# Data Knowledge Base for HENP Scientific Collaborations

**V A Aulov[1, a], M V Golosova[1], M A Grigorieva[1,2,b], A A Klimentov[3], S Padolski[3], T Wenaus[3]**

[1]Kurchatov complex of NBIC-technologies, National Research Centre Kurchatov Institute, 1 Akademika Kurchatova pl., Moscow 123182, Russia
[2]Tomsk Polytechnic University, 4a Usov Street, Building 19, Tomsk, Russia
[3]Brookhaven National Laboratory, PO Box 5000 Upton, NY 11973-5000, USA

[a]vasiliyaulov@gmail.com
[b]maria.grigorieva@cern.ch

**Abstract**. Contemporary scientific experiments produce significant amount of data as well as scientific publications based on this data. Since volumes of both are constantly increasing, it becomes more and more problematic to establish a connection between a given paper and the underlying data. However, such an association is one of the crucial pieces of information for performing various tasks, such as validating the scientific results presented in paper, comparing different approaches to deal with a problem or even simply understanding the situation in some area of science. Authors of this paper are working under the Data Knowledge Base (DKB) R&D project, initiated in 2016 to solve this issue for the ATLAS experiment at CERN. This project is aimed at developing of the software environment, providing the storage and a coherent representation of the basic information objects. In this paper authors present a metadata model developed for the ATLAS experiment, the architecture of the DKB system and its main components. Special attention is paid to the Kafka-based ETL subsystem implementation and mechanism for extraction of meta-information from the texts of ATLAS publications

## 1. Introduction

The life cycle of a scientific experiment has changed considerably during the last decades. Data and metadata related to the experiment must be stored and available for reprocessing for years or even decades, as scientists may need to consult it in order to plan further research, compare results to the ones achieved with different approaches, initiate reprocessing of some experimental data to maintain reproducibility. And while storing data is a problematic issue itself, but finding the required parts of it among the petabytes of scientific knowledge and then processing it in a proper way is even more complex problem. The Data Knowledge Base (DKB) project [1], dealing with the metadata of high energy physics (HEP) experiments (with ATLAS as an example), is one of the attempts to address this problem.

DKB project was started in 2016 as a joint effort by ATLAS experiment at CERN, Kurchatov Institute and Tomsk Polytechnic University. The initial idea of the project was to provide a fast and user-friendly access to relevant scientific information regarding the physical analysis -- data used in the research, hardware/software configuration, publications and so on.

## 2. Data Knowledge Base Architecture

One of the key points of DKB architecture (figure 1) is the possibility to interact with structured and unstructured (documentary) data sources. For example, plain text or PDF documents are considered as one of the forms of data representation and can be used as any other processing object. A special tool for metadata extraction from PDF documents, including papers and internal notes (drafts) -- PDFAnalyzer -- is one of the specific parts of DKB system.

Information, extracted from heterogeneous data sources is integrated into a number of internal DKB storages. Each of these storages provides optimization for specific types of search or aggregation requests. Continuous data integration is performed by Apache Kafka-based [2] ETL subsystem, discussed in section 4 of this paper.

Internal storages of DKB include:

- ***Ontological storage*** OpenLink Virtuoso [3]. It allows to store the meta-information in form of Semantic Web (representing objects with their properties and connections between them). Virtuoso internal storage contains metadata, integrated from CERN Document Server (CDS) [4] and search engine for the ATLAS Collaboration (GLANCE) [5], and information, extracted from PDF documents by PDFAnalyzer.
- ***ElasticSearch storage*** [6] provides indexing of metadata from Production System [7] and ATLAS Metadata Interface (AMI) [8], ensuring fast search and aggregation capabilities.
- ***Transitional storage***, based on Hadoop, is used to store the interim data processing results. For example, temporary storage of PDF documents from CDS.

DKB interfaces, providing access to these internal storages:

- JavaScript-based library Ontodia is used as the interface to Virtuoso, allowing to visualize, navigate and explore the data in the form of an interactive graph based on underlying data sources.
- Kibana runs on the top of the ElasticSearch storage, providing possibility to generate aggregated reports, perform data filtration and basic search requests.
- NodeJS interface provides full-text search within ElasticSearch by a set of keywords.

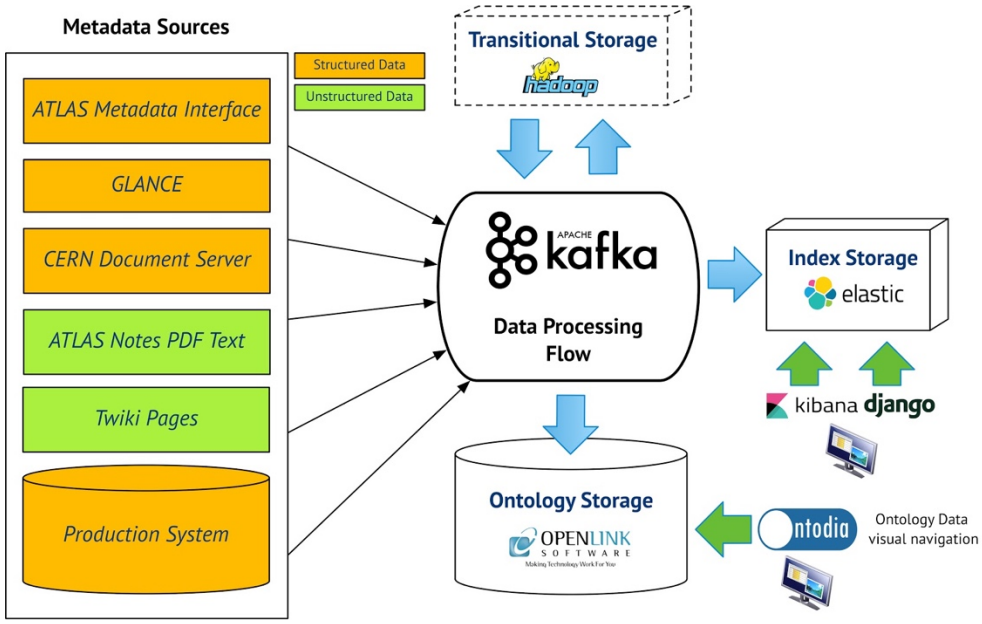


Figure 1 - DKB Architecture

## 3. ATLAS Ontological Data Model

To build a complete and coherent view of an experiment lifecycle various metadata must be integrated and interlinked. These metadata are strongly semantically connected and have a complex structure, but can be stored in loosely connected information subsystems of the experiment. With constantly growing volume of these metadata it is hardly possible to organize a relational storage, that would efficiently execute complex search requests, utilizing multiple interconnections (JOINs) between metadata items. The widely used technologies to solve the problem of strongly connected data are graph-based or ontology data model. In this study the ontological model of ATLAS experiment metadata was developed. It formalizes the description of scientific experiment phases, parameters, processes and other entities.

The main entities of this model are “Document”, “DataSample” and “PhysicsExperiment”. “Document” can be of several types: publication, internal document, journal issue, etc. All documents have attributes, corresponding to the ones in CDS meta-description, and obtained from PDF texts of these documents - experiment attributes and data samples. “Document” and “DataSample” can be associated to the “PhysicsExperiment” by a set of parameters: data samples, data processing project, Monte-Carlo simulation campaigns, keywords, physics group, the center-of-mass energy, integrated luminosity, data taking year.

“DataSample” is connected with “Task”, which corresponds to the real computational task in distributed data processing system (Production System). “Task” is the most informative object, containing the meta-information about software versions, detector architecture and calibration metrics, the number of processed events, task name, status and timestamps, input/output data samples, physics process metadata, hashtags and others. Connections between “Task”, “DataSample” and “Document” allow to gather complete information about physics analysis process.

## 4. ETL subsystem

The ETL subsystem of DKB, based on Apache Kafka, is organized as a set of independently developed modules, joined into a data flow topology.

Each of these modules implements a single ETL action: data extraction from an original source, transformation of the extracted data or load data into one of the internal storages.

Data *extraction modules* are run on a configured schedule by supervising Kafka Source Connector, which publishes extracted data to the corresponding Kafka topic(s) (a transitional storages between ETL stages).

Transformation process may consist of a number of steps, implemented as *transformation modules*, each responsible for a single logical operation: removing of excessive data, filtration, generation of surrogate keys or derived values, aggregations, retrieving additional information from other external sources (those that are not used as initial sources), and, finally, format conversion -- to prepare data for upload to the DKB integrated storages. These modules are organized into pipelines (or a more complicated topologies) by means of Kafka Streams library: every Kafka Processor is operating as a module supervisor, that makes sure that the module is running and ready to process input data as soon as it appear, taking care of restarting it in case of any failure and providing reliable communication mechanism between modules in standalone or distributed mode. These transformation topologies can be subscribed to one or more topics, and when new data are published by a Source Connector, transformation process begins with minimal delay. When a record, consumed from a source topic, passes through all the transformation steps, it is ready for the upload to some of DKB storages and goes to the topology sink (published to an output topic).

*Load modules*, final stages of ETL processes, responsible for data load to DKB internal storages, are subscribed to one or more of these output topics (now via supervising Kafka Sink Connectors). As soon as new data appears, load module start transferring it to the DKB storage -- and making available for user-oriented DKB services.

This ETL subsystem is oriented on a diverse development teams, as it allows to organize any ETL process as a combination of successive steps -- executable modules, sharing only input/output data

format and obeying common set of rules -- each of which can be developed independently, implemented by means of any program language, used as a standalone program or reused in a number of different ETL processes. The mentioned set of rules is required to unify chaining of the modules, and is simple enough to follow: standard input/output streams for input/output data, conventional message delimiter and simple flow control specification (sending control symbol at the end of message processing).

This subsystem also allows to minimize the delay between consuming information from original sources and placing transformed data to the internal storages.

Apache Kafka is used to address general tasks (such as organization of transitional storage between separate processing stage, operating in a distributed mode, restarting modules without massive data reprocessing, stateful processing, e.g. (such as data filtration or aggregation, etc) and to organize independent transformation modules into a topology of successive processing steps. Variety of Kafka configuration parameters allows to tune thoroughly the ETL subsystem as a whole and every ETL process individually, making it very flexible and adaptable for different use cases.

## 5. Metadata Extraction from PDF Documents

Scientific data analysis requires usage of the correct datasets (sets of data files) for the computational tasks and providing relevant information in scientific papers. There are ATLAS tools, such as AMI/GLANCE interface, intended for finding correlations between papers and datasets -- however, currently it can not be considered as a universal tool covering all documents and datasets. Therefore, alternative options were considered. One possibility for solving the task is provided by the ATLAS Internal Notes (the documents describing physics analysis in details) which are referenced by published papers. Though papers themselves can refer to the datasets being used for physics analysis, it is usually done in form of human-oriented descriptions -- unlike the Internal Notes, where datasets are specified explicitly.

Unfortunately, there is no general agreement to define the data samples in the Internal Notes. This makes the datasets automated extraction rather complicated. To deal with this obstacle, evaluation of existing tools for PDF processing was carried out. As a result, the tool called PDFMiner [9] was chosen to perform the basic functions such as converting PDF to text, while further analysis and processing were implemented by a module developed from scratch - PDFAnalyzer.

PDFAnalyzer can extract two categories of metadata from the texts -- experiment attributes and datasets. Experiment attributes -- data taking year, energy of the center of mass, integrated luminosity and Monte-Carlo production campaigns -- are found by means of regular expressions. Extracting datasets is a more complex procedure since they can be specified in several ways, most prevalent of which are text lists and tables.

- *Dataset names in text lists* are the easiest to find – since all the names are expected to comply with the ATLAS Dataset Nomenclature [10], regular expressions are usually a sufficient option to find them. However, nomenclature cannot be fully relied on due to reasons such as PDF to TXT conversion problems.
- *Dataset tables* usually contain only parts of dataset names (for example, dataset identificators). An algorithm for table reconstruction based on XML representation of the page layout, which explicitly states the position, size and font of each symbol, was developed.

The corpus of 133 ATLAS Internal Notes published between 2016 and 2017 was selected to test the developed program. Each document was analyzed, as well as checked manually whether it contains datasets or not. Out of 133 documents:

- 95 documents where datasets were found and extracted by the program;
- 12 documents contained datasets, but the program failed to detect them correctly;
- 26 contained no datasets and were correctly identified as such.

Therefore, roughly 89% of the documents with datasets in them were analyzed correctly. 12 remaining documents either have dataset names in text lists, but these names are specified only partially; or require better methods of table handling, which are currently being developed.

## Conclusion

A prototype of the described system provides continuous integration of metadata from heterogeneous sources, including extraction of information from unstructured (in terms of automated text analysis) PDF documents. This information is being gathered in the internal storages of DKB, and is accessible via number of interfaces, providing visualization of the metadata field in form of a connected graph (and navigation through it), google-like search and numerous aggregation tools. Further DKB development involves expansion of the ontological data model, addition of new metadata sources, development of services and data workflows, and advancing mechanisms of PDF documents analysis. The proposed scientific knowledge base will form a unified information field of any scientific experiment, and its architecture can be adapted to a variety of subject areas.

## Acknowledgements

The work was supported by the Russian Ministry of Science and Education under contract No.14.Z50.31.0024 and by the Russian Science Foundation under contract №16-11-10280.

## References

[1] M Grigorieva, V Aulov, A Klimentov and M Gubin 2016, Knowledge base of scientific experiment, *Open Systems. DBMS*, No. 4 of 2016, page 42
[2] Apache Kafka [Online]. Available: https://kafka.apache.org/ [accessed on: 27.02.2018]
[3] Erling O., Mikhailov I. RDF Support in the Virtuoso DBMS.. CEUR Proceedings, vol. 301. Proc. of the 1st Conference on Social Semantic Web, Leipzig, Germany, Sep 26- 28, 2007
[4] CERN Document Server [Online]. Available: https://cds.cern.ch [accessed on: 27.02.2018]
[5] ATLAS GLANCE [Online]. Available: https://atglance.web.cern.ch/atglance/ [accessed on: 27.02.2018]
[6] ElasticSearch [Online]. Available: https://www.elastic.co/products/elasticsearch [accessed on: 27.02.2018]
[7] K De, D Golubkov, A Klimentov, M Potekhin and A Vaniachine on behalf of the ATLAS Collaboration. Task Management in the New ATLAS Production System // Journal of Physics: Conference Series 513 (2014) 032078. doi:10.1088/1742-6596/513/3/032078. [Online]. Available: http://iopscience.iop.org/article/10.1088/1742-6596/513/3/032078/pdf [accesses on: 27.02.2018]
[8] ATLAS Metadata Interface [Online]. Available: http://ami.in2p3.fr/index.php/en/ [accessed on: 27.02.2018]
[9] PDFMiner [Online]. Available: http://euske.github.io/pdfminer/index.html [accessed on: 27.02.2018]
[10] ATLAS Dataset Nomenclature [Online]. Available: https://dune.bnl.gov/w/images/9/9e/Gen-int-2007-001_%28NOMENCLATURE%29.pdf [accesses on: 27.02.2018]