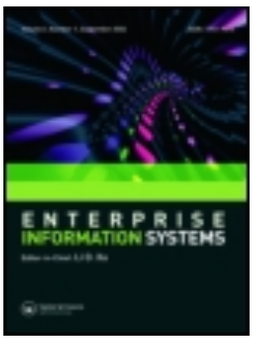

Enterprise Information Systems

ISSN: 1751-7575 (Print) 1751-7583 (Online) Journal homepage: http://www.tandfonline.com/loi/teis20

# Schema on read modeling approach as a basis of big data analytics integration in EIS

Slađana Janković, Snežana Mladenović, Dušan Mladenović, Slavko Vesković & Draženko Glavić

**To cite this article:** Slađana Janković, Snežana Mladenović, Dušan Mladenović, Slavko Vesković & Draženko Glavić (2018): Schema on read modeling approach as a basis of big data analytics integration in EIS, Enterprise Information Systems, DOI: 10.1080/17517575.2018.1462404

**To link to this article:** https://doi.org/10.1080/17517575.2018.1462404

Published online: 18 Apr 2018.

ENTERPRISE INFORMATION SYSTEMS, 2018
https://doi.org/10.1080/17517575.2018.1462404

Taylor & Francis
Taylor & Francis Group

ARTICLE



# Schema on read modeling approach as a basis of big data analytics integration in EIS

Slađana Janković, Snežana Mladenović, Dušan Mladenović, Slavko Vesković and Draženko Glavić

Faculty of Transport and Traffic Engineering, University of Belgrade, Belgrade, Serbia

**ABSTRACT**

Big Data analysis is the process that can help organizations to make better business decisions. Organizations use data warehouses and business intelligence systems, i.e. enterprise information systems (EISs), to support and improve their decision-making processes. Since the ultimate goal of using EISs and Big Data analytics is the same, a logical task is to enable these systems to work together. In this paper we propose a framework of cooperation of these systems, based on the schema on read modeling approach and data virtualization. The goal of data virtualization process is to hide technical details related to data storage from applications and to display heterogeneous data sources as one integrated data source. We have tested the proposed model in a case study in the transportation domain. The study has shown that the proposed integration model responds flexibly and efficiently to the requirements related to adding new data sources, new data models and new data storage technologies.

**ARTICLE HISTORY**
Received 14 September 2017
Accepted 2 April 2018

**KEYWORDS**
Big data analytics; data virtualization; schema on read; data warehouse; business intelligence system

## Introduction

A large number of new approaches and technological solutions in data modeling, storage, processing and analysis, grouped together under the common term 'Big Data', have the task of keeping under control the massive inflow of data and placing it in the service of organizations and individuals. The initial successful initiatives in the application of Big Data technologies soon gave rise to a problem known as Big Data integration. Big Data integration means any software integration involving the data characterized as Big Data, i.e. the data with at least one of the following features: volume, variety, velocity and veracity. According to Arputhamary and Arockiam (2015), there are two categories of Big Data integration, namely integration of several Big Data sources in Big Data environments and integration of the results of Big Data analysis with structured corporate data. This research is focused on addressing the second, above-mentioned category of the Big Data integration problem.

An Enterprise Information System (EIS) is an integrated information system with the basic task of providing the management with the necessary information. This research addresses two major challenges encountered by modern EISs in the sphere of data management in order to be qualified as 'integrated' as per the above definition. The promotion of business operation of organizations nearly always involves the introduction of new sources of corporate data. If new data sets fall into the category of Big Data, they require the application of Big Data storage, processing and analysis

**CONTACT** Slađana Janković s.jankovic@sf.bg.ac.rs Faculty of Transport and Traffic Engineering, University of Belgrade, Belgrade, Serbia

methods. To use new corporate Big Data sets in a business context, they have to be integrated with the existing corporate data sets, after which the integrated data should be subjected to Big Data analysis. The integration of the existing and new corporate data sets to create the subject of the future Big Data analysis is the first challenge to which this research will try to respond. The second challenge and the subject of this research is the integration of the results of Big Data analysis with EIS. This task has to be solved regardless of whether corporate or external data are the subject of Big Data analysis. External data, such as social media and web data, are increasingly used as the subject of Big Data analyses in order to examine user satisfaction, habits and needs etc.

Zdravković and Panetto (2017) highlighted that current challenges in EISs development are related to the growing need for flexibility caused by cooperation with other EISs. EISs environment has become very dynamic and variable not only in terms of collaboration with other EISs, but also in terms of availability of data sources. The research aims to offer a solution that would efficiently meet the following three key requirements: frequent appearance of new Big Data sources (either corporate or external), application of new data processing, analysis and visualization methods, and integration of structured (i.e. relational) and semi- and non-structured data sources. To solve the above problems, the schema alignment method of data integration has been selected. The traditional schema alignment method of data integration has been adapted to Big Data sources and methods of Big Data analysis by being based on the schema on read data modeling approach and data virtualization concepts. Schema on read means you create the schema only when reading the data. Structure is applied to the data only when it's read, this allows unstructured data to be stored in the database. Since it's not necessary to define the schema before storing the data it makes it easier to bring in new data sources on the fly. Data virtualization is any approach to data management that allows an application to retrieve and manipulate data without requiring technical details about the data, such as how it is formatted at source, or where it is physically located. The research also provides a technological framework for the implementation of the proposed integration model. It includes the following three technological environments: NoSQL databases, data virtualization servers and data integration tools.

The second section of this paper presents the reference literature review. In the third section, we propose and describe our Big Data analytics integration approach based on the 'data integration on demand' approach and the 'schema on demand' modeling approach. In order to evaluate our approach, we have implemented the proposed approach in a case study in the transportation domain. We have carried out the custom analysis of road traffic data on a Big Data platform and integrated it with the SQL Server database, Business Intelligence (BI) tool and traffic geo-application, according to the proposed integration approach. Finally, we will present our conclusions about the possibilities and constraints of our integration approach.

## Literature review

As pointed out in the introduction of the paper, this research does not deal with the integration of different Big Data sources on Big Data platforms but with the integration of the results of Big Data analysis with structured corporate data. For this reason, the literature review includes the data integration approaches and solutions that can be applied to Big Data sources as well as the existing EIS architectures.

For decades, there have been two main approaches to data integration, namely batch data integration and real-time data integration. Both approaches have secured a place for themselves in Big Data integration processes as well. From the data analytics perspective, Big Data systems support the following classes of applications: batch-oriented processing, stream processing, OLTP (Online Transaction Processing) and interactive ad-hoc queries and analysis (Ribeiro, Silva, and da Silva 2015). The batch data integration approach is used in batch-oriented processing applications, whereas the real-time data integration approach is used in stream processing, OLTP and interactive ad-hoc queries and analysis applications. An overview of the most important approaches and

solutions in the field of Big Data integration with EISs, both in the batch as well as the real-time mode, will be given in the text below.

### *Batch data integration for big data*

When data exchange between two systems is performed through periodic big file transfers on a daily, weekly or monthly basis, we call this batch data integration. In the era of the Internet of Things (IoT) and social media, i.e. the era of Big Data, this interval between two successive file transfers can be much shorter and measured in hours or even minutes. The transferred files include records with an unchangeable structure, which is adapted to the requirements of the system that receives them. This approach to integration is known as a 'tightly coupled' approach, because it implies that systems are compatible in terms of file and data format and that the format can only be changed if both systems simultaneously implement specific changes (Reeve 2013). The standard batch data integration process includes the following operations: extract, transform, and load (ETL). Today, there is a large number of commercial and open-source ETL tools (Alooma 2018). The main purpose of these tools is to upgrade and facilitate the warehousing, archiving, and conversion of data.

Big Data are most frequently raw data, which are 'dirty' and incomplete and therefore it is necessary to perform the operations of extracting, cleaning and data quality processing (Macura 2014; Chen and Zhang 2014) in order to work with them. In the Big Data context, ETL tools are used to extract, clean and transform raw data from Big Data platforms and NoSQL databases into a relational or another required form, as well as to load the results of Big Data analytics into Enterprise Data Warehouses (EDWs) (Florea, Diaconita, and Bologa 2015). The task can only be performed by ETL tools enabling the creation of interfaces according to both traditional data sources (relational databases, flat files, XML files, etc.) as well as Big Data platforms (Hortonworks Data Platform, Cloudera Enterprise, SAP HANA Platform, etc.) and NoSQL databases (MongoDB, Cassandra, HBase, Neo4j, etc.). Such commercial ETL tools include Informatica, Oracle Data Integrator, Alooma, SAS ETL and Altova MapForce. The major open-source tools of this type include Apache NiFi, Talend and Pentaho Data Integration.

Transformation as an operation can vary, ranging from an extremely simple operation to an inexecutable operation, and it may require the use of additional data collections. In the simplest case, it consists of the simple mapping of source fields to target fields, but most frequently it also includes operations such as aggregation, normalization and calculation. Some ETL tools, such as Altova MapForce, include a revolutionary interactive debugger to assist with the data mapping design.

Apache Hadoop is an open-source distributed software platform for storing and processing data. Central to the scalability of Apache Hadoop is the distributed processing framework known as MapReduce (Sridhar and Dharmaji 2013). According to the research done by Russom (2013), the main reason to integrate Hadoop into Business Intelligence or Enterprise Data Warehouse is the expectation from Hadoop to enable Big Data analytics. The basic advantage of Hadoop is the possibility to use advanced non-OLAP (Online Analytic Processing) analytic methods, such as data mining, statistical analysis and complex SQL. However, in addition to the fact that it can be used as an analytical sandbox, Apache Hadoop includes many components useful for ETL. For example, Apache Sqoop is a tool for transferring data between Hadoop and relational databases. When data are located in the Hadoop File System, they can be efficiently subjected to the ETL tasks of cleansing, normalizing, aligning, and aggregating for an EDW by employing the massive scalability of MapReduce (Intel Corporation 2013). In this way, the Apache Hadoop platform represents a powerful ETL tool enabling the integration of the results of Big Data analysis of structured and non-structured data in an EDW.

Research (Wang et al. 2016) has shown that the most important Big Data technologies that support batch data integration include the following: MapReduce, Hadoop (HDFS, Hive, HBase),

Flume, Scribe, Dryad, Apache Mahout, Jaspersoft BI Suite, Pentaho, Skytree Server, Cascading, Spark, Tableau, Karmasphere, Pig and Sqoop.

### *Real-time data integration for big data*

In many cases of data integration, the batch mode is unacceptable so that real-time or near real-time data integration has to be performed instead. Real-time data integration involves the transfer of much smaller quantities of data in one interaction, in the form known as a 'message' (Gokhe 2016). The quantity of data transferred in this way is limited and each interaction means ensuring security on all levels, the same as in batch data integration. Consequently, when it comes to larger quantities of data, real-time data movement is slower than batch data movement. The traditional 'point-to-point' interaction model means that there are direct 'tightly coupled' interfaces between each two systems which have to share data. The data from each data source have to be transformed as per the requirements of each target data format. If the number of systems which should be connected by an interface is n, the number of interfaces is (n * (n – 1))/2. The most significant and most important design pattern for architecting real-time data integration solutions is the 'hub-and-spoke' design for data interactions (Reeve 2013). The point of this interaction model is that data from all sources are transformed into a common, shared format, from which they are transformed into the target format. The number of interfaces for the connection of n systems is n in this case. From the technological point of view, the central segment of the real-time data integration solution is the implementation of an enterprise service bus (ESB). An enterprise service bus is an application used to coordinate the movement of data messages across different servers that may be running different technologies.

XML (eXtensible Markup Language) has been a de facto standard for the exchange of information in the past two decades and, consequently, it also plays a major role in the field of data integration. XML files are a typical example of semi-structured data (Gandomi and Haider 2015). Modern data integration software enables the transformation of data from XML files into other types of data warehouses (Big Data included) and vice versa. Other self-documenting data interchange formats that are popular include JSON (Java Script Object Notation).

Hadoop offers excellent performances in the processing of massive data sets, but query execution on the Hadoop platform (e.g. Hive queries) is measured in minutes and hours. This constitutes a great challenge in the integration of Hadoop into a real-time analytics environment. Intel and SAP have joined forces to tackle this challenge (Intel Corporation 2014). The Intel® Distribution for Apache Hadoop (IDH) is highly optimized for performance on Intel® architecture. Intel and SAP have enabled the generation of queries that will be efficiently executed on both platforms, SAP HANA as well as IDH.

Research (Wang et al. 2016) has shown that the most important Big Data technologies that support stream processing and real-time integration include the following: Kafka, Flume, Kestrel, Storm, SQLstream, Splunk, SAP Hana and Spark Streaming.

### *Schema alignment in big data integration*

The main task of data integration, regardless of whether it is traditional or Big Data integration, batch or real-time data integration, is to download the required data from their current warehouse, to change their format in order to be compatible with the destination warehouse and to place them at the target location (Loshin 2013). It is the challenges which data integration has to address that have changed. The three main steps in data integration include schema alignment, record linkage and data fusion. Schema alignment should respond to the challenge of semantic ambiguity, enabling the identification of attributes with the same meaning as well as those without it. Record linkage should find out which records refer to the same entity and which do not. Data

fusion should enable the identification of accurate data in an integrated data set in cases when different sources offer conflicting values.

Dong and Srivastava (2015, 35) underline that, 'schema alignment is one of the major bottlenecks in building a data integration system'. They believe that in the Big Data context, where the number of data sources is permanently on the rise and where source schemas are expected to change all the time, no up-to-date schema mappings are possible. In contrast, Gal (2011) speaks of the important role schema matching plays in the data integration life cycle. He believes that the Big Data challenges of variety and veracity can be dealt with by using schema matching, while the challenges of volume and velocity can be dealt with by using entity resolution (record linkage).

## Big data analytics integration framework

This section of the paper presents the framework for the integration of Big Data sources with structured data sources, which still form the backbone of EISs. In the previous section, we have seen that both the batch data integration approach as well as the real-time data integration approach have their advantages as well as disadvantages and, consequently, our goal has been to propose a model capable of supporting both integration methods.

In view of the fact that EISs are based on structured data (data warehouses, predefined business analytics and reports, etc.), we believe that variety and veracity constitute the key challenges in the integration of Big Data analysis and EISs. The integration framework we propose is therefore based on the upgrade of the model of application of the schema alignment (schema matching) method of data integration. The upgrade is expected to be the result of the application of the schema on read modeling approach and data virtualization concepts. In the text below, the two approaches will be first briefly outlined and then the reason why they have been selected explained.

### *Schema on read modeling approach in big data integration process*

Schema on write is a standard modeling approach, where we create a database schema and a database for a specific purpose, and then we enter data into the database. This means that the data must be adequately prepared for the developed schema. The schema on read approach involves storing raw data, and then, when we need it for a specific purpose, we create a schema while reading data from a data storage (Figure 1). Unlike schema on write, which requires you to expend time before loading the data, schema on read involves very little delay and you generally store the

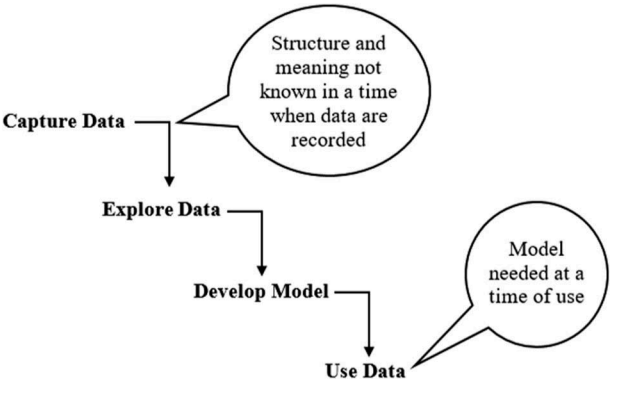


**Figure 1.** Schema on read modeling approach.

data at a raw level. In data-intensive computation problems data is the driver, not analytical human or machines. When the schema on read modeling approach is used, these very large data sets can be used multiple times in different ways, for various types of analysis. However, we believe that the schema on read modeling approach has a big potential not only in the field of Big Data analysis but also in the field of Big Data integration.

According to (EMC Education Services, ed 2015), the main phases of the data analytics life cycle include data discovery, data preparation, model planning, model building, communicate results and operationalize. However, in our experience, Big Data integration process, too, has to include almost all above phases, as shown in Figure 2. Consequently, we shall speak of the roles the schema on read modeling approach plays in all mentioned activities, as the phases of Big Data integration process:

- Phase 'discovery': at this stage, the schema on read modeling approach plays an important role in getting to know the team with data and the selection of appropriate data preparation methods.
- Phase 'data preparation': given that the possibilities of data transformation with ETL tools are nevertheless limited, the data in Big Data source systems have to be organized and formatted so as to be able to be transformed with ETL tools into the format required by EIS. The data in Big Data source systems can be prepared for ETL operations through adequate modeling. Data modeling when necessary, at the point of reading, is precisely what the schema on read modeling approach makes possible. In this way, ETL operations are more effectively realized using the schema on read modeling approach.
- Phase 'model planning': the schema on read modeling approach allows a deeper exploration of data and recognition of the relationships between individual variables.
- Phase 'model building': at this stage, the schema on read modeling approach has the most significant role, because it allows flexible creation, testing and changing of the models. In data integration process, the phases of 'model planning' and 'model building' can occur several times. They will definitely occur during the ETL operations and, if there is a data virtualization level, they will occur also during the creation of virtual tables.

Due to the above roles the schema on read modeling approach can play in Big Data integration process, we believe that this modeling approach is imperative for efficient Big Data integration.

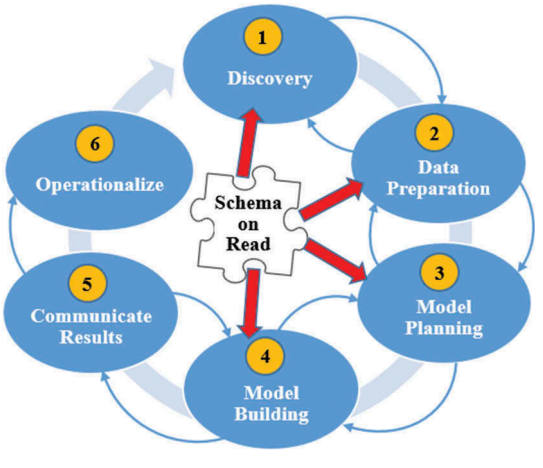


**Figure 2.** Schema on read modeling approach in Big Data integration lifecycle.

### *Data virtualization integration approach for big data analytics*

Big Data analytics is characterized by a permanent appearance of new data sources and new requirements regarding analytical models and methods, so that we have tried to adopt an integration approach likely to ensure a satisfactory degree of flexibility. We have recognized the data virtualization concept as a suitable basis for flexible 'on-demand' integration and multiple use of the same data, without copying.

As van der Lans (2012, 9) points out, 'Data virtualization is the technology that offers data consumers a unified, abstracted, and encapsulated view for querying and manipulating data stored in a heterogeneous set of data stores'. Basically, when data virtualization is applied, the middle layer that hides from an application most of the technical aspects on where and when data are stored is provided. Besides that, all data sources are shown as one integrated data source. Data virtualization is available in various implementation processes. Some of them include the following: a server for data virtualization, Enterprise Service Bus (ESB) architecture, placing data warehouse on the cloud, a virtual in-memory database and object-relational mappers.

We have concluded that all above phases of Big Data integration, which include data discovery, data preparation, model planning, model building, communicate results and operationalize, can be performed on data virtualization servers. This is not the case in other data virtualization implementation processes. Consequently, our approach to data virtualization implies the use of data virtualization servers. The main parts of a data virtualization server include source tables, mappings and virtual tables. Mappings represent the way to transform data from source tables to virtual tables. What makes virtualization servers powerful tools is the fact that source tables are not restricted to relational tables, but instead different data sources such as data generated by websites, the result of a web service call, a HTML page, a spreadsheet or a sequential file, can be used. Users can access virtual tables by using different APIs (Application Programming Interface), such as the JDBC/SQL interface, MDX (MultiDimensional eXpressions) and the SOAP-based interface. That means that same tables would be seen differently by different users.

According to (van der Lans 2012), a data virtualization server consists of a design module and a runtime module. When data consumers access the virtualization layer, they use the runtime module of a data virtualization server. The design module is an environment which data analysts and data model designers use to create concept definitions, data models, and specifications for transformation, cleansing and integration. Some data virtualization servers enable the creation of unbound virtual tables. That means that it is possible to create data models using them, and to join them with the real data source afterwards. The runtime module of a data virtualization server represents a virtual sandbox for data scientists and enables managed self-service reporting for business analysts.

At a time when new data sources appear on a daily basis, in order to ensure the understanding and integrity of data, it is very important to manage metadata. Metadata must be a link between the existing and new data sources. As Zdravković et al. (2015, 5) point out, 'the capability to interoperate will be considered as the capability to "semantically interoperate".' It is very important that data virtualization servers allow the entering and using of data models, glossaries and taxonomies.

The data virtualization integration approach can help in two ways in data integration processes enabling Big Data analytics. Firstly, data virtualization can help in the phases of data discovery and data preparation according to the requirements of different analytical models. Big Data analyses can include only external data or only internal historical data stored in an EDW, but they often require the integration of external and corporate data. Considering that we are talking about analyzing a huge amount of external data coming at a high speed, it makes no sense to consider the physical integration of data based on their copying into a single central data warehouse. Instead of that, Big Data analysis is performed on Big Data platforms and in NoSQL databases with

appropriate storage and processing performances. In that case, the required corporate data can be ensured on the data virtualization layer, according to the requirements of a specific Big Data analysis, and can then be exported to a Big Data platform, such as Hadoop. If data virtualization is conducted via a virtualization server, the required data are ensured by means of virtual tables. This means that no local copy of the selected data is made, but the data can instead be exported to different warehouses, in the form defined by a given virtual table. Data virtualization servers have built-in functions for the export of data to different warehouses, Big Data platforms included.

Secondly, the data virtualization integration approach can help in the phase of integration of the results of Big Data analytics and EIS. After becoming familiar with the available data sources, the operations of model planning and model building can be performed on a data virtualization server, similarly as in any database management system. Data models are designed by creating unbound virtual tables. Regrettably, at this point, not all data virtualization servers have this option. Once a virtual table is created, it can be linked with some external or internal data source. The design of virtual tables depends on the form of analysis results which should be integrated and the data model into which they should be integrated. We propose that the designing of virtual tables be based on the application of the schema alignment method and the available data virtualization concepts, such as nested virtual tables. Nested virtual tables are virtual tables created on top of other virtual tables. The schema alignment method and the way it is applied on a data virtualization server will be explained in detail in the next section.

## *Schema alignment based on schema on read and data virtualization*

Schema alignment is used when one domain includes several different source schemas, which describe it in different ways. The results of schema alignment include the following:

- a mediated schema, which provides a uniform view over heterogeneous data sources, covering the most important domain aspects;
- attribute matching, which matches attributes in all source schemas with the corresponding attributes in a mediated schema;
- schema mapping between each source schema and a mediated schema, specifying the semantic ties between the data described by source schemas and the data described by a mediated schema.

There are two classes of schema mappings: Global-as-View (GAV) and Local-as-View (LAV). GAV defines a mediated schema as a set of views over source schemas. LAV expressions describe source schemas as views over a mediated schema. We shall first define GAV and LAV schema mappings and then, by using these two formalisms, we shall give an example to show how the application of schema alignment method of data integration can be upgraded through the application of data virtualization concepts and the schema on read modeling approach. To demonstrate this, we have selected an example from the case study conducted to verify the proposed model. The case study is described in detail in the next section of the paper.

This is followed by the definitions of GAV and LAV schema mappings according to Doan, Halevy, and Ives (2012).

Definition 1 (GAV Schema Mappings). Let G be a mediated schema, and let $\bar{S} = \{S_1, \ldots, S_n\}$ be schemata of n data sources. A Global-as-View schema mapping $\bar{M}$ is a set of expressions of the form $G_i(\overline{X}) \supseteq Q(\overline{S})$, where

- $G_i$ is a relation in G,
- and appears in atmost one expression in $\overline{M}$, and $Q(\overline{S})$ is a query over the relations in $\overline{S}$

Definition 2 (LAV Schema Mappings). Let G be a mediated schema, and let $\bar{S} = \{S_1, \ldots, S_n\}$ be schemata of n data sources. A Local-as-View schema mapping $\bar{M}$ is a set of expressions of the form $S_i(\overline{X}) \supseteq Q_i(G)$, where

- $Q_i$ is a query over the mediated schema G, and
- $S_i$ a source relation and it appears in at most one expression in M.

The example: The backbone of the EIS architecture consists of an enterprise data warehouse (EDW), a data virtualization server and a business intelligence (BI) tool. This particular EIS is used by a road maintenance organization. We shall extract the relations modeling the road network, EDW.Road and EDW.Road_section, from the EDW schema. The problem in hand is to integrate new data sources, the Big Data analysis results and new reports to be created in the BI tool with the existing EIS. There are two new data sources: one stores the road traffic data, the other stores the data on automatic traffic counters monitoring traffic. The new reports should enable the visualization of Big Data analysis results over integrated data. The traffic data are stored in TXT files. In view of the fact that TXT files are semi-structured and that they contain a large amount of data that is constantly growing, they are warehoused on the Big Data platform HDFS (Hadoop Distributed File System). The following three tasks have been identified:

- Data on traffic counters, which are small in volume and do not change often, should be integrated with EIS on the data warehouse level.
- Data on traffic flow volume and structure, which will be the result of Big Data analysis, should be integrated with EIS on the corporate data model level.
- The new reports should be integrated with EIS on the corporate data model level.

What we are interested in are the Road and RoadSection relations, which belong to the EDW:
EDW.Road(RoadID, RoadName, RoadCategory),
EDW.RoadSection(SectionID, SectionName, RoadID, SectionLength).

The first task will be solved by adding a new relation to the EDW system and by linking it to the RoadSection relation. The new relation is Counter:

EDW.Counter(Location, Longitude, Latitude, SectionID, Type).

The second task requires a far more complex solution. The integration of a new data source with EIS on the corporate data model level will be performed through the successive multiple application of the schema alignment method. The results of the application of this method will be implemented on a data virtualization server, by creating virtual tables and nested virtual tables. We have adopted a top-down approach to this problem. This means that we first analyze the end goal to be achieved through integration. The end goal is a data schema as required by new reports. Since this data schema should be a common, uniform view over the EDW and the Big Data source, it will be designed as a mediated schema by using GAV schema mappings. Its relations will be nested virtual tables (NVT_Counter and NVT_AADT), created as views over virtual tables (VT_Road, VT_Section, VT_Counter, VT_Traffic). The virtual tables VT_Road, VT_Section, VT_Counter and VT_Traffic will be created as unbound virtual tables. Their role is very important. At this point, they will enable the application of GAV schema mappings and the creation of a virtual mediated schema. The following expressions describe the above GAV schema mappings:

Mediate.NVT_Counter(Location, Longitude, Latitude, RoadName, SectionName) ⊇
VT_Road(RoadID, RoadName),
VT_Section(SectionID, SectionName, RoadID),
VT_Counter(Location, Longitude, Latitude, SectionID).
Mediate.NVT_AADT(Location, Year, AADT, AADT_D1, AADT_D2) ⊇

VT_Traffic(Location, Year, AADT, AADT_D1, AADT_D2, AADT_A0, AADT_A1, AADT_A2, AADT_B1, AADT_B2, AADT_B3, AADT_B4, AADT_B5, AADT_C1, AADT_C2, AADT_X).

The AADT field represents Annual Average Daily Traffic, while AADT_D1 and AADT_D2 represent AADT by vehicle movement direction. The other fields represent AADT by vehicle categories.

In the next phase, by using LAV schema mappings, the unbound virtual tables VT_Road, VT_Section and VT_Counter are linked with the corresponding EDW relations. The EDW schema represents a mediated schema in this case. The following expressions describe the above LAV schema mappings:

VT_Road(RoadID, RoadName) ⊆
EDW.Road(RoadID, RoadName, RoadCategory)
VT_Section(SectionID, SectionName, RoadID) ⊆
EDW.RoadSection(SectionID, SectionName, RoadID, SectionLength)
VT_Counter(Location, Longitude, Latitude, SectionID) ⊆
EDW.Counter(Location, Longitude, Latitude, SectionID, Type)

Using LAV schema mappings, source schemas are created for the Big Data source (BD) based on VT_Traffic. The virtual table schema VT_Traffic represents a mediated schema in this case. The following expressions describe the above LAV schema mappings:

BD.AADT(Location, Year, AADT) ⊆VT_Traffic(Location, Year, AADT, AADT_D1, AADT_D2, AADT_A0, AADT_A1, AADT_A2, AADT_B1, AADT_B2, AADT_B3, AADT_B4, AADT_B5, AADT_C1, AADT_C2, AADT_X)

BD.AADTByDirections(Location, Year, AADT_D1, AADT_D2) ⊆VT_Traffic(Location, Year, AADT, AADT_D1, AADT_D2, AADT_A0, AADT_A1, AADT_A2, AADT_B1, AADT_B2, AADT_B3, AADT_B4, AADT_B5, AADT_C1, AADT_C2, AADT_X)

BD.AADTByCategories(Location, Year, AADT_A0, AADT_A1, AADT_A2, AADT_B1, AADT_B2, AADT_B3, AADT_B4, AADT_B5, AADT_C1, AADT_C2, AADT_X) ⊆ VT_Traffic(Location, Year, AADT, AADT_D1, AADT_D2, AADT_A0, AADT_A1, AADT_A2, AADT_B1, AADT_B2, AADT_B3, AADT_B4, AADT_B5, AADT_C1, AADT_C2, AADT_X)

Once schemas for the Big Data sources BD.AADT, BD.AADTByDirections and BD. AADTByCategories are designed, the designing of Big Data analysis begins so as to get the results described in the above schemas. This is when the schema on read modeling approach comes into play. It is applied to a Big Data source in situations when one knows what kind of data schema is required. In other words, the data on a Big Data platform are organized according to the schema derived through the successive application of GAV and LAV schema mappings. Once a Big Data source is created according to the above schemas, it should be linked with a data virtualization server. After that, the unbound virtual table VT_Traffic is linked with the real Big Data source. This solves the task of integrating the results of Big Data analysis with EIS on the corporate data model level.

The third task, integration of new reports with EIS on the corporate data model level, will be simply solved by linking the BI tool with the virtual schema Mediate on a data virtualization server.

We can say now that the key factors of the proposed model of Big Data integration include in the following:

- a top-down approach to solving the integration problem, i.e. starting with reports and moving down to data sources,
- application of GAV schema mappings in order to create a uniform view over the domain – a mediated schema, using the concept of unbound nested virtual tables on a data virtualization server,
- application of LAV schema mappings in order to create the required local and external data source schemas, using the concept of unbound nested virtual tables on a data virtualization server,

- application of the schema on read modeling approach in creating data schemas for Big Data sources, derived by using the above combined GLAV (Global-as-Local-as-View) schema mapping approach.

Although some authors, such as Dong and Srivastava (2015), believe that schema alignment is not an appropriate Big Data integration method, we have shown that it can be effectively implemented using unbound nested virtual tables and bound virtual tables on the data virtualization server.

### *Big data analytics integration scenarios*

Between the enterprise information system and the Big Data analytic tool a two-way data exchange is necessary. In Big Data analysis for business purposes, apart from data originating from external sources, such as sensor data, data generated by various machines, social networking data etc., corporative data are used, too. Corporative data that are used in Big Data analysis or are crossed with Big Data analysis results frequently appear on their own as a result of some predefined analysis in a business intelligence system. Thus, it is necessary to enable integration of corporative data and other data that are the object of Big Data analysis. One corporative data part, which is archived and traditionally used for business reporting, is structured. However, a significant part of corporative data are semi-structured and unstructured data.

On the other hand, external sources generate heterogeneous data that are stored in different types of data storages. The amount of external data that are of interest for corporative analysis as a rule increases. The results of Big Data analysis should become available to business analysts and other business users, and sometimes even end users, such as buyers, service users, etc. This can be achieved through data integration or through integration on the report level. Integration of corporate data, external data and Big Data is done in the phase of preparing input data for various advanced Big Data analysis techniques. After Big Data analysis is completed, it is necessary to integrate the results of the analysis with the corporate data. Big Data analysis scenarios can be different. Only the data analyzed on a Big Data platform can be analyzed without the use of corporate data. In this case, the only remaining task is to integrate the Big Data analysis results with EIS. The Big Data analytics integration framework we suggest allows us to integrate Big Data analysis and EIS on three levels: data warehouse level, corporate data model level and report level (Figure 3). The example described in the previous section demonstrates all three levels of integration, as shown in Figure 3. It has been mentioned earlier that all data integration phases can be conducted on the data virtualization server. Consequently, as seen in Figure 3, integration on the corporate data model level is performed directly between the Big Data platform and the data virtualization server, without the mediation of ETL tools.

Integration on the data warehouse level means that the Big Data platform is used to design schema on read which is identical to the one segment of the data warehouse model. Data from the Big Data platform can be obtained, transformed and loaded into data warehouse tables by using some ETL tool. It has been mentioned earlier that, among other things, the goal of the schema on read approach to modeling Big Data is to prepare Big Data so that ETL operations could be more efficient. As seen in Figure 3, the ETL tool is linked with one of the 'schemas on read' on the Big Data platform. In the case of data warehouse level of integration, the first four phases of the Big Data integration process from Figure 2: discovery, data preparation, model planning and model building are executed on the Big Data platform, or within ETL tools, and most often combined in both environments (Figure 3). The last two phases from Figure 2: communicate results and operationalize, are executed in the data warehouse (Figure 3). This kind of integration is suitable for batch-oriented Big Data analysis which is repeated periodically (monthly, quarterly, yearly) or on demand (Figure 4).

Integration on the corporate data model level can be carried out in two ways. The first method involves the prior preparation of the organization and storage of data on the Big Data platform and the creation of schemas on read according to the corporate data model. The difference between this method of integration and integration on the data warehouse level is that, in this way, the integration is done on the virtual level. The data virtualization server connects virtual tables derived from internal data sources and virtual tables generated from external – Big Data sources (schemas on read). In the case of integration on the corporate data model level, the first two phases of the Big Data integration process from Figure 2: discovery and data preparation are executed on the Big Data platform (Figure 3). The remaining four phases from Figure 2: model planning, model building, communicate results and operationalize, are realized on the data virtualization server (Figure 3). The second method involves the implementation of the schema on read modeling approach only within the design module of the data virtualization server. This means that by designing unbound virtual tables, a data model is created, which is subsequently associated with real data sources.

The key stages of the schema on read modeling approach are Explore Data and Develop Model (Figure 1). Both of these phases, according to our integration framework, can be performed on Big Data platforms over Big Data sources, but also on the data virtualization server, over integrated internal and external data sources. This is shown by schemas on reads in the form of a puzzle puzzle segment in Figure 3.

If we observe the three mentioned levels of integration, only integration on the corporate data model level enables all types of Big Data analysis applications: batch-oriented processing, stream processing, OLTP (Online Transaction Processing) and interactive ad-hoc queries and analysis (Figure 4).

Integration on the report level means creating schemas on read on Big Data platforms. These schemas are created with the aim of representing the data sources for the predefined reports and are designed so as to suit the reports' requirements. In the case of integration on the report level, the first four phases of the Big Data integration process from Figure 2: discovery, data preparation, model planning and model building are executed on the Big Data platform (Figure 3). The remaining two phases from Figure 2: communicate results and operationalize, are executed on the BI tool (Figure 3). This kind of integration is used for the following Big Data analysis applications: batch-oriented processing, stream processing and OLTP (Figure 4).

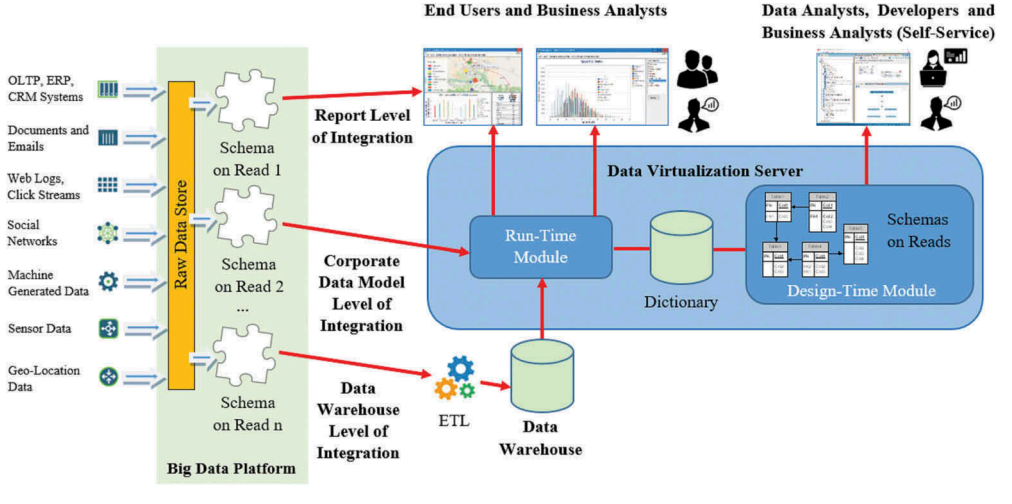


**Figure 3.** Proposed framework for Big Data analytics integration in EISs.

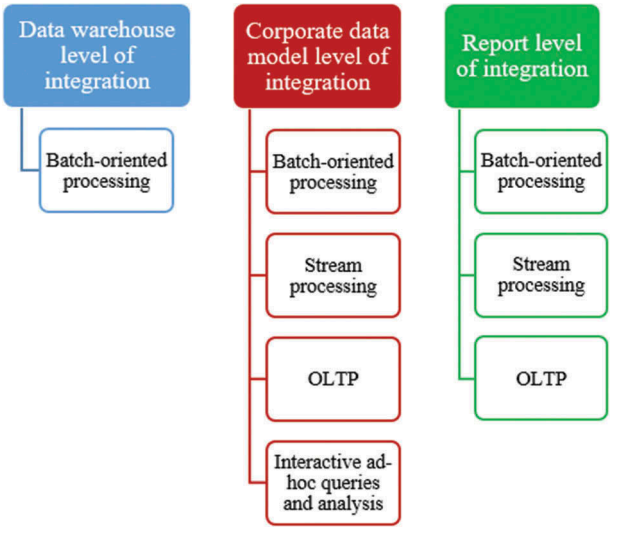


**Figure 4.** Levels of integration and Big Data analysis applications.

In existing batch data integration solutions, which are based only on the use of ETL tools, phases: discovery, data preparation, model planning and model building, include copying and temporary storage of large amounts of data in the data staging area. Our integration framework does not envision data staging area, because these Big Data integration phases are performed either on the Big Data platform, or at the virtual level on the server for data virtualization. If these four phases are implemented on the Big Data platform, our integration framework does not exclude the use of some existing solutions, such as Apache Hadoop components useful for ETL.

When it comes to real-time data integration scenarios, our integration framework does not exclude existing ESB-based solutions. On the contrary, our approach enables the development of a traditional ESB approach, by the implementation of the 'hub-and-spoke' design on the data virtualization server. As described in the previous section, data from all sources are transformed into a unified shared format called the mediated schema.

If we do not want to store permanently the data in a data warehouse, integration on the data warehouse level can be replaced with integration on the corporate data model level. Additionally, integration on the report level can be replaced with integration on the corporate data model level. The prerequisite for that is to imply data virtualization as an integration approach.

As the needs of business analysts and data analysts are becoming similar, the proposed approach enables the integration of reporting and analytical tools with enterprise data warehouse and external data sources. Depending on the categories of Big Data analytics use cases and the specific needs and skills of a particular user, the proposed framework enables the following integration scenarios:

(1) integration on the data warehouse level, for data analysts and developers;
(2) integration on the corporate data model level, for business analysts (self-service analysis), data analysts and developers;
(3) integration on the report level, for end users, business analysts, data analysts and developers;
(4) integration on the corporate data model level and data warehouse level, for data analysts and developers;

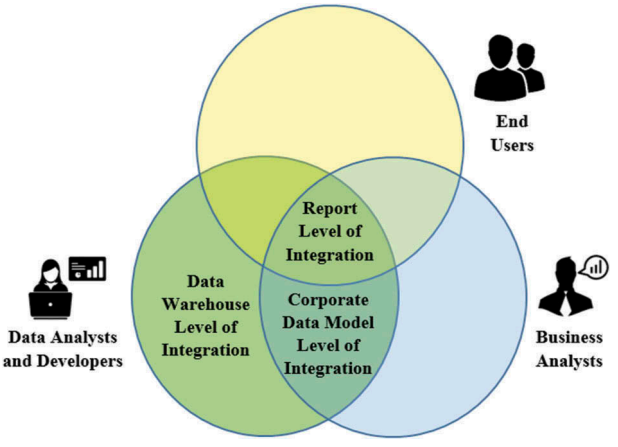


**Figure 5.** Levels of integration from Big Data analytics use cases point of view.

(5) integration on the corporate data model level and report level, for business analysts (self-service analysis), data analysts and developers;
(6) integration on the data warehouse level and report level, for data analysts and developers, and
(7) integration on the corporate data model level, the data warehouse level and the report level, for data analysts and developers.

The integration scenarios appropriate for particular user categories are presented in Figure 5.

## Implementation of integration framework in transportation domain

Traffic data are an excellent example of heterogeneous data that are continuously coming, making a demand for Big Data storage and analysis. Excellent tailor-made traffic data are the best basis for excellent transportation models (Janković et al. 2016a). We want to provide the traffic engineers and authorities with pre-attributed maps tailored to their specific needs. For the analysis of traffic flow, the traffic engineers calculate the indicators on an annual basis. For example, Annual Average Daily Traffic (AADT), along with its main characteristics of composition and time distribution (minutes, hourly, daily, monthly, yearly), is the basic and key input to the traffic-technical dimensioning of road infrastructure and road facilities. This parameter is used in capacity analysis, level of service analysis, cost benefit analysis, safety analysis, environmental assessment impact analysis of noise emission and air pollution, analyses of pavement construction, as well as for the static calculation of road infrastructure objects, traffic forecasting, etc.

To count the traffic at the specified locations on the state roads in the Republic of Serbia, 391 inductive loop detectors were used (Lipovac et al. 2015). These detectors are QLTC-10C automatic traffic counters (ATC). The case study included the analysis of traffic data in ten locations on the state roads and streets in the city of Novi Sad, Serbia, which the traffic counters generated during 2015. In order to have sensor data, it is necessary to link them to the traffic infrastructure data. As two different data categories exist, namely one that is continually generated and the other that is changed rarely, we recognized the need to process them differently. The traffic data that are continually generated in our case study are analyzed on the Big Data platform, while the data related to the traffic infrastructure are stored in the local relational database. Obviously, there is a need for their integration. In this study, we have integrated Big Data analytics with the existing EIS, first traditionally, without a data virtualization layer, then by using a data virtualization server.

Before developing the integration solution for our use case, we needed to go through the following phases:

(1) A relational data model was developed and the SQL server database STATE ROADS created. These enable storing the data on the state road reference system in the Republic of Serbia and the data on the automatic traffic counters used on these roads. The most important entities of the relational model are the following: road, road section, intersection, automatic traffic counter, etc.
(2) Each automatic traffic counter generated 365 text files in 2015. Each file contained about 10,000 records on average, so that the collected data amounted to 10 □ 365 □ 10,000 = 36,500,000 records.
(3) For the storage and processing of traffic data, the Apache Hadoop platform was chosen. Using the Apache Ambari user interface, on the Hortonworks Sandbox – a single-node Hadoop cluster, with the help of Apache Hive data warehouse software and HiveQL query language, a Hive database named TRAFFIC ANALYSIS was created.
(4) An ETL application was designed to 'clean up' the text files of any invalid records generated by traffic counters. Also, for each counter, this application consolidated the content of all 365 .txt files into a single text file which generated ten large .txt files. After that, we uploaded each of the ten large .txt files into the HDFS (Hadoop Distributed File System). White (2015) did useful work on HDFS. Using HiveQL query language we 'filled' Hive database tables with the data from the .txt files that are stored on HDFS.

### *Integration approach without data virtualization*

The traditional integration solution – without data virtualization – is presented in Figure 6. This integration solution was implemented in the following phases:

(5) We carried out numerous HiveQL queries on the Hadoop TRAFFIC ANALYSIS database resulting in useful information on traffic volumes, traffic structure, vehicle speeds, etc. (Janković et al. 2016b). HiveQL has a powerful technique known as Create Table As Select (CTAS). This type of HiveQL queries allow us to quickly derive Hive tables from other tables in order to build powerful schemas for Big Data analysis. This data modeling approach is known as schema on read. Schemas of Hive tables are designed so as to be joined to the relational model of the local SQL server database. This enables the integration of Big Data analytics with EIS on the corporate data model level. The query results include traffic volume and traffic safety indicators for each counting place: AADT, AADT by directions and vehicle categories, Monthly Average Daily Traffic (MADT), average speed of vehicles, $85^{th}$ percentile of vehicle speed, percentage of vehicles that exceed the speed limit, average speeding, etc.
(6) In the IDE Microsoft Visual Studio 2015, a Windows Forms geo-application called Traffic Counting was developed. It has the following features:
   - An intuitive GUI that allows the traffic engineers to define the query parameters and start executing the queries against Hive tables on the Hadoop database TRAFFIC ANALYSIS and tables from the local SQL server database STATE ROADS. This enables the integration of Big Data analytics with EIS on the report level. Access to the Hadoop database TRAFFIC ANALYSIS from the Windows Forms geo-application Traffic Counting was enabled with the help of Hortonworks ODBC Driver for Apache Hive.
   - A GUI for graphical and tabular visualization of query results and their geo-location. For the geo-location of query results in the Traffic Counting application, we used Bing Maps and OpenStreetMaps.

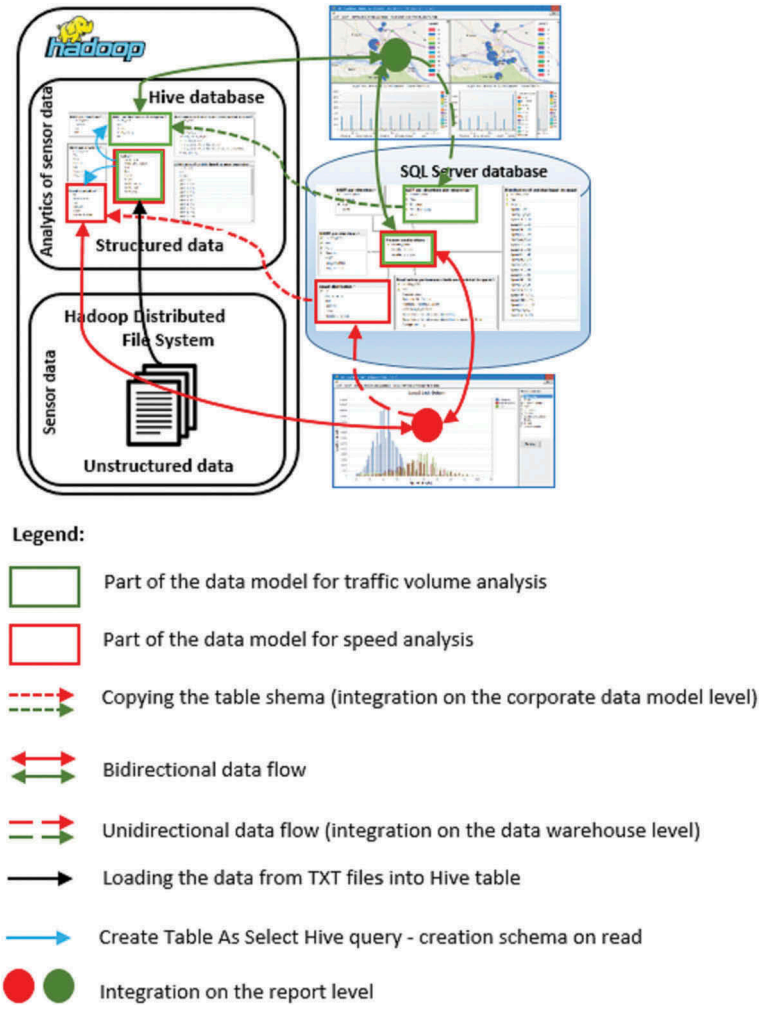


**Figure 6.** Big Data analytics integration solution without data virtualization.

(7) The results of the query of the Hadoop database TRAFFIC ANALYSIS were stored in the SQL Server database STATE ROADS with the help of Hortonworks ODBC Driver for Apache Hive and the Windows Forms geo-application Traffic Counting. This enabled the integration of Big Data analytics with EIS on the data warehouse level.

### *Integration approach based on data virtualization*

The architecture of the integration solution based on data virtualization is presented in Figure 7. This integration solution includes the first phase of the traditional integration approach, while its second and third phase differ from the traditional approach:

(2) A virtual data source was created on the Denodo Express 6.0 data virtualization platform, by virtualizing and integrating data from the local SQL Server database STATE ROADS and the Hadoop database TRAFFIC ANALYSIS (Figure 8). In this way, the data on volume, structure and speed of traffic flow that are generated on the Big Data platform are connected with the locally stored data on state roads in the Republic of Serbia. This enables the integration of Big

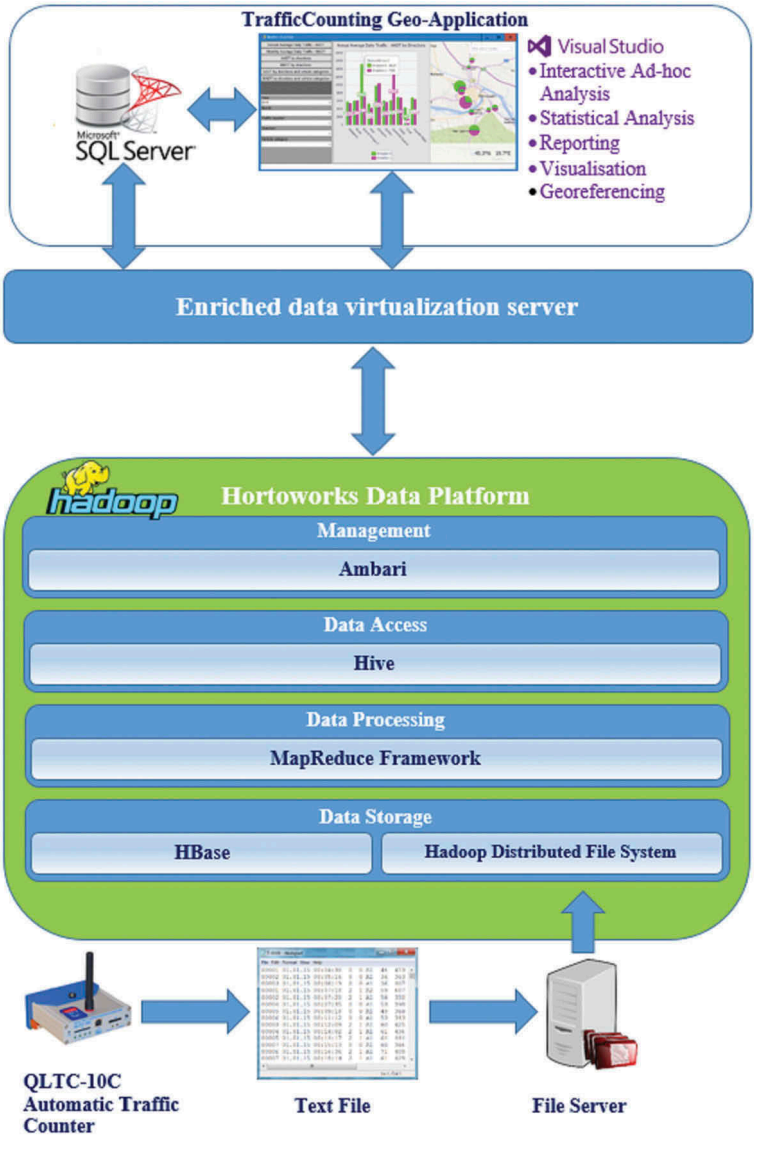


**Figure 7.** Architecture of the proposed Big Data analytics integration solution.

Data analytics with EIS on the corporate data model level. In Figure 9, a tree view and a relationship view of data schemas created by combining (merging) fields from the local and Big Data sources are shown. One should notice that the local data source is a relational table, and a non-relational table that does not even have the primary key. The query results based on which field combining from the mentioned heterogeneous data sources is performed are presented in Figure 10.

(3) The Traffic Counting geo-application, which was developed during the sixth phase of the traditional approach to integration, was linked to the unique virtual data source on the Denodo Express 6.0 data virtualization platform. In this way, the Traffic Counting geo-application uses the results of Big Data analysis from the Hadoop database TRAFFIC ANALYSIS and data from the local SQL Server database STATE ROADS, integrated on the

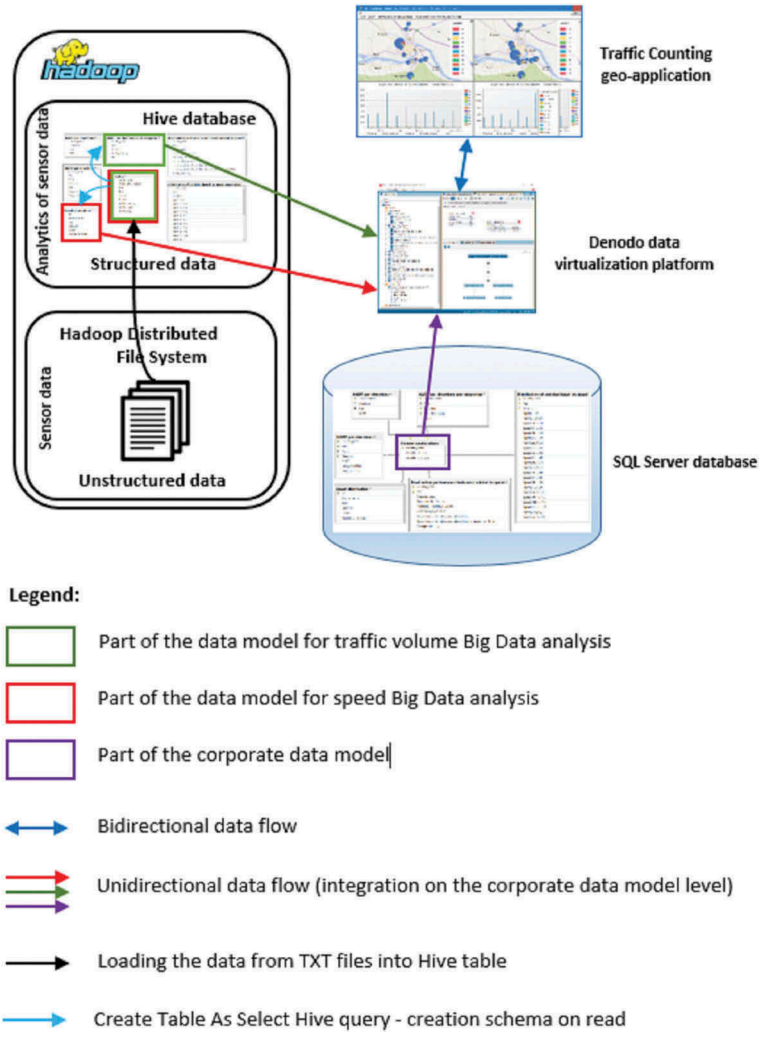


**Figure 8.** Big Data analytics integration solution based on data virtualization.

data virtualization platform. Figure 11 shows one window from the Traffic Counting geo-application that displays average speeding for each counting place. As seen in Figure 11, visualization is achieved on the tabular and graphical level and the maps.

## Conclusions

The continuous emergence of new data sources, data models, database management systems and data integration platforms, coupled with the pronounced need for the self-service analytics used by business analysts, makes it increasingly necessary to integrate Big Data analytics with traditional EISs on demand. IFAC TC5.3 Technical Committee for Enterprise Integration and Networking of the International Federation for Automatic Control has recognized the most serious challenges that must be solved in the Next Generation Enterprise Information System (NG EIS) development. The following have been selected as its key required features: omnipresence, a model-driven architecture and openness. 'In the ideal scenario, NG EIS will become a software shell, a core execution

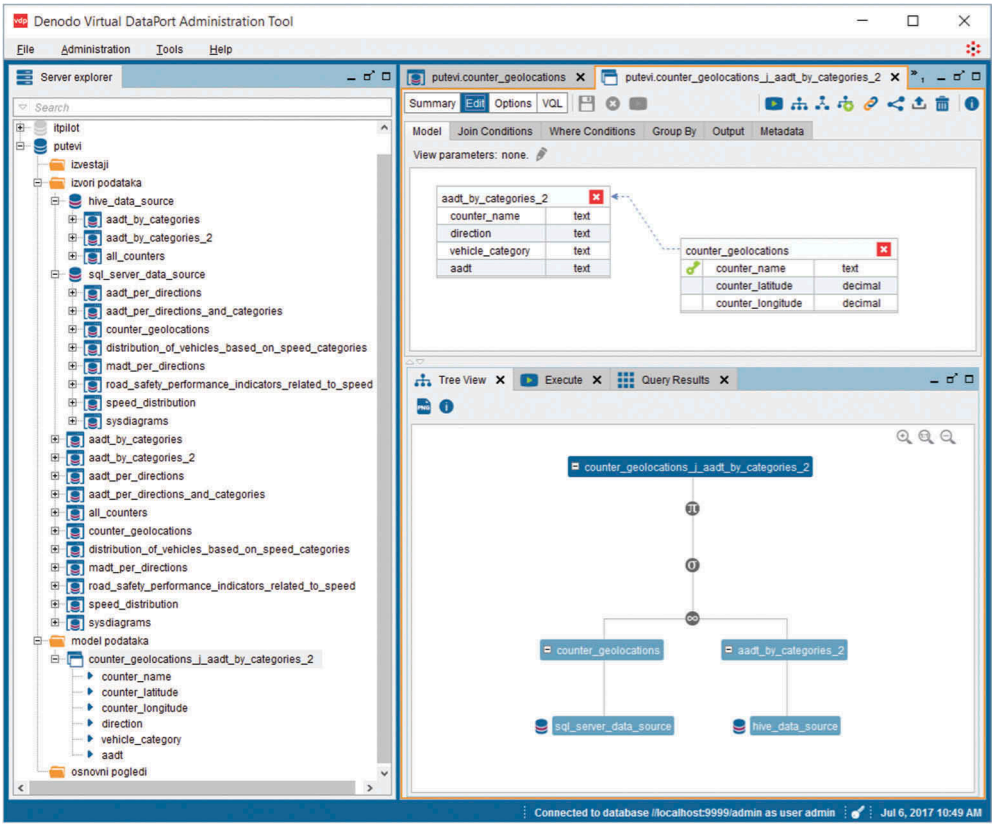


**Figure 9.** Tree view and relationships view of data source on data virtualization platform.

environment with the integrated interoperability infrastructure. Such an environment is foreseen as a highly flexible and scalable, deployable on any and every platform, using the external models and services infrastructure, exclusively or on a sharing basis.' (Zdravković and Trajanović 2015). Our approach to integration of Big Data analytics with EISs is based on a flexible appending of external models and their joining with the existing corporate data model on the virtual level. In this research, an approach that enables flexible integration from heterogeneous external sources and Big Data analytics with EISs is developed. The key drivers of our integration approach include flexibility, the reuse of raw/atomic data and querying multiple data stores and types at once.

The proposed Big Data analytics integration framework enables seven integration scenarios, which can include integration on the corporate data model level, on the data warehouse level and on the report level. Only integration on the corporate data model level enables all kinds of Big Data analysis applications, which include batch-oriented processing, stream processing, OLTP and interactive ad-hoc queries and analysis. Integration on the data warehouse level enables integration in Big Data analysis applications based on batch-oriented processing. Integration on the report level enables integration in Big Data analysis applications based on batch-oriented processing, stream processing and OLTP. All integration scenarios start with the designing of schemas for data analysis at the time of reading raw Big Data sources. Schemas on read are designed so as to be integrated into the existing relational corporate data models and/or the existing business reports, taking into account the structure of the source data files.

From the point of view of Big Data analytics use cases, integration scenarios can be divided into three categories. For end users, integration scenarios that include integration on the report level

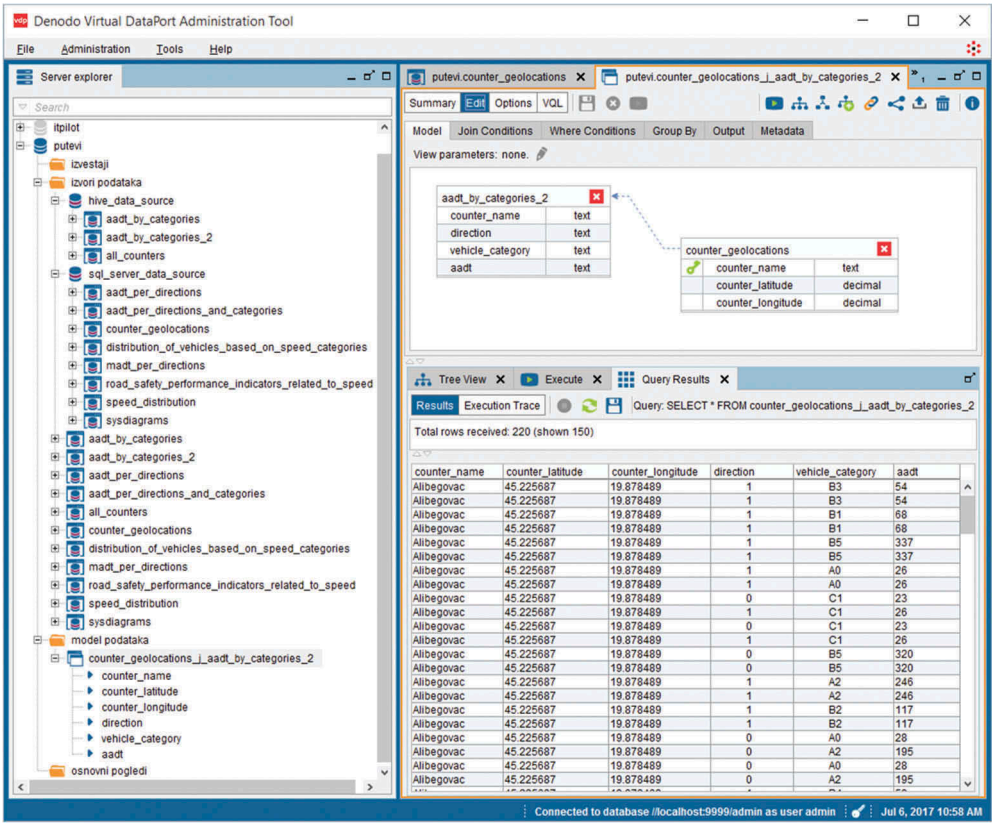


**Figure 10.** Query results view on data virtualization platform.

are appropriate. For business analysts and business reporting, integration scenarios that include integration on the corporate data model level and/or on the report level are appropriate. For data analysts, data discovery and developers, integration scenarios that include integration on all three levels, namely the corporate data model level, the data warehouse level and the report level, are appropriate.

The case study conducted has confirmed that the use of a data virtualization layer offers numerous advantages. These can be classified into three groups. The first group of advantages comes into play if the user accesses only one data source and it consists of the following: a data virtualization layer with the capability of language translation and API supported by a language data warehouse and API suitable for data users, independence from data source technologies (in the era of the IoT and Big Data, the possibility of exchanging a non-SQL data warehouse with a SQL warehouse is very important), and minimal negative user influence of data warehouse performance. The second group of advantages is connected to metadata specification, such as: a table structure, cleansing and transformation operations, aggregation and similar. When data virtualization is used metadata specification is implemented only once and it is not necessary to copy it for several data users. In other words, data users share and use metadata specifications on multiple occasions, with which they achieve more simple table structures, centralized data transformation, centralized data cleansing, simplified application development, more consistent application behavior and more consistent results. The third group refers to data integration from multiple data sources and includes the following: a unified approach to different types of data warehouses (SQL Server database, Excel worksheets, index sequential files, NoSQL databases, XML files, HTML web

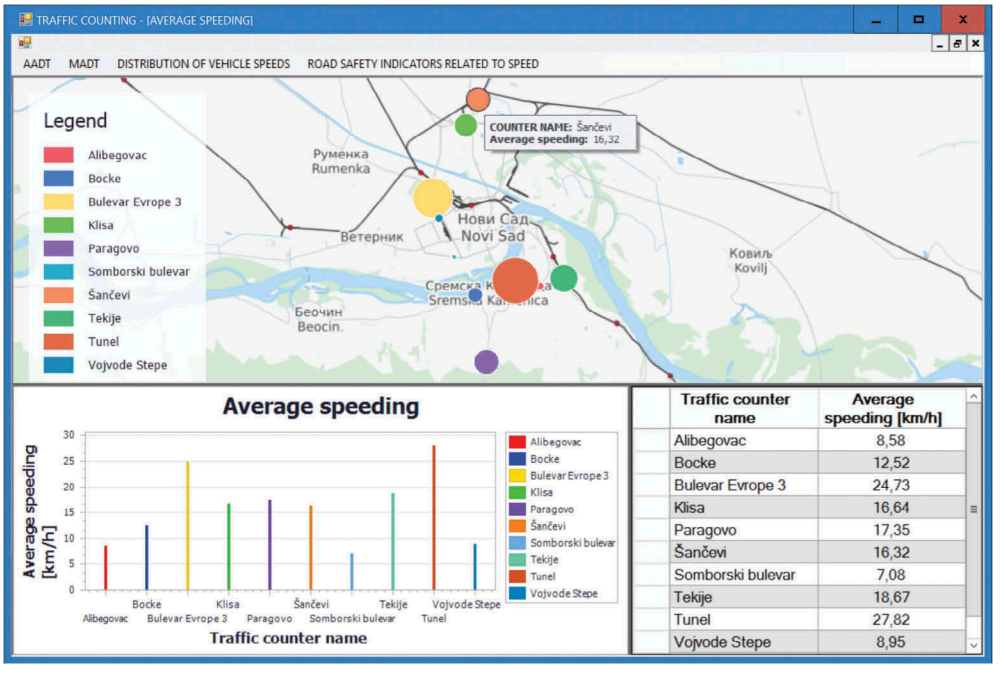


**Figure 11.** Traffic Counting geo-application – Average Speeding window.

pages, etc.), centralized data integration and sharing of integration programming code, consistent report results and efficient distributed data access.

In view of the positive experiences gained while using a data virtualization platform, the authors' future research will focus on the use of the above platform in the integration of Big Data analytics with NoSQL databases, such as column and key-value databases.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## Funding

This paper has been partially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia project under No. 36012, and the project under the name "Novel Decision Support tool for Evaluating Strategic Big Data investments in Transport and Intelligent Mobility Services – NOESIS". NOESIS project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769980. The data generated by automatic traffic counters have been provided by the company MHM - Project from Novi Sad.

## References

Alooma. 2018. "ETL Tools." January 4. https://www.etltools.net/

Arputhamary, B., and L. Arockiam. 2015. "A Review on Big Data Integration." *International Journal of Computer Applications Proceedings on International Conference on Advanced Computing and Communication Techniques for High Performance Applications* 5: 21–26.

Chen, C. L. P., and C. Y. Zhang. 2014. "Data-Intensive Applications, Challenges, Techniques and Technologies: A Survey on Big Data." *Information Sciences* 275: 314–347. doi:10.1016/j.ins.2014.01.015.

Doan, A., A. Halevy, and Z. Ives. 2012. *Principles of Data Integration*. Waltham: Morgan Kaufmann.

Dong, X. L., and D. Srivastava. 2015. *Big Data Integration (Synthesis Lectures on Data Management)*. Williston: Morgan & Claypool Publishers.

EMC Education Services, ed. 2015. *Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*. Indianapolis: John Wiley & Sons.

Florea, A. M. I., V. Diaconita, and R. Bologa. 2015. "Data Integration Approaches Using ETL." *Database Systems Journal (VI)* 3: 19–27.

Gal, A. 2011. *Uncertain Schema Matching*. (Synthesis Lectures on Data Management). Williston: Morgan & Claypool Publishers

Gandomi, A., and M. Haider. 2015. "Beyond the Hype: Big Data Concepts, Methods, and Analytics." *International Journal of Information Management* 35: 137–144. doi:10.1016/j.ijinfomgt.2014.10.007.

Gokhe, P. 2016. "Enterprise Real-Time Integration." E-book. http://www.enterpriserealtimeintegration.com/enterprise-real-time-integration/

Intel Corporation. 2013. "Extract, Transform, and Load Big Data with Apache Hadoop." White Paper Big Data Analytics. https://software.intelcom/sites/default/files/article/402274/etl-big-data-with-hadoop.pdf

Intel Corporation. 2014. "Real-Time Big Data Analytics for the Enterprise." White Paper Intel® Distribution for Apache Hadoop. https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/big-data-hadoop-real-time-analytics-for-the-enterprise-paper.pdf

Janković, S., D. Mladenović, S. Mladenović, S. Zdravković, and A. Uzelac. 2016a. "Big Data in Traffic." In *Proceedings of the First International Conference Transport for Today's Society – TTS 2016*, edited by M. M. Todorova, 28–37. Bitola, Macedonia: Faculty of Technical Science.

Janković, S., S. Zdravković, S. Mladenović, D. Mladenović, and A. Uzelac. 2016b. "The Use of Big Data Technology in the Analysis of Speed on Roads in the Republic of Serbia." In *Proceedings of the Third International Conference on Traffic and Transport Engineering - ICTTE Belgrade 2016*, edited by O. Čokorilo, 219–226. Belgrade: City Net Scientific Research Center.

Lipovac, K., M. Vujanić, T. Ivanišević, and M. Rosić. 2015. "Effects of Application of Automatic Traffic Counters in Control of Exceeding Speed Limits on State Roads of Republic of Serbia." In *Proceedings of the 10th Road Safety in Local Community International Conference*, edited by Proff. K. Lipovac and M. Nešić, 131–140. Belgrade: Academy of Criminalistic and Police Studies.

Loshin, D. 2013. *Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph*. Waltham: Elsevier.

Macura, M. 2014. "Integration of Data from Heterogeneous Sources Using ETL Technology." *Computer Science* 15 (2): 109–132. doi:10.7494/csci.2014.15.2.109.

Reeve, A. 2013. *Managing Data in Motion*. Waltham: Elsevier.

Ribeiro, A., A. Silva, and A. R. da Silva. 2015. "Data Modeling and Data Analytics: A Survey from A Big Data Perspective." *Journal of Software Engineering and Applications* 8: 617–634. doi:10.4236/jsea.2015.812058.

Russom, P. 2013. *Integrating Hadoop into Business Intelligence and Data Warehousing*. Renton, WA: Data Warehousing Institute.

Sridhar, P., and N. Dharmaji. 2013. "A Comparative Study on How Big Data Is Scaling Business Intelligence and Analytics." *International Journal of Enhanced Research in Science Technology & Engineering* 2 (8): 87–96. -izbaciti.

van der Lans, R. F. 2012. *Data Virtualization for Business Intelligence Systems*. Waltham: Elsevier.

Wang, H., Z. Xu, H. Fujita, and S. Liu. 2016. "Towards Felicitous Decision Making: An Overview on Challenges and Trends of Big Data Technologies." *Information Sciences* 367–368: 747–765. doi:10.1016/j.ins.2016.07.007.

White, T. 2015. *Hadoop: The Definitive Guide*. Sebastopol, CA: O'Reilly Media.

Zdravković, M., F. Luis-Ferreira, R. Jardim-Goncalves, and M. Trajanović. 2015. "On the Formal Definition of the Systems' Interoperability Capability: An Anthropomorphic Approach." *Enterprise Information Systems* 11 (3): 389–413. doi:10.1080/17517575.2015.1057236.

Zdravković, M., and H. Panetto. 2017. "The Challenges of Model-Based Systems Engineering for the Next Generation Enterprise Information Systems." *Information Systems and e-Business Management* 15 (2): 225–227. doi:10.1007/s10257-017-0353-z.

Zdravković, M., and M. Trajanović. 2015. "On the Runtime Models for Complex, Distributed and Aware Systems" In *Proceedings of the 5th International Conference on Information Society and Technology – ICIST 2015*, edited by M. Zdravković, M. Trajanović, and Z. Konjović, 236–240. Kopaonik, Serbia: Society for Information Systems and Computer Networks.