Data Quality Centric Application Framework for Big Data

Venkat N. Gudivada*, Dhana Rao †, and William I. Grosky ‡
*Department of Computer Science, East Carolina University, USA
†Department of Biology, East Carolina University, USA
‡Department of Computer and Information Science, University of Michigan - Dearborn, USA
email: gudivadav15@ecu.edu, raodh16@ecu.edu, and wgrosky@umich.edu

***Abstract*—Risks associated with data quality in Big Data have wide ranging adverse implications. Current research in Big Data primarily focuses on the proverbial harvesting of low hanging fruit and applications are developed using the Hadoop Ecosystem. In this paper, we discuss the risks and attendant consequences emanating from data quality in Big Data. We propose a data quality centric framework for Big Data applications and describe an approach to implementing it.**

***Keywords*—*Big Data*; *Data Quality*; *Data Analytics*; *Application Framework*.**

I. Introduction

Big Data, which has emerged in the last five years, has wide ranging implications for the society at large as well as individuals at a personal level. It has the potential for groundbreaking scientific discoveries and power for misuse and violation of personal privacy. Big Data poses research challenges and exacerbates data quality problems [1][2].

Though it is difficult to precisely define Big Data, it is often described in terms of five Vs – Volume, Velocity, Variety, Veracity, and Value. *Volume* refers to the unprecedented scale and velocity refers to the speed at which the data is being generated. The heterogeneous nature of data – unstructured, semi-structured, and structured – and associated data formats refers to the *variety* dimension. Typically Big Data goes through several data transformations from its inception before reaching the consumers. *Veracity* refers to data trustworthiness and *data provenance* is one way to specify veracity. Finally, the *value* dimension refers to unusual insights and actionable plans that are derived from Big Data through analytic processes.

A. Apache Hadoop Ecosystem

Currently, most of Big Data research is focused on issues related to volume, velocity, and value. These investigations primarily use the Hadoop Ecosystem which encompasses Hadoop Distributed File System (HDFS), a high-performance parallel data processing engine called Hadoop MapReduce, and various tools for specific tasks. For example, *Pig* is a declarative language for ad hoc analysis. It is used to specify dataflows to extract, transform, and load (ETL) process and analyze large datasets. Pig generates MapReduce jobs that perform the dataflows and thus provides a high level abstract interface to MapReduce. The *Pig Latin* enhances the Pig through a programming language extension. It provides common data manipulation operations such as grouping, joining, and filtering. *Hive* is a tool for enabling data summarization, ad hoc query execution, and analysis of large datasets stored in HDFS-compatible file systems. In other words, *Hive* serves as a SQL-based data warehouse for the Hadoop Ecosystem.

The other widely used tools in the Hadoop Ecosystem include Cascading, Scalding, and Cascalog. *Cascading* is a popular high-level Java API that hides many of the complexities of MapReduce programming. *Scalding* and Cascalog are even higher level and concise APIs to Cascading, accessible from Scala and Clojure, respectively. While Scalding enhances Cascading with matrix algebra libraries, Cascalog adds logic programming constructs.

Hadoop Ecosystem tools for the velocity dimension are the Storm and Spark. *Storm* provides a distributed computational framework for event stream processing. It features an array of spouts specialized for receiving streaming data form disparate data sources, incremental computation, and computing metrics on rolling data windows in real-time. Like Storm, Spark supports real-time stream data processing and provides several additional libraries for database access, graph algorithms, and machine learning.

Amazon Web Services (AWS) Elastic MapReduce (EMR) is a cloud-hosted commercial offering of the Hadoop Ecosystem from Amazon. Microsoft's StreamInsight is a commercial product for stream data processing with focus on complex event processing applications.

B. Industry Driven Big Data Research

Big Data research is primarily industry driven. Naturally, the focus is on the proverbial harvesting of the low-hanging fruit due to economic considerations. Research investigations in variety (aka data heterogeneity) and *veracity* dimensions are in the initial phases and tools are yet to emerge. However, data heterogeneity has been studied since 1980s in the database context,

 ISBN: 978-1-61208-457-2

where the focus has been on database schema integration and distributed query processing. These investigations assumed that the data is structured and all the component databases use one of the three data models – relational, network, and hierarchical. However, in the Big Data context, the data is predominantly unstructured, and semi-structured and structured data is derived using a processing framework such as the Hadoop MapReduce.

Typically Big Data is obtained from multiple vendor sources. Maintaining data provenance [3] – record of original data sources and subsequent transformations applied to the data – plays a central role in data quality assurance. Veracity investigations are beginning to appear under the umbrella term *data provenance* [4][5].

C. Data Quality Issues in Big Data

The *value* facet of Big Data critically depends on upstream data acquisition, cleaning, and transformation (ACT) tasks. Especially with the emerging Internet of Things (IoT) technologies, more and more data is machine generated. While some of the IoT data is generated under controlled conditions, most of it is created in environments which are subjected to fluctuations and thereby the quality of data can vary in an unpredictable manner. For example, the operating environment of wireless sensor and smart camera networks is subject to weather.

Data quality in ACT tasks has a direct bearing on volume, velocity, variety, and veracity facets of Big Data. Though data quality has been studied for over two decades, these investigations focused on database and information systems. Data quality assurance is the ultimate biggest challenge for Big Data management. Currently, data quality assurance requires intensive manual cleaning efforts. This is neither feasible nor economically viable.

In this paper, we discuss data quality issues in the Big Data context. We propose a data quality centric reference framework for developing Big Data applications. We also describe how this framework can be implemented using open source tools.

The remainder of the paper is structured as follows. Risks and implications of data quality are discussed in Section II. Data quality centric application framework for Big Data is described in Section III. Considerations for implementing this framework are discussed in Section IV. Finally, Section V concludes the paper.

II. Data Science, Risks and Implications of Data Quality

The ability to effectively process massive datasets has become integral to a broad range of scientific investigations. Data Science is a new interdisciplinary academic area with data driven approaches to problem solving as the foundation. Big Data has the potential to fundamentally affect all walks of life.

A. Data Science

Big Data is a double-edged sword. On the one hand, it enables scientists to overcome problems associated with small data samples. For example, it enables relaxing the assumptions of theoretical models, avoids over-fitting of models to *training data*, effectively deals with noisy training data, and provides ample *test data* to validate models.

Halevy, Norvig and Pereira [6] argue that the accurate selection of a mathematical model ceases its importance when compensated by *big enough* data. This insight is particularly important for tasks that are ill-posed for mathematically precise algorithmic solutions. Such tasks abound in natural language processing including language modeling, part-of-speech tagging, named entity recognition, and parsing. Ill-posed tasks also occur in applications such as computer vision, autonomous vehicle navigation, image and video processing.

Big Data enables a new paradigm for solving ill-posed problems by managing the complexity of the problem domain through building simple but high quality models by harnessing the power of massive data. For example, in the imaging domain, an image can be recovered by simply averaging successive image frames that are highly corrupted by a normally distributed Gaussian noise.

B. Big Data Risks

On the flip side, Big Data poses several challenges for personal privacy. It provides opportunities for using it in ways that are different from the original intention [7]. Cases of Big Data misuse abound. González et al. [8] describe how individual human mobility patterns can be accurately predicted. Their study tracked position history of 100,000 anonymized mobile phone users over a six-month period. Contrary to the existing theories, this study found that human trajectories show a high degree of temporal and spatial regularity and individual travel patterns collapse into a single spatial probability distribution. They conclude that despite the diversity of peoples' travel history, they follow simple reproducible travel patterns. In other words, there is a correlation between spatio-temporal history and a person's identity.

Another case in point is how the retailer Target used its customers' purchase history and other information to accurately predict their shopping needs. This information is used to issue relevant coupons and improve sales [9]. Davis describes how to balance the benefits of big data innovation with the risk of harm from unintended consequences in [10].

Big Data also entails negative and in some cases even disastrous results, if the insights discovered are based on poor quality data. As we go about performing our daily activities, we are inevitably generating a data trail – for example, location tracking though GPS in mobile phones. This data is captured and stored persistently

 ISBN: 978-1-61208-457-2

along with meta data, such as temporal information. It is possible to reconstruct a person's life history by fusing together seemingly disparate data acquired from multiple sources. This information can be further enhanced to gain insight into the behavior and activities of people, which in turn can be used to create new products and services. With only a little effort, personal data can be acquired, cleaned, aggregated, analyzed, sold, and repurposed [10].

C. Data Quality Issues

As indicated earlier, data quality issues have been studied for over two decades in the context of corporate data governance, enterprise data warehousing, and the Web. However, in the Big Data context, the following issues are either unique to Big Data or their severity is more pronounced: streaming data, disparate data types and multiple data vendors, preponderance of machine generated unstructured data, and integration difficulties. The problem of flight delay prediction illustrates the case in point. Solutions to this problem consider historical data, current weather, departure time, departing city, and other concurrent flights. These data can be obtained from multiple sources including FlightView Flight Tracker, Flight Aware, Flightwise Tracker Pro, and Orbitz. These sources use different terminology and their data conflicts with each other. Data quality issues that arise in Web data in the context of two applications – Stock Markets and Airline Flights – is investigated in [?].

Many organizations acquire massive datasets from diverse data vendors to complement their internally generated data. Usually, the data acquired from the vendors is produced without any specific application or analysis context. Therefore, the perceived meaning of the data varies with the intended purpose [11]. This necessitates defining data validity and consistency in the context of intended use. Big Data life cycle is relatively long. Another issue raised by this is the inconsistency between the recent copy of the vendor supplied data and the previous version copy of the same data. The latter has been modified to conform to intended use-specific validity and consistency checks.

In summary, the grand challenge for Big Data applications is developing automated tools for resolving data quality issues. Currently, Big Data analysis requires significant manual cleansing of input data.

III. Data Quality-centric Framework for Big Data Applications

We have investigated the requirements of telemedicine, environmental monitoring, and agriculture domains to help us define the data quality-centric framework for extracting value from Big Data. These requirements necessitate the framework to feature high performance computing cluster driven data analytics and knowledge extraction capabilities to harvest value from data. Data analytics and knowledge extraction involves managing complex and heterogeneous data and using advanced data fusion and information extraction algorithms. More specifically, automated tools are needed for processing and analyzing structured, semi-structured, unstructured data.

The structure of the framework is shaped by the tasks that are canonical across Big Data applications. Figure 1 shows task chain activities. We refer to this structure as Data Quality-centric Framework (DQF).

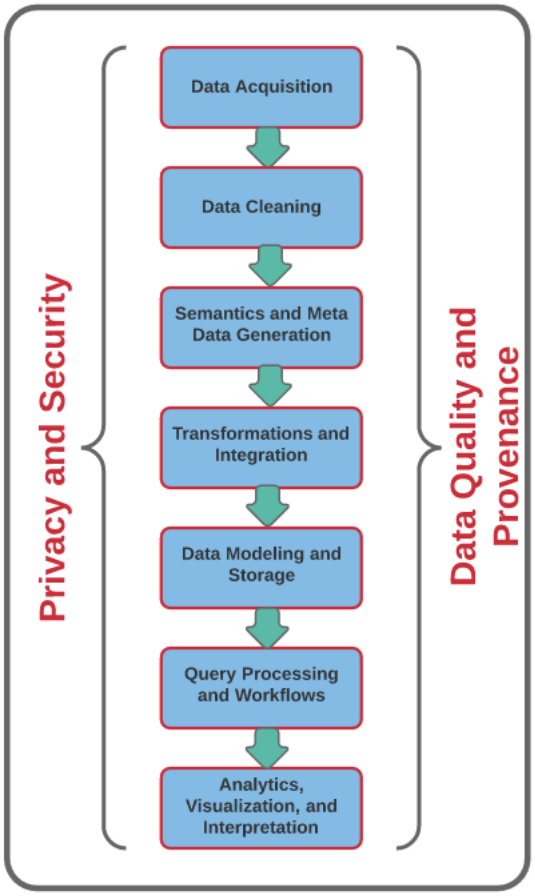

Figure 1. Workflows in data quality-centric framework for Big Data applications

A. Data Acquisition

Data acquisition devices can vary across the spectrum ranging from IoT, to wireless camera and sensor networks. Some of these devices are very simple in that they simply transmit the quantized data from the sensors. In addition to the sensed data, some devices will add meta data such as spatio-temporal and provenance data. Other devices may be much more sophisticated in that they employ sampling at Nyquist rate or variable sampling depending on the environmental conditions. Some devices may even apply real-time in-situ processing to detect anomalies and outliers and transmit only that data that has significance for intended data use.

 ISBN: 978-1-61208-457-2

Some data capture devices compress and transmit data using lossy or lossless data compression algorithms. Another important aspect of data acquisition is the scale dimension. Measurement theory specifies four levels or *scales* for assigning values to variables – nominal, ordinal, interval, and ratio. The chosen scale determines the type of processing that can be performed on the data.

B. Data Cleaning

This is one of the most investigated areas of data quality and provides approaches and algorithms for inferring missing data, resolving conflicting and inconsistent data, detecting integrity constraint violations, and detecting and resolving outliers. Duplicate detection and elimination are also important data cleaning tasks.

Ganti and Sarma [12] describe a set of data cleaning tasks in an abstract manner to enable developing solutions for the common data cleaning tasks. They also discuss a few popular approaches for developing such solutions. They take an operator-centric approach for developing a data cleaning platform. The operators are customizable and serve as building blocks for data cleaning solutions. Other works in this direction include [13], [14].

C. Semantics and Meta Data Generation

Bulk of the Big Data is unstructured in the form of video, images, audio, graphics, tweets, blogs, and natural language text. Information extraction techniques are used to turn unstructured data into semi- and structured data. For example, word boundary and sentence detection in spoken text, parts-of-speech tagging, parsing, named entity recognition, and coreference resolution are fundamental tasks in generating semi-structured representation from spoken and written text [15], [16].

D. Data Transformations and Integration

Once the unstructured data is transformed into semi- and structured representations, associated data from multiple sources is fused to link related data. This task is referred to by various names including record linking, entity resolution, and data matching. For example, recognizing various pictures of the same person generated under different conditions as one and the same is entity resolution in image data. Record linking may lead to unexpected privacy violations. For example, various pieces of information about an individual viewed in isolation may not entail privacy violation. However, if these pieces are fused together using record linking techniques, this may lead to serious privacy violations.

Traditionally, Extract, Transform, and Load (ETL) tools [17] have been used for transformations task that involve structured data. ETL tools enable rule-based data transformations in batch processing mode and are capable of transforming data formats and detecting anomalies and outliers.

E. Data Modeling and Storage

In relational database systems, data is uniformly modeled as relations. However, in the Big Data context, such a simple data model does not suffice. An assortment of new data models are used for Big Data and solutions based on such models are named NoSQL systems [18]. These systems can be grouped into classes, and each one meets the needs of a Big Data application category. Several data models for NoSQL systems have emerged during the last few years [19][20]. The new data models include key-value [21], column-oriented relational [22], column-family [23], document-oriented [24], and graph-based [25] [26]. The data modeling challenge in Big Data context is how to model heterogeneous data which requires multiple data models. It is more natural and practical to model the heterogeneous data using a collection of data models. Database-as-a-Service model [27] integrates a collection of data models and provides a unified interface.

Of late, the concept of data lakes is gaining popularity. A data lake is a storage repository that holds a vast amount of raw data in their native formats. Hadoop HDFS is often used for implementing data lakes since it is inherently better suited for storing large volumes of heterogeneous data with varying data formats.

F. Query Processing and Workflows

This component of the DQF addresses query processing and optimization, programmatic interfaces, and defining and executing workflows. MapReduce and distributed computing principles are used in realizing this component.

NoSQL databases for Big Data are expected to provide five basic operations: Create (insert), Read (retrieve), Update (modify), Delete, and Search (CRUDS). The read operation retrieves data based on a precise match as in relational databases. In contrast, the search operation provides functionality similar to a Web search engine — the query is often imprecise and incomplete and the retrieved results are based on similarity measures and full-text search. For example, document data model based NoSQL systems provide full-text search by integrating with search engines and libraries such as Solr, Lucene, and EaslticSearch.

NoSQL systems' query languages vary a spectrum from procedural to declarative. Query languages and client interfaces are influenced by the data model and underlying storage engine. For example, ad hoc queries in MongoDB (a document-oriented NoSQL system) are expressed as *map* and *reduce* functions written in Javascript. On the other hand, Cassandra (a column family NoSQL system) provides a SQL-like query language called Cassandra Query Language (CQL).

Big Data workflows are similar to the conventional workflows. However, Big Data workflows introduce additional complexity which arises from disparate data types, semi-structured and unstructured data, and dramatic

 ISBN: 978-1-61208-457-2

increase in storage and processing capacities. Given the volume and velocity of data, it should be possible to resume a failed workflow rather than starting all over.

G. Analytics, Visualization, and Interpretation

This is the final component of the DQF and features functionality for Big Data analytics to discover and visualize actionable insights. It involves automatic hypothesis generation and testing and visual analytics. The latter facilitates analytical reasoning through interactive exploration using visual interfaces with human in the loop.

Big Data analytics enables several functions including hypothesis testing, population inferencing, inferencing about individuals in the population, profile construction, and outlier discovery. Hypotheses are statements that need validation. Hypotheses are formulated based on predictions from theory, heuristics, or hunches. In some cases, though controversial, hypotheses are automatically generated. The goal of hypothesis testing is to determine whether a hypothesis is supported by the available data.

Attributes characterize entities in the population. *Population inferencing* determines whether or not correlations exist between certain attributes among entities in the population. *Inferencing about individuals* may reveal, for example, whether or not an individual has been exhibiting consistent behavior over a period of time.

Profile construction is used to identify key characteristics that describe population classes. For example, what are the key characteristics of impulsive buyers?

Outliers are those entities in the population whose characteristics are drastically different from rest of the population. One may simply discard outliers assuming that they have risen due to data quality errors, or they may signify a new trend. Explaining an outlier often leads to valuable insights into the problem domain.

In all of the above tasks, data quality plays a critical role. The quality of inferences obtained is affected by the upstream activities - data acquisition, cleaning, semantics and meta data generation, transformations and integration.

H. Privacy, Security, Data Quality and Provenance

These four facets pervade all the components of the DQF. *Differential privacy* is essential for Big Data. Just like authorization of entitlements in an application, differential privacy provides user access to data based on their job roles. Protecting the rights of privacy is a tremendous challenge. In 2013 alone, there were more than 13 million identity thefts in the United States [28]. Encryption in both hardware and software, and round the clock monitoring of security infrastructure are critical to protecting privacy.

A related issue is the notion of *personally identifiable information*, which is difficult to define precisely. Furthermore, as data goes through various transformations, it becomes even more difficult to identify and tag personally identifiable data elements. It has been shown that even anonymized data can often be re-identified and attributed to specific individuals [29].

Data perturbation is a technique for privacy preservation mainly used for electronic health records (EHR). It enables data analytics without compromising privacy requirements. It is considered as a more effective approach for privacy preservation of EHR compared to de-indentification and re-identification procedures.

Two types of data perturbation methods are suitable for EHR – *probability distribution* and *value distortion* approaches. In the first approach, the original data is replaced by data which is taken from the same distribution sample or from the distribution itself. In the second approach, the data is perturbed by adding randomly generated multiplicative or additive noise.

Security is implemented using access control and authorization mechanisms. *Access control* refers to ways in which user access to applications and databases is controlled. Databases limit access to those users who have been authenticated by the database itself or through an external authentication service, such as Kerberos [30]. *Authorization* controls what types of operations an authenticated user can perform. Access control and authorization capabilities of relational database systems have evolved over four decades. In contrast, data management for Big Data applications is provided by NoSQL systems [18]. Some systems provide limited security capabilities and others assume that the application is operating in a trusted environment and provide none.

As data goes through various processing steps, provenance is tracked and managed. Data provenance [31] is an issue that has received little or no attention from a security standpoint. As various transformations are applied to the data, metadata associated with provenance grows in complexity. The size of provenance graphs increases rapidly which makes analyzing them computationally expensive [4].

IV. Implementing the Framework

Shown in Figure 2 is a reference architecture for implementing the DQF of Figure 1. This modular architecture enables mix and match best of the breed components for its implementation. The architecture is generic, configurable, and lends itself for implementation using stable and field-tested open source software components. We refer to this as Data Quality-centric Framework Architecture (DQFA).

A. Remotely Deployed Wireless Sensor and Camera Networks

These are input capture devices which are connected though a wireless network. They are deployed in remote rural and mountainous communities that are not served by broadband networks. Wireless networks data is

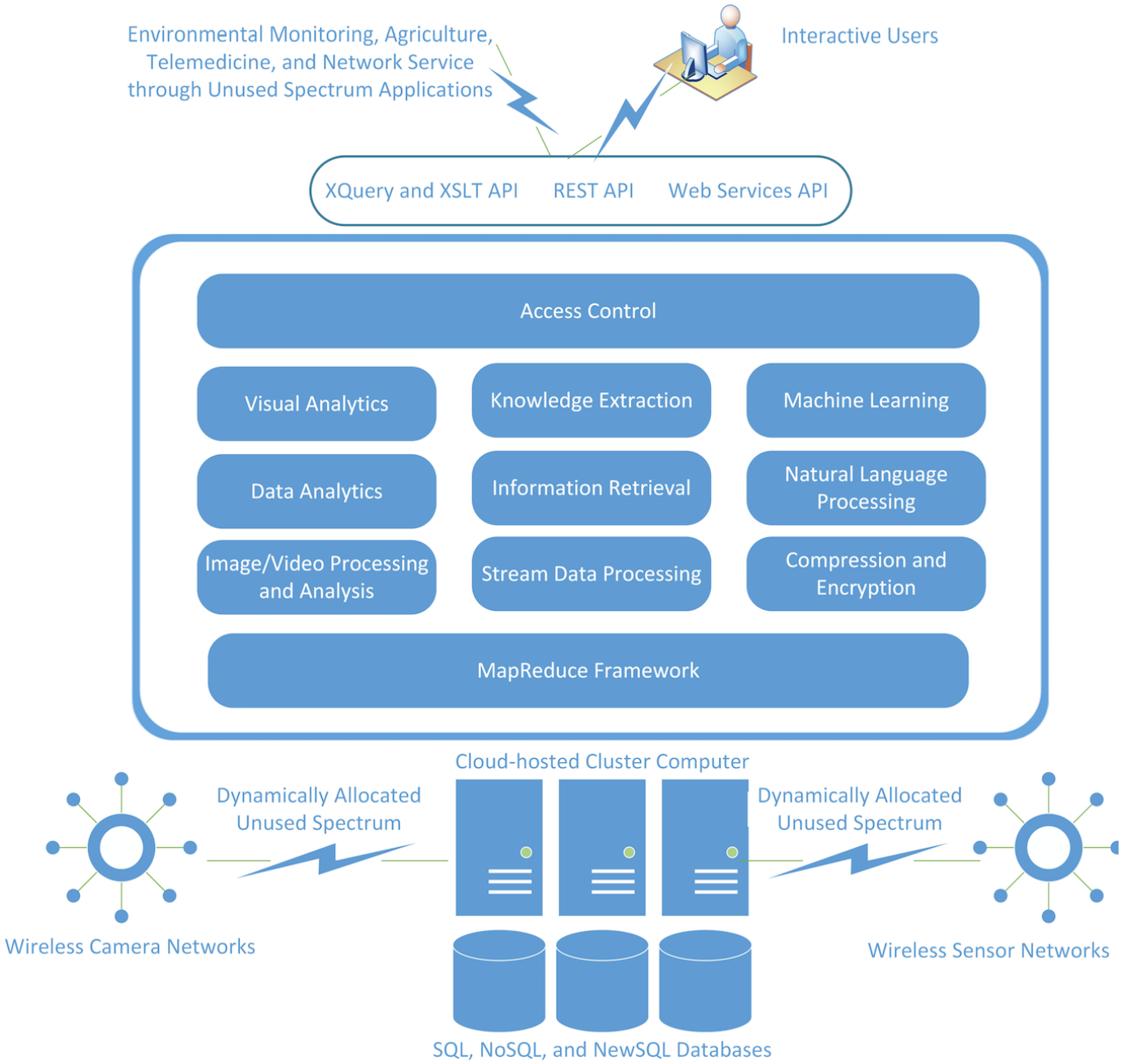

Figure 2. Implementing the data quality-centric framework architecture

transmitted to the cloud-hosted cluster computers using a dynamically allocated unused spectrum.

Innovative uses of unused spectrum (aka white spaces) is gaining momentum in the US [32]–[34]. In addition to its current primary use as rural broadband, other uses of white spaces include connectivity Web for IoT, monitoring of oil and gas exploration and drilling, utilities monitoring [35], and smart grids [36], [37].

B. Cloud-hosted Cluster Computer

Because of huge data volumes and the need for both batch and interactive processing, a cloud-hosted cluster computing platform will be used for implementing the DQFA. Each node in the cluster is self-contained and acts independently to remove single point of resource contention or failure. In this shared-nothing architecture, nodes share neither memory nor disk storage.

C. SQL, NoSQL, and NewSQL Databases

Until recently, Relational Database Management Systems (RDBMS) were the mainstay for managing all types of data. Underlying the RDBMS is the relational model for structuring data and SQL query language for data manipulation and retrieval. Though RDBMS are a perfect

fit for many applications, they maybe less suitable or an expensive for certain applications. An array of new systems for data management have emerged in recent years to address the needs of such applications.

Currently there are over 300 systems for data management and new ones are introduced routinely. They are referred to by various names including NoSQL, NewSQL, Not Only SQL, and non-RDBMS. By design, these new database systems do not provide all the RDBMS features. They principally focus on providing near real-time reads and writes in the order of billions and millions, respectively. The DQFA will leverage these advances for data storage and processing.

D. MapReduce Framework

MapReduce is computational paradigm for computing arbitrary functions on massive datasets in parallel if the computation fits a three-step pattern: map, shard and reduce. The *map process* is a highly parallel one comprised of several processes. Each one processes a different segment of data and produces (key, value) pairs. The *shard process* collects the generated pairs, sorts and partitions them. Each partition is assigned to a different *reduce* process, which produces one result. The DQFA infrastructure will feature MapReduce framework to speed up both batch and interactive jobs.

E. Compression and Encryption

Massive data volumes require compression as a means to reduce storage requirements. Encryption converts data into unreadable form to ensure data integrity and confidentiality. Some DQFA-driven applications require compression and encryption to meet regulatory compliance enforced by government and industry standards organizations. Tools we will consider for this task include Basic Compression Library, Google's Zopfli Compression Algorithm, LZ4, and LZ4_HC.

F. Stream Data Processing

The ubiquity of networked sensors is leading to sensorization of the real world. For example, some environmental monitoring applications generate streaming data. A concomitant effect is the emergence of many novel monitoring and control applications that require high-volume and low-latency processing.

Due to tremendous data volume, stream data is not stored in its entirety. First, the data is analyzed to determine which subset of it meets specified patterns and anomalies. Only such data is stored and processed further. The DQFA will provide an engine for stream data processing. Open source software to consider for this task include Apache Storm and Apache Spark.

G. Image/Video Processing and Analysis

Smart camera networks generate image and video data streams. It is not practical to store all streaming data. Techniques such as statistical sampling and abnormal event detection are required to identify image and video data that has informational value. Open source libraries to consider for this task include NIH's ImageJ, OpenCV, ImageMagick, CImg, Scipy, and Java Advanced Imaging (JAI).

H. Data Analytics

The set of tools required for data analytics is vast and varied. They encompass domain-independent descriptive and inferential statistics, as well as domain-specific processes and tools. Open source libraries to consider for this task include GNU Scientific Library (GSL), Computational Geometry Algorithms Library (CGAL), NumPy and SciPy.

I. Visual Analytics

Visual Analytics is an emerging area which integrates the analytic capabilities of the computer and the abilities of the human analyst [38], [39]. It is the science of analytical reasoning facilitated by visual interactive interfaces. High performance computing is the backbone of Visual Analytics.

Visual Analytics offers great potential for uncovering unexpected and hidden insights in heterogeneous healthcare data, which may lead to ground-breaking discoveries and profitable innovation [40]–[42]. Open source libraries to consider for this task include Flare, Gephi, Google Vis, Graph Viz, IVTK, D3.js, and JGraph.

J. Information Retrieval

Information Retrieval (IR) deals with modeling and retrieving of information from semi-structured and unstructured documents [43]. They provide full-text indexing and support various types of search including Boolean search and document-structure based search. They also rank the search results.

IR capability is essential for the DQFA to enable information fusion for knowledge extraction. Open source libraries to consider for this task include Apache OpenNLP, Stanford NLP, NLTK, Apache Lucene, Apache Solr, ElasticSearch, and Splunk.

K. Natural Language Processing

Natural Language Processing (NLP) based querying complements IR search [16]. NLP tools are available for part-of-speech tagging, named entity recognition, parsing, abstracting and summarization, text and speech generation, and machine translation.

NLP tools will be used in DQFA to extract information from unstructured documents and to enable knowledge extraction. These tools will also be used for providing flexible and natural interfaces for user interaction. Open source libraries to consider for this task include Natural Language Toolkit (NLTK), Stanford CoreNLP, WordNet, SRILM, Apache Lucene, MontyLingua, and tm.

 ISBN: 978-1-61208-457-2

L. Machine Learning

Machine learning algorithms are central to knowledge extraction. They include algorithms for basic statistics, feature extraction and transformation, classification and regression, clustering, dimensionality reduction, and optimization [44]. Machine learning libraries that we will consider include PyML, Apache Mahout, MLib, dlibml, WEKA, and scikit-learn.

M. Knowledge Extraction

Knowledge extraction is a domain-dependent task [45]. It involves creating knowledge from disparate sources of data and information represented in forms such as structured relational databases, semi-structured document databases, text corpora, image and video collections, semantic annotations, XML, RDF, and ontologies. Open source tools that we will consider for this task include AIDA, AlchemyAPI, Apache Stanbol, DBPedia Spotlight, FOX, FRED, and NERD.

N. Data Provenance

As computing has become distributed, the need to ensure security and privacy of data has increased greatly. In the proposed DQFA, data is created, processed, propagated, and consumed by diverse domain scientists, belonging to different security domains.

Secure data provenance is a key technology to ensure analysis results are reproducible by recording the lineage of data and information transformation processes. Tools that we will consider for this task are Pentaho Kettle, eBioFlow, PLIER, and SPADE.

O. Access Control

This component is used for user rights management. Access control provides two important functions. First, the identity of entities accessing the DQFA is confirmed. This step is referred to as *authentication*. Second, the confirmed entities are restricted to perform only those functions that are authorized. This step is referred to as *authorization*. We will consider tools such as OpenDJ, OpenIDM, OpenAM, DACS, and Shibboleth for implementation of this component.

P. Interfaces, Application Programming and User Access

We envision DQFA to provide several interfaces to promote flexible access to its services.

1) Interactive Users: Interactive users engage in exploratory style interaction with the system and expect real-time response. For example, visual analytics requires active participation of the domain scientists to discover patterns and formulate hypotheses.

2) Web Services API: This API will be designed to expose DQFA services to other applications. It enables applications to communicate and exchange data without concern for programming language, operating system, and network protocol issues.

3) REST API: Representational State Transfer (REST) is a minimal overhead Hypertext Transfer Protocol (HTTP) API for interacting with DAIS infrastructure. REST uses four HTTP methods – GET (for reading data), POST (for writing data), PUT (for updating data) and DELETE (for removing data).

4) XQuery and XSLT API: XQuery provides a declarative means for querying, updating, and transforming semi-structured and unstructured data mostly in the form of hierarchically structured XML documents. XQuery contains a superset of XPath expression syntax to address specific parts of an XML document. An extension to the XQuery/XPath language specifies how full-text search queries be specified as XQuery functions. XSLT is another declarative language for specifying how to transform an XML document into another.

XPath, XQuery, XQuery/XPath Full-text Search are all W3C standards. XSLT 3.0 has W3C Last Call Working Draft status. XQuery and XSLT API entail several advantages to the DAIS infrastructure. They include reduced applications development time through the use of standards, performance gain through elimination of data mappings between application layers by using the same data model, and enabling nontechnical staff to perform development and maintenance work.

V. Conclusions

Data quality plays a critical role in Big Data applications. As data goes through various transformations and meanders from upstream to downstream applications, data quality errors propagate and accumulate. These error have the potential to cause detrimental consequences for an organization or individual. The data quality centric application framework model we proposed is intended to serve as a reference model to promote data quality research in Big Data context. Our future research direction is to implement this framework.

References

[1] V. Gudivada, D. Rao, and V. Raghavan, *Big Data Analytics*. Elsevier, 2015, ch. Big Data Driven Natural Language Processing Research and Applications, pp. 203 – 238.

[2] V. Gudivada, R. Baeza-Yates, and V. Raghavan, "Big data: Promises and problems," *IEEE Computer*, vol. 48, no. 3, pp. 20–23, Mar. 2015.

[3] P. Buneman, J. Cheney, W.-C. Tan, and S. Vansummeren, "Curated databases," in *Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, ser. PODS '08. New York, NY, USA: ACM, 2008, pp. 1–12.

[4] Y.-W. Cheah, "Quality, retrieval and analysis of provenance in large-scale data," Ph.D. dissertation, Indianapolis, IN, USA, 2014.

[5] J. Cheney, P. Buneman, and B. Ludäscher, "Report on the principles of provenance workshop," *SIGMOD Rec.*, vol. 37, no. 1, pp. 62–65, Mar. 2008.

[6] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 8 – 12, 2009.

[7] M. R. Wigan and R. Clarke, "Big data's big unintended consequences," *Computer*, vol. 46, no. 6, pp. 46–53, Jun. 2013.

[8] M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, Jun. 2008.

 ISBN: 978-1-61208-457-2

[9] C. Duhigg. How companies learn your secrets. [retrieved: December, 2015]. [Online]. Available: http://www.nytimes.com/2012/02/19/magazine/shopping-habits.html?_r=0

[10] K. Davis, *Ethics of Big Data: Balancing Risk and Innovation*. O'Reilly Media, Inc., 2012.

[11] D. Loshin. Understanding big data quality for maximum information usability. [retrieved: December, 2015]. [Online]. Available: http://www.dataqualitybook.com

[12] V. Ganti and A. D. Sarma, *Data Cleaning: A Practical Perspective*, ser. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2013.

[13] J. W. Osborne, *Best Practices in Data Cleaning: A Complete Guide to Everything You Need to Do Before and After Collecting Your Data*. SAGE Publications, 2012.

[14] Q. E. McCallum, *Bad Data Handbook: Cleaning Up The Data So You Can Get Back To Work*. O'Reilly Media, 2012.

[15] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, 2009.

[16] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Prentice Hall, 2009.

[17] M. Casters, R. Bouman, and J. van Dongen, *Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration*. Wiley, 2010.

[18] V. Gudivada, D. Rao, and V. Raghavan, "Renaissance in data management systems: Sql, nosql, and newsql," *IEEE Computer, forthcoming*.

[19] solid IT. Knowledge base of relational and NoSQL database management systems. [retrieved: December, 2015]. [Online]. Available: http://db-engines.com/en/ranking

[20] A. Schram and K. M. Anderson, "Mysql to nosql: Data modeling challenges in supporting scalability," in *Proceedings of the 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity*, ser. SPLASH '12. New York, NY, USA: ACM, 2012, pp. 191–202.

[21] R. Gandhi, A. Gupta, A. Povzner, W. Belluomini, and T. Kaldewey, "Mercury: Bringing efficiency to key-value stores," in *Proceedings of the 6th International Systems and Storage Conference*, ser. SYSTOR '13. New York, NY, USA: ACM, 2013, pp. 6:1–6:6.

[22] Z. Liu, S. Natarajan, B. He, H.-I. Hsiao, and Y. Chen, "Cods: Evolving data efficiently and scalably in column oriented databases," *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 1521–1524, Sep. 2010.

[23] A. Lakshman and P. Malik, "Cassandra: A structured storage system on a p2p network," in *Proceedings of the Twenty-first Annual Symposium on Parallelism in Algorithms and Architectures*, ser. SPAA '09. New York, NY, USA: ACM, 2009, pp. 47–47.

[24] P. Murugesan and I. Ray, "Audit log management in mongodb," *2014 IEEE World Congress on Services*, pp. 53–57, 2014.

[25] R. Angles, "A comparison of current graph database models," *2014 IEEE 30th International Conference on Data Engineering Workshops*, pp. 171–177, 2012.

[26] I. Robinson, J. Webber, and E. Eifrem, *Graph Databases*. O'Reilly, 2013.

[27] D. Agrawal, A. El Abbadi, F. Emekci, and A. Metwally, "Database management as a service: Challenges and opportunities," in *Data Engineering, 2009. ICDE '09. IEEE 25th International Conference on*, March 2009, pp. 1709–1716.

[28] United Credit Service. Identity theft; will you be the next vitcim? [retrieved: December, 2015]. [Online]. Available: https://ucscollections.wordpress.com/2014/03/06/identity-theft-will-you-be-the-next-victim/

[29] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," *2013 IEEE Symposium on Security and Privacy*, vol. 0, pp. 111–125, 2008.

[30] S. T. F. Al-Janabi and M. A. S. Rasheed, "Public-key cryptography enabled kerberos authentication," in *Developments in E-systems Engineering (DeSE), 2011*. IEEE Computer Society, Dec 2011, pp. 209–214.

[31] U. Braun, A. Shinnar, and M. Seltzer, "Securing provenance," in *Proceedings of the 3rd Conference on Hot Topics in Security*, ser. HOTSEC'08, 2008, pp. 4:1–4:5.

[32] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh, "White space networking with wi-fi like connectivity," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 27–38, Aug. 2009.

[33] R. Chandra, T. Moscibroda, P. Bahl, R. Murty, G. Nychis, and X. Wang, "A campus-wide testbed over the tv white spaces," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 15, no. 3, pp. 2–9, Nov. 2011.

[34] R. Chandra, "White space networking beyond the tv bands," in *Proceedings of the Seventh ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, ser. WiNTECH '12, 2012, pp. 1–2.

[35] C.-S. Sum, H. Harada, F. Kojima, Z. Lan, and R. Funada, "Smart utility networks in tv white space," *IEEE Communications Magazine*, vol. 49, no. 7, pp. 132–139, 2011. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5936166

[36] G. Nychis, B. DeBruhl, and H. Tang, "Demo: Tv white space networking capabilities and potential with an embedded & open-api platform," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '14, 2014, pp. 309–312.

[37] S. W. Oh, F. Chin, and S. G. Kerk, "Tv white-space for smart grid," in *Proceedings of the 4th International Conference on Cognitive Radio and Advanced Spectrum Management*, ser. CogART '11, 2011, pp. 56:1–56:5.

[38] P. Alzamora, Q. V. Nguyen, S. Simoff, and D. Catchpoole, "A novel 3d interactive visualization for medical data analysis," in *Proceedings of the 24th Australian Computer-Human Interaction Conference*, ser. OzCHI '12. New York, NY, USA: ACM, 2012, pp. 19–25.

[39] T. E. Hansen, J. P. Hourcade, A. Segre, C. Hlady, P. Polgreen, and C. Wyman, "Interactive visualization of hospital contact network data on multi-touch displays," in *Proceedings of the 3rd Mexican Workshop on Human Computer Interaction*, ser. MexIHC '10. San Luis Potos, S.L.P. Mexico, Mxico: Universidad Politcnica de San Luis Potos, 2010, pp. 15–22.

[40] J. J. Caban and D. Gotz, "2011 workshop on visual analytics in healthcare: understanding the physician perspective," *SIGHIT Rec.*, vol. 2, no. 1, pp. 29–31, Mar. 2012.

[41] F. Fischer, F. Mansmann, and D. A. Keim, "Real-time visual analytics for event data streams," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ser. SAC '12. New York, NY, USA: ACM, 2012, pp. 801–806.

[42] D. Gotz and J. Sun, "Ieee visweek workshop on visual analytics in health care 2010," *SIGHIT Rec.*, vol. 1, no. 1, pp. 31–32, Mar. 2011.

[43] C. D. Manning, P. Raghavan, and H. Sch´utze, *Introduction to Information Retrieval*. Wiley, 2010.

[44] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning From Data*, 2012.

[45] R.-S. Chen, C.-C. Chang, and I. Chi, "Ontology-based knowledge extraction-a case study of software development," in *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006. Seventh ACIS International Conference on*, Jun. 2006, pp. 91–96.

 ISBN: 978-1-61208-457-2