Visualizing Large Graphs Out of Unstructured Data for Competitive Intelligence Purposes

Zakaria Boulouard, Lahcen Koutti, Nihal Chouati, Amine El Haddadi, Bernard Dousset, Anass El Haddadi, Fadwa Bouhafer

Visualizing Large Graphs Out of Unstructured Data for Competitive Intelligence Purposes

Zakaria Boulouard[1(✉)], Lahcen Koutti[1], Nihal Chouati[2], Amine El Haddadi[3], Bernard Dousset[3], Anass El Haddadi[4], and Fadwa Bouhafer[4]

[1] LabSIV, Faculty of Sciences, Ibn Zohr University, Agadir, Morocco
zboulouard@gmail.com, lkoutti@yahoo.fr
[2] LabTEC, National School of Applied Sciences, Tangier, Morocco
chouati.nihal89@gmail.com
[3] SIG, Institut de Recherche en Informatique de Toulouse, Toulouse, France
amine.elhaddadi@gmail.com, dousset.bernard@gmail.com
[4] Mathematics and Computer Sciences Department,
National School of Applied Sciences, Al Hoceima, Morocco
anass.elhaddadigmail.com, fadwa.bouhafer@gmail.com

Abstract. In the information era, people's lives are deeply impacted by IT via exposure to social media, emails, RSS feed, chats, web pages, etc. Such data is considered very valuable nowadays since it may help companies to better their strategies. For example, companies can analyse their customers' trends or their competitors marketing interventions and adjust their strategies accordingly. Several decisional tools have been developed but most of them rely on relational databases. This makes it difficult for decision makers to take advantage of unstructured data which today represents more than 85% of the available data. Thus, there is a rising need for a suitable management process of unstructured data through collecting, managing, transferring and transforming it into a meaningful informed data. This paper will introduce a new tool for Big Unstructured Data for the Competitive Intelligence named Xplor EveryWhere (XEW). It will also describe the enhancement brought to its newest feature XEWGraph. This tool, or as described later on the paper, this "Service", offers the decision makers the possibility to have a better user experience regarding large graph visualization on their web browsers as well as their mobile devices.

Keywords: Big data · Unstructured data · Competitive intelligence system · Graphs · Graph visualization · Large graphs

1 Introduction

People nowadays rely on information technology in their daily lives, resulting in petabytes of shared data. This can present a huge opportunity for companies wishing to improve their presence in the market as well as their strategy as a whole. The use of the traditional decisional tools, mostly based on relational databases, has showed its limits in analyzing such data since, according to Merrill Lynch; more than 85% of all business information exists as unstructured data commonly appearing in emails, memos, notes

from call centers and support operations, news, user groups, chats, reports, letters, surveys, white papers, marketing material, research, presentations and web pages (Blumberg and Atre [3]). Plejic et al. [24] have described unstructured data as often being part of a document's text body, or content not included in structured data management systems. Common examples where we can find unstructured data are emails, maps, reports, contracts, images, movies, spreadsheets, web content and presentations. As we can see, unstructured data comes also in different forms. Thus, the user will encounter many issues in handling these data which will further require extra programming and coding (Yafooz et al. [32]).

This paper will explore several ways to deal with unstructured data, then present the process proposed by the Competitive Intelligence System Xplor EveryWhere (XEW). This tool starts by collecting unstructured data, synthetizes it and then represents it to the decision makers in the most understandable way possible. The produced synthetized information takes often a relational form based on the connections between actors, semantic networks, etc. Representing this information as a graph may ease its analysis for non-experts since understanding a network's (graph's) structure helps understanding the way its components interact. This need has urged the development of a new service to be proposed in XEW. This service, called "XEWGraph" will display related unstructured data as a graph accessible from the user's browser or mobile phone.

It is important to mention that visualization in existing systems is not satisfactory when it comes to readability, be it in the global structure or in the detailed analysis of local communities, especially when it comes to complex graphs. XEWGraph proposes a new approach based on reducing the distance between related nodes, and as such, gaining more display space.

The rest of this paper is structured as follows: Sect. 2 will give a definition of Big Data. Section 3 describes the significance and methods of managing unstructured data. Section 4 gives a presentation on XEW. Section 5 will talk about large graph visualization and present the enhancement brought to XEW's newest service XEWGraph and the team's new approach regarding large graph visualization.

2 Big Data

The growing amount of data has made the researchers as well as the professionals set standards defining what is called "Big Data". Until recently, this trend has been defined as shown in Fig. 1.

This model has recently been extended to 5 V explained by Lomotey and Deters [19] as follows:

- **Volume:** The actual size of data keeps growing in an exponential rate. It is believed that the amount of data produced within the last two years is more than the total electronic data ever created.
- **Variety:** The data being generated comes in heterogeneous formats and from multiple sources, besides having no standard schema to contain either semi-structured or unstructured data.

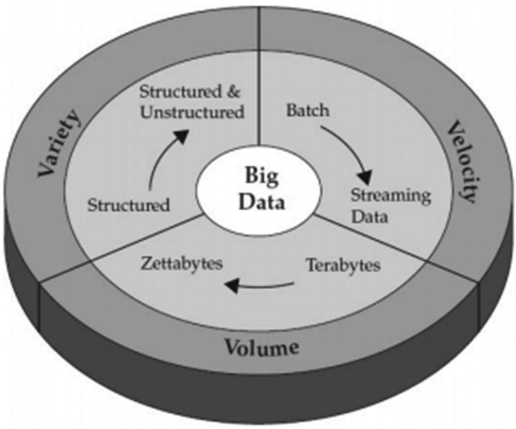


Fig. 1. The initial 3Vs of Big Data (Zikopoulos et al. [34])

- **Velocity:** The concept of data sets (batches) is very fast moving to data streams due to the speed of data coming in and going out.
- **Value:** Identified by Lomotey and Deters [19] as the cost. This definition is based on the fact that enterprises are in possession of various data that have different price values.
- **Veracity:** Getting the "noise" out of the data will guarantee its quality and will make us sure that the data we get is the one we want.

3 Significance and Methods of Managing Unstructured Data

Business routine cases before, used to require analyzing gigabytes of documentary data (big data) on a daily basis. This need has arisen much more nowadays due to these facts (Geetha and Mala [10]):

- 80% of all entity data is unstructured.
- Amount of unstructured data doubling every two months.
- Computer's inability to manage content based data still remains a problematic.
- Most sophisticated Artificial Intelligence tools are still unable to realize proper analysis.
- Text analysis technologies seem better at data reduction than actual data analysis.

In this section, we will present several methods of managing unstructured data, especially textual and image data.

3.1 Using Relational Databases

Relational Database Management Systems (RDBMS) have a powerful and robust data structure for managing, organizing, and retrieving structured data (Doan et al. [6]).

This made the the most preferred way to manage data in the business world for over 30 years. Yet, the exponential growth of unstructured data made it difficult for RDBMS to keep up. Besides its enormous size, it comes in different shapes and without any constraints or rules. Examples of that can be textual documents in directories, emails, reports, and online news articles. Gupta [31] pointed out that 80% of information is often stored in text documents, thus the urgency of a suitable management process for unstructured data.

Most of the techniques proposed so far are based on mapping unstructured data to structured data.

Abdullah and Ahmad [1] suggested that this mapping should be according to the following four steps process:

- **Extraction:** It is about identifying the format and the source of unstructured data. It has two main activities:
  - *Entity Extraction:* It is the process of extracting entities found within unstructured data such as names, dates, places, etc.
  - *Fact Extraction:* It is the process of understanding the information about the facts from the unstructured data (contacts, issues, content, etc.) which is important for integration purposes.
- **Classification:** It is a process where unstructured data is classified or categorized according to their nature and format. Four main data classes have been identified (Text, Image, Audio, and Video).
- **Repositories Development:** The main activity in this process is the preparation and development of individual repositories to store all identified unstructured data.
- **Data Mapping:** It has two main activities:
  - *Preparation of the subject:* It comes from the study of the business needs and the organizational interests.
  - *Mapping:* It requires involving metadata as a linkage to create association between unstructured data with the thematic topic. The content of the metadata is defined earlier based on the organizational needs.

Meanwhile, Yafooz et al. [33] have pointed out three methods for managing unstructured data using a relational database approach: by creating a database schema, by developing a new data model, or by query search:

(1) *Database Scheme:* A database scheme is a description of the entities in a database.

Mansuri and Sarawagi [21] have proposed a technique that establishes a connection among unstructured data in a relational database. Their technique is based on two stages: First, the named entities are extracted. Second, the extracted entities are matched with an entity that already exists in the database table or in the same table.

Similarly, Tari et al. [28] have proposed an intermediate repository for an incremental information extraction framework with a relational database management system to avoid repeating several information extraction processes in biomedical textual articles. Figure 2 demonstrates system architecture of incremental repository. change the default, adjust the template as follows:

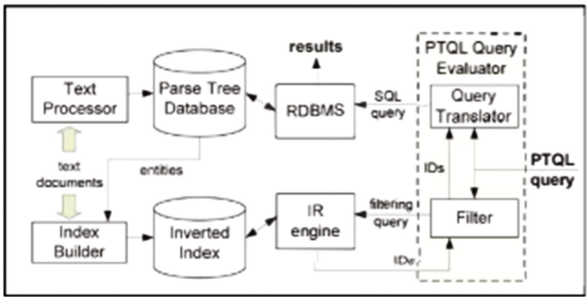


Fig. 2. System architecture of incremental repository (similarly, Tari et al. [28])

Chu et al. [5] developed an extraction architecture based on a database scheme that incrementally extracts structured information from textual data for further queries.

(2) *Data Model*: A data model is a data structure used to organize data. Doan et al. [6] have introduced the Unstructured Database Management System (UDMS) which is based on a data model called Data Generation and Exploitation (DGE). DGE interacts with three main elements: System, Data, and Users. Liu et al. [18] developed AUDR which focuses on managing multimedia file based on a tetrahedral data model.

Commercial Database vendors have two methods:

- *Traditional:* Storing textual data in a variable with a link to the file stored separately.
- *Modern:* Introducing Binary Large Objects (Oracle) and FileStreams (Microsoft).

(3) *Query-Based Approach:* It is a front-end database technique used to retrieve data from text database.

There are attempts to run SQL queries on unstructured data (Extract then Query, Query then Extract, and Keyword Search).

- *Extract then Query:* It is an offline process that focuses on extracting structured data from text files then storing them into the database.
- *Query then Extract:* It retrieves only the relevant document.
- *Keyword Search:* It is utilized to search for a term or a word in a text document or a search engine.

3.2 Using XML

Abidin et al. [2] have proposed an approach for capturing unstructured data in web pages, classifying them, transforming them into XML format and then saving them into a multimedia database. Their prototype was built based on a framework of five layers:

- User.
- Interface: It is an interaction medium between the Source and the User that allows manipulating the data in the web page.
- Source: It consists of a huge amount of useful data in the form of Structured, Semi-Structured or Unstructured web pages.
- XML: In this layer, the results of the classification process will be placed into a structured XML document.
- Storage: The storage system used in this layer is a multimedia database.

They have considered classification as the most important step in the process, especially when it comes to the data extraction. They have identified four classes (Text, Image, Video, and Audio) and each of these classes has several subclasses which represent the detail category of a particular data.

The classification process was based on the DOM Tree Technique in order to find the correct data in the HTML document. Some of the unnecessary nodes, such as script, style, or other customized nodes were filtered which has minimized the unnecessary information during the extraction process.

3.3 Using NoSQL

NoSQL is a new generation database management systems introduced to solve problems and limitations of RDBMS such as performance and managing large amounts of data. They are designed to be implemented in a distributed environment; the workload is thus divided among several machines.

It is to consider though that Yafooz et al. [33] have considered NoSQL management systems are still unable to fill for the RDBMS since their databases lack the most important database properties which are: Atomic, Consistent, Isolated, and Durable (ACID). The databases known as ACID are guaranteed to achieve successful database transactions. Sequeda and Miranker [27] add to that the fact that NoSQL databases are products of different vendors so their query styles vary, and in order to aggregate the data from all of these sources, a mush up service has to be deployed obliging the developer to study different APIs and deal with different structures of returned data.

On the other hand, Lomotey and Deters [19] believed in the potentials of NoSQL and suggested that it just needs a proper data mining tools based on it. They have proposed a tool based on two algorithms: Parallel Search and Bloom Filtering.

4 Mining Unstructured Data Approach in the Competitive Intelligence System XEW

After a great number of strategic analyses that were already conducted using the softwares Tétralogie and Xplor V1, the result was that the final users of analysis products need, along with the macroscopic view, some microscopic analyses on the already identified elements (competitiveness, markets, new products or processes, potential partners, etc.) or to discover others. In hindsight, many experts or decision

makers need more details on the traditional elements of their environment, especially concerning their specific vocabulary, the actors and markets around them, as well as the alliances they plan.

Thus, the idea is to keep adopting the proposed Xplor model and to complete its macroscopic analyses by an advanced online model XEW that enhances the obtained information using statistical overlaps, incremental classifications or multidimensional analyses. Our goal is to favor the information's extraction according to the general context and non-exclusively by decrypting the contents of separate documents. This makes it possible to retrieve, from a known element (actor, keyword), all or some of its related information (teams, collaboration, concepts, rises, associate keywords, etc.) using advanced filtering concepts.

The XEW prototype helps running strategic analyses on information corpuses coming from all various sources such as online bases (scientific publications, patents, portals, directories), CDs, visible and invisible web, newspapers, internal bases, RSS feed, social networks, etc. and gives the decision makers the possibility to run their own investigations without the assistance of a senior analyst or expert.

Its applications are very diverse:

- Identification of themes and actors of the field.
- Demonstrating the development and cooperation strategies.
- Proposing scenarios for the technologic evolution (innovation).
- Extracting weak signals.
- Consulting updated information in real time thanks to the web services.
- Make up "field" information during salons, customer visits, or meetings.
- Asking for urgent specific information to be online.

The CI model XEW relies on a four level decisional architecture, as presented in Fig. 3.

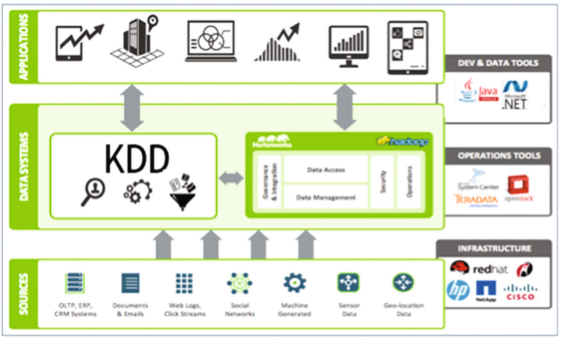


Fig. 3. XEW architecture

4.1 XEW Sourcing Service (XEW-SS)

This service allows searching, collecting, and processing the data from different sources. This requires consideration of a multimodal fusion able to consider the heterogeneity, the imprecision and the uncertainty of multisource data. This fusion awareness ensures mastering the knowledge and the information and consequently, eases the decision making. XEW-SS processes the information heterogeneity from different sides:

- *Semantic content:* scientific, technical, etc.
- *Structural:* From highly structured (Patents), to unstructured (emails).
- *Language (multilingualism):* Chinese, Arabic, etc.
- *Support format:* Word, HTML, PDF, etc.
- *Size:* Definition of the information unit to be analyzed (information granularity).

This architectural level's objective is to provide a complete description of the multisource data process. The techniques used in this level rely on web services dedicated to every source of information.

4.2 Meta-model of Unstructured Data

The multidimensional model aims to identify all the relations of existing dependencies between different variables from the subject of analysis. These relations are defined by co-occurence matrices which indicate the simultaneous presence of the methods of two qualitative variables in a document.

We have altered these matrices by introducing a third temporal variable (Year, Month, Days, Hours), which consists in indicating the presence of a certain relation, in a certain moment.

Example. Figure 4 presents a formed multidimensional presentation of collaborations between scientists in cells and of three edges graduated respectively according to the sets of their research themes, the organizations they are affiliated to and the publication

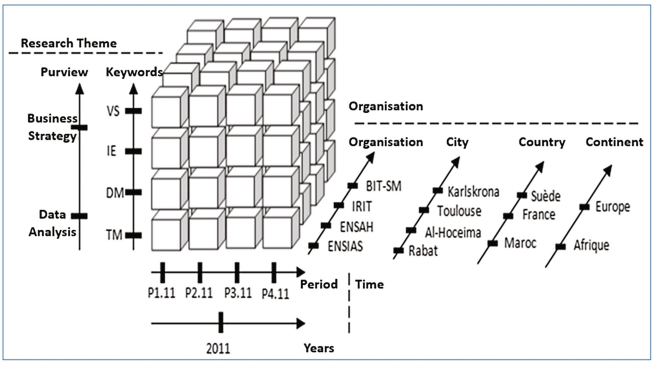


Fig. 4. Laboratory collaboration

dates of their respective articles. This presentation is not limited to three axes but spreads into a meta-meta-model where the number of axes is able to go to several tens.

4.3 Homogenization of the Information Source

The final objective is to obtain a unified sight on the collected sources, which will be used throughout the process of analysis. This sight must be:

- Homogeneous: shared by the various data whatever their sources.
- Reduced: to facilitate and accelerate the treatment of information.
- Able to facilitate the analysis of any type of information and to restore it within very short times in order to answer to the competitive intelligence requirements.

This unified sight associated with the targeted corpus corresponds to a logical structured representation, presenting its whole collections in the form of a warehouse of strategic data. This homogenization process is described in (Fig. 5).

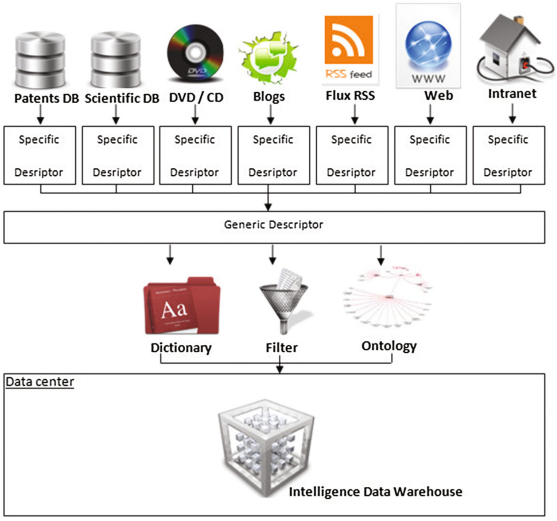


Fig. 5. Information proceeding subsystem

This service is a storage space called "XEW Data Warehousing Services – XEWDWS" which allows, on its first level, to have a unified view of the target corpus, extraction and storage of incoming data, would it be structured, semi-structured or unstructured and represent it in a multidimensional form. The second level is about the data warehouse creation processes: from the classical SQL to NoSQL (MongoDB, Neo4j, GraphDB, HBase, etc.).

4.4 XEW Big Data Analytics Service (XEW-BDAS)

This service allows making multidimensional analyses by adapting data mining algorithms to Big Data. It is based on the parallelism of the algorithms developed in the XEW system, as well as other open source tools such as Weka or R.

5 Graph Visualization in CIS XEW

Competitive Intelligence (CI) is a set of coordinated actions of search, treatment and distribution of useful information helping stakeholders in the process of decision making. In the opposite of industrial espionage, CI is a legal process of competitiveness where the source of information is the external business environment.

The concept of CI is a bit wide and it is necessary to define a specific framework based on a multidisciplinary approach. The CI process definition adopted while developing XEW was based on three concepts:

- Strategic Analysis: Defines the informational needs of a company on its environment in order to ease decision making.
- Environment Analysis: It is the process of collecting, processing, analysing and diffusing the useful information in order to respond to the expressed needs.
- Information System: Used as a support to the activities of data collection, analysis and reporting.

Graphs are considered among the most powerful tools when it comes to visualizing trends, making them a research orientation with gravitas [4]. They can display any type of information and thus, can respond to the needs of Competitive Intelligence.

It is to mention that, opposite to the typical Business Intelligence (BI) graphs, which are based on structured data, CI graphs are based on unstructured, even massive, data. This brings the necessity to define layout algorithms or solutions that may be able to visualize a large amount of data, and in other words, large graphs.

A study conducted by Hu and Shi [14] has toured several techniques of large graph visualization and categorized them into four different major models:

- **Spring-Electric Model:** This model inculdes Force Directed Placement along with the algorithms derivating from Eads works [7]. When it comes to visualizing massive data, FDP tends to fall into local minima due to the great amount of nodes and the repulsive charges they emit. Several solutions were proposed in order to address this problem, especially the approaches of Tunkelang [29] and Quigley [26] along with the multilevel approach where a sequence of smaller and smaller graphs is generated from the original graph. Every graph of these captures the information concerning the connectivity of its parent. Once the smallest graph possible is generated, it gets refined and adjusted.
- **Stress Model:** In this model, instead of minizing the attractive/repulsive energy within nodes, we try to minimize it within the edges (represented as springs). As an example, Kamada and Kawai [16] have proposed an algorithm that minimizes the stress energy within the edges by bringing the distance between the nodes to the ideal length of the spring connecting them, and since the latter depends on the distances

between all the couples of nodes, they should be all calculated by finding the shortest paths between all these couples. In this case, scaling would be extremely expansive. Several teams tried to solve this problem, such as Hadany et Harel [10] who proposed a solution to better the speed of the Kamada-Kawai [16] algorithm by accelerating its convergence, while Gajer et al. [9] have proposed a multilevel approach similar to the one precited.

- **High Dimensional Embedding:** This algorithm, also known as «HDE» [11] affects coordinates to nodes in k-dimension space, then projects it into a regular 2-D or 3-D space.
- **Algorithms Based on the Spectral Information of the Laplacian:** Hall [12] remarked that several node positioning problems could be brought back to problems of defining positions which would minimze the weighted sum of the squared distances between the nodes. Hu and Shi [14] have mathematically represented this proposition using this notation:

$$\sum_{i \leftrightarrow j} w_{ij} \left\| x_i - x_j \right\|^2, such\ as \sum_{k=1}^{|V|} x_k^2 = 1$$

«x_i» is the 1-D coordinate of the node «i».

This function can also be rewritten as:

$$\sum_{i \leftrightarrow j} w_{ij} \left\| x_i - x_j \right\|^2 = x^T L_w x,$$

$$such\ as\ x = \left\{ x_1, x_2, \ldots, x_{|V|} \right\}$$

$$and\ L_w\ is\ the\ weighted\ Laplacian\ matrix$$

The solution «x» to this minimization problem is the eigenvector which has the smallest positive eigenvalue of the weighted Laplacian matrix «L_w».

Koren et al. [17] have proposed an algorithm that brings a quick solution this problematic based on a multilevel approach, but it still keeps the weakness of Hall's algorithm regarding sparse graphs.

In XEW, the massive data is collected, analysed and organized according to the steps mentioned earlier. The connected data is later filtered and stored in a Graph oriented datamart which will be the source of our graphs.

5.1 Graph Visualization

According to Purchase [25], a well spread graph should provide an explicit vision on the relationships between the presented entities, which may help the decision maker have a quick understanding of the graph and take the useful information out of it.

In order to respond to this growing need, several graph representations (or layouts) have been suggested. Tutte [30], as one of the pioneers of this field, proposed to lay down the first nodes on a plan, then the later ones on the barycenters of their neighbors.

Eads [7] has suggested a model called "Spring Layout" where the nodes are given an initial positioning and then, the edges (represented as springs) would bring the nodes back to an equilibrium position corresponding to a global energy minimum. This work has been improved later on by Fruchterman and Reingold [8] who introduced the Force Directed Placement (FDP) in which the attractive force of the spring between two neighboring nodes is proportional to the squared distance between them. The attraction force is then expressed as:

$$F_a = -\frac{d^2(n_1, n_2)}{K}$$

K is a parameter related to the nominal edge length of the final layout.

On the other hand, the repulsive force between any node on any other node is inversely proportional to the distance between these two nodes. It is expressed as:

$$F_r = -\frac{K^2}{d(n_1, n_2)}$$

Fruchterman and Reingold thought later about reducing the complexity of this algorithm by partitioning the drawing space into a cell grid in order to calculate the local repulsive energy between nodes in neighboring cell. This procedure could cause several calculation errors since it neglects the repulsive forces that may exist between non neighboring nodes.

Tunkelang [29] and Quigley [26] could find a solution to this problem by introducing quadtrees. A quadtree is a grouping of nodes which could be presented as a "Super-Node" with which we can approximate the total repulsive force of the nodes it contains. If a group of nodes is far enough from a certain node, the group of nodes is then considered a super-node. Other methods were proposed like minimizing the spring energy between links, etc.

5.2 Benchmarking

Several trials for algorithms have been launched in order to have a better graph visualization in a Competitive Intelligence context but those who got our attention were Gephi and VisuGraph.

(1) *Gephi:* Developed by Jacomy et al. [15], it is a free software for analyzing and visualizing data in form of graphs. It helps you get maximum information from the data, isolate its most important factors, detect inconsistencies and errors, etc. It imports data from different sources and displays it in the form of graphs.

This tool comes as an executable and offers a plugin management system, programming APIs and a graph visualization based on the most common algorithms.

The main visualization algorithm adopted by Gephi, as mentioned in Jacomy et al. [15] is ForceAtlas 2 (FA2). It is a Force Directed Layout (FDL) algorithm developed

by the Gephi team. It simulates a graph as a physical system in which nodes repel each other (Repulsive Force) while links attract back the connected nodes (Attraction Force).

The basic expression of the Attractive Force (F_a) between two connected nodes, according to the model adopted by Gephi team, is equal to the distance between these nodes, which would be represented as:

$$F_a = d(n_1, n_2)$$

The Repulsive Force (F_r) between two nodes, according to the same model, depends on the nodes degrees. This allows the highly connected nodes to be more centered while the less connected ones (called leaves) are repelled to the suburbs. The Repulsive Force is expressed as:

$$F_r = K_r \frac{\deg(n_1 + 1) * \deg(n_2 + 1)}{d(n_1, n_2)}$$

K_r can be fixed by the settings and the (+1), different from Noack's expression (Noack [22]), is added in order to make sure that even the nodes with a 0 deg can have a repulsive force.

The combination of these two forces creates a movement that converges to an equilibrium position which would help interpreting the data. A node's position depends on the other nodes and on the links connecting it with them. This algorithm eases the visual interpretation of the data structure under study. However, it doesn't take the nodes attributes under consideration during the positioning process, which would be a problem if those attributes were actually meant to be the initial coordinates. In (Fig. 6), we can see a screenshot of Gephi.

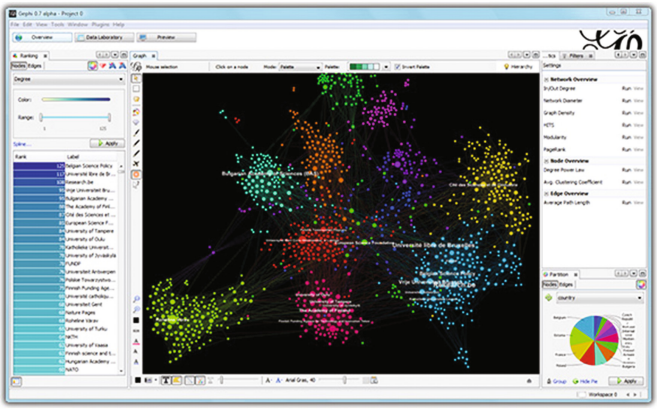


Fig. 6. Screenshot of Gephi

(2) *VisuGraph:* Developed by Loubier [20], it was particularly interesting by its approach regarding mass data visualization. This approach can be subdivided to 2 principal axes:

- Graph Visualization.
- Evolutionary Aspect: By taking the time variable into consideration while drawing the graph.

(a) Graph Visualization: Loubier has proposed a slight modification of the FDP by setting the attraction and replusion forces as:

The attraction force: $F_a(u,v) = \frac{\beta * d_{uv}^{\alpha_a}}{K}$

"β" is a constant parameter.

"d_{uv}" is the distance between the nodes u and v.

"α_a" is a parameter used to increase or decrease the attraction between the two pre-cited nodes.

"K" is calculated according to the dimensions of the drawing space: $K = \sqrt{\frac{L*l}{N}}$

"L" is the window's length, and "l" is its width.

The repulsion force: $F_r(u,v) = \frac{\alpha_r * K^2}{d_{uv}^c}$

"c" is a constant parameter.

«α_r» is a parameter used to increase or decrease the repulsion between the nodes u and v.

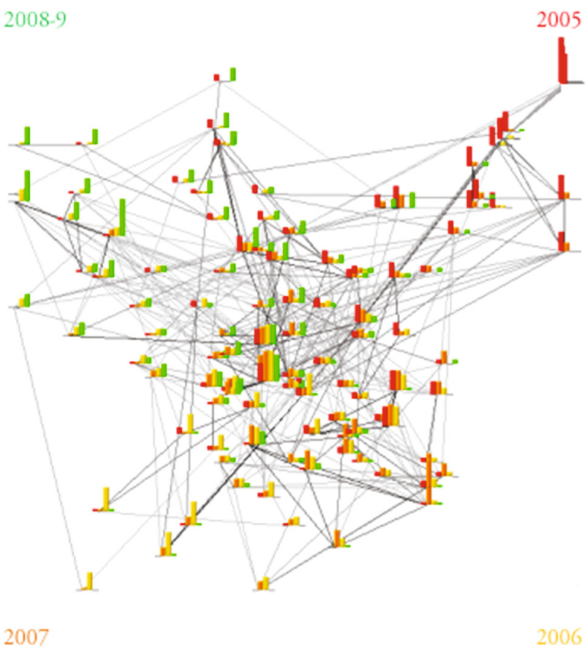


Fig. 7. Screenshot of VisuGraph

(b) Evolutionary aspect (Time Slices): Loubier has remarked that when the analysis is time dependant, a graph can send mixed signals or misinterpreted information. Thus, she proposed a graph presentation based on "Time Slices". Every slice represents a certain period of time.

Taking the temporal dimension into account in the graph visualization goes in two steps:

- First, a global graph representing all periods is drawn.
- Second, some virtual nodes representing the time slices are scattered around the drawing space and the graph nodes are positioned near the virtual nodes which represent the time slice they belong to; as depicted in (Fig. 7).

5.3 XEWGraph

XEWGraph is a new module for the Competitive Intelligence System Xplor EveryWhere dedicated to visualizing big data in the form of graphs representing, for instance, social networks, semantic networks or even strategic alliances networks.

The main objective of this tool, is to give the decision makers a better user experience when it comes to large graph visualization. Thus, several different approaches were adopted while developing it.

First, we adopted the Force Directed Placement as defined by Fruchterman and Reingold [8] and we bettered it by an approach inspired by the hypergraphs. That resulted with an "out of the box" categorization.

The need for that approach came from the fact that XEWGraph, being a part of Xplor EveryWhere which is web and mobile oriented, has suffered from many problems when the graphs get larger.

The hypergraph approach has helped for sure, but it wasn't enough. The results of the FDP itself needed to be optimized.

In order to do so, an algorithm that would be executed after the FDP was proposed. Its purpose would be to replace the nodes in order to gain more space on the screen and to better represent the weights of the edges.

The algorithm is similar to the FDP as described by Hu and Shi [14] except in the function to be applied on the nodes referred to as FForce.

(1) *FForce function:* It describes a combination between the attractive and the repulsive forces by taking the weights of the nodes and the edges into consideration while drawing.

It is expressed as follows:

$$f_{ij} = -\frac{\mathrm{a}}{\mathrm{d}} + \mathrm{b} * (\mathrm{d} - \mathrm{d}_0) + \mathrm{c} * \left(\frac{\mathrm{d}}{\mathrm{d}_0} - 1\right)^3$$

«f_{ij}» is the FForce applied between the nodes «*i*» and «*j*».
«d» is the euclidian distance between the nodes «*i*» and «*j*».
«d_0» is the spring's rest length.
«a», «b» and «c» are coefficients that were given the following values:

$$\mathrm{a} = \mathrm{d}_0, \mathrm{b} = \mathrm{c} = \frac{w_j * \mathrm{v}_{ij}}{w_i + w_j}$$

«a» adjusts the size of the graph according to the screen size.
«w_i» and «w_j» are the weights of the nodes «*i*» and «*j*» respectively.
«$\frac{w_j}{w_i + w_j}$» differenciates the action of the node «*j*» according to its weight.
«v_{ij}» is the value of the graph's matrix corresponding to the nodes «*i*» and «*j*» respectively. It makes the force proportional to the value of the link (case of weighted links).

The weight «w_i» of a node «*i*» is expressed by the maximum value «$\max(v_{i.})$» of its corresponding row in the graph's matrix.

The weight of an edge connecting two nodes «*i*» and «*j*» is expressed by the value «v_{ij}» of the graph's matrix in the corresponding row and column respectively.

The term «$\frac{\mathrm{a}}{\mathrm{d}}$» expresses the repulsive force.

The term «$\mathrm{b} * (\mathrm{d} - \mathrm{d}_0) = \frac{w_j * \mathrm{v}_{ij}}{w_i + w_j} * (\mathrm{d} - \mathrm{d}_0)$» expresses the linear effect of the attraction on the edge (assimilated to a coil spring).

The term «$\mathrm{c} * \left(\frac{\mathrm{d}}{\mathrm{d}_0} - 1\right)^3 = \frac{w_j * \mathrm{v}_{ij}}{w_i + w_j} * \left(\frac{\mathrm{d}}{\mathrm{d}_0} - 1\right)^3$» expresses the non-linear effect of the attraction on the edge which would spread the nodes and at the same time limit the graph's expansion.

(2) *FForce algorithm:* As mentioned earlier, it is highly inspired from the FDP algorithm as described in the pseudo-code by Hu and Shi [14]. Our algorithm is described in the pseudo-code (Algorithm 1).

The «break» term expresses the node's displacement increment.

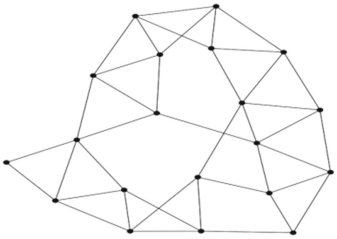
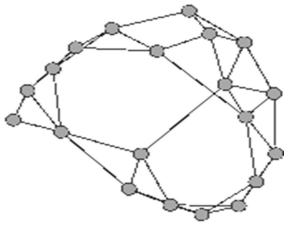
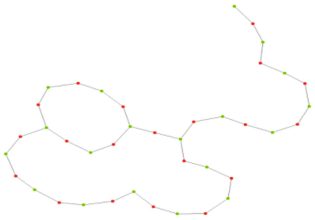
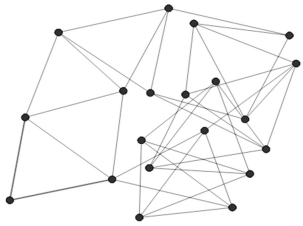
This algorithm is set to run right after the FDP in order to better the nodes equilibrium positions and at the same time, to gain more space on the screen. It must be mentioned that applying the algorithm alone gave a rendering that is close to the FDP and that better results were diagnosed by applying the FDP first then applying the FForce later.

Algorithm 1 FForceAlgorithm(G, x, tol, K)

input:graph $G = \{V, E\}$, initial positions x , tolerance tol , and nominal edge length K
set break = initial break value
repeat
    $x_0 = x$
    For($i \in V$) {
        $f = 0$ //f is a 2-D or 3-D vector
        $f \leftarrow f +$ FForce(i, j)
        $x_i \leftarrow x_i + break * f$
    }
until ($|x - x_0| < tol * K$)

The following paragraph will describe a test of the performance of the FDP coupled with FForce.

Table 1. Comparing the layouts performance for graph sized 20 nodes and 40 links

Graph size : 20 Nodes and 40 Links	
FFDP	**ForceAtlas2**
	
T-FDP	**FDP alone**
	

(3) *Testing:* In order to test the performance of this combination, called «FFDP», it was put into comparison with, the FDP applied alone, then with VisuGraph's T-FDP [20] and Gephi's ForceAtlas2 [15].

The testing sample is composed of 3 graphs with different sizes: the first graph has 20 nodes and 40 links, the second one has 100 nodes and 500 links, while the third one has 1000 nodes and 50000 links. Those graphs can be found among Gephi's test dataset available in the following link under the name «small_world»:

https://github.com/medialab/benchmarkForceAtlas2/blob/master/dataset.zip (last checked 28/02/2016).

The next Tables 1, 2, and 3 show the results of the comparison we proposed.

In the first graph, FFDP gave a better nodes dispositions, closely seconded by ForceAtlas2, while the visualizations provided by raw FDP and T-FDP were completely far from the expected result. In the second graph, raw FDP and T-FDP gave a practically similar representation, while ForceAtlas2 was able to gather the node in at least two collections of nodes. FFDP was able to bring out a third less obvious

Table 2. Comparing the layouts performance for graph sized 100 nodes and 500 links

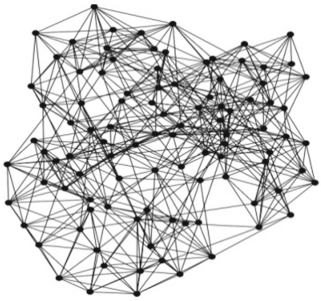
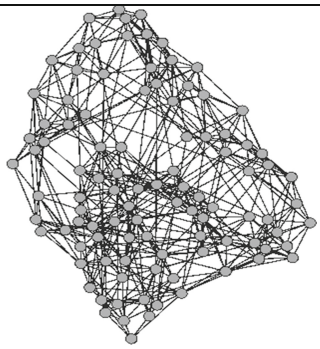
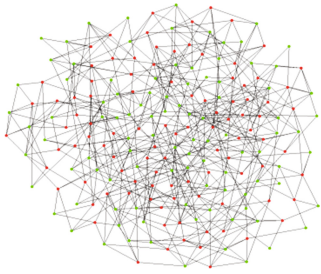
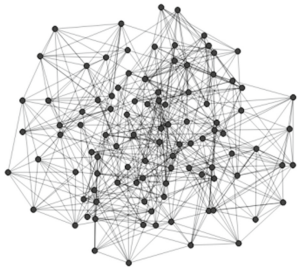
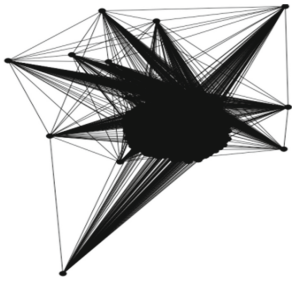
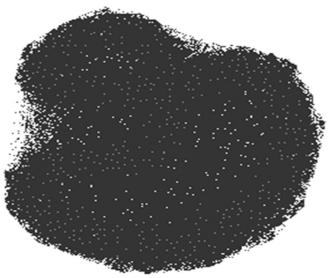
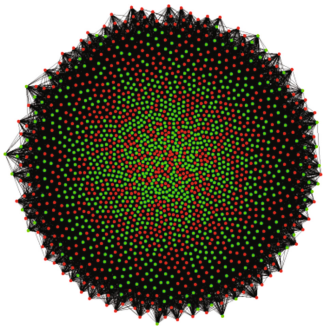
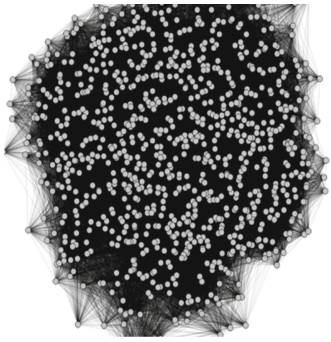
Graph size : 100 Nodes and 500 Links	
FFDP	**ForceAtlas2**
	
T-FDP	**FDP alone**
	

Table 3. Comparing the layouts performance for graph sized 1000 nodes and 50000 links

1000 NODES AND 50000 LINKS

Graph size : 1000 Nodes and 50000 Links	
FFDP	**ForceAtlas2**
	
T-FDP	**FDP alone**
	

collection in the center. In the third, largest graph, raw FDP and T-FDP were able to make a few nodes pop out of the cloud but it wasn't enough to get a proper reading on the graph. ForceAtlas2 was able to highlight two collections of nodes, while FFDP managed to highlight more.

The results rendered by FFDP are then clustered and categorized using the hypergraph approach. This gave two advantages, the first one is to be able to draw lighter, web destined graphs with a general view and then have a deeper view on more specific details according to the decision maker's needs. The second advantage is the ability to display these graphs within smaller screens such as smartphones. Figure 8 describes a clustered (a), then expanded (b) hypergraph representing the the collaboration between research teams that have published papers within previous edtions of the colloquium on Scientific and Technological Strategic Intelligence.

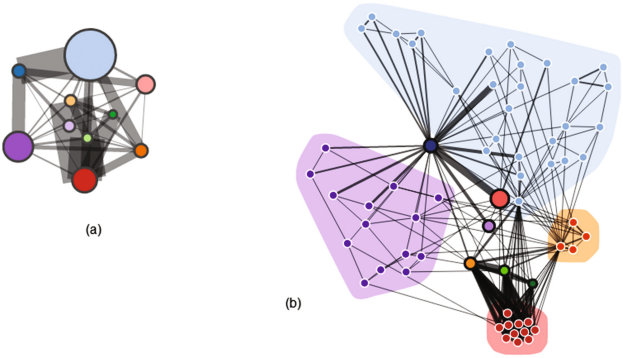


Fig. 8. Clustered (a) then expanded (b) hypergraph

Loubier's work on the "Time-Slices" inspired us to come up with another approach when it comes to taking the time variable into consideration. Indeed, a time slider was proposed with the idea of retrieving the time periods from the data and then, taking the decision maker in a "Time Travel" as he slides through the time periods. This will will give him an idea on the evolution of, for example, the collaboration between research teams throughout the last decade. In Fig. 9, you will see screenshots of the previous graph with a time slider but taken from smartphones.

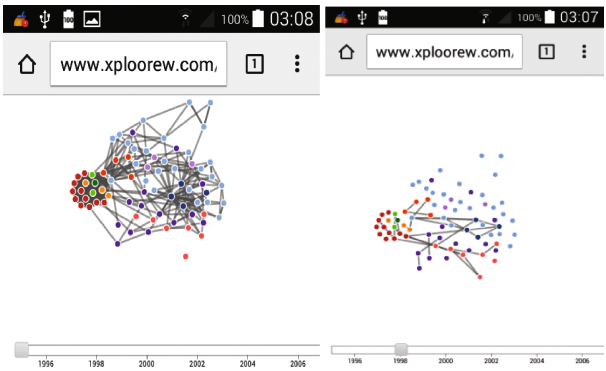


Fig. 9. a. The global graph. b. The graph situation in 1998

6 Conclusion and Perspectives

This paper has presented the services offered by the Competitive Intelligence tool Xplor EveryWhere (XEW), as well as the approach adopted for visualizing large graphs using the service XEWGraph.

This approach was based on the enhancement of the FDP by combining it with a function called «FForce». It describes a combination between the attractive and the repulsive forces by taking the weights of the nodes and the edges into consideration while drawing.

The results of this approach are later clustered and then sliced according to the time variable using the hypergraph and the time slice approaches.

Another approach will be adopted in order to better the visualization of larger graphs within XEWGraph, which is prepositioning. Indeed, if the server gives the nodes some initial coordinates that would bring them to a close to best position, it would ease the visualization process executed client-side.

This prepositioning approach can be bettered by adding a third coordinate and thus proposing a 3-D representation of the graphs within navigators and smartphones by taking advantage of the WebGL engine.

References

1. Abdullah, M.F., Ahmad, K.: The mapping process of unstructured data to structured data. In: 3rd Conference on Research and Innovation in Information Systems, ICRIIS (2013)
2. Abidin, S.Z.Z., Idris, N.M., Husain, A.H. Extraction and Classification of Unstructured Data in Web Pages for Structured Multimedia Database via XML. IEEE (2010)
3. Blumberg, R., Atre, S.: The problem with unstructured data. DM Rev. **13**, 42–46 (2003)
4. Spence, R.: The issues. In: Information Visualization: Design for Interaction, 2nd edn., pp. 16–28. ACM Press, New York (2007)
5. Chu, E., Baid, A., Chen, T., Doan, A., Naughton, J.: A relational approach to incrementally extracting and querying structure in unstructured data. In: Proceedings of the 33rd International Conference on Very Large Databases, vol. VLDB Endowment (2007)
6. Doan, A., Naughton, J.F., Baid, A., Chai, X., Chen, F., Chen, T., Chu, E., DeRose, P., Gao, B., Gokhale, C., Huang, J., Shen, W., Vuong, B.Q.: The case for a structured approach to managing unstructured data. arXiv preprint arXiv:0909.1783 (2009)
7. Eads, P.: A heuristic for graph drawing. Congr. Numer. **42**, 149–160 (1984)
8. Fruchterman, T.M.J., Reingold, E.M.: Graph Drawing by Force-Directed Placement. Software: Practice and Experience, pp. 1129–1164. Wiley, Software (1991)
9. Gajer, P., Goodrich, M.T., Kobourov, S.G.: A Fast Multidimensional Algorithm for Drawing Large Graphs. Lecture Notes on Computer Sciences, pp. 211–221. Springer, Berlin (2000)
10. Geetha, S., Mala, G.S.A.: Effectual extraction of data relations from unstructured data. In: 3rd International Conference on Sustainable Energy and Intelligent System, VCTW (2012)
11. Hadani, R., Harel, D.: A Multi-scale Algorithm for Drawing Graphs Nicely. Discrete Applied Mathematics, pp. 3–21. Elsevier, Amsterdam (2001)
12. Hall, K.M.: An r-dimensional quadratic placement algorithm. Manag. Sci. Informs J. Comput. **17**(3), 219–229 (1970)
13. Harel, D., Koren, Y.: High dimensional embedding. J. Graph Algorithms Appl. Brown Univ. **8**(2), 195–214 (2004)
14. Hu, Y., Shi, L.: *Visualizing large graphs*, pp. 115–136. Wiley Interdisciplinary Reviews: Computational Statistics. Wiley Periodicals Inc., New York (2015)

15. Jacomy, M., Venturini, T., Heymann, S., Bastian, M.: ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. PLoS ONE **9**, 1–12 (2014)
16. Kamada, T., Kawai, S.: An Algorithm for Drawing General Undirected Graphs. Information Processing Letters, pp. 7–15. Elsevier, Amsterdam (1989)
17. Koren, Y., Carmel, L., Harel, D.: Ace: a fast multiscale eigenvectors computation for drawing huge graphs. In: Proceedings of the IEEE Symposium on Information Visualization (InfoVis 2002), pp. 137–144 (2002)
18. Liu, X., Lang, B., Yu, W., Luo, J., Huang, L.: AUDR: an advanced unstructured data repository. In: 6th International Conference on Pervasive Computing and Applications (ICPCA). IEEE (2011)
19. Lomotey, R.K., Deters, R.: Topics and terms mining in unstructured data stores. In: 16th International Conference on Computational Science and Engineering. IEEE (2013)
20. Loubier, E.: Analyse et visualisation de données relationnelles par morphing de graphe prenant en compte la dimension temporelle. Ph.D. thesis, IRIT, Paul Sabatier University (2009)
21. Mansuri, I.R., Sarawagi, S. Integrating unstructured data into relational databases. In: Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006. IEEE (2006)
22. Noack, A.: An energy model for visual graph clustering. In: Proceedings of the 11th International Symposium on Graph Drawing (GD 2003). LNCS, vol. 2912, pp. 425–436. Springer, Berlin (2004)
23. Noack, A.: Energy models for graph clustering. J. Graph Algorithms Appl. **11**(2), 453–480 (2007)
24. Plejic, B., Vujnovic, B., Penco, R.: Transforming unstructured data from scattered sources into knowledge. In: IEEE International Symposium on Knowledge Acquisition and Modeling Workshop, 2008. KAM Workshop 2008, pp. 924–927 (2008)
25. Purchase, H.C.: Performance of layout algorithms: comprehension, not computation. J. Visual Lang. Comput. **9**(6), 647–657 (1998)
26. Quigley, A.: Large Scale Relational Information Visualization, Clustering, and Abstraction. Ph.D. Thesis, Department of Computer Science and Software Engineering, University of Newcastle, Australia (2001)
27. Sequeda, J., Miranker, D.P.: Linked Data. Linked Data tutorial at Semtech. (2010). http://fr.slideshare.net/juansequeda/linked-data-tutorial-at-semtech-2012
28. Tari, L., Tu, P.H., Hakenberg, J., Chen, Y., Son, T.C., Gonzalez, G., Baral, C.: Parse tree database for information extraction. IEEE Trans. Knowl. Data Eng. (2010).http://www.public.asu.edu/∼cbaral/papers/tkde10.pdf
29. Tunkelang, D.: A numerical optimization approach to general graph drawing. Ph.D. Thesis, Carnegie Mellon University (1999)
30. Tutte, W.T.: How to draw a graph. In: Proceedings of the London Mathematical Society, pp. 743–767 (1963)
31. Vishal Gupta, G.S.L.: A survey of text mining technics and applications. J. Emerg. Technol. Web Intell. **1**(1), 60–76 (2009)
32. Yafooz, W.M.S., Abidin, S.Z.Z., Omar, N.: Towards automatic column-based data object clustering for multilingual databases. In: IEEE International Conference on Control System, Computing and Engineering (ICCSCE), IEEE (2011)
33. Yafooz, W.M.S., Abidin, S.Z.Z., Omar, N., Idrus, Z.: Managing unstructured data in relational database. In: IEEE Conference on Systems, Process & Control (ICSPC) (2013)
34. Zikopoulos, P.C., Eaton, C., DeRoos, D., Deutsch, T., Lapis, G.: Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. McGraw-Hill, New York (2012)