CEPIS UPGRADE is the European Journal for the Informatics Professional, published bi-monthly at <http://cepis.org/upgrade>

Publisher
CEPIS UPGRADE is published by CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>), in cooperation with the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática,* <http://www.ati.es/>) and its journal Novática

CEPIS UPGRADE monographs are published jointly with Novática, that publishes them in Spanish (full version printed; summary, abstracts and some articles online)

CEPIS UPGRADE was created in October 2000 by CEPIS and was first published by Novática and INFORMATIK/INFORMATIQUE, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies)

CEPIS UPGRADE is the anchor point for UPENET (UPGRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:
- inforewiew, magazine from the Serbian CEPIS society JISA
- Informatica, journal from the Slovenian CEPIS society SDI
- Informatik-Spektrum, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- ITNOW, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- Mondo Digitale, digital journal from the Italian CEPIS society AICA
- Novática, journal from the Spanish CEPIS society ATI
- OCG Journal, journal from the Austrian CEPIS society OCG
- Pliroforiki, journal from the Cyprus CEPIS society CCS
- Tölvumál, journal from the Icelandic CEPIS society ISIP

Editorial correspondence: Llorenç Pagés-Casas <pages@ati.es>
Advertising correspondence: <info@cepis.org>

Subscriptions
If you wish to subscribe to CEPIS UPGRADE please send an email to info@cepis.org with 'Subscribe to UPGRADE' as the subject of the email or follow the link 'Subscribe to UPGRADE' at <http://www.cepis.org/upgrade>

ISSN 1684-5285

Monograph of next issue (October 2011)
"Green ICT"

(The full schedule of **CEPIS UP**GRADE is available at our website)

CEPIS UPGRADE
The European Journal for the Informatics Professional
http://cepis.org/upgrade
Vol. XII, issue No. 3, July 2011

Monograph
Business Intelligence
(published jointly with Novática*)
Guest Editors: *Jorge Fernández-González and Mouhib Alnoukari*

UPENET (UPGRADE European NETwork)

CEPIS NEWS

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

Business Intelligence Solutions: Choosing the Best solution for your Organization

Mahmoud Alnahlawi

With the increased awareness regarding the importance of Business Intelligence (BI), a wide array of platforms and tools have come to existence to answer companies demand. Choosing the right tools depends on the specific needs and goals that an organization is trying to optimize, along with the nature of its data and analysis requirements. In this paper different aspects and goals of the business intelligence architecture are described. The way how the Architecture Trade-off Alternative Method (ATAM) can be used to evaluate different vendors and platforms is presented too.

Keywords: Architecture, Business Intelligence, Data Warehousing, LATAM, Systems Design, Software Evaluation.

"The world's total production of information amounts to about 250 megabytes for each man, woman, and child on earth. It is clear that we are all drowning in a sea of information. The challenge is to learn to swim in that sea rather than drown in it."
Peter Lyman and Hal R. Varian

Author

Mahmoud Alnahlawi is an experienced software architect based in Palo Alto, California, USA. He built scalable data systems that range from petabyte-scale data warehouses for offline processing to near realtime data pipelines. His area of expertise include data modeling, real-time document indexing, relational and NoSql databases, distributed processing and cloud data discovery. He has played many critical roles in Fortune-500 companies as well as small startups to solve data problems related to web click stream analysis, sponsored and display advertising as well as security log management. <alnahlawi@gmail.com>

1 Introduction

Both the amount of data and its processing are growing at a very fast pace. More so, academia and industry are continuously trying to find out new ways to harness the power of the data and use it to derive meaning insights that drive and direct innovation in different areas. The uptrend of both phenomena have triggered a proliferation of platforms and tools that aim to solve the problem of storing, processing and presenting the data to facilitate the innovation.

Although well accepted architectures of building a robust data warehousing and business intelligence solution have been around for a long time, having vast solutions on the market requires diligent, systematic and thorough analysis of existing products along with their respective tradeoffs.

The first main question to be asked when looking into building a new business intelligence project is: Should the platforms and tools be built it in house or should off-the-shelf products be used? Many organizations underestimate what it takes to build an end-to-end Business Intelligence Solution. It seems as if building everything in house will always be cheaper than adopting external ones. They end up wasting many cycles of time and resources or even worse, cancel the project. A company needs to clearly articulate the gaps in the existing products that prevent it from adopting them, along with detailed plans of how the gaps are going to be closed.

One of the major dimensions is the budget that can be allocated for the project and the cost of the end-to-end solution. Prices of products vary widely as well as the pricing model. Some companies charge per license seat, others per CPU. Some have unlimited usage for an annual fee or a one time payment.

The other major factor is the reporting requirements that the solution needs to address. Is canned reporting sufficient or do analysts need ad-hoc and interactive reporting slicing and dicing the data by different dimensions? What is the skill set of the users of the product? Are they proficient in SQL and Excel or do they need easy and intuitive user interfaces to work with?

The size and type of data to be analyzed also plays a big role in determining the best option. If the data is very large, it is crucial to pick a tool that can support parallel execution for both Extract-Transform-Load (ETL) and reporting. A slow performing system discourages users and results in overall failure of the project. Scalability is also very important. Picking a solution that not only meets the organization's current needs but also can handle projected data growth and increase usage in a timely manner.

In this paper, a high level overview of different areas needed for building a business intelligence solution is first given, followed by an overview of the Architecture Tradeoff Analysis Method (ATAM) developed by the Software En-

> “ Both the amount of data and its processing are growing at a very fast pace ”

gineering Institute at the Carnegie Mellon University, USA. Next, important quality attributes needed for building a solid business intelligence systems are given. Lastly a sample Utility Tree for a business intelligence system is created to help organizations make the proper platform or vendor decision that meets their goals and requirements.

2 Background

The background section of this paper is broken up into two sub-sections. The first describes an over all architecture for building data warehousing and business intelligence solutions; and the second focuses on describing ATAM.

2.1 Data Warehousing and Business Intelligence Architecture

Ralph Kimball is one of the original architects of data warehousing and business intelligence systems. He described a high level data warehousing and business intelligence architecture which contains three main areas: Operational Source Systems, Data Staging Area, and the Data Presentation Area. Below is an overview of each area.

2.1.1 Operational Source Systems

Rather than being a part of the warehousing and business intelligence system, an operational source system is the input to the warehouse. Often in the initial stages of the design and requirement gathering phases go into assessing and understanding source systems. Two major activities are needed with respect to source systems. The first activity is performing gap analysis on the source system to determine whether all requirements can be filled by it. The second is detail data profiling which is needed to detect possible data quality issue and give requirements to the detail ETL design.

2.1.2 Data Staging Area

The data staging area is where most of the development time and QA stages are typically spent. The staging area is where the ETL (i.e. extract, transform, and load) is completed. The best analogy to the staging area is a closed kitchen where only experienced chefs are allowed to enter. They prepare and cook the data before it is served to the dining area- presentation area. In the staging area, data is first extracted from the source systems. Source systems can have a variety of interfaces such as data hosted in relational database systems or log files generated by a web server. The extraction process may also involve a complex collection system which can collect data from thousands of machines in geographically distributed locations. Once the data is finally in the staging area, different types of transformations are applied to it. Such transformations include cleansing, where erroneous data is detected and possibly corrected; integration, in which desperate data sources are joined together to give an end to end perspective on the data; and aggregation, where the data is summarized and grouped in different ways to facilitate analysis. The staging area is usually a very complex and dynamic environment. It is imperative that it is available, reliable and operable.

2.1.3 Data Presentation Area

After the data has been cleansed, transformed and integrated, it is finally loaded into the data presentation area. The presentation area is analogous to the dining area in the restaurant metaphor that was used above. Data in the presentation area will be accessed in many different ways and by different types of users. A good presentation area may and often does contain many sub-systems that are specialized for different types of users. It services product managers and business owners interested in the Key Performance Indicators (KPIs) of their products. It is where the scientists go to mine the data for interesting and insightful trends. The presentation area may need to handle requests that are expected to return in less than a minute, to queries that can run for hours processing very large and detailed data.

Additionally, different from the staging area, the presentation area is not a closed environment, The presentation area needs to be able to handle different access roles and ensures that the data hosted within is protected and only allowed users can get access to protected information.

The presentation area requires other types of data management as well such as Retention Management, Discovery, Online Analytical Processing (OLAP), Reporting and Visualization tools.

2.2 ATAM – Architecture Tradeoff Analysis Method

The ATAM method shows how well an alternative satisfies different business requirements and how business requirements impact each other. The ATAM method requires a well documented component level architecture along with well defined business requirements.

Business requirements are represented in terms of Qual-

> “ There is a a proliferation of platforms and tools that aim to solve the problem of storing, processing and presenting the data to facilitate the innovation ”

> “Should the platforms and tools be built it in house or should off-the-shelf products be used?”

ity Attributes – things that stakeholders of the product care most about. Quality Attributes are represented in what is called an Utility Tree. An Utility Tree is defined as a hierarchical, tree structure with general broad categories at the first level. Each category is then divided into sub-categories. At the lowest level of the tree are the scenarios. Scenarios represent specific requirements of the architecture that has to be detailed, unambiguous and measurable. Describing Quality Attributes in terms of scenarios is essential since they eliminate ambiguity and give concrete requirements for the development team and test cases of the quality assurance team. A scenario consists of six parts: 1. source, 2. stimulus, 3. environment, 4. artifact, 5. response and 6. response measure. Source describes who generated the stimulus, whether it is System A, User X or bug 123. Stimulus is an event or a condition that needs to be handled by the system. Environment is the state of the system during which the stimulus takes place. Artifact is the part of the system that was impacted by the stimulus. Response is desired behavior of the system after or during the stimulus. And finally, response measure is way to test that the desired response actually took place. For example, consider the following scenario under the performance Quality Attribute of a reporting system:

- Source of stimulus = Users
- Stimulus = 100 users login simultaneously
- Environment = new data is being loaded into the reporting system database
- Artifact = read load of the database is increased
- Response = system should handle load gracefully
- Response Measure: Each report should finish and data returned to the requester within five minutes of report request time.

Once the Utility Tree is constructed, the scenarios are prioritized by the architect using feedback from all stakeholders of the project. It is important to include users, developers, testers and system operators in the process of assigning priorities to ensure that all viewpoints are represented. Once prioritization is complete, the architect then documents how well each alternative, such as Vendor A, handles each scenario, such as automatic error recovery or failover. After this is complete, each scenario will have a priority a score for each alternative.

Once the Quality Attributes have been defined, a mapping between the different scenarios and the different architectural decision or alternative is constructed. Essentially, each architectural decision, such as using platform A, is given a a rank for how well it handles the scenario.

Once the prioritization and the assessment phase is done, analysis of the architecture is ready to take place. The analysis phase identifies for each scenario and each alternative a set of sensitivity points, tradeoffs points, risks and non-risks. Sensitivity points are Quality Attributes or scenarios that are impacted by choosing one alternative over another. Tradeoff points that are doing well on by an alternative implies doing poor on another scenario. Risks are tradeoff points that may result in an undesirable behavior based on the scenarios and non-risks are tradeoff points that are deemed safe with respect to scenarios.

Detailed documentation and examples of the Architecture Tradeoff Analysis Method ATAM, as well as alternatives to it such as Cost Benefit Analysis Method(CBAM) and Microsoft’s Lightweight Architecture Alternative Assessment Method (LAAAM) can be found online at the Software Engineering Institute website, <http:// www.sei.cmu.edu>.

3 Choosing the Proper Solution for your Organization

In this section we will go some of the main quality attributes that should be considered to shape your decision and guide you towards the right solution for your organization. You should take the quality attributes listed in this section and come up with sub-categories and scenarios that are applicable to your organization’s need and requirements. Once that is done, we give a sample Utility Tree that can be used for evaluating how well different vendors meet the different scenarios and aid in making the optimal decision.

3.1 Availability

Availability determines how system deals with failures and has a big impact on the architecture of the system and it’s associated cost and time. The first and largest availability question is what count of Business Continuity Plan (BCP) does your warehouse require. Business continuity planning requirements specify how the system should react in the

> “The size and type of data to be analyzed also plays a big role in determining the best option”

event of large outages such as an earthquake destroying an entire data center. The requirements, and therefore the architecture, vary by organization. Some require ZERO downtime especially if the business intelligence solution is used by production customer facing systems. Other analytical or internal facing systems have more relaxed recovery requirements than can be multiple days. Different vendors have built BCP solutions than range from automatic backup to tape to real-time replication of data over TCP networks.

Typically availability is described by nines – 90, 99, 99,9 etc. Companies with high availability requirements target five nine availability goals, meaning the system has to be up and functional 99.999% of the time allowing for only 5.26 minutes of downtime per year. Going to six nines, allows for only 31.5 seconds of downtime per year! For a system to obtain such high availability numbers, there are minimum requirements that need to be met. The solution should have no single points of failure (SPOF) which is a component whose failure result in the failure of the entire system. There should be ability to provide live updates – updates while the system is up and running. The system needs to be fault tolerant, which is the ability of the system to operate gracefully, with possible degradation of service but not loss of it, in the event of failure of one ore more of its components.

3.2 Scalability

Scalability is one of the major differentiators amongst vendors (along with Performance). Scalability measures the ability of the system to handle large amount of work without performance degradation. Scalability can be defined for sub-systems of the business intelligence architecture and can have different measure of requirements. For example, the presentation tools and OLAP solution need to scale for a certain number of concurrent users. It also needs to scale for certain number of predefined reports or aggregations. The storage sub-system of the presentation area need to scale for a given number of bytes, certain number of rows per table and certain number of concurrent queries.

Different ETL and data storage and processing platforms have different solutions for scalability. Its important to assess how the vendor techniques meet the requirements of your organization. There are two different techniques for handling scalability, vertical or horizontal scale. Vertical scale is the ability to add more resources to a single machine such as increasing memory or CPU. Horizontal scale means adding more machines to a distributed system. Horizontal scale allows for using commodity and cheaper machines instead of specialized and expensive ones. Horizontally scalable systems require shared storage with high throughput access to the data. Tradeoffs between horizontal and vertical scaling models involve high-cost-of scale for hardware vs. high number of machines which might be hardware to manage and operate. Also, larger number of machines consume more power and more data center real estate.

Additionally, data bases have a different techniques that facilitate both vertical and horizontal scaling. A common technique that is supported by almost all vendors is partitioning. Vendors may differ however by the maximum number of allowed partitions. Also, they may offer different partitioning schemes such as range or hash partitioning. Databases have also different threading implementations that allow them to handle vertical scale differently.

3.3 Performance

Performance is extremely important to the success of the business intelligence project. Yet, performance is a very vague and ambiguous term. It relates to many aspects of the system. Scenarios are most helpful for performance requirements. Make sure to specify exact user cases and what is the expected and acceptable response from the system.

One measure of performance is latency – the total time taken by the system from when a request is made until the response is received by the requester. Latency cuts across all aspects of the presentation area. For example, a user of the system logs in to the reporting portal and runs a report. There is latency between the machine of the user and the server hosting the application portal. The application server then typically issues a query against the database system hosting the data. The database server has a given latency for responding to the query which is made up of many smaller latencies, such as the latency to read a block from disk, network latency between different machines in a distributed system or latency by the CPU to add two integers. Scenarios are defined at the perceived performance level which is the visible latency to the user independent of all internal latencies of the system. The architect ensures that the proposed solutions meets the latency scenarios empirically by building different prototypes or proof of concepts.

Throughput is another aspect of performance and it's measured in things per second. It states how many operations, requests, records or queries per second a system can handle. The Transaction Processing Performance Counsel defines a set of performance benchmarks that are vendor independent and publishes performance number of various platforms. It is important to understand the different benchmarks and how vendors being considered for the Business

> “The ATAM method shows how well an alternative satisfies different business requirements and how business requirements impact each other”

> “An Utility Tree is defined as a hierarchical, tree structure with general broad categories at the first level”

Intelligence solution performance on the benchmark.

Different vendors also have varying methods and techniques for enhancing performance of queries. There are different indexing techniques such as b-tree index or bitmap index which is very suitable for a dimensional model design typically used for implementing data warehouse and decision support systems. Other techniques involve passing hints in the SQL statement that tells the query engine the degree of parallelism to use for executing a given query or the join algorithm that is most suitable for the data. Partitioning is also a tool for increasing performance of the data warehouse and is used to prune data and only scan partitions that satisfy the criteria of the query dramatically decreasing the amount of I/O the system has to do.

Some vendors store the data in column oriented fashion, columnar databases, that are essentially a way of vertically partitioning the data. Columnar oriented databases increase performance by only scanning columns that are needed for the execution of the query, either projected or included in the where clause of the query. They also have the advantage of better compressing the data given that columns contain similar values in closer proximity to each other which results in better compression.

Lastly, some vendors rely on proprietor hardware to enhance query performance. Some relational operations are pushed down to the hardware layer resulting in much better performance. Such techniques include pushing filters to hardware so that disk controllers only return data that satisfy a where clause.

3.4 Operability

After a business architecture solution has been built and push to production it lives there for a long time. Most data warehouses have teams dedicated service engineering and database administration teams working tirelessly to ensure that the system is meeting its availability and performance requirements. A well designed system is one that treats operability features as first class citizens. Everything has to be automated, monitored, self-healing and self configuring.

The staging area of the solution is where most of the data processing happens. Therefore, great attention needs to be paid for the operability of the stating area. A very important component to the operability of the staging area is a work-flow management system. Work-flow managers allow developers to express ETL processing as an acyclic direct graph where nodes are processing jobs and edges are dependencies. Advantages of such modeling has enormous benefits to the operability of the ETL system. They typically come with a graphical user interface that allows the operator to inspect the progress, or lack of progress, of the ETL pipeline. They also ensure that processing jobs run in the correct order and that in the case of failures only subset of the pipeline is re-executed.

Another important aspect is alerting ability. Alerting should involve complex event processing that ensures that the right amount of alerts are being sent. Over alerting results in thrashing of operator and possibly loss of important alerts. Different vendors allow for different types of alerting such as paging, e-mailing, integration with ticketing systems or graphical user interfaces. They also allow for different severity levels of alerts such as Info or Critical. They monitor the application, the platform, the different services and the hardware of the end to end solution.

Operability of the presentation area is just as important as operability of staging area. There are numerous jobs constantly running in the staging area. Data needs to be backed up, retention policies has to be applied, indexes need to built and rebuilt. Cube in the OLAP system should automatically be triggered for reprocessing.

An advantage of using one vendor for the Staging and Presentation area is integrated monitoring and work-flow solution. Typically more mature vendors have an end to end solution with integrated monitoring and work-flow system.

One last important aspect to keep in mind while reviewing different vendors for operability is related to Quality Attributes and how well they handle rolling upgrades – specially in a distributed environment. Rolling upgrades are the ability to push new software versions to the production environment without having to bring down the system. Distributed systems have the added complication of ensuring that all components of the system are running compatible and consistent versions.

3.5. Time to Market

Time to market is very important factor for determining the best business intelligence (BI) strategy to follow. Most of the time, in addition to cost, time to market is the barrier for implementing a BI solution in house.

Since most of the development and testing efforts are in the staging area of the business intelligence solution, time to market features have to be carefully evaluated for ETL vendors. Many ETL vendors offer features that allow for rapid development of ETL processing. Such features include an extensive library of transformation and processing nodes. They give the ability to compose complex data pipeline by chaining together pre-built processing components, and allowing for description of metadata based ETL using logical mappings of attributes and transformations.

Many vendors also allow for flexible and automatic schema evolution and metadata driven ETL where new col-

Category	Sub Category	Scenario
Availability	Business Continuity	*Source:* Nature *Stimulus:* Earthquake destroys data center *Environment:* Typical load *Artifact:* entire system is destroyed *Response:* • No data loss • Recover within 2 hours *Response Measure:* simulate failure, switch to new geographical location, run report on old system and new system
Availability	Fail Over	*Source:* Computer machine *Stimulus:* An ETL processing machine loses power during processing *Environment:* ETL job in progress *Artifact:* Particular job fails and intermediate data is in inconsistent state *Response:* System should recover automatically *Response Measure:* manually shut down down one of the ETL processing machines. ETL job should recover with no manual intervention
Scalability	Data Size	*Source:* Users of website *Stimulus:* a new feature on website increases page view to 100 million in on hour *Environment:* ETL system is running at 80% of its capacity *Artifact:* Double the number of rows in the input web server logs *Response:* add new hardware results in no changes to response ETL finish time *Response Measure:* Duration time of ETL processing jobs
Scalability	Concurrent Queries	*Source:* Product managers *Stimulus:* Due to a new product launch, all product managers are running the 20 product managers are running the same report at the same time *Environment:* ETL load has finished for the day *Artifact:* Load is increased on the OLAP tool as well as the DBMS *Response:* Only 20% degradation in response time *Response Measure:* Run 20 simultaneous reports and measure run time
Performance	Query Response Time	*Source:* User of reporting system *Stimulus:* A user query asks for one day of data to be reported on *Environment:* Normal conditions *Artifact:* Query sent to DBMS for processing *Response:* Only required horizontal application is processed *Response Measure:* See execution plan and compare to running time of query that accesses all partitions
Performance	Query Response Time	*Source:* User of reporting system *Stimulus:* A user query asks an aggregation that only uses subset of columns *Environment:* Normal conditions *Artifact:* Query sent to DBMS for processing *Response:* Only required columns are scanned and aggregated *Response Measure:* See execution plan and compare to running time of query that accesses all columns
Operability	Terminal failure	*Source:* Un-handled error condition in ETL job *Stimulus:* A job in ETL pipeline has failed *Environment:* ETL cleansing and transformation stage *Artifact:* ETL completely stopped *Response:* Error is reported on monitoring console, operator is alerted via a pager, problem is manually rectified, operators resumes work-flow from point of failure *Response Measure:* Simulate failure in processing job by removing input data in the middle of processing and measure the end to end time it takes to resume the pipeline
Operability	Upgrades	*Source:* Service engineering team *Stimulus:* A new ETL software version needs to be rolled out to production *Environment:* ETL jobs are running

Table 1 (Part 1 of 2): Example of Utility Tree including Scenarios.

Category	Sub Category	Scenario
Time to Market	Flexibility	*Source:* Upstream changes *Stimulus:* A new pass through column is added to the web logs *Environment:* Normal conditions *Artifact:* Input schema changed *Response:* Output Schema changed with the additional column *Response Measure:* Amount of time spent developing, testing and deploying new software
Time to Market	Modifiability	*Source:* Product manager *Stimulus:* A change to the the transformation applied to one of the columns *Environment:* Normal conditions *Artifact:* ETL code needs to be modified *Response:* New code is deployed *Response Measure:* Amount of time spent developing, testing and deploying new software
Compliance	SOX	*Source:* Government auditors *Stimulus:* SOX Audit *Environment:* Normal conditions *Artifact:* Login and Log out reports requested *Response:* Generate required reports within 2 days and no additional resources *Response Measure:* Time it takes to generate and validate the required reports and the number of people used to work on the task
Compliance	A29	*Source:* Upstream changes *Stimulus:* A new private and personally identifiable information attribute is added to one of the source systems *Environment:* Normal conditions *Artifact:* Additional column is added and additional transformation is needed *Response:* Personal Information is converted to anonymous one and stored in the presentation area *Response Measure:* No personal information stored in presentation area
Data Quality	Error Detection and Correction	*Source:* Upstream data quality issue *Stimulus:* A non-nullable attribute has a null value *Environment:* ETL load in progress *Artifact:* Error detection code is triggered *Response*: Reject malformed record and log it in a separate store *Response Measure:* Verify that the malformed record is rejected and logged
Data Quality	Metric Reporting	*Source:* Upstream data quality issue *Stimulus:* 10 input records were malformed *Environment:* Normal conditions *Artifact:* 10 records are rejected and stored *Response:* Run report on data and see that there are 10 rejected records broken down by reason of rejection *Response Measure:* Simulate input and run report

Table 1 (Part 2 of 2): Example of Utility Tree including Scenarios.

ing where number of partitions can be determined and adjusted dynamically based on the input systems. Some vendors facilitate team based development by supporting integration with source control systems allowing multiple developers to work on the project easily at the same time.

Some features provided by vendors also reduce the time it takes to maintain the ETL application by providing auto-documentation features, custom annotations of different processing jobs, automatically generated lineage report that can be used as a manual for the users of the data as well as new developers and impact assessment of changes to the system such as a data type change for one of the attributes could result in changes to only a subset of the processing jobs.

3.6 Compliance

Organizations have different standards that they need to comply with depending on the nature of the data they posses and the type of analytics they perform on it. For example, business intelligence solutions that are used for revenue recognition and reporting need to adhere to the Sarbanes–Oxley Act of 2002, also known as SOX. SOX compliance applies to publicly traded US companies and is a result of financial scandals affecting companies such and Enron and costing people billions of dollars.

> “The main factors for an organization to make a decision on the best Business Intelligence (BI) strategy to follow: availability, scalability, performance, operability, time to market, compliance, and data quality”

SOX compliance requires companies to document and show the flow of transactions. From a data warehousing perspective, this translates to the ability of extract lineage out of the ETL processing jobs. It also requires detailed reports about user activities such as login/logout events. Every access to the data needs to be documented along with the type of access such as read, write or delete. This is needed to ensure that the data has not been tempered with after it has been published by the ETL process. System events such as startup and shut down or changes to the system time or audit log need to be tracked to ensure that the ETL code has not been changes without proper authorization and approvals. Also tracking of account management and user group changes needs to be tracked to ensure that only authorized users have access to the data with the right permissions.

This requires all components of the warehousing and business intelligence solution to have detailed security and auditing features as well as comprehensive and structured logging to facilitate the generation of required SOX report.

Another form of compliance requirements are requirements for protecting user privacy. This is specially needed by companies that collect user behavior or financial data. The European Privacy Directive, specially A29, requires companies to not retain any user personally identifiable data such as browser cookies, IP address or searches that the user performed on their site. Companies usually handle this by converting the private information to anonymous values that are used to identify an unique anonymous person, instead of a login name, or aggregate the information to an appropriate level such as zip code instead of IP address. Some vendors have some pre-built components that allows such transformations or have the ability for the application developer to plug in their own transformation functions in the form of a UDF – user defined function.

3.7 Data Quality

Data quality measures the consistency and accuracy of the data. It is used to determine how fit the data is to be used for decision making. Data with poor quality is considered worse than no data at all since it leads to the wrong decision making.

It is the responsibility of the staging area to ensure that the quality of the data is of high standards before publishing into the presentation area. And, it is the responsibility of the presentation area to keep data quality metrics and expose them to the users of the data.

Most data quality issues are a result of bad data from the input systems. It is best to deal with the data quality issue at the source. In addition to that, ETL vendors have features that allow the detection of bad data and configurable actions to be taken when encountering it. Options include the ability to reject and log the bad data to be analyzed and possibly corrected offline, the ability to correct or nullify bad data or the ability to halt the ETL process until the data quality issue is investigated by an operator (not recommended). It is important to verify that the ETL vendor of choice meets your data quality issues handling requirements. The ETL system also needs to aggregate the number of data quality issues encountered and publish them with the final datasets to be consumed by different users.

In the presentation area, the BI tools need to show data quality metrics to users. This can be added to all reports as as custom aggregation. Some BI tools also allow users to collaborate in a discussion about the and its quality.

3.8 Sample Business Intelligence Utility Tree

Table 1 is an example Utility Tree, represented in tabular format, for a business intelligence system. It highlights the different important quality attributes and give an example of scenarios.

4 New Trends

Although open source software has been around for a long time, it only recently became used widely as part of BI solutions. Most notably is Hadoop, an Apache based open source java implementation of Map/Reduce framework. Although Hadoop has not yet reached version 1.0, it is being used in over one hundred companies that are listed on the Hadoop page of the Apache website, <http://hadoop.apache.org/>. More so, there are different sub-projects of Hadoop that focus on making it more productive and suitable for business intelligence projects such as PIG, a procedural query and processing language on top of Hadoop. Hive is an SQL implementation on top of Hadoop and Oozie

is a workflow manger for Hadoop based jobs.

Additionally, many companies are using in-house or external cloud computing techniques to process their data. Cloud computing with regard to Business Intelligence solution entails ease of provisioning new hardware resources (scalability and performance), geographical location independence (availability), and automatic and live deployment (Maintainability).

5 Summary

Before discussing alternatives for implementing a Business Intelligence solutions, it is important that the quality attributes are documented and reviewed by all stockholders of the project. Quality attributes serve as a medium of communication across multiple teams. It also helps document and serve as a reference for the rationale and reasons behind the decisions that were made. After building the quality tree, spending time writing detailed scenarios. Get all stakeholders to review them and participate in the prioritization process. Make sure that one person is ultimately responsible for assigning the priorities for scenarios otherwise consensus on priorities maybe impossible to achieve. Spend time researching different technologies available on the market and determine how well they meet the different scenarios. Document your findings, review them and move on to implementation.

Bibliography

- L. Bass, P. Clements, and R. Kazman Rick. Software Architecture in Practice. Second Edition. Addison Wesley, 2003.
- R. Kimball and M. Ross. The data warehouse toolkit: the complete guide to dimensional modeling. Wiley Publishing Inc. 2002.
- R. Kimball and J. Caserta. The data warehouse ETL toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. Wiley Publishing Inc., 2004.
- P. Lyman, and H R. Varian. How Much Information? The Journal of Electronic Publishing, Vol. (6), Number 2, 2000.