayer) and Smart Homes (AGT).

1.2 Goal

In HOBBIT, we address the issues mentioned above by the following means:

1. **Define benchmarks** for domains of industrial relevance in Europe that make use of BLD. In the process, we will circumvent the hurdle of making real industrial data public by deploying mimicking algorithms. These will allow configuring synthetic data generators so as to compute data streams that display the same variables as industry data while being fully open and available for evaluation without restrictions.
2. **Determine the KPIs** for processing BLD by collaborating with stakeholders from relevant industry sectors as well as H2020 and FP7 Big Data projects, such as BigDataEurope, GeoKnow, LDBC, DIACHRON, GrowSmarter and Peer Energy Cloud.
3. **Create an open task-driven benchmarking platform** to evaluate the performance of BLD processing systems on standardized hardware and provide yearly evaluation campaigns. The platform will support the generation of open, human- and machine-readable reports on the evaluation campaign results. The published data will include all experiment metadata, as well as fine-grained results for the different KPIs, overall results for tools and the correlation between the tool results and the KPIs for diagnostics.
4. **Organize yearly evaluation campaigns**, using the platform and the industry-defined KPIs. We will target industry-led events for the acquisition of more datasets as well as the extension of industrially relevant KPIs.
5. **Create a self-sustainable HOBBIT association** for the continuation of the project activities even after the end of the project.

1.3 Overview and Components

The idea behind HOBBIT is summarized in Figure 2. The project began by gathering requirements, benchmark ideas and key performance indicators from the community. Overall, the results suggest that while storage benchmarks are the most important, all benchmarks foreseen in HOBBIT (and many more) are of importance to the community.[2] In addition to benchmarks and performance indicators, we also aimed to collect supplementary datasets from the community. This ongoing effort is documented in the project CKAN,[3] in which the project has already been able to amass 19 datasets of relevance with

[2] For more details on the results of the surveys and interviews with community members, please see the HOBBIT Deliverable 1.2.1 at https://project-hobbit.eu/about/deliverables/

[3] https://hobbit.iminds.be/dataset

more datasets coming up in Year 2.

The datasets collected, benchmarks and performance indicators were combined with the project-internal expertise to devise a family of benchmarks and key performance indicators, which is presented in Section 2. Moreover, they served as the foundation for the specification of the architecture of the HOBBIT benchmarking platform, which is described in Section 3 of this document. The first version of the platform is foreseen to be released in 3 months and will be used to run the first HOBBIT challenges, of which an overview is given in Section 4.

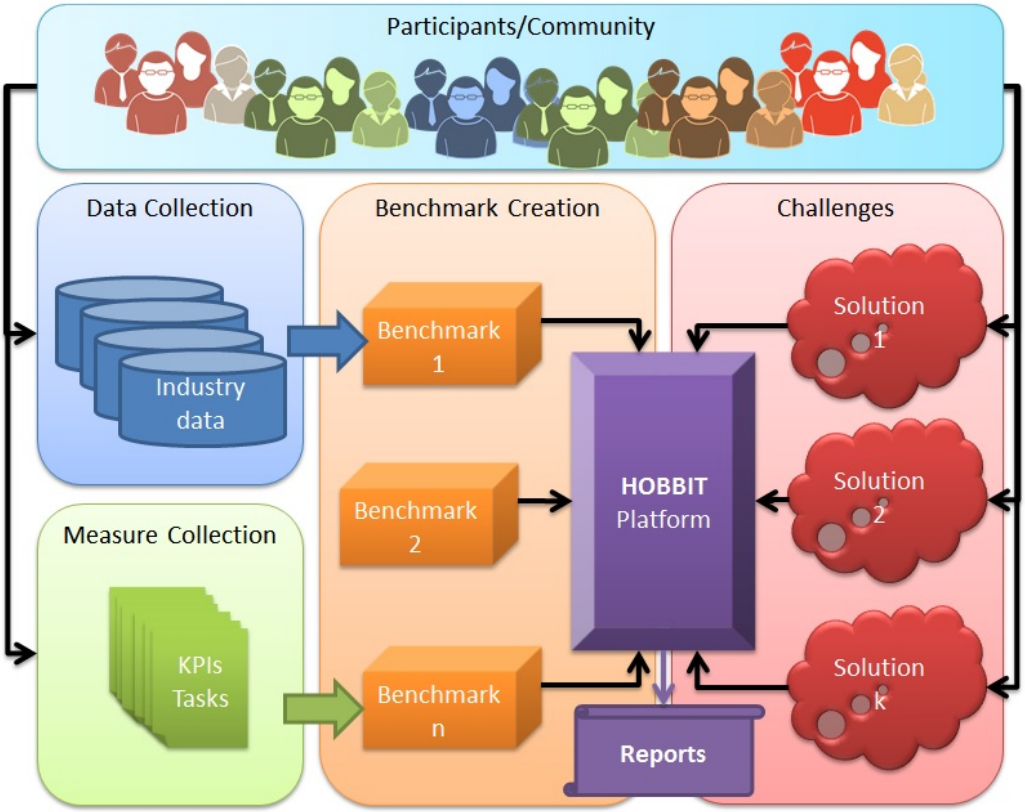

Figure 2: Overview of HOBBIT

2 Benchmarks

Eight benchmarks are currently foreseen within the HOBBIT project, of which a summary is given in Table 1. The following subsections aim to give an understandable overview of these benchmarks with only the necessary technical details. Note that we wrote the descriptions such as to ensure that the readers do not need to read through every benchmark description but can simply choose to select the description they are interested in.

2.1 Data Acquisition

The constant growth of data in velocity and volume has increased the need to integrate and process data using efficient and scalable storage approaches. The task of a storage system is two-fold: (1) retrieve and store the data and (2) process multiple users questions (queries) in parallel.

Benchmark	Short Description
Data Acquisition	Evaluate storage solutions that deal with the ingestion of streams of RDF data
Knowledge Extraction	Test the performance (runtime and accuracy) of entity recognition and linking frameworks over streams of unstructured data (text)
Link Discovery	Go beyond mere instance matching and check how well tools performs on other types of links (e.g., geospatial links) when faced with large amounts of data
Structured Machine Learning	Study the performance of machine Learning techniques (i.e., performance and runtime) on streams of structured data (e.g., RDF)
Data Storage	Stress test storage solutions for RDF when faces with realistic scenarios such as being the backend of a social network
Versioning	Check how well storage solutions deal with storing evolving data available in several versions and performing queries on and across these different versions
Question Answering	Evaluate the performance of data access solutions that can answer questions in natural language as well as keyword queries on large amounts of data
Faceted Browsing	Test storage solutions w.r.t. their performance as backends of data browsers

Table 1: Overview of HOBBIT benchmarks

In most real-time applications, such as financial transactions or predictive maintenance, both tasks must be completed in parallel with a minuscule latency. The aim of this benchmark is to measure the performance of such systems in terms of efficiency and completeness when faced with streams of input data. To achieve this goal, we study the behaviour of existing systems when faced with data of increasing volume and velocity. In order to emulate a realistic scenario, the data is generated from one or multiple resources and is inserted into a storage system simultaneously. Then, queries are used to test the system's ingestion performance and storage abilities.

The current status of the benchmark system for data ingestion is as follows:

1. Data is obtained from one resource in the form of statements and is ordered and stored in files using a time stamp. The time stamp indicates the point time that a statement was generated.
2. Statements with the same time stamp are inserted into a storage system simultaneously. The data ingestion between statements of different time stamps is delayed by a dilatation factor. The dilatation factor decreases as more sets of statements are inserted into the system. The final set of statements with the same time stamp has a dilatation value of 0.
3. After a set of statements of different time stamps are inserted into the storage system, a query is performed against the system. The goal of this query is to check if the last statement was successfully inserted.

4. The emulation stops when there are no more statements to be inserted.

Our future goals consists of the following steps:

1. We will test the ingestion performance of a storage system by deploying datasets that vary in volume (size of statements and timestamps).
2. We aim to use dilatation factors based on the real time differences between the various statement, in order to benchmark the system within a particular time interval.
3. We shall use streaming data from multiple resources.
4. We will vary the queries to cover different proportions of inserted statements.

2.2 Knowledge Extraction

A considerable portion of the information in data on the Web is still only available in unstructured form, i.e., without predefined formal structure. The goal of this benchmark is to evaluate how well knowledge acquisition frameworks for unstructured data perform. In particular, the benchmark will test the performance of systems that implement approaches for analyzing unstructured data streams (Twitter, RSS feeds, etc.).

A generic data generator that produces unstructured natural-language data streams was needed. Therefore, the work on this benchmark started with the analysis of natural language data streams (e.g., messages from social networks, content of crawled web pages). The analysis was based on the syntactical and semantical characteristics of this data. The different characteristics of the data streams were stored and used as input for the generic data generator. The generator produces natural language data streams that have a structure similar to that of the data on which it was trained, e.g, tweets.

The task generator for unstructured streams will generate tasks aiming at the extraction of structured data from the given stream of unstructured data. These tasks can range from the recognition of known entities inside the text to the extraction of new, unknown entities and their properties. For this task, we aim to reuse parts of the GERBIL project. For every type of task generated by the task generator for unstructured data, there will be at least one key performance indicator (e.g. recall, precision, f-measure, throughput) that has to be computed to evaluate the generated results. Thus, every task generator needs an evaluation module that can calculate this key performance indicator based on the generated result and a given gold standard.

The created benchmark has to be integrated into the HOBBIT platform.

We performed the following in the first year:

- GERBIL reuse for this task.
- Mimicking Twitter streams.
- Verbalization of knowledge base facts.
- Prototype integration of the benchmark into the HOBBIT platform.

We aim for the following goals in the second year:

- Integration of extraction benchmarks.

  - Recognition of entities.
  - Extraction of properties between entities.

- Expand the prototype integration of the benchmark into the HOBBIT platform.
- Dockerization of the benchmarks.
- Run the benchmarks.

2.3 Link Discovery

The number of datasets published in the Web of Data as part of the Linked Data Cloud is constantly increasing. The Linked Data paradigm is based on the unconstrained publication of information by different publishers, and the interlinking of Web resources across knowledge bases. In most cases, the cross-dataset links are not explicit in the dataset and must be automatically determined by using Instance Matching (IM) tools. The large variety of techniques requires their comparative evaluation to determine which one is best suited for a given context. Performing such an assessment generally requires well-defined and widely accepted benchmarks to determine the weak and strong points of the proposed techniques and/or tools.

A number of real and synthetic benchmarks that address different data linking challenges have been proposed for evaluating the performance of such systems. So far, only a limited number of link discovery benchmarks target the problem of linking geo-spatial entities such as PABench [1]. The objective of this task is to propose a synthetic and scalable benchmark that will test both the correctness of the results of instance matching systems and their performance and scalability. More specifically, our objective is to develop benchmarks for spatial and streaming data for assessing the performance in terms of precision, recall and f-measure of linking systems. The benchmark will extend LANCE [3], a domain-independent instance matching benchmark generator that supports value-, structure-based and semantic-aware test cases, the latter taking into account schema information expressed in RDFS and OWL to address spatial data.

For the development of the benchmark we are going to work with TomTom data that record car trajectories. A trajectory is a set of points (denoted by longitude, latitude), a time stamp and a speed. The benchmark will comprise of a Source Dataset (SD) and a Target Dataset (TD) that will be produced by transforming labels and trajectories from the source dataset. The current status of the benchmark is the following:

- We are introducing labels for each of the trajectory points in a TomTom dataset using GoogleMaps API.
- Given such as TomTom dataset, we apply value-based transformations to the labels of trajectory points using LANCE to obtain the target dataset. An Instance Matching tool is then called to decide whether the two trajectories are *equal*.

During the first year of the project we studied Allen's Relations[4] and DE-9IM[5] relations that are used for modeling temporal and spatial data in single and two-dimensional spaces respectively and adjust those on spatial data. Currently, we are designing test cases for our benchmark to accommodate

[4] http://cse.unl.edu/ choueiry/Documents/Allen-CACM1983.pdf
[5] https://download.tuxfamily.org/tuxgis/geodescargas/Tutorial_DE9IM_in_PostGIS.pdf

Relation	Explanation
Equals	The Geometries are topologically equal
Disjoint	The Geometries have no point in common
Intersects	The Geometries have at least one point in common (the inverse of Disjoint)
Touches	The Geometries have at least one boundary point in common, but no interior points
Crosses	The Geometries share some but not all interior points, and the dimension of the intersection is less than that of at least one of the Geometries
Overlaps	The Geometries share some but not all points in common, and the intersection has the same dimension as the Geometries themselves
Within	Geometry A lies in the interior of Geometry B
Contains	Geometry B lies in the interior of Geometry A (the inverse of Within)

Table 2: DE-9IM relations

DE-9IM relations and, more specifically, the relations shown in Table 2. We are focusing on the following scenarios, which are being designed and implemented for the first version of the linking benchmark. For each DE-9IM relation we present the test cases that we are designing and implementing.

- **Equal** (same as Allen's *Equal*): We identify here the following test cases:
  - **C1**: Keep arrival and departure points in SD , remove a subset of intermediate points
  - **C2**: Keep arrival and departure points in SD , insert additional intermediate points with error in long, lat.
- **Disjoint** Create a random target trajectory in the TD that has no common points with the source trajectory in the SD
- **Intersects/Crosses** We identify here the following test cases:
  - **C1**: Create a target trajectory that has one common point with the source trajectory
  - **C2:** Create a target trajectory that has more than one common points with the source trajectory
- **Overlaps** Create a target trajectory that shares at least one sub-path with the source trajectory.
- **Within** Create a target trajectory that is a sub-path of the source trajectory without the two trajectories being equal and without having the same departure and arrival points. (same as Allen's *During*)
- **Contains** Create a target trajectory that is a super-path of the source trajectory without the two trajectories being equal and without having the same departure and arrival points. (same as Allen's *During Inverse*)

DE-9IM relation *Touches* cannot be captured by the available TomTom datasets.

2.4 Structured Machine Learning

The value of machine learning in a modern world cannot be overstated. Many applications that we are using on a daily basis (email, web search, etc.) incorporate some machine learning components.

The ability to benchmark such components in a reliable and reproducible way will improve their quality which will not go unnoticed.

This task (Structural Machine Learning Benchmark) is aimed at developing a benchmark for a certain type of machine learning that operates on a structured data. The availability of a structured data has increased over the past years. The structure can be seen as a background knowledge — an additional input to a learner providing some insight into the data.

The last year we were focusing on the part of benchmark that addresses background knowledge modeled on using Semantic Web languages like RDF and OWL and we have done the following:

- Settled on KPIs for the first version of the benchmark (latency, throughput).
- Developed a mimicking algorithm. The algorithm is capable of learning a statistical model of real-world sensor measurements dataset and of using that model to generate simulated datasets preserving some important properties of the original dataset.
- Implemented data generator that incorporates the mimicking algorithm. The data generator is a highly configurable software and can generate a stream or files of measurements according to its configuration.
- We were working on ACM DEBS Grand Challenge organization. Our structured machine learning benchmark will be the main benchmark of the challenge and the HOBBIT platform will be the main evaluation platform of the challenge.

In the nearest future we are planning the following:

- To finish implementation of the benchmark.
- To investigate other important types of data structures (e.g., graph) and machine learning algorithms behavior on them in order to include them in the next version of benchmark.
- To continue to work on implementation of other benchmarks (e.g., sensor stream benchmark)

2.5 Data Storage

In recent years, the huge expansion of the Linked Data Web in data volume has increased the need for triple stores to process more and more data in a shorter time. The systems should be able to handle a growing amount of triples and show its potential how they can be enlarged in order to accommodate that growth. Some of the applications requests fast and reliable responses from data stores, usually in an interactive time (less than a second), regardless of the scale of dataset. The other types of the applications (e.g. Business Intelligence applications) can tolerate longer process times because of their complex logic and/or a lot of data that has to be accessed and taken into consideration. The aim of this task is to develop a benchmark that can be used to measure how well the system under test is regarding to both types of queries already mentioned (interactive and BI ones).

As a starting point for our benchmark, we used LDBC Social Network Benchmark developed in LDBC project[6]. Workloads are designed to mimic the different usage scenarios found in operating a real social network site. Each workload defines a set of queries and query mixes, designed to stress the systems under test in different choke-point areas, while being credible and realistic. In previous

[6] http://www.ldbcouncil.org/

months, we focused on the Interactive workload, which reproduces the interaction between the users of the social network by including lookups and transactions that update small portions of the data base. These queries are designed to be interactive and target systems capable of responding such queries with low latency for multiple concurrent users. Later on, the BI queries will be added.

In the first year of the project, we finished the following tasks:

- Formulation of queries in SPARQL language
- Validation of the queries
- Modification of the data generator, a software that contains the mimicking algorithm which are used to generate the data needed by the benchmark
  - Fixed and validated TTL serialization
  - Changes in the data schema and distributions, making the dataset more realistic, and similar to real-world RDF datasets

Future work includes:

- To run the full benchmark on different scale factors, and provide results of these experiments
- Mutual comparisons between different scales, and with SQL implementation of the benchmark
- Modifications of the queries, related to the new version of data generator
- Dockerization of the benchmark and porting on HOBBIT platform

2.6 Versioning

The open nature of the web implies that these changes typically happen without any warning, centralized monitoring, or reliable notification mechanism; this raises the need to keep track of the different *versions* of the datasets and introduces new challenges related to assuring the quality and traceability of Web data over time.

The objective of this task is to propose a synthetic and scalable benchmark based on LDBC's SPB 2.0 benchmark to test the ability of systems to store and query different versions of an evolving dataset. In the first year of the project we performed the following tasks:

- Studied the changes between the different versions of the broadly used datasets such as the Gene Ontology (GO), DBpedia, Experimental Factor Ontology (EFO), Ontology of Genes and Genomes (OGG), Medical Subject Headings (MSH), Foundational Model of Anatomy (FMA), Atlas RDF Ontology (ATLAS). We focused on high and low level changes both at the schema and instance level such as addition and deletion of schema classes and properties, modification of class and property hierarchies, instantiation of resources under classes and schema properties, addition/deletion of property domain and range and finally the addition and deletion of comments and labels. In our analysis of the aforementioned datasets and versions thereof we found that the majority of changes focused on modification of comments and labels of instances.

  The purpose of this study was to comprehend interesting changes in real datasets in order to address those in the versioning benchmark that we will propose in HOBBIT.

- We have re-designed SPB Data Generator in order to decouple it from any triple store where schema information was stored and used for data generation.
- We have designed and implemented the first version of the data generator for our versioning benchmark that is based on the SPB generator. We modified the data generator to produce versions of an input dataset by taking into account the number of versions requested by the user. The generation of versioned data follows the principles of SPB data generator and more precisely (a) clustering (b) correlation and (c) random tagging of entities. In this first version of the data generator we are focusing on the addition of creative works (i.e., metadata descriptions of journalistic assets) between the different versions.
- We have collected Linked Data datasets from publishing organizations such as the NYTimes to comprehend the changes between versions of datasets from the semantic publishing domain. We expect that the changes will be similar to the ones we witnessed in the datasets we discussed earlier.

Future work includes:

- Study of the collected Linked Data datasets from the Publishing Domain to understand the types of changes across the versions thereof.
- Study of DBpedia query logs to identify the cross version user queries. On the basis of those queries we are going to design the query workload that we will use in our benchmark to check the performance of versioning systems regarding query performance.

2.7 Question Answering

The past years have seen a growing amount of research on question answering over large-scale RDF data. At the same time, the growing amount of data has led to a heterogenious data landscape. The purpose of this work package is to provide up-to-date benchmarks for assessing performance and accuracy of question answering approaches that mediate between users, expressing their information need in natural language, and large-scale background knowledge data (i.e. DBpedia). The evaluation platform for this benchmarks will be built upon the open source GERBIL QA benchmarking platform[7]. Concerning question answering tasks, the main task is:

Given one or several RDF datasets as well as additional knowledge sources and natural language questions or keywords, return the correct answers or a SPARQL query that retrieves these answers.

Additional tasks are going to tackle multilingual question answering over DBpedia such that answers can be retrieved from an RDF data repository given an information need expressed in a variety of languages (including English, German, Dutch, French, Spanish, Italian, Romanian, Persian and Hindi), hybrid question answering that requires the integration both from RDF and textual data sources and large-scale question answering including a mass amount of automatically derived questions taking in to account not only a system's accuracy on answers but also the time needed to retrieve these answers.

The current status of the benchmark system for question answering is as follows:

- Both the training questions and test questions from previous question answering challenges have been extracted together with their answers and DBpedia SPARQL queries.

[7] http://gerbil-qa.aksw.org/gerbil

- Instance data from DBpedia including their class mappings have been obtained, converted and stored.
- A question generating algorithm aims to identify instance data from DBpedia in the challenge questions, annotates the text span and maps the respective instance's class to it. At the same time, the algorithm also alters the questions' SPARQL query accordingly.
- A question template is derived from the challenge questions, such that the identified instance data span is replaced with a placeholder for the instance's class.

Our future goals consists of the following steps:

- We will connect the GERBIL QA benchmarking platform to HOBBIT and build an evaluation environment to run the question answering benchmark assessments.
- We will derive a mass amount of questions from the templates by replacing the class placeholder by all instances' text spans that are members of the respective class.
- Inspect the results and investigate possible disambiguation steps to reach a higher quality level where instances have been ambiguous. Test if the answers are feasible in the underlying dataset.
- Once a level of maturity has been reached, both the generated mass questions and their queries could possibly be fed to a machine learning algorithm. We will investigate this possibility.

2.8 Faceted browsing

Faceted browsing stands for a session-based and state-dependent interactive method for query formulation over a multi-dimensional information space. It provides a user with an effective way for exploration through a search space. After having defined the initial search space, i.e., the set of resources of interest to the user, a browsing scenario consists of applying (or removing) filter restrictions of object-valued properties or of changing the range of a number-valued property.

As a well-established example for an implementation of faceted browsing consider an online shopping portal where the search space could be a certain type of clothes and, amongst others, the facets could consist of size, color and price. Using the mentioned filtering operations aimed to select items with desired properties, the user browses from state to state, where a state consists of the currently chosen facets, their corresponding facet values and the current set of instances satisfying all chosen constraints.

The goal of the task on faceted browsing is to check existing solutions for their capabilities of enabling faceted browsing through large-scale structured datasets, that is, it analyses their efficiency in navigating through large datasets, where the navigation is driven by intelligent iterative restrictions. We develop browsing scenarios through a dataset, which reflect an authentic use-case and challenge participating systems on different points of difficulty. To distinguish several solutions, we measure the performance relative to dataset characteristics, such as overall size and graph characteristics.

The following is the current state of this project task.

(i) We collected the choke point of faceted browsing, that is, the difficulties that arise for a system in enabling efficient browsing through structured datasets.

(ii) We explored our possibilities to develop benchmarking scenarios which resemble realistic browsing scenarios.

(iii) We investigated the characteristics of several datasets for their suitability to benchmark systems according to the choke points collected in (i).

The next steps are as follows:

- We enrich the transport dataset of linked connections to enable the examination of participating system on the choke points that we have collected.
- We write a series of SPARQL queries which simulate the anticipated browsing scenarios.
- We integrate the benchmark to the HOBBIT platform.

3 Platform and Results

The HOBBIT evaluation platform builds upon the open-source platform GERBIL benchmarking platform,[8] which resulted from collaboration between 7 institutions across Europe. It is created so as to ensure that:

1. The benchmarks are easy to use.
2. New benchmarks can be easily created and added to the platform by third parties.
3. The evaluation can be scaled out to large datasets and on distributed architectures.
4. The publishing and analysis of the results of different systems can be carried out in a uniform manner across the different benchmarks.

All benchmarks will run directly in the platform, which will provide standard interface for the integration of novel benchmarks as well as for evaluating external tools. Consequently, the platform will build the basis for HOBBIT's challenges, during which developers will evaluate system prototypes as well as complete systems. In general, the challenges will consist of two phases: In the first phase of the challenges (training phase), the developers will train their systems based on training settings of the data generators (the training settings will lead to the automatic generation of small datasets). After a successful completion of the training phase, an exact description of the deployment conditions for the full-scale evaluation will be provided to the tool developers. With this predefined hardware configuration for all tools within a benchmark, we will ensure a consistent full-scale deployment of participating frameworks across the challenges. The second phase of each challenge (test phase) will then consist of an evaluation of the tools in conditions representing real usage at industrial scale with respect to volume, velocity and variability. During the challenges, we will gather further requirements, datasets, KPIs and relevant tasks from the participants and use these to improve the benchmarks periodically. The HOBBIT platform will be able to publish regular reports containing the results of different systems for our benchmark and an analysis of these results.

During the first year, we developed the architecture of the platform and published it as Deliverable 2.1. The architecture of the platform is shown in Figure 3. It can be seen that the platform can be divided into three parts.

[8] http://gerbil.aksw.org

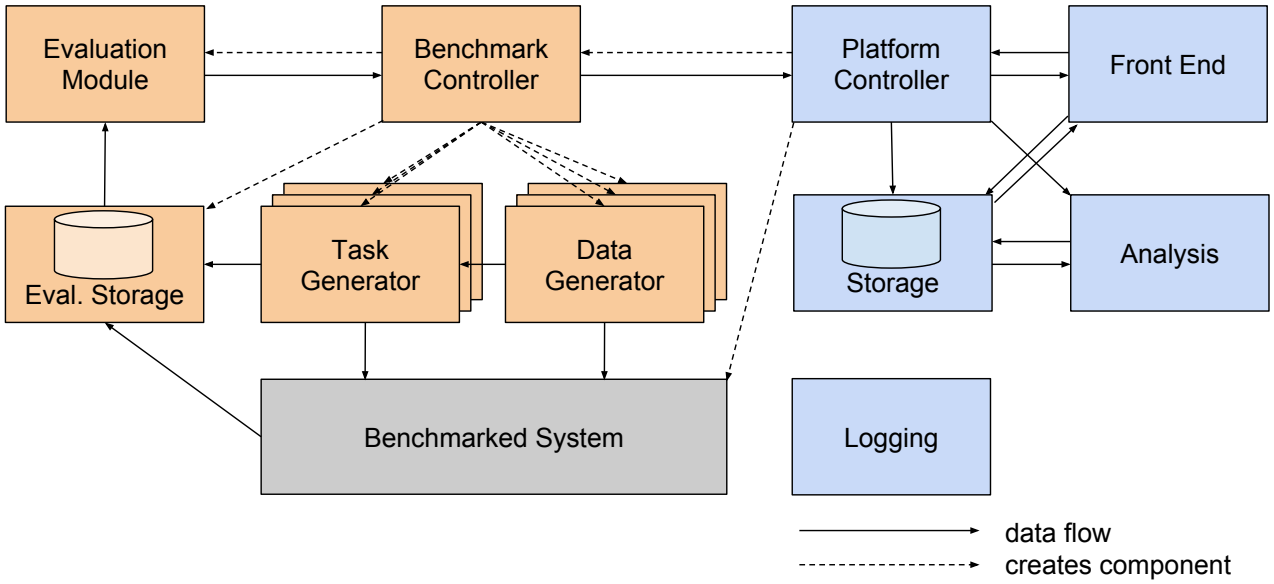

Figure 3: HOBBIT Platform architecture.

The first part (blue) comprises the platform components. It can be seen that the platform offers a front end for interacting with a user. The storage component contains the results of the experiments and challenges while the analysis component uses the data to find relations between the features of the benchmark or the benchmarked system and the evaluation results. The platform controller that executes the benchmarking of systems while the logging system enables the monitoring of the single components.

The second part (orange) comprises the components of a benchmark. The data generator contains the mimicking algorithm which are used to generate the data needed by the benchmark. The task generator create the tasks that have to be fulfilled by the benchmarked system and store the expected responses in the evaluation storage. Additionally to the expected responses, the evaluation storage receives the responses of the system. The evaluation module loads the responses from the evaluation storage and compares the expected responses with those from the benchmarked system. The benchmark controller orchestrates the single components and forwards the results of the benchmark experiment to the platform controller.

The third part (grey) comprises the benchmarked system. All components are deployed as Docker containers and communicate via RabbitMQ. This ensures that the components can interact with each other even if a benchmarked system has been written in a different programming language or needs a different runtime environment than the benchmark or the evaluation platform.

In the following, the implementation of the single mimicking algorithms is described.

3.1 Print Machine Data

The printing machine, in particular an offset printing machine, is a specific machine type in the domain of production industry. It usually consists of different parts like a feeder, different printing units, optional coating units and a delivery system. The machine operation is divided in several printing jobs, which represent orders or parts of an order of customers. The data currently used for mimicking are the event data generated during machine operation. A printing job usually starts with a start-job

event and ends with the finish-job event. In between these events several other events occur. Most of them are standard events within the operation, others indicate issues.

In order to mimic these data USU data scientists analyzed the original machine data. Based on the time lag between different events, the probability distribution function has been determined. From this the cumulative probability distribution function was calculated. These functions are used within the mimicking algorithm to generate the mimicking data.

Additionally to the mimicking of the data, USU developed a schema for an event ontology representing the machine events and their meta-data. This ontology is used to generate a semantic representation of the mimicked data. The implementation was done using the programming language Python. For interoperability reason within HOBBIT we implemented a RESTful web-service. As parameter it accepts the start date and time of the machine data mimicking, the number of printing jobs as well as a seed to produce pseudo random numbers. The implementation allows several formats for representing the semantic data including the common formats RDF/XML, Turtle, N-Tripple and JSON-LD. Finally we created deployment scripts in order to deploy the mimicking service as Docker image.

The algorithm for generating mimicked data is generic. It can be applied to other machine types as well, where event data are generated. However, the main effort still is the identification of the distribution functions.

3.2 IT Data

For the IT Data Use Case we concentrate on log data from an Apache Cassandra cluster, run by USU. The cluster currently consists of 7 nodes with a total of 42 CPUs, 140GB RAM and 5.2TB of storage. The goal of the benchmark will be to predict bottleneck or even breakdowns under certain load scenarios.

The data provided and to be mimicked consists of the key areas where we will want to capture and analyze:

1. Throughput, especially read and write requests;
2. Latency, especially read and write latency;
3. Disk usage, especially disk space on each node;
4. Garbage collection frequency and duration and
5. Errors and overruns, especially unavailable exceptions which indicate failed requests due to unavailability of nodes in the cluster.

Additionally, as the Cassandra Cluster is run under a Apache Spark computing cluster, we will mimic number and frequency of Spark jobs as well, as this data might lead to prediction of the upcoming workload for the Cassandra cluster.

Like in the Use Case of Printing Machine Data, the mimic algorithm feature a REST-API and support different formats, like DF/XML, Turtle, N-Tripple and JSON-LD. The algorithm for generating mimicked data is generic. It can be applied to other machine types as well, where event data are generated. However, the main effort still is the identification of the distribution functions.

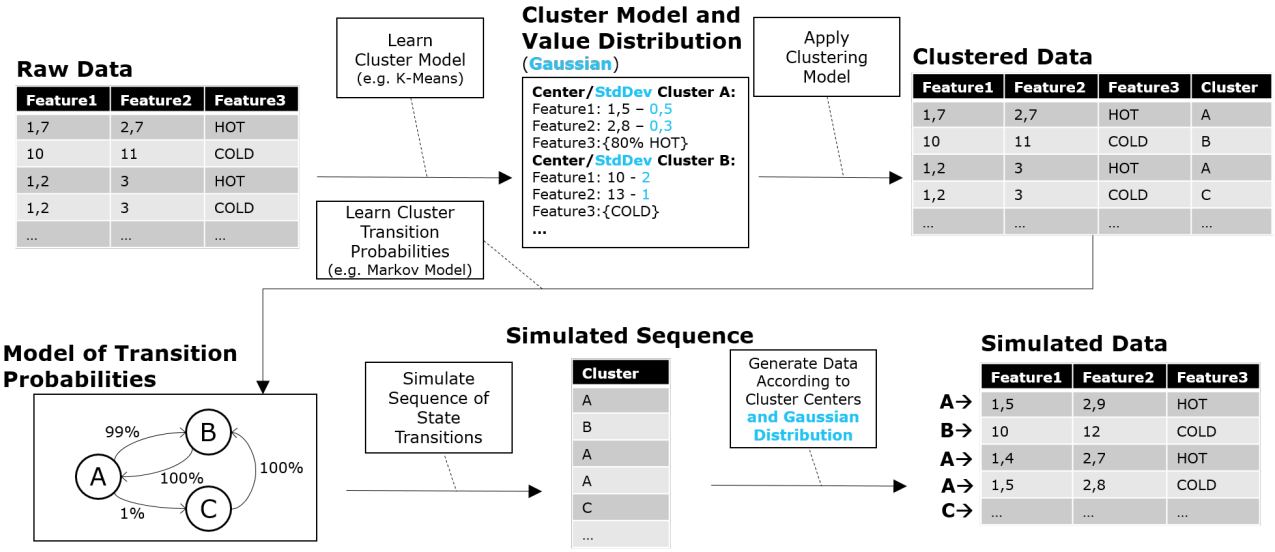

Figure 4: Steps to mimik stateful dimensions in the Weidmüller dataset.

3.3 Weidmüller

The Weidmüller dataset consists of readings taken from sensors deployed on a plastic injection molding machine. The sensors can measure various parameters of the production process: distance, pressure, time, frequency, volume, temperature, time, speed, force, etc. Each measurement is a timestamped, 120 dimensional vector consisting of values of different types, like fractional, decimal, but mostly fractional values

The first step of our mimicking approach is to automatically classify all dimensions in three groups: constant, trending phases and stateful. A constant dimension has only one value for all data instances. A trending phase is the dimension that exhibits ascending or descending growth. All the other dimensions are considered stateful.

On the second step we take each individual dimension and apply mimicking techniques based on how the dimensions has been classified. For a constant dimension we take a random constant which is not far away from the original dimension's value. For a trending phase we take the first value, an increment value and produce every next value as the previous value + increment. At random time moments we subtract some other value. For a stateful dimension we follow a more sophisticated scheme that allows us to mimic states and state transitions found in the original data (see Figure 4). This scheme includes the following steps:

1. Clustering of the dimension with the k-means algorithm using an automatically computed k for this dimension. This allows us to assign a cluster to each data instance in the dimension.
2. For each cluster we compute the mean value and the standard deviation.
3. Iterating throughout the dimension we compute one step cluster transition probabilities (Markov model).

On the third step we use this model to generate simulated data. In order to so, we randomly walk through the Markov model. When we enter a new vertex in the Markov model we generate a data instance using the cluster parameters (mean value and standard deviation) associated with the visited vertex.

3.4 Twitter Dataset

The Twitter dataset[9] is derived from 1 million real tweets that were generated in June 2009. To ensure that we do not divulge any personal information, we used (1) a Markov model to generate text that resembles tweets and abide by the density distribution of words in tweets and (2) a tweet time distribution model that allows scaling up the number of agents generating tweets as well as the distribution of time for tweets. Therewith, we can ensure that the behavior of systems that ingest our tweets is similar to that of systems which ingest real tweets generated by the same number of users over the same period of time. A simple ontology which describes tweets by the user who generated them, the time at which they were generated and their content is used in the dataset.

3.5 Transport Data

A significant portion of people use public transport for their travels. The countless public transport services worldwide, combined with their usage, lead to an enormous source of information. Many public transport companies worldwide provide this data using the GTFS standard[10], which can be converted to Linked Data using the Linked Connections framework [2]. Such data is an ideal source for the benchmarking of systems because of its time and space dimensions. These datasets contain geospatial information about stops, temporal information about transit schedules and the interlinking between both.

In many cases, benchmarking requires the ability to create synthetic datasets with specific properties of any given size. This is why we provide a public transport dataset generator that is able to create realistic public transport areas, networks and schedules. The generator can be configured to produce countless of synthetic datasets using a wide range of parameters.

4 Challenge Preparations

During the first year of the project, the preparations for the challenges that will run during 2017 have taken place. A thorough investigation of the existing challenges and workshops have been done, in order to identify the most appropriate events, where HOBBIT challenges could take place. We ended up aiming to organize five challenges in three different events. In particular, three challenges will be organized in ESWC conference[11], one at the ISWC conference[12] and one challenge will be organized in collaboration with ACM DEBS Grand Challenge. Below, a short description of the challenges is given.

4.1 The Mighty Storage Challenge

The aim of the mighty storage challenge is to test the performance of solutions for SPARQL processing in aspects that are relevant for modern applications. Hence, we aim to benchmark systems that deal with the benchmarks presented in Sections 2.1, 2.5, 2.6 and 2.8. These include ingesting data, answering queries on large datasets and serving as backend for applications driven by Linked Data. The proposed challenge will test the systems against data derived from real applications and with

[9] https://github.com/renespeck/TWIG
[10] https://developers.google.com/transit/gtfs/
[11] http://2017.eswc-conferences.org/
[12] http://iswc2017.semanticweb.org/

realistic loads. An emphasis will be put on dealing with changing data in form of streams or updates. This version of the challenge (which we aim to run periodically for at least 3 years) will comprise the following tasks: (1) RDF data ingestion, (2) data storage, (3) versioning and (4) browsing. In essence,

- Task 1 will measure how well systems can ingest streams of RDF data.
- Task 2 will measure how data stores perform with different types of queries.
- Task 3 will measure how well versioning and archiving systems for Linked Data perform when they store multiple versions of large datasets.
- Task 4 will check existing solutions for how well they support applications that need browsing through large datasets.

A corresponding proposal is currently submitted to ESWC 2017 and is being evaluated.

4.2 QALD Challenge

The Question Answering over Linked Data (QALD) challenge targets the works on question answering (see Section 2.7) aims at providing an up-to-date benchmark for assessing and comparing systems that mediate between a user, expressing his or her information need in natural language, and RDF data. It thus targets all researchers and practitioners working on querying Linked Data, natural language processing for question answering, multilingual information retrieval and related topics.

The main goal is to gain insights into the strengths and shortcomings of different approaches and into possible solutions for coping with the heterogeneous and distributed nature of Semantic Web data.

The challenge will focus on the following three tasks:

- Task 1: Multilingual question answering over DBpedia
- Task 2: Hybrid question answering
- Task 3: Large-Scale Question answering over RDF

A corresponding proposal is currently submitted to ESWC 2017 and is being evaluated.

4.3 OKE Challenge

The OKE challenge has the ambition to provide a reference framework for research on Knowledge Extraction from text for the Semantic Web by re-defining a number of tasks (typically from information and knowledge extraction), taking into account specific SW requirements. The OKE challenge defines three tasks, each one having a separate dataset:

- Entity Recognition, Linking and Typing for Knowledge Base population
- Class Induction and entity typing for Vocabulary and Knowledge Base enrichment
- Web-scale Knowledge Extraction by Exploiting Structured Annotation.

Task 1 consists of identifying Entities in a sentence and create an OWL individual representing it, link to a reference KB (DBpedia) when possible and assigning a type to such individual. Task 2 consists in producing rdf:type statements, given definition texts. The participants will be given a dataset of sentences, each defining an entity (known a priori). Task 3 will be based on one of the largest, publicly available collections of triples extracted from HTML pages (provided by the Web Data Commons project). A corresponding proposal is currently submitted to ESWC 2017 and is being evaluated.

4.4 Ontology Alignment and Evaluation Initiative

The tasks proposed target the benchmark described in Section 2.3 and will focus on the different types of spatial object representations and will be provided with different severity levels for the applied transformations. In these transformations, objects may keep their representation, they may change their geometry, type or attributes, merge with other objects, or can completely disappear. This is a scenario that stems from the heterogeneous datasets (in structure and semantics) used to describe geo-spatial entities. The produced tasks will be used by IM tools that implement string-based as well as topological approaches for identifying matching entities. The IM frameworks will be evaluated for both accuracy (precision, recall and f-measure) and scalability. Furthermore, the results will be made available in both human and machinereadable form for further processing. Since Lance is schema-agnostic, contrary to PABench, it will be used to produce benchmarks for different (source) ontologies to accommodate the different requirements that stem from a variety of applications. A corresponding proposal will be submitted to ISWC 2017.

4.5 DEBS Grand Challenge

The 2017 ACM DEBS Grand Challenge[13] is the challenge for the benchmark described in Section 2.4. This challenge is the seventh in a series of challenges which seek to provide a common ground and uniform evaluation criteria for a competition aimed at both research and industrial event-based systems. The goal of the 2017 DEBS Grand Challenge competition is to evaluate event-based systems for real-time analytics over high velocity and high volume data streams. The focus of the 2017 Grand Challenge is on the analysis of the RDF streaming data generated by digital and analogue sensors embedded within manufacturing equipment. Specifically, the scenario of this year's Grand Challenge focuses on the the detection of anomalies in the behavior of the manufacturing equipment based on the machine learning-based classification. The data set for the 2017 DEBS Grand Challenge as well as the automated evaluation platform are provided by HOBBIT.

5 Dissemination Activities

The dissemination activities of HOBBIT are in the following dimensions:

- Organization of workshops for benchmarking challenges
- Raising awareness for the project

These activities are strongly interconnected with the coordination and consolidation of the relevant communities in each phase of the Big Data value chain - the creation and establishment and

[13] Page for the 2016 challenge at http://www.ics.uci.edu/~debs2016/call-grand-challenge.html. Page of the 2017 challenge under construction at this point in time

maintenance of these communities play a crucial role for the success of the project.

Workshops for benchmarking challenges play a crucial role in promoting the benchmarks that will be developed by HOBBIT and will help out in disseminating the results of the project both in academia and industry. We decided to follow two strategies for campaigns: In fields for which established campaigns exist already we will extend those with industrial-strength benchmarks. Otherwise, we will integrate the campaigns with top-level conferences or other large events. Here, we will especially focus on presenting the results of our evaluation during industry-led events. Organisations such as Big Data vendors, as well as groups from Academia will participate in the relevant workshops in order to present (a) their systems and (b) their data as well as the results from running the HOBBIT benchmarks with their systems. Section 4 present in detail the challenges that we have organized for evangelising the HOBBIT benchmarks.

In order to raise awareness and build communities both from academia and industry we engaged in several activities such as:

- the creation of the project's website (https://project-hobbit.eu/) to advertise and promote the results of the project.
- the design of a project fact sheet that was published on the HOBBIT website from the first month of the project, flyers, broschures, banners as well as the design of 2 versions of posters that have been used in both academic events such as conferences and workshops (ESWC 2016, ISWC 2016) as well as in industry gatherings (EDF 2016, ApacheCon 2016).
- the publication of press releases in the languages at least of the consortium partners i.e, Greek, German and Dutch on the project's website in addition to newsletters to promote the results of HOBBIT to the community.
- set up of a Twitter channel to tweet the events the members of the HOBBIT consortium are participating (of academic or industrial nature) (@hobbit_project) where we have tweeted 324 posts, we have 316 followers, 40.000+ impressions, and more than 1170 profile visits.
- the set up of a Slideshare account where we publish the slides from talks and presentations from the events that members of the consortium attend (http://www.slideshare.net/hobbit_project) where we have uploaded 24 slides and we have more than 3.000 views in the last year.
- the set up a Bibsonomy account to upload the papers that we author at the HOBBIT project where we have uploaded 21 publications and tagged them with the tag @projecthobbit.

Members of the HOBBIT consortium have authored and presented fifteen papers in addition to presenting four tutorials related to benchmarking in major conferences and workshops in the field of Semantic Web and Big Data Processing such as ESWC 2016, ISWC 2016, ICSS 2016, EKAW 2016, ECAI 2016. Members of the consortium have also authored journal papers that were published in the Web Semantics: Science, Services and Agents on the World Wide Web (2016) and Semantic Web journals.

HOBBIT members also organized the 1st International Workshop on Benchmarking Linked Data (BLINK) that was held in conjunction with ISWC 2016. The workshop was successful, attracted many attendees who had interesting discussions.

6 Outreach Activities

The HOBBIT community building efforts are ongoing. As a strategy, we chose to attend and present the project at academic and industrial meetings so as to gather feedback from the community but also awake interest into the upcoming HOBBIT association. Our efforts have led to a total of 135 parties being contacted directly over the first 12 project months. In particular, we sent out a survey[14] to gather requirements from the community on benchmarking. Some of the results of this survey were presented at the Extended Semantic Web Conference 2016 and the European Data Forum 2016 and are described in the corresponding deliverable.[15] A second survey is planned to be sent out before the end of the first project year. This survey will gather information from the community to determine interest in services the HOBBIT association may offer, and to determine the types of services that the HOBBIT association should offer.

The total audience having heard about the project is now larger than 2,000 individuals (in events such as the Big Data Apache Conference 2016,[16] the International Sematic Web Conference 2016[17] and many more, see Table 3). Our focus was clearly European, with more than 80% of our contacts being located in Europe. However, we still aimed to gather information from other continents and now have approximately 5% of our contact in the Americas and 3% in Africa. We aim to continue building up our contacts and reach out directly to at least 250 people by the end of the project. We aim to work with these companies and academic partners to create the initial members of the HOBBIT association and extend these members gradually even after the end of the project. A corresponding outreach strategy is being prepared and will be deployed throughout 2017.

Event	**Participants**
Association for Computational Linguistics (ACL) 2016	≈ 300
Apache Big Data Conference (ApacheCon)	≈ 600
CEBIT 2016	≈ 300[18]
European Data Forum (EDF) 2016	≈ 400
Extended Semantic Web Conference (ESWC) 2016	≈ 300
International Sematic Web Conference (ISWC) 2016	≈ 400

Table 3: Excerpt of events where HOBBIT was presented

7 Summary

With this document, we aimed to give the interested public an overview of the past, ongoing and future activities in the HOBBIT project. We have been able to reach out successfully to the community and to gather benchmark requirements, datasets and key performance indicators. These matched well with the foreseen activities, leading to the consortium continuing the foreseen benchmarks without a

[14] https://docs.google.com/forms/d/1yFTroiYmdJfhQiUqVw0FYgqmbiPtL6FjcE6_J28o-gs/edit

[15] https://project-hobbit.eu/wp-content/uploads/2016/11/D1.2.1_Requirements_Specification_from_the_Community.pdf

[16] http://events.linuxfoundation.org/events/apache-big-data-europe/program/schedule

[17] http://iswc2016.semanticweb.org/

need for change. The creation of the benchmarks and data generator is ongoing and will be completed on time for the challenges. These will be the first outing of the benchmarking platform, which will allow for a fair evaluation of Big (Linked) Data processing technologies. Completing the benchmarks and mimicking algorithms, organizing and running the challenges and continuing to gather a corresponding community around HOBBIT will be at the center of the activities foreseen for next years. So far, the project is well on track to reaching all the goals described in the introduction. For more information, please visit our Web page at http://project-hobbit.eu. There, detailed information on the progress of the project and contact details are available.

References

[1] B. Berjawi and F. Duchateau and F. Favetta and M. Miquel and R. Laurini. Pabench: Designing a taxonomy and implementing a benchmark for spatial entity matching. In *GeoProcessing*, 2015.

[2] Pieter Colpaert, Alejandro Llaves, Ruben Verborgh, Oscar Corcho, Erik Mannens, and Rik Van de Walle. Intermodal public transit routing using linked connections. In *Proceedings of the 14th International Semantic Web Conference: Posters and Demos*, 2015.

[3] T. Saveta, E. Daskalaki, G. Flouris, I. Fundulaki, and A. Ngonga Ngomo. LANCE: Piercing to the Heart of Instance Matching Tools. In *ISWC*, 2015.