ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 8, August 2018

Data Visualization and Visual Design Patterns: Based on Big Data Approach

DR. ARCHANA VERMA

Assistant Professor, Computer Science & Engineering , Bipin Tripathi Kumaon Institute of Technology, Dwarahat, Uttarakhand, India

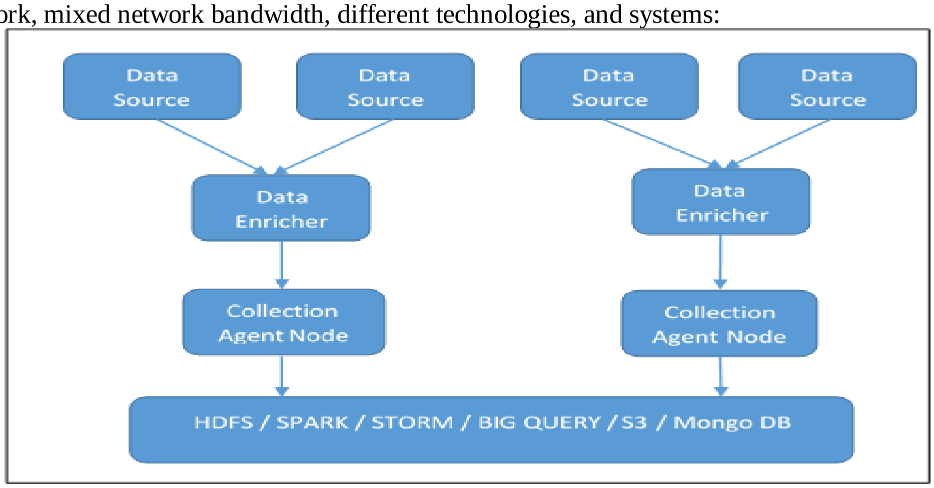
ABSTRACT: Design patterns have provided many ways to simplify the development of software applications. Now that organizations are beginning to tackle applications that leverage new sources and types of big data, design patterns for big data are needed. These big data design patterns aim to reduce complexity, boost the performance of integration and improve the results of working with new and larger forms of data.This article intends to introduce readers to the common big data design patterns based on various data layers such as data sources and ingestion layer, data storage layer and data access layer. Enterprise big data systems face a variety of data sources with non-relevant information (noise) alongside relevant (signal) data. Noise ratio is very high compared to signals, and so filtering the noise from the pertinent information, handling high volumes, and the velocity of data is significant. This is the responsibility of the ingestion layer. The common challenges in the ingestion layers are as follows: Multiple data source load and prioritization,Ingested data indexing and tagging,Data validation and cleansing, Data transformation and compression

KEYWORDS: design patterns, big data, visualization, approach, complexity, data storage, noise ratio

I.INTRODUCTION

Multisource extractor

An approach to ingesting multiple data types from multiple data sources efficiently is termed a Multisource extractor. Efficiency represents many factors, such as data velocity, data size, data frequency, and managing various data formats over an unreliable network, mixed network bandwidth, different technologies, and systems:



ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 8, August 2018

The multisource extractor system ensures high availability and distribution. It also confirms that the vast volume of data gets segregated into multiple batches across different nodes. The single node implementation is still helpful for lower volumes from a handful of clients, and of course, for a significant amount of data from multiple clients processed in batches. Partitioning into small volumes in clusters produces excellent results.Data enrichers help to do initial data aggregation and data cleansing. Enrichers ensure file transfer reliability, validations, noise reduction, compression, and transformation from native formats to standard formats. Collection agent nodes represent intermediary cluster systems, which helps final data processing and data loading to the destination systems.[1]The following are the benefits of the multisource extractor:

- Provides reasonable speed for storing and consuming the data
- Better data prioritization and processing
- Drives improved business decisions
- Decoupled and independent from data production to data consumption
- Data semantics and detection of changed data
- Scaleable and fault tolerance system

The following are the impacts of the multisource extractor:

- Difficult or impossible to achieve near real-time data processing
- Need to maintain multiple copies in enrichers and collection agents, leading to data redundancy and mammoth data volume in each node
- High availability trade-off with high costs to manage system capacity growth
- Infrastructure and configuration complexity increases to maintain batch processing

Multidestination pattern

In multisourcing, we saw the raw data ingestion to HDFS, but in most common cases the enterprise needs to ingest raw data not only to new HDFS systems but also to their existing traditional data storage, such as Informatica or other analytics platforms. In such cases, the additional number of data streams leads to many challenges, such as storage overflow, data errors (also known as data regret), an increase in time to transfer and process data, and so on.The multidestination pattern is considered as a better approach to overcome all of the challenges mentioned previously. This pattern is very similar to multisourcing until it is ready to integrate with multiple destinations (refer to the following diagram). [2]The router publishes the improved data and then broadcasts it to the subscriber destinations (already registered with a publishing agent on the router). Enrichers can act as publishers as well as subscribers:

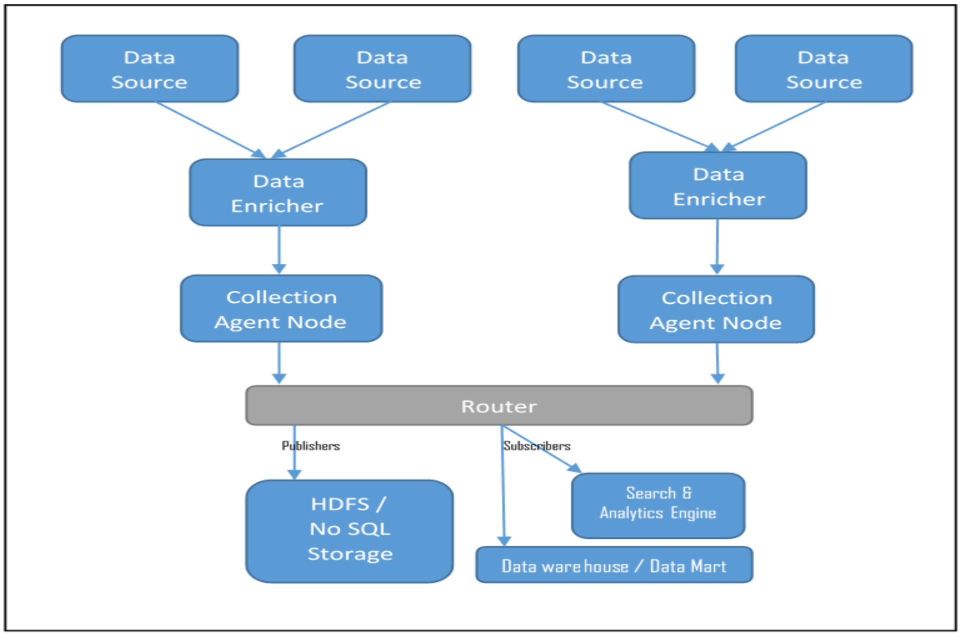
ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 8, August 2018



Deploying routers in the cluster environment is also recommended for high volumes and a large number of subscribers.The following are the benefits of the multidestination pattern:

- Highly scalable, flexible, fast, resilient to data failure, and cost-effective
- Organization can start to ingest data into multiple data stores, including its existing RDBMS as well as NoSQL data stores
- Allows you to use simple query language, such as Hive and Pig, along with traditional analytics
- Provides the ability to partition the data for flexible access and decentralized processing[3]
- Possibility of decentralized computation in the data nodes
- Due to replication on HDFS nodes, there are no data regrets
- Self-reliant data nodes can add more nodes without any delay

The following are the impacts of the multidestination pattern:

- Needs complex or additional infrastructure to manage distributed nodes
- Needs to manage distributed data in secured networks to ensure data security
- Needs enforcement, governance, and stringent practices to manage the integrity and consistency of data

Protocol converter

This is a mediatory approach to provide an abstraction for the incoming data of various systems. The protocol converter pattern provides an efficient way to ingest a variety of unstructured data from multiple data sources and different protocols.The message exchanger handles synchronous and asynchronous messages from various protocol and handlers as represented in the following diagram. It performs various mediator functions, such as file handling, web services message handling, stream handling, serialization,[4] and so on:

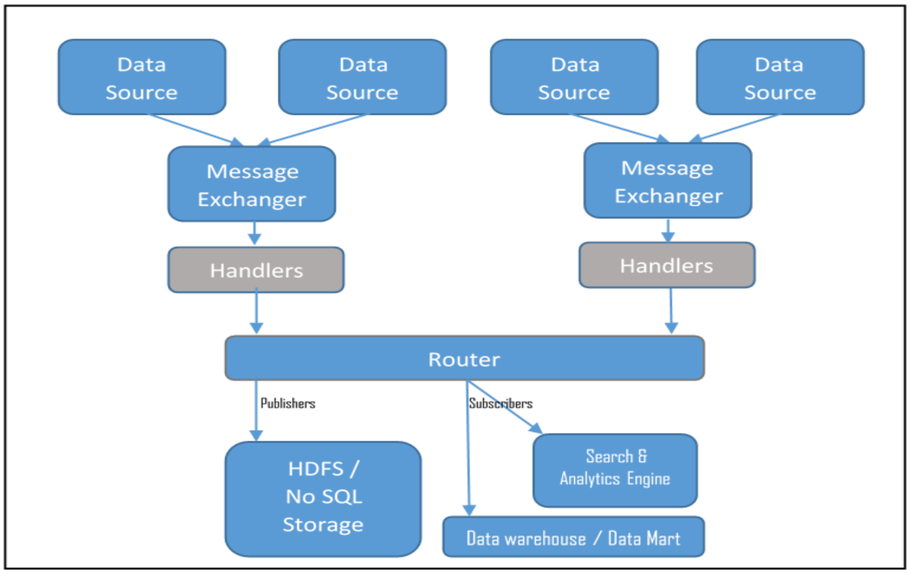
ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

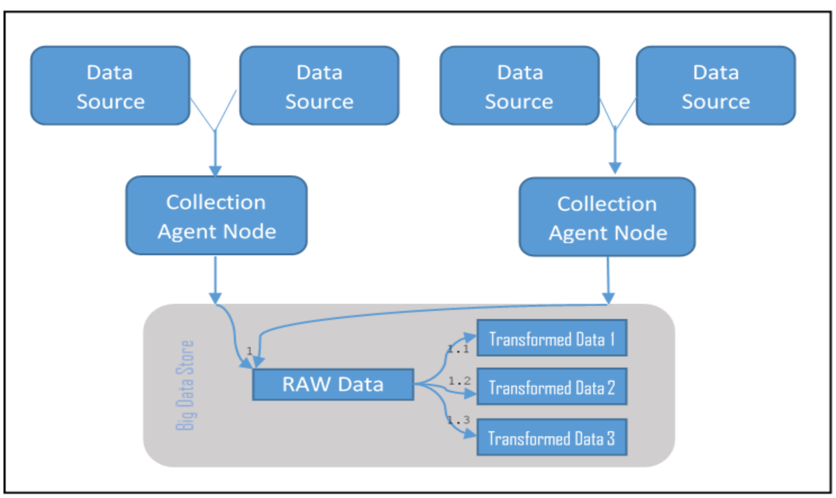
Vol. 7, Issue 8, August 2018



In the protocol converter pattern, the ingestion layer holds responsibilities such as identifying the various channels of incoming events, determining incoming data structures, providing mediated service for multiple protocols into suitable sinks, providing one standard way of representing incoming messages, providing handlers to manage various request types, and providing abstraction from the incoming protocol layers.[5]

Just-In-Time (JIT) transformation pattern

The JIT transformation pattern is the best fit in situations where raw data needs to be preloaded in the data stores before the transformation and processing can happen. In this kind of business case, this pattern runs independent preprocessing batch jobs that clean, validate, corelate, and transform, and then store the transformed information into the same data store (HDFS/NoSQL); that is, it can coexist with the raw data:



ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 8, August 2018

The preceding diagram depicts the datastore with raw data storage along with transformed datasets. Please note that the data enricher of the multi-data source pattern is absent in this pattern and more than one batch job can run in parallel to transform the data as required in the big data storage, such as HDFS, Mongo DB, and so on.Real time streaming pattern-Most modern businesses need continuous and real-time processing of unstructured data for their enterprise big data applications.Real-time streaming implementations need to have the following characteristics:

- Minimize latency by using large in-memory
- Event processors are atomic and independent of each other and so are easily scalable
- Provide API for parsing the real-time information
- Independent deployable script for any node and no centralized master node implementation[6]

The real-time streaming pattern suggests introducing an optimum number of event processing nodes to consume different input data from the various data sources and introducing listeners to process the generated events (from event processing nodes) in the event processing engine.Event processing engines (event processors) have a sizeable in-memory capacity, and the event processors get triggered by a specific event. The trigger or alert is responsible for publishing the results of the in-memory big data analytics to the enterprise business process engines and, in turn, get redirected to various publishing channels (mobile, CIO dashboards, and so on).Big data Workload patterns help to address data workload challenges associated with different domains and business cases efficiently. The big data design pattern manifests itself in the solution construct, and so the workload challenges can be mapped with the right architectural constructs and thus service the workload.Workload design patterns help to simplify and decompose the business use cases into workloads. Then those workloads can be methodically mapped to the various building blocks of the big data solution architecture.Data storage layer is responsible for acquiring all the data that are gathered from various data sources and it is also liable for converting (if needed) the collected data to a format that can be analyzed. The following sections discuss more on data storage layer patterns.[7]

ACID versus BASE versus CAP-Traditional RDBMS follows atomicity, consistency, isolation, and durability (ACID) to provide reliability for any user of the database. However, searching high volumes of big data and retrieving data from those volumes consumes an enormous amount of time if the storage enforces ACID rules. So, big data follows basically available, soft state, eventually consistent (BASE), a phenomenon for undertaking any search in big data space.Database theory suggests that the NoSQL big database may predominantly satisfy two properties and relax standards on the third, and those properties are consistency, availability, and partition tolerance (CAP).With the ACID, BASE, and CAP paradigms, the big data storage design patterns have gained momentum and purpose. We will look at those patterns in some detail in this section.[8] The patterns are:

- Façade pattern
- NoSQL pattern
- Polyglot pattern

Façade pattern-This pattern provides a way to use existing or traditional existing data warehouses along with big data storage (such as Hadoop). It can act as a façade for the enterprise data warehouses and business intelligence tools.In the façade pattern, the data from the different data sources get aggregated into HDFS before any transformation, or even before loading to the traditional existing data warehouses:

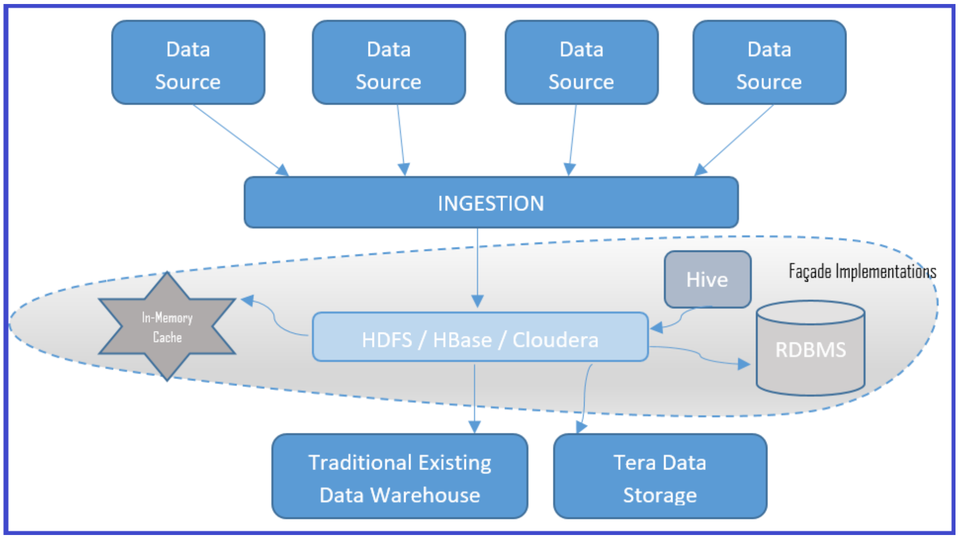
ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 8, August 2018



The façade pattern allows structured data storage even after being ingested to HDFS in the form of structured storage in an RDBMS, or in NoSQL databases, or in a memory cache. The façade pattern ensures reduced data size, as only the necessary data resides in the structured storage, as well as faster access from the storage.
NoSQL pattern-This pattern entails getting NoSQL alternatives in place of traditional RDBMS to facilitate the rapid access and querying of big data. The NoSQL database stores data in a columnar, non-relational style. It can store data on local disks as well as in HDFS, as it is HDFS aware. Thus, data can be distributed across data nodes and fetched very quickly.[9]

Let's look at four types of NoSQL databases in brief:

- Column-oriented DBMS: Simply called a columnar store or big table data store, it has a massive number of columns for each tuple. Each column has a column key. Column family qualifiers represent related columns so that the columns and the qualifiers are retrievable, as each column has a column key as well. These data stores are suitable for fast writes.
- Key-value pair database: A key-value database is a data store that, when presented with a simple string (key), returns an arbitrarily large data (value). The key is bound to the value until it gets a new value assigned into or from a database. The key-value data store does not need to have a query language. It provides a way to add and remove key-value pairs. A key-value store is a dictionary kind of data store, where it has a list of words and each word represents one or more definitions.
- Graph database: This is a representation of a system that contains a sequence of nodes and relationships that creates a graph when combined. A graph represents three data fields: nodes, relationships, and properties. Some types of graph store are referred to as triple stores because of their node-relationship-node structure. You may be familiar with applications that provide evaluations of similar or likely characteristics as part of the search (for example, a user bought this item also bought… is a good illustration of graph store implementations).[10]
- Document database: We can represent a graph data store as a tree structure. Document trees have a single root element or sometimes even multiple root elements as well. Note that there is a sequence of branches, sub-branches, and values beneath the root element. Each branch can have an expression or relative path to determine the traversal path from the origin node (root) and to any given branch, sub-branch, or value. Each branch may

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

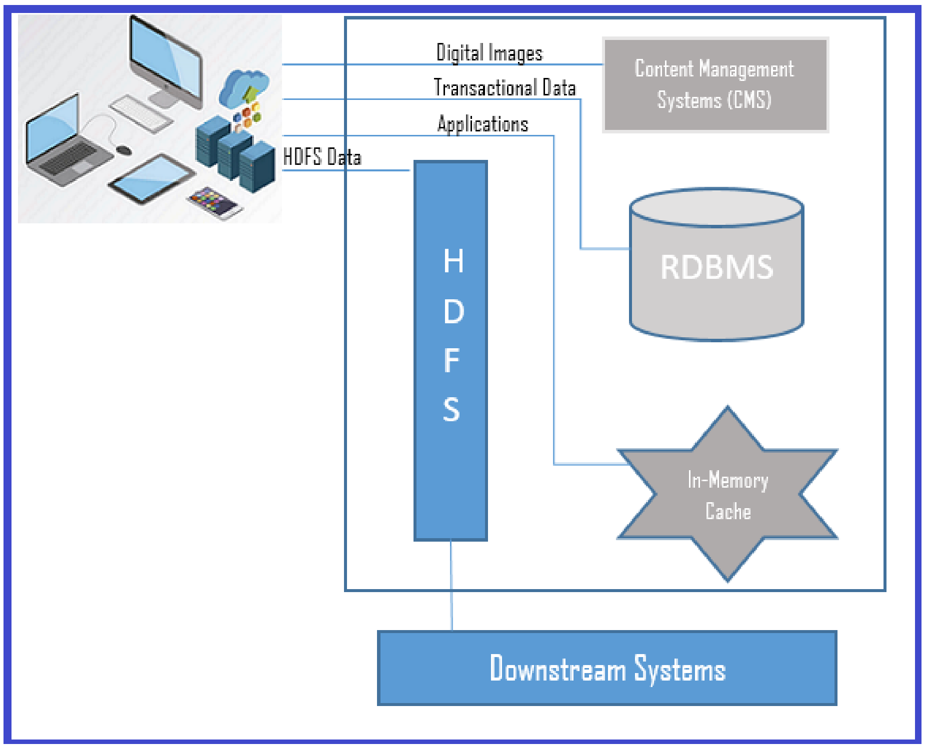
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 8, August 2018

have a value associated with that branch. Sometimes the existence of a branch of the tree has a specific meaning, and sometimes a branch must have a given value to be interpreted correctly.

Polyglot pattern-Traditional (RDBMS) and multiple storage types (files, CMS, and so on) coexist with big data types (NoSQL/HDFS) to solve business problems.Most modern business cases need the coexistence of legacy databases. At the same time, they would need to adopt the latest big data techniques as well. Replacing the entire system is not viable and is also impractical.[11] The polyglot pattern provides an efficient way to combine and use multiple types of storage mechanisms, such as Hadoop, and RDBMS. Big data appliances coexist in a storage solution:



The preceding diagram represents the polyglot pattern way of storing data in different storage types, such as RDBMS, key-value stores, NoSQL database, CMS systems, and so on. Unlike the traditional way of storing all the information in one single data source, polyglot facilitates any data coming from all applications across multiple sources (RDBMS, CMS, Hadoop, and so on) into different storage mechanisms, such as in-memory, RDBMS, HDFS, CMS, and so on.[12]

Data access layer-Data access in traditional databases involves JDBC connections and HTTP access for documents. However, in big data, the data access with conventional method does take too much time to fetch even with cache implementations, as the volume of the data is so high.

So we need a mechanism to fetch the data efficiently and quickly, with a reduced development life cycle, lower maintenance cost, and so on.Data access patterns mainly focus on accessing big data resources of two primary types:

- End-to-end user-driven API (access through simple queries)
- Developer API (access provision through API methods)

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

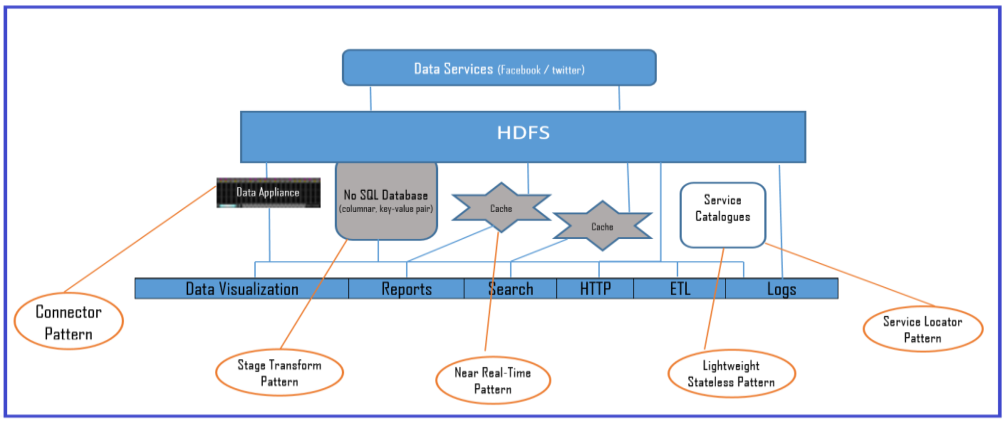
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

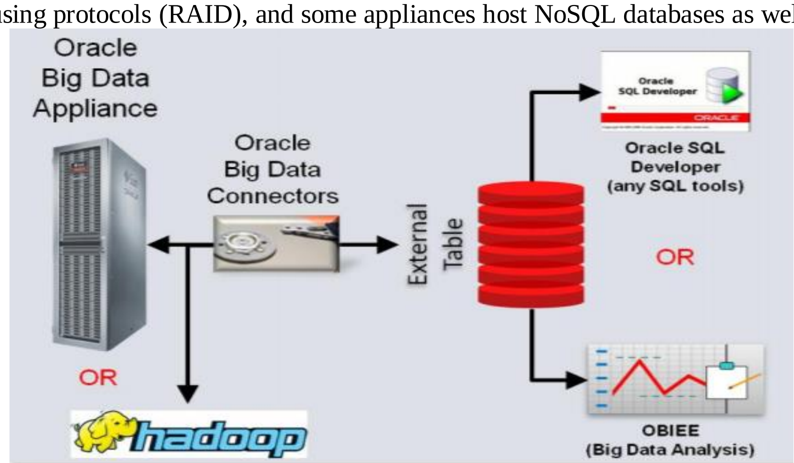
Vol. 7, Issue 8, August 2018

In this section, we will discuss the following data access patterns that held efficient data access, improved performance, reduced development life cycles, and low maintenance costs for broader data access:

- Connector pattern
- Lightweight stateless pattern
- Service locator pattern
- Near real-time pattern
- Stage transform pattern[13]



The preceding diagram represents the big data architecture layouts where the big data access patterns help data access. We discuss the whole of that mechanism in detail in the following sections.
Connector pattern-The developer API approach entails fast data transfer and data access services through APIs. It creates optimized data sets for efficient loading and analysis. Some of the big data appliances abstract data in NoSQL DBs even though the underlying data is in HDFS, or a custom implementation of a filesystem so that the data access is very efficient and fast.The connector pattern entails providing developer API and SQL like query language to access the data and so gain significantly reduced development time. As we saw in the earlier diagram, big data appliances come with connector pattern implementation. The big data appliance itself is a complete big data ecosystem and supports virtualization, redundancy, replication using protocols (RAID), and some appliances host NoSQL databases as well.[3]



ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

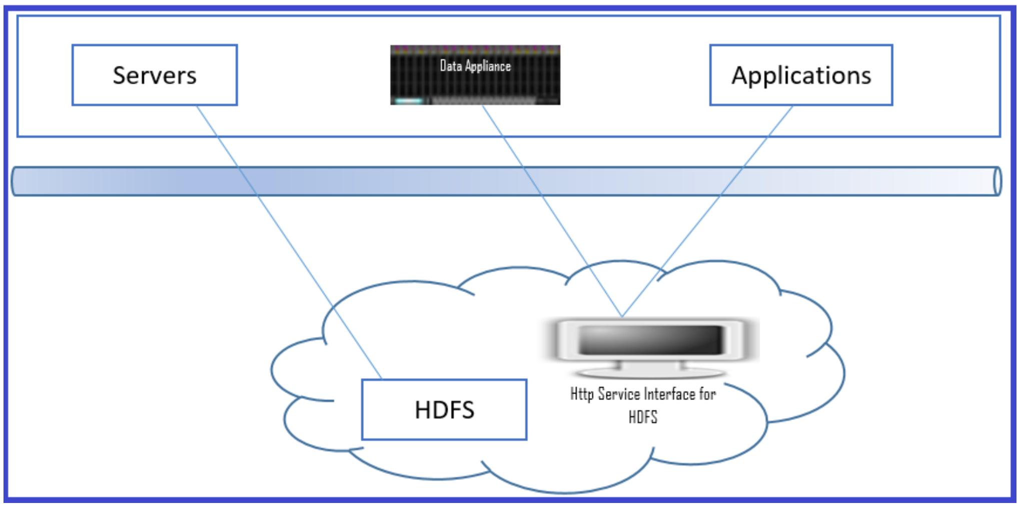
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 8, August 2018

The preceding diagram shows a sample connector implementation for Oracle big data appliances. The data connector can connect to Hadoop and the big data appliance as well. It is an example of a custom implementation that we described earlier to facilitate faster data access with less development time.Lightweight stateless pattern-This pattern entails providing data access through web services, and so it is independent of platform or language implementations. The data is fetched through restful HTTP calls, making this pattern the most sought after in cloud deployments. WebHDFS and HttpFS are examples of lightweight stateless pattern implementation for HDFS HTTP access. It uses the HTTP REST protocol. The HDFS system exposes the REST API (web services) for consumers who analyze big data. This pattern reduces the cost of ownership (pay-as-you-go) for the enterprise, as the implementations can be part of an integration Platform as a Service (iPaaS):



The preceding diagram depicts a sample implementation for HDFS storage that exposes HTTP access through the HTTP web interface.Near real-time pattern-For any enterprise to implement real-time data access or near real-time data access, the key challenges to be addressed are:

- Rapid determination of data: Ensure rapid determination of data and make swift decisions (within a few seconds, not in minutes) before the data becomes meaningless
- Rapid analysis: Ability to analyze the data in real time and spot anomalies and relate them to business events, provide visualization, and generate alerts at the moment that the data arrived

Some examples of systems that would need real-time data analysis are:

- Radar systems
- Customer services applications
- ATMs
- Social media platforms
- Intrusion detection systems[5]

Storm and in-memory applications such as Oracle Coherence, Hazelcast IMDG, SAP HANA, TIBCO, Software AG (Terracotta), VMware, and Pivotal GemFire XD are some of the in-memory computing vendor/technology platforms that can implement near real-time data access pattern applications:

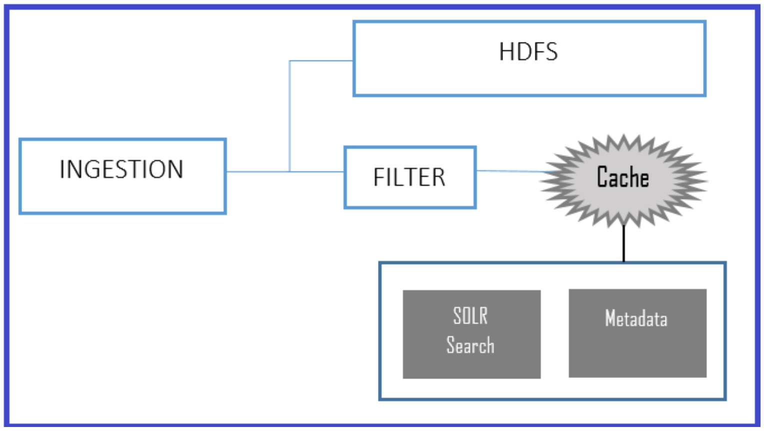
ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

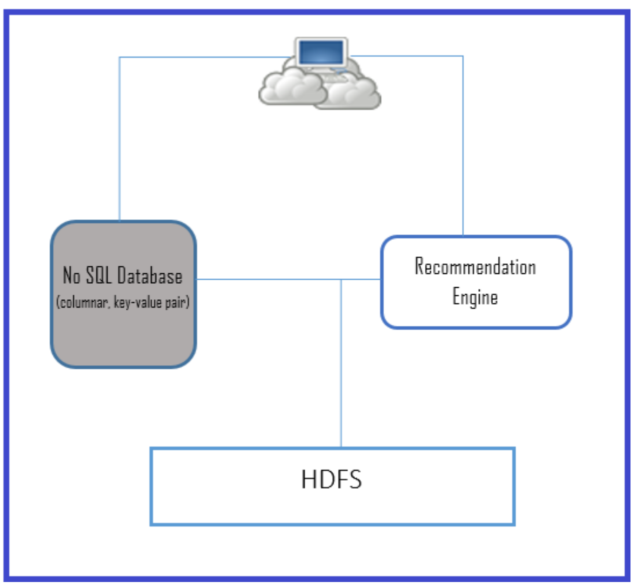
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 8, August 2018



As shown in the preceding diagram, with multi-cache implementation at the ingestion phase, and with filtered, sorted data in multiple storage destinations (here one of the destinations is a cache), one can achieve near real-time access. The cache can be of a NoSQL database, or it can be any in-memory implementations tool, as mentioned earlier. The preceding diagram depicts a typical implementation of a log search with SOLR as a search engine.Stage transform pattern-In the big data world, a massive volume of data can get into the data store. However, all of the data is not required or meaningful in every business case[7]. The stage transform pattern provides a mechanism for reducing the data scanned and fetches only relevant data.HDFS has raw data and business-specific data in a NoSQL database that can provide application-oriented structures and fetch only the relevant data in the required format:



Combining the stage transform pattern and the NoSQL pattern is the recommended approach in cases where a reduced data scan is the primary requirement. The preceding diagram depicts one such case for a recommendation engine where we need a

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 8, August 2018

significant reduction in the amount of data scanned for an improved customer experience.The implementation of the virtualization of data from HDFS to a NoSQL database, integrated with a big data appliance, is a highly recommended mechanism for rapid or accelerated data fetch.[9]

II.DISCUSSION

Data visualization is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from. The main goal of data visualization is to make it easier to identify patterns, trends and outliers in large data sets. The term is often used interchangeably with others, including information graphics, information visualization and statistical graphics.Data visualization is one of the steps of the data science process, which states that after data has been collected, processed and modeled, it must be visualized for conclusions to be made. Data visualization is also an element of the broader data presentation architecture (DPA) discipline, which aims to identify, locate, manipulate, format and deliver data in the most efficient way possible.Data visualization is important for almost every career. It can be used by teachers to display student test results, by computer scientists exploring advancements in artificial intelligence (AI) or by executives looking to share information with stakeholders. It also plays an important role in big data projects. As businesses accumulated massive collections of data during the early years of the big data trend, they needed a way to get an overview of their data quickly and easily. Visualization tools were a natural fit.Visualization is central to advanced analytics for similar reasons. When a data scientist is writing advanced predictive analytics or machine learning (ML) algorithms, it becomes important to visualize the outputs to monitor results and ensure that models are performing as intended. This is because visualizations of complex algorithms are generally easier to interpret than numerical outputs.Data visualization provides a quick and effective way to communicate information in a universal manner using visual information. The practice can also help businesses identify which factors affect customer behavior; pinpoint areas that need to be improved or need more attention; make data more memorable for stakeholders; understand when and where to place specific products; and predict sales volumes.[11]

Other benefits of data visualization include the following:

- the ability to absorb information quickly, improve insights and make faster decisions;
- an increased understanding of the next steps that must be taken to improve the organization;
- an improved ability to maintain the audience's interest with information they can understand;
- an easy distribution of information that increases the opportunity to share insights with everyone involved;
- eliminate the need for data scientists since data is more accessible and understandable; and
- an increased ability to act on findings quickly and, therefore, achieve success with greater speed and less mistakes.

Data visualization and big data

The increased popularity of big data and data analysis projects have made visualization more important than ever. Companies are increasingly using machine learning to gather massive amounts of data that can be difficult and slow to sort through, comprehend and explain. Visualization offers a means to speed this up and present information to business owners and stakeholders in ways they can understand.Big data visualization often goes beyond the typical techniques used in normal visualization, such as pie charts, histograms and corporate graphs. It instead uses more complex representations, such as heat maps and fever charts. Big data visualization requires powerful computer systems to collect raw data, process it and turn it into graphical representations that humans can use to quickly draw insights.[12]

While big data visualization can be beneficial, it can pose several disadvantages to organizations. They are as follows:

- To get the most out of big data visualization tools, a visualization specialist must be hired. This specialist must be able to identify the best data sets and visualization styles to guarantee organizations are optimizing the use of their data.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 8, August 2018

- Big data visualization projects often require involvement from IT, as well as management, since the visualization of big data requires powerful computer hardware, efficient storage systems and even a move to the cloud.
- The insights provided by big data visualization will only be as accurate as the information being visualized. Therefore, it is essential to have people and processes in place to govern and control the quality of corporate data, metadata and data sources.[13]

Examples of data visualization
In the early days of visualization, the most common visualization technique was using a Microsoft Excel spreadsheet to transform the information into a table, bar graph or pie chart. While these visualization methods are still commonly used, more intricate techniques are now available, including the following:

- infographics
- bubble clouds
- bullet graphs
- heat maps
- fever charts
- time series charts

Some other popular techniques are as follows:
Line charts. This is one of the most basic and common techniques used. Line charts display how variables can change over time.Area charts. This visualization method is a variation of a line chart; it displays multiple values in a time series -- or a sequence of data collected at consecutive, equally spaced points in time.Scatter plots. This technique displays the relationship between two variables. A scatter plot takes the form of an x- and y-axis with dots to represent data points.Treemaps. This method shows hierarchical data in a nested format. The size of the rectangles used for each category is proportional to its percentage of the whole. Treemaps are best used when multiple categories are present, and the goal is to compare different parts of a whole.Population pyramids. This technique uses a stacked bar graph to display the complex social narrative of a population. It is best used when trying to display the distribution of a population.[10]

Common use cases for data visualization include the following:
Sales and marketing. Research from market and consumer data provider Statista estimated $566 billion was spent on digital advertising in 2017 and that number will cross the $700 billion mark by future. Marketing teams must pay close attention to their sources of web traffic and how their web properties generate revenue. Data visualization makes it easy to see how marketing efforts effect traffic trends over time.

Politics. A common use of data visualization in politics is a geographic map that displays the party each state or district voted for.
Healthcare. Healthcare professionals frequently use choropleth maps to visualize important health data. A choropleth map displays divided geographical areas or regions that are assigned a certain color in relation to a numeric variable. Choropleth maps allow professionals to see how a variable, such as the mortality rate of heart disease, changes across specific territories.[11]

Scientists. Scientific visualization, sometimes referred to in shorthand as SciVis, allows scientists and researchers to gain greater insight from their experimental data than ever before.
Finance. Finance professionals must track the performance of their investment decisions when choosing to buy or sell an asset. Candlestick charts are used as trading tools and help finance professionals analyze price movements over time,

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 8, August 2018

displaying important information, such as securities, derivatives, currencies, stocks, bonds and commodities. By analyzing how the price has changed over time, data analysts and finance professionals can detect trends.
Logistics. Shipping companies can use visualization tools to determine the best global shipping routes.[11]
Data scientists and researchers. Visualizations built by data scientists are typically for the scientist's own use, or for presenting the information to a select audience. The visual representations are built using visualization libraries of the chosen programming languages and tools. Data scientists and researchers frequently use open source programming languages -- such as Python -- or proprietary tools designed for complex data analysis. The data visualization performed by these data scientists and researchers helps them understand data sets and identify patterns and trends that would have otherwise gone unnoticed.

The science of data visualization
The science of data visualization comes from an understanding of how humans gather and process information. Daniel Kahn and Amos Tversky collaborated on research that defined two different methods for gathering and processing information.[12]
System 1 focuses on thought processing that is fast, automatic and unconscious. This method is frequently used in day-to-day life and helps accomplish:

- reading the text on a sign;
- solving simple math problems, like 1+1;
- identifying where a sound is coming from;
- riding a bike; and
- determining the difference between colors.

System 2 focuses on slow, logical, calculating and infrequent thought processing. This method is used in one of the following situations:

- reciting a phone number;
- solving complex math problems, like 132 x 154;
- determining the difference in meaning between multiple signs standing side by side; and
- understanding complex social cues.[13]

Data visualization tools and vendors
Data visualization tools can be used in a variety of ways. The most common use today is as a business intelligence (BI) reporting tool. Users can set up visualization tools to generate automatic dashboards that track company performance across key performance indicators (KPIs) and visually interpret the results.The generated images may also include interactive capabilities, enabling users to manipulate them or look more closely into the data for questioning and analysis. Indicators designed to alert users when data has been updated or when predefined conditions occur can also be integrated.Many business departments implement data visualization software to track their own initiatives. For example, a marketing team might implement the software to monitor the performance of an email campaign, tracking metrics like open rate, click-through rate and conversion rate.As data visualization vendors extend the functionality of these tools, they are increasingly being used as front ends for more sophisticated big data environments. In this setting, data visualization software helps data engineers and scientists keep track of data sources and do basic exploratory analysis of data sets prior to or after more detailed advanced analyses.The biggest names in the big data tools marketplace include Microsoft, IBM, SAP and SAS. Some other vendors offer specialized big data visualization software; popular names in this market include Tableau, Qlik and Tibco.While Microsoft Excel continues to be a popular tool for data visualization,[12] others have been created that provide more sophisticated abilities:

- IBM Cognos Analytics
- Qlik Sense and QlikView
- Microsoft Power BI

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 8, August 2018

- Oracle Visual Analyzer
- SAP Lumira
- SAS Visual Analytics
- Tibco Spotfire
- Zoho Analytics
- D3.js
- Jupyter
- MicroStrategy
- Google Charts

III.RESULTS

Linkurious Enterprise makes the complex connected data within your favorite graph database easily accessible and intelligible to any business analyst. Customize the look and feel of the application, implement workflows or alerts, and watch your organization uncover hidden insights and make smarter decisions faster than ever before. The days of chasing information scattered across different tools and tabs are over. With Linkurious Enterprise's rich visualizations, you have a unified and comprehensive view of the entities and relationships that are the most relevant to complete your analysis at scale. So you can easily identify new insights or patterns in complex connected data that would otherwise go unnoticed. With powerful yet intuitive ways to filter or display your data and automate your exploration, you can quickly remove non-relevant information and get to the bottom of your exploration. Both technical and non-technical users alike have the autonomy to easily navigate through their graph data to gather key insights and make more informed decisions, all before lunch. On-premise or in the cloud, Linkurious Enterprise is a no-code graph visualization and analytics tool that can be deployed in a matter of hours rather than months. It is compatible with enterprise security and can be seamlessly integrated with your existing technology through APIs. attern extraction algorithms are enabling insights into the ever-growing amount of today's datasets by translating reoccurring data properties into compact representations. Yet, a practical problem arises: With increasing data volumes and complexity also the number of patterns increases, leaving the analyst with a vast result space.[11] Current algorithmic and especially visualization approaches often fail to answer central overview questions essential for a comprehensive understanding of pattern distributions and support, their quality, and relevance to the analysis task. To address these challenges, we contribute a visual analytics pipeline targeted on the pattern-driven exploration of result spaces in a semi-automatic fashion. Specifically, we combine image feature analysis and unsupervised learning to partition the pattern space into interpretable, coherent chunks, which should be given priority in a subsequent in-depth analysis. In our analysis scenarios, no ground-truth is given. Thus, we employ and evaluate novel quality metrics derived from the distance distributions of our image feature vectors and the derived cluster model to guide the feature selection process. We visualize our results interactively, allowing the user to drill down from overview to detail into the pattern space and demonstrate our techniques in two case studies on Earth observation and biomedical genomic data. The quantification of patterns in visualizations is an active research field with broadly two distinctive approaches: Either pattern measures are computed from the data space or the image space. Image-based quality metrics have the advantage that a direct correspondence to the human perceptual system is imminent. Following this argumentation, we can also assess the limits of our approach: an image-based pattern analysis can only work if the pattern space is clearly defined and distinguishable; i.e., the patterns can be discerned computationally and perceptually (human). Moreover, image-based pattern analysis builds on the assumption that the applied visualization technique can express the pot. complex data patterns. For relational and bivariate data, several research papers have considered this question. For other visualization types a structured pattern space examination remains future research. Another limitation results from the choice of our data analysis machinery. Our quality scores rely on internal quality measures derived from feature vector distances. As, for example, the choice of the dissimilarity calculation has an impact on the result interpretation. More research in this direction needs to be devoted to developing (more) dissimilarity score agnostic approaches.[12]

IV.CONCLUSIONS

Before jumping into the term “Data Visualization”, let’s have a brief discussion on the term “Data Science” because these two terms are interrelated. But how? Let’s understand. So, in simple terms, “Data Science is the science of analyzing raw data using statistics and machine learning techniques with the purpose of drawing conclusions about that information“.

Advantages of Data Visualization
1. Better Agreement: In business, for numerous periods, it happens that we need to look at the exhibitions of two components or two situations. A conventional methodology is to experience the massive information of both the circumstances and afterward examine it. This will clearly take a great deal of time.
2. A Superior Method: It can tackle the difficulty of placing the information of both perspectives into the pictorial structure. This will unquestionably give a superior comprehension of the circumstances. For instance, Google patterns assist us with understanding information identified with top ventures or inquiries in pictorial or graphical structures.
3. Simple Sharing of Data: With the representation of the information, organizations present another arrangement of correspondence. Rather than sharing the cumbersome information, sharing the visual data will draw in and pass on across the data which is more absorbable.
4. Deals Investigation: With the assistance of information representation, a salesman can, without much of a stretch, comprehend the business chart of items. With information perception instruments like warmth maps, he will have the option to comprehend the causes that are pushing the business numbers up just as the reasons that are debasing the business numbers. Information representation helps in understanding the patterns and furthermore, different variables like sorts of clients keen on purchasing, rehash clients, the impact of topography, and so forth. [10]
5. Discovering Relations Between Occasions: A business is influenced by a lot of elements. Finding a relationship between these elements or occasions encourages chiefs to comprehend the issues identified with their business. For instance, the online business market is anything but another thing today. Each time during certain happy seasons, like Christmas or Thanksgiving, the diagrams of online organizations go up. Along these lines, state if an online organization is doing a normal $1 million business in a specific quarter and the business ascends straightaway, at that point they can rapidly discover the occasions compared to it.
6. Investigating Openings and Patterns: With the huge loads of information present, business chiefs can discover the profundity of information in regard to the patterns and openings around them. Utilizing information representation, the specialists can discover examples of the conduct of their clients, subsequently preparing for them to investigate patterns and open doors for business. [13]

REFERENCES

[1] C. Ware, Information Visualization, Perception for Design, Third Edition. Morgan Kaufman. 2012.
[2] C. Alexander, S. Ishikawa, and M. Silverstein, A pattern language. Oxford University Press. 1977
. [3] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. 1995.
[4] S.K.. Card, T.P Moran, and A. Newell, The Psychology of HumanComputer Interaction, London: Lawrence Erlbaum Associates. 1983.
[5] J. R Anderson, M. Matessa, and C. Lebiere,. ACT-R: A theory of higher level cognition and its relation to visual attention. Human-Computer Interaction, 12(4), 439-462. 1997.
[6] P. Pirolli, and S. Card, The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In Proceedings of International Conference on Intelligence Analysis (Vol. 5). 2005.
[7] N. Elmqvist, and J.D. Fekete, Hierarchical Aggregation for Information Visualization: Overview, Techniques,and Design Guidelines. IEEE Transactions on Visualization and Computer Graphics. 16(3) 439-454. 2010.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 8, August 2018

[8] F. van Ham, , and A. Perer. Search, Show Context, Expand on Demand": Supporting Large Graph Exploration with Degree-of-Interest. Visualization and Computer Graphics, IEEE Transactions on 15, no. 6 953-960. 2009.
[9] R. Becker, and W.S. Cleveland, Brushing scatterplots. Technometrics, 29, 127-42.
[10] G.W. Furnas,. Generalized fisheye views. Vol. 17, no. 4. ACM, 1986.
[11] C. Ahlberg, and B. Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays." In Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence, pp. 313-317. ACM, 1994.
[12] R. Amar and J. Stasko. A knowledge task-based framework for design and evaluation of information visualizations. In Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on, pp. 143-150. IEEE, 2004.
[13] D. Jonker, S. Langevin, N. Bozowsky and W. Wright, Aperture: An Open Web 2.0 Visualization Framework, 46th Hawaii International Conference on System Sciences, 201.