Sensors & Transducers, Vol. 195, Issue 12, December 2015, pp. 69-76

Sensors & Transducers

http://www.sensorsportal.com

Big Data Information Technology and Data Space Architecture

[1, 2] Natalya SHAKHOVSKA, [2] Oleh VERES, [2] Yuri BOLUBASH and [3] Liliana BYCHKOVSKA

[1] The University of Economy in Bydgoszcz, Faculty of Applied Studies, Poland,
[2] Lviv Polytechnic National University, Institute of Computer Science and Information Technologies, S. Bandery Str., 12, Lviv, 79013, Ukraine
[3] University of Computer Sciences and Skills, Lodz, Poland
[1] Tel.: +38 (032) 258-23-91, fax: +38 (032) 2582663
E-mail: natalya.shakhovska@byd.pl

Received: 6 October 2015 /Accepted: 30 November 2015 /Published: 30 December 2015

Abstract: The article dwells upon the basic characteristic properties of Big data technologies. There are proposes and describes the elements of the generic formal big data model. It is analyzed the peculiarities of the application of the proposed model components. Big Data as Data Space vector is described. There is constructed Data Space architecture. There is specific debug operation Data Space as a complex system. The query translator in Backus/Naur Form is given. *Copyright © 2015 IFSA Publishing, S. L.*

Keywords: Architecture, Data, Big Data, Data Space, Information technology, Information products, Model.

1. Introduction

The main problems that arise in the data processing is the lack of analytical methods suitable for use because of their diverse nature the need for human resources to support the process of data analysis, high computational complexity of existing algorithms for analysis and rapid growth of data collected. They lead to a permanent increase in analysis time, even with regular updating of hardware servers and also arise need to work with distributed database capabilities which most of the existing methods of data analysis is not used effectively. Thus, the challenge is the development of effective data analysis methods that can be applied to distributed databases in different domains. It is therefore advisable to develop methods and tools for data consolidation and use them for analysis.

Big data is the term increasingly used to describe the process of applying serious computing power – the latest in machine learning and artificial intelligence – to seriously massive and often highly complex sets of information (cited from 4/2013 the Microsoft Enterprise Insight Blog).

Big Data information technology is the set of methods and means of processing different types of structured and unstructured dynamic large amounts of data for their analysis and use of decision support [7-11].

There is an alternative to traditional database management systems and solutions class Business Intelligence. This class attribute of parallel data processing (NoSQL, algorithms MapReduce, Hadoop) [1-2, 15-16]. Defining characteristic of Big Data is the amount (*V*olume, in terms of value of the physical volume), speed (*V*elocity in terms of both

growth rate and needs high-speed processing and the results), diversity (*V*ariety, in terms of the possibility of simultaneous processing of different types of structured and semi-structured data) [12-14].

Variety (Complexity) defines as using:

- Relational Data (Tables/Transaction/ Legacy Data);
- Text Data (Web);
- Semi-structured Data (XML);
- Graph Data (Social Network, Semantic Web, RDF);
- Streaming Data;
- A single application can be generating/collecting many types of data;
- Big Public Data (online, weather, finance, etc).

There such types of *V*alue in Big Data as statistical, events, correlations ect.;

- Velocity (Speed) of Big Data,
- Data is begin generated fast and need to be processed fast;
- Online Data Analytics;
- Late decisions regard with missing opportunities.

The fifth V in Big Data is Veracity, because we have uncertainty due to data inconsistency and incompleteness, ambiguities, latency of data.

Bill Inmon considers the concept of "big data" as a new information technology [17]. Big Data is a technology that has the following features:

- It is possible to process a very large volume of data;
- The data media are inexpensive;
- Data are managed by the "Roman Census" method;
- Data managed with the Big Data are unstructured.

It is hard to find the industry to which the problem of Big Data would be irrelevant. The ability to handle the large volumes of information, analyze the relationships between them and to make informed decisions, on the one hand, carries the potential for the companies from different verticals to increase the attractiveness and profitability, efficiency. On the other hand, this is a great opportunity for the additional income to the vendor partners, i.e. the integrators and consultants.

2. Generalized Formal Model of the Big Data Technologies

Generalizing the listed definitions of the "big data" term as an information technology, it is possible to develop a formal model in the form of a quartet:

$$BD = \langle Vol_{\text{BD}},\ Ip,\ A_{\text{BD}}, T_{\text{BD}} \rangle, \quad (1)$$

where Vol_{BD} is the set of volume types; Ip is the set of types of data sources (information products); A_{BD} is the set of techniques of Big Data analysis; T_{BD} is the set of Big Data processing technologies.

Hinchcliff divides the approaches to the Big Data into three groups depending on the volume:

$$Vol_{\text{BD}} = \{Vol_{\text{FD}},\ Vol_{\text{BA}},\ Vol_{\text{DI}}\}, \quad (2)$$

where Vol_{FD} is the Fast Data: their volume is measured in terabytes; Vol_{BA} is the Big Analytics; they are petabyte data; Vol_{DI} is the Deep Insight; it is measured in exabytes, zettabytes.

Groups differ among themselves not only in the operated volumes of data, but also in the quality of their processing solutions.

Fast Data processing does not provide new knowledge, its results are in line with the prior knowledge and enable to assess the process executions, to see better and in more detail what is happening, to confirm or reject some hypotheses. The velocity used for this technology must grow simultaneously with the growth of data volume.

Tasks solved by means of Big Analytics are used to convert the data recorded in the information into new knowledge. The system is based on the principle of the "supervised learning".

Deep Insight provides an unsupervised learning and the use of modern analytic methods and various visualization techniques. At this level the knowledge and mechanism elicitation is possible. They are a priori unknown.

Processing information from different expressive power types of information sources, namely structured, semistructured, and unstructured is necessary for the Big Data technology. A set of information products is divided into three blocks:

$$Ip = \langle St, SemS, UnS \rangle, \quad (3)$$

where $St = \langle\ DB,\ DW\ \rangle$ is the structured data (databases, warehouses); $SemS = \langle\ Wb,\ Tb\ \rangle$ is the semi-structured data (XML, electronic worksheets); $UnS = \langle\ Nd\ \rangle$ is the unstructured data (text).

There are operations and predicates on this vector and its separate elements that provide the transformation of various vector elements into each other; combining elements of the same type; search for items by the keyword.

Computer programs are becoming closer to the real world in all its diversity, hence the growth in the volume of input data and hence the need for their analytics, moreover, in the mode maximally close to the real time. The convergence of these two trends has led to the emergence of an approach of Big Data Analytics.

Today, there is $A_{\text{BD}} = \{A_{\text{i}}\}$ set of different methods for the data set analysis, which are based on the tools borrowed from statistics and informatics (e.g. machine learning). The list is not exhaustive, but it reflects the most demanded approaches in various industries. Of course, the larger volume and

diversified array are subject to analysis, the more accurate and relevant data are obtained at the output.

Methods and techniques of the analysis applied to the Big Data are identified in the McKinsey report [18, pp. 27-31].

On the phases of the Big Data processing the following technologies are used:

$$T_{\text{BD}} = \left\langle T_{\text{NoSQL}}, T_{\text{SQL}}, T_{\text{Hadoop}}, T_{\text{V}} \right\rangle, \quad (4)$$

where T_{NoSQL} is the technology of NoSQL databases; T_{Hadoop} is the technology that ensures the massively-parallel processing; T_{SQL} is the technology of the structured data processing (SQL database); T_{V} is the technology of the Big Data visualization.

Association's models between entities and characteristics for various categories Nosql database T_{NoSQL} consist of:

- Key-value model;
- Bigtable data model;
- The object-document model;
- The graph data model;
- The XML document-oriented model;
- RDF-graph model.

The data model "key-value" (another name is column DB) described as:

$$KV = \{< f, e >\}, \quad (5)$$

where f is the key that accepts a unique value in each pair, e is the value that corresponds to this key. The keys can be complex and support virtually unlimited value semantics.

Signature of this model looks like:

$$O = \langle \pi, \sigma \rangle, \quad (6)$$

where π is the attributes projection operation (key or value), σ is the operation of attributes selection. These operations are classified as reading operations.

Examples of real reads:

- Get (key)
- MultiGet
  - MultiGetIterator, StoreIterator (major key)
  - Subrange (keyFirst, keyLast)
  - Depth.CHILDREN_ONLY
- MultiGetKeys
  - Subrange (keyFirst, keyLast)
  - Depth.CHILDREN_ONLY

An example of column type database is Cassandra.

BigTable from Google was designed for version distributed storage of large amounts of data with following characteristics:

- Not full relational data model;
- It supports dynamic control over data placement.

The base Bigtable data model is simple: rows, columns and timestamps.

$$BigTable = \{< r, c, t >\} \quad (7)$$

A search engine can be used as names of rows from the Internet addresses of documents and column names (e.g. the content of a document can be stored in the column «content:», and links to subsidiary pages are saved in the pages of «anchor:»). Another example is Google maps, which consist of a billion images, each of which details particular geographical areas of the planet. The maps in Bigtable Google are structured as follows:

- Each line corresponds to one geographical segment;
- Columns are images consists of segments;
- Different columns contain images with different detail.

If several columns stored data of same type, these columns form a family in Bigtable model. We can use family column conveniently least to compress homogeneous data, thereby reducing the volume. This column is a family unit of data access.

Rows of Bigtable (the maximum length can be up to 64 kilobytes) are also important. Access operation to the row is atomic (it means that while one program refers to the row, no other has no right to change the data in the family column for that row).

The contents of the pages on the Internet are constantly changing. To accommodate these changes each copy of the data stored in the column is assigned a temporary mark (timestamp). The timestamp in Bigtable is a 64-bit number which can encode the time and date, as required by the client programs. For example, the timestamp for copies of Web pages in the column «contents:» is the date and time of the creation of these copies. Using temporary tags, applications can specify in Bigtable search, for example, only the most recent copy of the data.

So, for the subject area of any Google service, you can create your own data map in Bigtable, which contains a number of rows, and it is unique to this domain set of families. The inevitable duplicates the data in the columns are arranged for a temporary tag. All of this points to a complete lack of support of the ACID properties.

But the main advantage of this approach is that such a database is easy to cut into independent pieces and distribute a set of servers. The rows are sorted in alphabetical order and are divided into ranges.

The object-document model is described by tuple:

$$\begin{aligned} OD = \{< f_0, < f_1 : e_1, f_2 : e_2, f_n : e_n, \\ f_{n+1} : d_1, f_2 : d_2, f_{n+l} : d_l >\}, \end{aligned} \quad (8)$$

where f_0 is the document identification, $f_1..f_m$ are the document characteristics, $e_1..e_m$ are the atomic values of characteristics $f_1..f_m$, $d_1..d_l$ are the links to another documents, $d_i = e(f_i)$.

We use object operation in this model.

Operation of determining element nodes is described as

$$v(f_i) = \{C\} \cup \{od_i \mid i = \overline{1,n}\} \cup \cup \{e(f_i) \mid i = \overline{0,n+l}\}, \quad (9)$$

where C is the collection of documents od_i.

Operation of determining element node's values is described as $v(f_i) = \{e_{ij} \mid i = \overline{1,n}, j = \overline{0,m+l}\}$, where e_{ij} is value of attribute f_i.

Let us describe relations over operands.

The relations of "element-element" determined between documents and collection:

$$OD \times C \rightarrow EE \quad (10)$$

The relations of "element-attribute" is given as:

$$f_i \times OD \rightarrow EA \quad (11)$$

The relations of "element-link" is given as:

$$f_i \times d_j \rightarrow ER \quad (12)$$

The relations of "element-data" is given as:

$$f_i \times e_j \rightarrow ED \quad (13)$$

Examples of this type of database is MongoDB and CouchDB.

The graph data model is given as:

$$O = \langle ID, A, z, r \rangle, \quad (14)$$

where ID is the set of identifications and nodes; A is the set of directed marked arcs (p, l, c), $p, c \in ID$, l is the «row-stamp», the record (p, 1, c) means, that we have relation l between arcs p and; z is the function for each arc's $n \in ID$ transforming in atomic or complex value, $z : n \rightarrow v$; V is the node arc.

The structure of the XML-document consisting of nested tag elements is well known. It's differs from the graph model is mainly in the interpretation of tags and labels:

- The label in graphs used to designate relations between circuit data elements and tags are not need to refer to the element;
- The XML document-oriented model required that each (non-text) data element was identifying sign.

Also XML data is translated into structure "tree", and this is a partial case of graph models.

The semistructured data in the graph models for XML types are described by attributes ID, IDREF, IDREFS. These types allow you to organize storage cross-references in XML-type elements <e_{id}, val_{ie}> (<ID element, value>) and attributes <*label*, e_{id}> (<label, value>).

There are several types of data in RDF-graph models: RDF / XML, N3, Turtle, RDF / JSON.

The resource description of RDF data is a triple "subject" - "predicate" - "object". We must define the set to U (Universal Resource Identifier, URI, unified resource identifier), elements f, set B {Black nodes}, set L {Literal} (RDF-literals) $B \in e, L \in e$ and the set (f, e (f), e), where f is subject, e (f) is predicate and e is object.

SN-architecture (Shared Nothing Architecture) is often stated as the basic principle of the Big Data processing that provides massively-parallel, scalable processing without degradation of hundreds or thousands of processing nodes [18, pp. 31-33]. Thus, except for the technologies such as *NoSQL*, *MapReduce*, *Hadoop*, *R* considered by most analysts, McKinsey also considers Business Intelligence technologies and SQL supported relational database management system in terms of Big Data applicability.

NoSQL (Not only SQL) in computer science is a number of approaches aimed at implementing database warehouses which differ from the models used in the traditional relational DBMS with access to the data by SQL means. It applies to the databases in which an attempt is made to solve the problems of scalability and availability at the expense of data atomicity and consistency.

MapReduce is an engine of parallel processing [16]. *MapReduce* is a model of distributed computing introduced by Google, which is used for parallel computing on very large data sets in computer clusters. *MapReduce* is a computing framework for the distributed task sets using a large number of computers that make up the cluster.

Hadoop is a project of the Apache Software Foundation, the freely available set of utilities, libraries and frameworks for the development and execution of the distributed programs running on the clusters of hundreds or thousand nodes. It is developed in Java within the computing *MapReduce* paradigm according to which the application is divided into a large number of identical elementary tasks that are feasible on the cluster nodes and naturally given in the final result. *Hadoop* is a capability set with open source code. *Hadoop* handles the large data caches breaking them into smaller, more accessible to applications and distribution across multiple servers for the analysis. *Hadoop* processes requests and generates the requested results in significantly less time than the old school of software analytics; it often takes minutes instead of hours or days.

R is a programming language for the statistical data processing and graphics as well as the free software computing environment with open source code in the GNU project. R supports a wide range of statistical and numerical methods and has good extensibility via packages. Packages are libraries for specific functions or special applications.

On the one hand, because of its heterogeneity and continuous growth, Big Data demand non-standard approaches in storage and processing. To work effectively it is required integrated solutions for monitoring, filtering, structuring and searching of hierarchical relationships. On the other hand, using Big Data, you can watch the huge set of variables and based on the provided information identify global trends and conclusions, considering the specific situation in the future.

Thus, the specificity of Big data (the presence of diverse set of sources, data doubling, ambiguity describing data sources) leads to the fact that the indeterminacy in traditional relational databases considered within a relationship and could occur at the level of attribute and tuple-level attitude in this case extends through the perception of the user information on the entire data space. Therefore, for processing indeterminacy in the Big data must use a different approach, the need for the use of which has not had in relational databases and data warehouses

Data Space is a block vector comprising a plurality of information products subject. It consists of the set of information products of subject area [1]. Above this vector and its individual elements defined operations and predicates that provide:

- Converting various elements of the vector at each other;
- Association of one type;
- Search items by keyword.

This thesis describe Big Data in Data Space architecture.

3. Data Space Architecture

Data Space is a block vector, containing of [19-20]:

- Structured *St*;
- Semi-structured *SemS*;
- Unstructured *UnS* information products;
- Operation over blocks, and operations over block vectors ΩP;
- And the set of predicates Ω:

$$DS = \langle \boldsymbol{Ip}, \Omega_{\mathrm{P}}, \Omega_{\mathrm{F}} \rangle \quad (15)$$

Above this vector and its individual elements there are defined operations and predicates that provide:

- Conversion of various elements of the vector into each other;
- The union of one type;
- Search items by keyword.
- Data models supported in Data Space will form a hierarchy according to their expressive power:
- Relational;
- Multidimensional;
- Object-relational model;
- Object-oriented model;
- Extended markup language data (XML) with the scheme;
- Description of the environmental resources (RDF);
- A standard means of describing relationships between data objects – ontology described using Web Ontology Language OWL;
- Structured text (including HTML);
- Semi-structured text.

Null-predicate Ω_{F0} returns TRUE, if for the given information product Ip its data structures are known, and FALSE otherwise.

Comparison Predicate (relation) of the information Resources returns TRUE, if data structure of both information products are same.

Binary operations over block vectors are advanced set-theoretic union and intersection operations.

The set of unary operations on data space return the result as the change of the data space state.

Data Space architecture consists of several levels, such as data level, manage the level and the metadescribe level (Fig. 1).

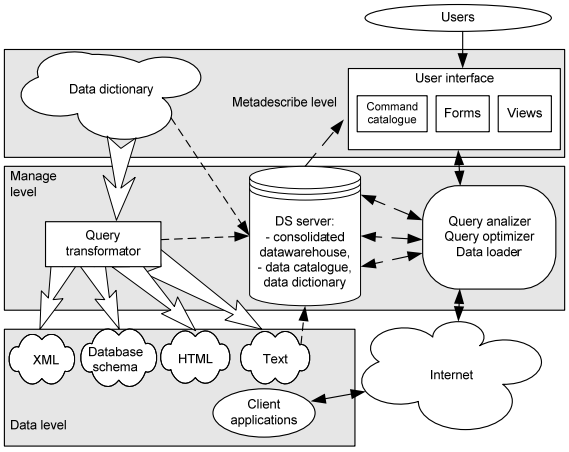

Fig. 1. Data Space architecture.

The module structure of Data Space is described in Fig. 2.

Data level consists of information products of Data Space. Data level in Fig. 2 described as a cloud.

Manage level consists of modules for Data Space organization and manage [1-2]:

- Module for user permission determination (by user authorized procedure);
- Query transformation module (by interpretation method);
- Module for working with metadata (by find's operation as query to metadata);
- Sources access by type module (by standard data exchange protocols usage);

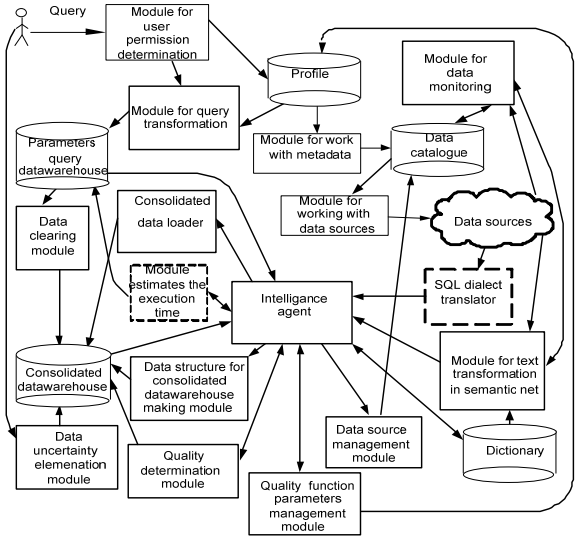

Fig. 2. Data Space structure.

- Module for text transformation in semantic net (by the semi-structured data analysis method);
- Intelligent Agent (based on the formal description of intelligent agent determine the structure of the data source, the algorithm of the intelligent agent);
- Data structure for consolidated Data Warehouse making module (based on the method of construction of consolidated data repository schemes and work smart agent determine the structure of the data source);
- Consolidated data loader (based operations consolidation, data consolidation method);
- Module purification data (based on advanced operators cut, coagulation operator, method of forming a system of norms and criteria, method of analysis, filtering and converting input data);
- Data uncertainty elimination module (based on the method of application of classification rules and modified operator eliminate uncertainty in the network structure of the consolidated data. The method of construction schemes consolidated data repository and work smart agent determine the structure of the data source);
- Quality determination module;
- Quality function parameters management module (based methods control elements data space based on the function of the quality and levels of trust);
- Data source management module;
- Module for data monitoring;
- Module estimates the execution time (based on the standard of fixing runtime) SQL dialect translator (by the SQL description).

Level control models of the platform are maintenance Data Space.

The meta descriptions level containing all the basic information about the data sources and methods to access them. Also, there are defining methods of data processing: for structured sources such operations as selection, grouping, etc., for semi-structured and unstructured - definition of a structure or search by keyword.

In addition, the Data Space also provides data storage for storing user profiles and temporary storage request parameters. It is very important for Big Data requirements.

4. Technological Aspects of Big Data Realization

Cloud computing is the technology of data processing, where software is provided as a user of Internet services. The user has access to data, but cannot control and should not care about the infrastructure, operating system and proprietary software, with which it works. Cloud computing defined new features, capabilities, operational/usage models and actually provided a guidance for the new technology development.

Several options may form the provision for the use of computing power and databases datacenter: Saa, Paa, Haa, Iaa, Caa - as Internet services. S, P, H, I, C – Software, Platform, Hardware, Infrastructure, Communication – respectively, the software, platform, hardware, infrastructure, communications. Platform paradigm of Cloud Computing consists of components: virtualization, SAAS, SOA (Service-Oriented Architecture) and SS (Software Services).

Cloud technologies allow for infrastructure virtualisation and its profiling for specific data structures or to support specific scientific workflows.

Reference Model for Big Data is presented on Fig. 3.

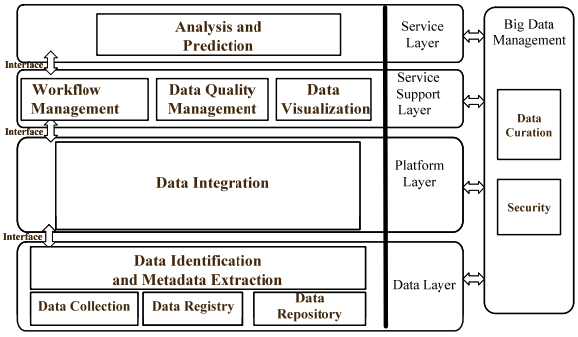

Fig. 3. Reference Model for Big Data.

5. Language Elements Description

For translator building, we must describe elements of the query language in Data Space. We used Backus/Naur Form (BNF).

```
<letter>::=a|b|c|d|e|f|g|h|i|j|k|l|n|m|o|p|q
|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|N
|M|O|P|Q|R|S|T|U|V|W|X|Y|Z
```

```
<keyword>  ::=  (<keyword>)  |<letter>  |  <
keyword>
<number>  ::= 0|1|2|3|4|5|6|7|8|9
<object>  ::= <data catalogue element>
<par>      ::= <the synonym of data catalogue
element >
<param>                                          ::=
<keyword>[{<keyword>|<number>}]
<num>     ::= <number>[{<number>}]
<expr>    ::= <operand> [{<op> <operand>}]
<operand> ::=» («<expr>»)» | <num> | <param>
[«[«<expr>»]»]
<op>       ::= <grteq>
<inv>      ::= <logicalop> | «*» | «/»
<type>     ::= «SUM» | «COUNT» | «AVG»
<logicalop>::= «<» | «>» | «>=» | «<=» | «=»
| «<>» | [<op>]
<whereop>  ::=  «where»  «(»  <object>  [«:»
<par>] {«,»<object> [«:» <par>] }«)»
<whoop>   ::= «who» «(» <object> [«:» <par>]
{«,»<object> [«:» <par>] } «)»
<howop>   ::= «how» «(» <object> [«:» <par>]
{«,»<object> [«:» <par>] } «)»
<Seop>       ::=  «Se»  «(»<object>[«:»<par>]
[«Agg»<type>]  {«,»  <object>  [«:»  <par>]
[«Agg» <type>] }«)»
<whatop>    ::=  «what»  «(»  <object>  [«:»
<par>] [«,»<object> [«:» <par>] ]«)»
<whichop>  ::=  «which»  «(»  <object>  [«:»
<par>] [«,»<object> [«:» <par>] ]«)»
<Semantop>::=  «Semant»  «(»  <object>  [«,»
<object> ]«)»
<Consop>    ::=  «Cons»  «(»<object>  [«:»  {
<par> <operator> <param>}] «)»
<profileop>  ::=  «where»  «(»  <object>  [«:»
<num>] {«,»<object> [«:» <num>] }«)»
<Unionop>    ::=  «Union»  «(»  <object>  [«:»
<par>]  {«,»<object> [«:» <par>] } «)»
<Unionop>    ::=  «Union»  «(»  <object>  [«:»
<par>]  {«,»<object> [«:» <par>] } «)»
<Intersop>   ::= «Inters» «(» <object> [«:»
<par>]  {«,»<object> [«:» <par>] } «)»
< Differop> ::= «Differ» «(» <object> [«:»
<par>]  {«,»<object> [«:» <par>] } «)»
```

For developing the portal as an architectural pattern there is used pattern Model-View-Controller (MVC).

Model-View-Controller (MVC) is architectural pattern, which is used in the design and development of software. It is used to separate data (model) from the user interface (view) so that the user interface changes minimally affect the operation of the data, and changes in the data model could be conducted without changing the user interface. The purpose of the template is flexible design software, which should facilitate further changes or expansion programs, and provide an opportunity for reuse of individual components of the program.

The architectural pattern MVC divides the program into three parts. In the triad of responsibilities Component Model (Model) is a data storage and software interface to them. View (View) is responsible for the presentation of these data to the user. Controller (Controller) manages components, receiving signals as a response to user actions, and reporting changes-component model.

Model encapsulates core data and basic functionality of their treatment. Also component model does not depend on the process input or output. Component output view can have several interconnected domains, such as various tables and form fields, in which information is displayed. The functions of the controller is monitoring the developments resulting from user actions (change of the mouse, pressing buttons or entering data in a text field).

6. Conclusions

So, Big Data are not a speculation, but a symbol of the coming technological revolution. The need for the analytical effort with Big Data will significantly change the face of the IT industry and stimulate the emergence of the new software and hardware platforms.

To achieve the desired goal, the development of a formal model of the Big Data information technology is made and its structural elements are described.

Today for the analysis of large volumes of data the most advanced methods are used: artificial neural network; predictive analytics, statistics and Natural Language Processing. Also, we use the methods that engage human experts, crowdsourcing, A / B testing, sentiment analysis, and the like. To visualize the results the known methods are applied, for example, tag clouds and completely new Clustergram, History Flow and Spatial Information Flow.

In this paper there is projected Data Space architecture and instrumentation tools for practical realization. There are chased program tools for variant data integration realization. The realization specification is described. There are described language tools and user interface realization.

References

[1]. Roger Magoulas, Lorica Ben, Big data: Technologies and techniques for large scale data, Release 2.0, 2009.

[2]. D. Kossmann, J. P. Dittrich, Personal Data Spaces, 2006, http://www.inf.ethz.ch/news/focus/res_focus/feb_2006/index_DE

[3]. J. Hooman, J. van de Pol, Equivalent semantic models for a distributed Dataspace architecture, Chapter in Formal Methods for Components and Objects, *Springer Berlin Heidelberg*, Vol. 2852, 2003, pp. 182-201.

[4]. The Open Archives Initiative Protocol for Metadata Harvesting Protocol Version 2.0 of 2002-06-14. http://www.openarchives.org/OAI/2.0/openarchives protocol.htm

[5]. V. I. Gritsenko, A. A. Ursatev, Information technologies, trends, ways of development, *Control Systems and Machines*, No. 5, 2001, pp. 3-20 (in Ukrainian).

[6]. T. White, Hadoop: The Definitive Guide, *O'Reilly Media*, 2012, pp. 3.

[7]. Martin Hilbert, Big Data for Development: From Information - to Knowledge Societies, SSRN Scholarly Paper No. ID 220514, *Social Science*

Research Network, Rochester, NY, 2013. http://papers.ssrn.com/abstract=2205145
[8]. IBM What is Big Data? — Bringing big data to the enterprise. www.ibm.com, 2013.
[9]. Oracle and FSN: Mastering Big Data: CFO Strategies to Transform Insight into Opportunity, 2012.
[10]. A. Jacobs, The Pathologies of Big Data, *ACM Magazine Queue*, Vol. 7, Issue 6, 2009.
[11]. R. Magoulas, B. Lorica, Introduction to Big Data, Release 2.0, *O'Reilly Media*, Sebastopol, CA, 2009.
[12]. C. Snijders, U. Matzat, U.-D. Reips, 'Big Data': Big gaps of knowledge in the field of Internet, *International Journal of Internet Science,* Vol. 7, Issue 1, 2012, pp. 1-5. http://www.ijis.net/ijis7_1/ijis7_1_editorial.html.
[13]. D. Laney, 3D Data Management: Controlling Data Volume, Velocity and Variety, *Gartner*, 2001.
[14]. M. Beyer, Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data, *Gartner*, 2011. Archived from the original on 10 July 2011.
[15]. D. Laney, The Importance of 'Big Data': A Definition, *Gartner*, 2012.
[16]. Stonebraker M., Abadi D., DeWitt D. J., Madden S., Pavlo A., Rasin A., MapReduce and Parallel DBMSs: Friends or Foes, *Communications of the ACM,* Vol. 53, Issue 1, 2012, pp. 64-71.
[17]. Inmon W. H., Big Data – getting it right: A checklist to evaluate your environment, *DSSResources.COM,* 2014. http:// dssresources.com/ papers/features/ inmon inmon01162014.htm/
[18]. Manyika James, *et al*., Big data: The next frontier for innovation, competition, and productivity, *McKinsey Global Institute*, May 2011. 156 p.
[19]. Shakhovska N. B., Features Data spaces modelling, *Proceedings of the National University* "Lviv Polytechnic". – Lviv: Izd Nat. Univ "Lviv. Polytechnic", *Computer Engineering and Information Technology*. No. 608, 2008, pp. 145-154. (in Ukrainian).
[20]. Shakhovska N. B., Veres O. M., Bolubash Yu. Ja., Bychkovska L., Data Space architecture for Big Data managering, in *Proceedings of the Xth Conference on Computer Science & Information on Technologies (CSIT'15)*, Lviv, 14-17 September 2015, pp. 184-187. (in Ukrainian).