The Change from Data Warehousing to Business Intelligence: The Concept of Data Sources' Dynamicity

Prof. D. Saad M. Darwish
Professor of Computer Science
Department of Information Technology
Institute of Graduate Studies and Research - Alexandria University
Saad.darwish@alexu.edu.eg

Dr. Reham Moustafa Essa
Doctor of Computer Science
At the Higher Institute of Management
And Information Technology in Kafr El-Sheikh
Moustafa.reham@yahoo.com

Abstract:

The capacity to collate data and present information efficiently in real time and the ability to make it readily accessible to everyone has been a major catalyst for many organizations to embrace globalization and improve productivity. Companies are requiring more regular in-depth analytical review of the data within their repositories to have a better understanding of their business environment and competitiveness capability. The ability to generate valuable insights based on business data is termed business intelligence. The approaches to generate these insights are commonly the function of business analytics. Business analytics is the use of analysis techniques and decision rules to provide business users with critical insights of the operational and performance characteristics in every aspects of the business. Data sources dynamicity can be classified into data updates, schema and instance changes, and constraint modifications. Current data warehouse systems manage data updates. Yet, they cannot follow schema and instance changes and constraint modifications. This paper analyses schema and instance changes resulted from dynamically changing external data sources. We call for applying the external data source schema changes to a data warehouse.. Furthermore, the use of data warehousing and data mining techniques in business analytics and how they support the use of business intelligence in e-businesses are discussed. The distinction between data warehousing, business analytics and business intelligence is presented. Suggestions on how electronic businesses can leverage on data warehousing to enhance their competitiveness are discussed.

Keywords: Business Intelligence, Data Warehousing, Data Mining, Dynamicity of Data Sources

1- Introduction

Data mining is a collective term used to describe different analysis techniques such as statistics, artificial intelligence and machine learning that are employed to scan huge amounts of data found in the organization's databases or online databases [21]. Its purpose is to identify patterns in the data set. These data sets could come from a single database or could come from integrated data established in a data warehouse. The key here is that one is searching for a pattern or relationship among different data groups [18]. By uncovering patterns in the data set, data mining can predict, associate and cluster events, products, or customers in a more effective manner so that the organization could provide better products or services to the customers or improve the efficiency of their operations [26]

Business analytics is used to describe the entire function of applying skills, technologies, standard practices, and algorithms related to data mining and data collation methods to generate valuable information, usually presented in highly readable format so that managers can make business decisions and to control and manage their business operations. When the business analytics function is efficiently and effectively executed, it may become a core competence for the organization in the form of valuable business intelligence that will support the strategic actions undertaken by the organization [18].

Data mining is usually applied in the back-end of the business analytics function while the front-end of business analytics function consists of executive reporting metrics and collated information. In applying data mining, the business analytics professional act like a data detective, analyzing data to better understand an organization’s current and past situation, predict future outcomes and act effectively. Customers' current and past transaction behaviors in sales, servicing and selection choices can also be scrutinized and grouped either in clusters. Event sequencing and product portfolio determination can also be studied using data mining. [26]

It is well known that Data Warehouses (DWs) are focused on decision support rather than on transaction support, and that they are prevalently characterized by an On-Line Analytical Processing (OLAP) workload. Traditionally, OLAP applications are based on multidimensional modeling that intuitively represents data under the metaphor of a cube whose cells store events that occurred in the business domain. Adopting the multidimensional model for DWs has a two-fold benefit [3]. The approaches to DW design are usually classified in two categories [27]. Data-driven approaches design the DW starting from a detailed analysis of the data sources; user requirements impact on design by allowing the designer to select which chunks of data are relevant for decision making and by determining their structuring according to the multidimensional model [14]. Requirement-driven approaches start from determining the information requirements of end users, and how to map these requirements onto the available data sources is investigated only a posteriori [15,24].

As in distributed databases, in distributed data warehousing a new phase needs to be added to the design method: the one for designing the distribution, from both the architectural and the physical points of view. During architectural design, general decisions will be taken about which distribution paradigm (P2P, federation, grid) better suits the requirements, how to deploy the DW on the infrastructure, which communication protocols to use, etc. On the other hand, the physical point of view mainly addresses how to fragment the DW and how to allocate fragments on the different sites in order to maximize local references to data and to take advantage of the intrinsic parallelism arising from distribution, thus optimizing the overall performance. Indeed, distribution is particularly useful in contexts where new data marts are often added, typically because of company mergers or acquisitions. In this case, the most relevant issue is related to integration of heterogeneous data marts [29].

The data warehouse is intended to combine autonomous and heterogeneous External Data Sources (EDSs). EDSs may differ from proprietary applications and legacy systems to modern relational, object or object relational database systems. They may involve plain text, XML documents, spreadsheets, news wires or multimedia contents. All EDSs usually vary in data models, require different user interfaces, and offer different functionality. Autonomy means that the EDSs autonomously and fully control its data. Heterogeneity indicates that the EDSs use different data models and different user and programming interfaces. Local autonomy and heterogeneity of EDSs mean that they were developed independently and are not aware of the integration issues [28]. An important result of the autonomy of EDSs is that they may evolve in time independently and that they change their data and schemas without being controlled from the global data warehouse level. The current data warehousing systems interact only with the first kind of changes, namely, with the content changes. The content changes of EDSs are detected and propagated to the data warehouse.

Business analytics is a major part of business intelligence. Business analytics is directly aided by data warehousing and business intelligence. Business intelligence is mainly analyzing data and collection of knowledge and applying them to various different methods. However, rare research has investigated the data warehouse refresh under schema changes and constraints changes. Integrating data from evolving EDSs puts more loads on the maintenance and evolution of data warehouse systems. These challenges can be classified into four groups: (1) Modifying data warehouse schema according to data source schema changes. (2) Modifying middleware level according to data source schema changes. (3) Modifying data warehouse content according to data source schema changes. (4) Modifying data warehouse content according to data source semantic changes. Most data warehouse models assumed, that data sources and data warehouse schema are static and that only the data changes. However, this Assumption doesn't hold in the real-world applications. Changes occur frequently both in EDSs and in the data warehouse schema and instance. After such change queries touching data affected by the change begin to yield incorrect results. Contemporary data warehouses are unable to handle such changes, which obstacles their functionality. [26]

This paper is planned as follows. In Section 2 we introduce the definition of ETL architecture and processes. In Section 3, we deliberate the effect of EDSs changes on the data warehouse schema, the data warehouse content, and the development of the middleware under those changes. Section 4 demonstrates the development of the sample data warehouse and its middleware under EDSs changes. In Section 6, we concluded with a summary and a further research agenda. [8]

2- Business Intelligence Concepts

Business Intelligence Concepts Initially, BI was coined as a collective term for data analysis tools. Meanwhile, the understanding broadened towards BI as an encompassment of all components of an integrated decision support infrastructure. In BI systems, data from OLTP is combined with analytical front ends to “present complex and competitive information to planners and decision makers”. A central component of BI systems is the data warehouse (DW), which integrates data OLTP for analytical tasks. From the managerial approach, BI is seen as a process in which data from within and out the organization are consolidated and integrated in order to generate information that would facilitate quick and effective decision-making [12]. The role of BI here is to create an informational environment and process by which operational data gathered from transactional systems and external sources can be analyzed and to reveal the “strategic” business dimensions. Figure1 illustrates the two main components of BI applications, and their relation with operational applications.

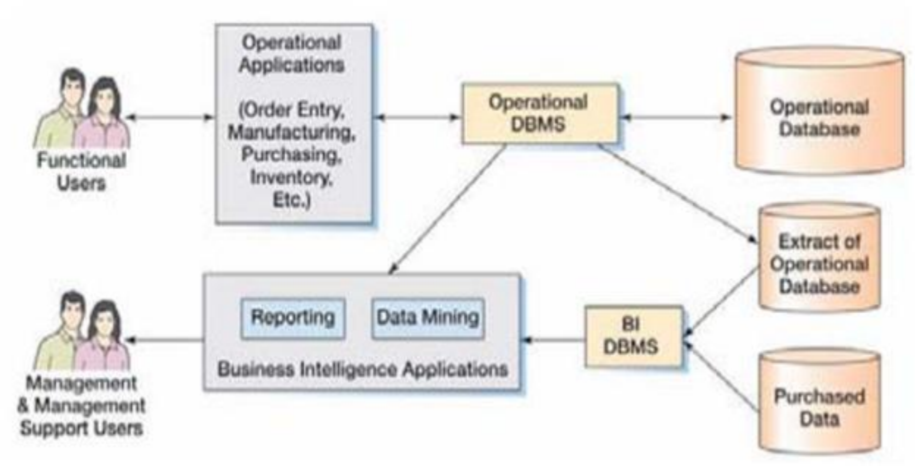

Figure1: Relationships of Operational and Business Intelligence Applications (Source: Kroenke, 2006) [34]

Intelligence means reducing a huge volume of data into knowledge through a process of filtering, analyzing and reporting information. The technological approach presents BI as a set of tools that supports the storage and analysis of information. Basically, the term business intelligence has two different meanings when related to intelligence. The first is the human intelligence or the capacity of a common brain applied to business affairs. Business Intelligence has become a novelty, the applications of human intellect and new technologies like artificial intelligence is used for management and decision making in different business related problems. The second is the information which helps raise currency in business. The intelligent knowledge gained by experts and efficient technology in managing organizational and individual business [2].

The focus is not on the process itself, but on the technologies that allow the recording, recovering, manipulation and analysis of information. BI, that can be defined as the process of turning data into information and then into knowledge to satisfy the managers' request for efficiently and effectively analyzing the enterprise data in order to better understand the situation of their business and for improving the decision process[22]. BI represents the step beyond DWing, it includes classic DW but requires contributions from many other research fields like economy and artificial intelligence. The authors believe that the two DW features that mainly limit the classic architecture are (1) the batch-update mechanism and (2) the limits in analysis capabilities induced by OLAP. Overcoming these problems requires a broad rethinking of the architecture that leads to new solutions that requires new research issue to be addressed.

Reporting technology in BI contains much more functionalities than just distribution of information.BI distinguishes three main types of reporting tools, namely production reporting tools, desktop report writers and managed query tools. Production reporting tools are used to generate operational reports. As these reports cover large amount of data, queries are processed in batch mode. In contrast, desktop report writers enable users to design queries and reports quite simply and easily on their desktop, When complex source data need to be accessed, managed query tools are to be applied.[4]

Business Analytics is a major part of business intelligence. Business analytics is directly aided by data warehousing and business intelligence. Business analytics is a term used in context with the entire process which involves application of skills, technology and different algorithms of data warehousing. Business analysis produce valuable information to help managers make better decisions regarding their business and have proper control on their business operations. There are two main faces of business analytics function, the back-end where the main application of data warehousing takes place and the front-end is a collation of diverse information and executive reporting metrics. If we can effectively execute the business analytics function, it may result in becoming the core competence for an organization containing valuable business intelligence which can support an organization in taking strategic and efficient actions in business [26].

2.1 Business Performance Management (BPM).

BPM helps organizations to optimize business performance by encouraging process effectiveness as well as efficient use of financial, human, and material resources. BPM includes DW but it also requires a brand new set of solutions that rely on different technologies and deeply impact on the overall architecture of the BI platform [13]. Note that, in comparison with operational databases, temporal issues are more pressing in DWs since queries frequently span long periods of time; thus, it is very common that they are required to cross the boundaries of different versions of data and/or schema. Describing BPM [36] requires to understand how management is carried out within a process-oriented enterprise where, beside the classical organizational structure, a set of inter-division processes are present. The organizational structure is a hierarchy of divisions, aimed at defining their duties and responsibilities, and is usually organized on three different levels. At the strategic level, the global strategy of the enterprise is decided. The tactical level is usually composed by multiple divisions, each controlling a set of functions; the decisions taken here are related to the corresponding functions and must comply with the strategy defined at the upper level. Finally, at the operational level, the core activities are carried out; the decision power is limited to optimizing the specific production activities in accordance with the main strategy. The peculiar features that distinguish BPM from classical DW-based BI are: [19]

- Users: the users of BPM systems are still decision makers, but at the tactical and operational levels.
- Delivery time: Decisions at the lower levels must be faster than the strategic ones, thus the freshness of information must be set accordingly
- Information coarseness and lifetime: information circulated in BPM systems is usually more detailed than in DW systems, since it concerns single events related to specific tasks.

3- Data Warehouse and ETL architecture

The strategy for performing effective data mining over data warehousing generally encompasses four phases. In today's information intensive business environment, the availability of data is not an issue in most organizations. However, the formats in which these data are represented may not all be suitable for data mining purpose. Thus, the first phase of data mining strategy is to prepare the data. Data preparation is a critical activity and is often the most time consuming activity in data mining. It involves converting unsuitable data formats into specific formats that lend themselves readily for data mining. Data preparation may include tasks such as data cleaning (how to deal with missing data), data transformations (converting data values through normalization, mapping, and/or aggregation), and data reduction (combining data that involves large number of variables into a smaller set of variables) [26].

Once data preparation is completed, the second phase involves selecting appropriate data mining techniques to search for patterns in the data sets. Depending on the objective of the data mining exercise, one would most likely use any one of the primary data mining techniques such as association, clustering and classification or estimation/prediction. Each approach invokes a particular algorithm that will systematically search for specific forms of pattern in the data sets. In the third phase, the results generated by data mining have to be interpreted and the model it created has to be assessed for accuracy, validity and/or relevance. If the model proved to be accurate, valid and/or relevant, it will then be used as a decision making tool in business applications. If the assessment of the model concluded that it is insignificant, the data mining exercise can be repeated using a bigger sample data set or alternatively using new data attributes. In the fourth phase, once the association, clustering or classification model has been assessed and is deemed to be accurate, valid and/or relevant, deployment of business plans guided by the model will follow [18].

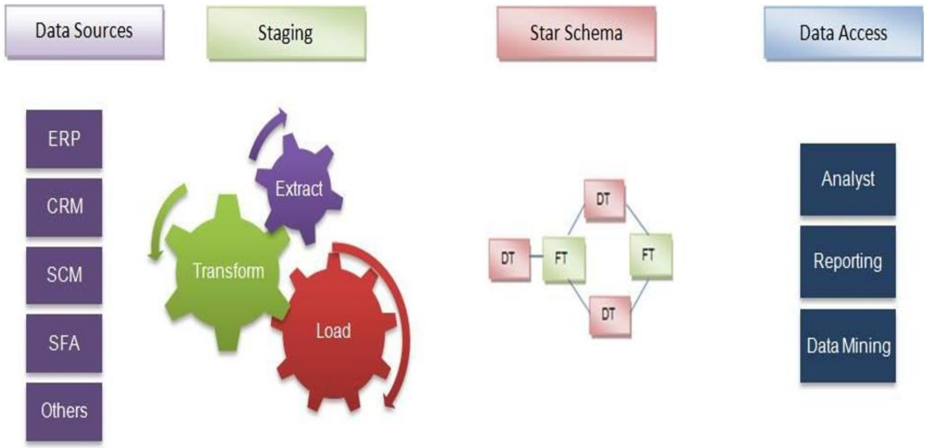

Fig.2: architecture of a data warehouse [11]

The Data warehousing is the process of collecting data to be stored in a managed database in which the data are subject-oriented and integrated, time variant, and nonvolatile for the support of decision making [5]. Fig. 2 shows the arcticture of data warehousing. Data from the different operations of a corporation are reconciled and stored in a central repository (a data warehouse) from where analysts extract information that enables better decision making. The Data can then be aggregated or parsed, and sliced and diced as needed in order to provide information. The Warehouse data are non-volatile in that data that enter the database are rarely, if ever, changed once they are entered into the warehouse. The data in the warehouse are read-only updates or refresh of the data occur on a periodic, incremental or full refresh basis [17]. According to Basaran reveals some of the DW characteristics to include the following [4] (1) It is subject-oriented. (2) It is non-volatile. (3) It allows for integration of various application systems. It supports information processing by consolidating historical data. (4) Data is stored in a format that is structured for querying and analysis. (5) Data is summarized. DWs usually do not keep as much detail as transaction-oriented systems.

One of the key processes in the DW system is integration with operational environment. The data is gathered from environment into DW through integration type called ETL. The data is extracted from the source systems, transformed, quality assured and finally transferred to the history store for long time storage. Abbreviated term, the ETL stands for: Extraction – get the data from data source system; Transformation – manipulate the data until it fits the business needs and history store data model; Load – persist data into target database (data warehouse, data mart) [23]. The ETL Layer consists of the following modules as illustrated in Fig. 3:

- ETL Processes – contains the data movement pieces of the architecture. These modules perform the main functions of the ETL Layer in extracting data from the source systems, validating input data, transforming the source data according to the business rules and loading the Enterprise Data Warehousing (EDW).
- Operational Processes (Workflow) – contains the operational maintenance pieces of the ETL Layer that supports the scheduling and automation of ETL jobs and cleanup of the files in the staging area
- Framework – contains the utilities and scripts for common functions of logging, metadata capture, process control, and automation utilities used by components in the ETL Layer modules – ETL Processes and Operational Processes.

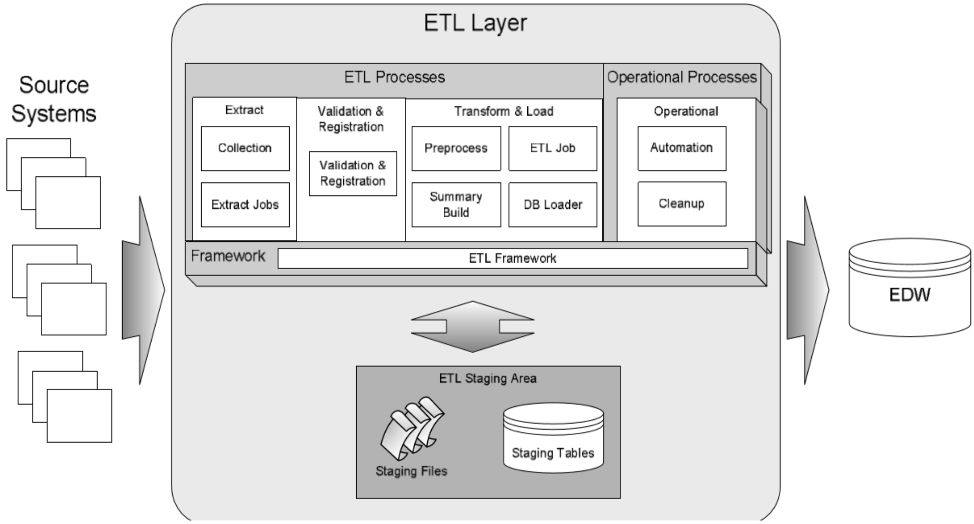

Figure 3: ETL Architecture [23].

Within database design modeling, there are three levels of data modeling. They are conceptual, logical, and physical. Conceptual design manages concepts that are close to the way users perceive data; logical design deals with concepts related to a certain kind of DBMS; physical design depends on the specific DBMS and describes how data is actually stored. The main goal of conceptual design modeling is developing a formal, complete, abstract design based on the user requirements. DW logical design involves the definition of structures that enable an efficient access to information. The designer builds multidimensional structures considering the conceptual schema representing the information requirements, the source databases, and non-functional (mainly performance) requirements. [4]

Most DW methods agree on the opportunity for distinguishing between a phase of conceptual design and one of logical design [14]. Conceptual design aims at deriving an implementation-independent and expressive conceptual schema for the DW, according to the chosen conceptual model, starting from the user requirements and from the structure of the source databases. Logical design takes the conceptual schema and creates a corresponding logical schema on the chosen platform by considering some set of constraints (e.g., concerning disk space or query answering time). Several methods (e.g.,) [31] also support a phase of physical design that addresses all the issues specifically related to the suite of tools chosen for implementation – such as indexing and allocation. From the functional point of view, the relationships between these phases can be summarized as in Figure 4 (in practice, this process will likely include feedback loops that allow to re-enter previous phases).

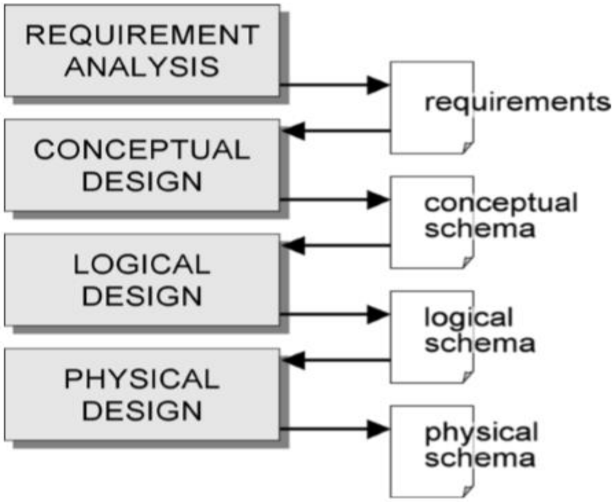

Figure 4: The core phases in DW design [16]

In the literature, conceptual modeling for DWs has been tackled from mainly two points of view so far: (1) Multidimensional modeling. The existing approaches may be framed into three categories: extensions to the Entity-Relationship model (e.g., [10]), extensions to UML (e.g., [32]), and ad hoc models (e.g., [14]), (2) Modeling of ETL. The focus is to model the ETL process either from the functional [25], the dynamic, or the static point of view. Though several conceptual models have been proposed, none of them has been accepted as a standard so far, and all vendors propose their own proprietary design methods. The main reasons for this, we argue, can be summarized as follows: (i) there is still no agreement from both the research and industrial communities about which are the most relevant multidimensional properties to be modeled; (ii) although the conceptual models devised are semantically rich, some of the modeled properties cannot be expressed in the target logical models, so the translation from conceptual to logical is incomplete; and (iii) commercial CASE tools currently enable designers to directly draw logical schemata, thus no industrial push is given to any of the models. Though most conceptual models for DWs in the literature do not address security, lately some interesting proposals were devised which define specific authorization and security models. [1]

Once the conceptual modeling phase is completed, the overall task of logical modeling is the transformation of conceptual schemata into logical schemata that can be optimized for and implemented on a chosen target system. With respect to fact modeling, there still is a semantic gap between advanced conceptual data models and relational or multidimensional implementations of data cubes. Consequently, future research is necessary to bridge this semantic gap, i.e., to preserve all information captured by advanced conceptual multidimensional models in logical implementations [38]. To this end, research could either investigate how to enrich meta-data for tool support in a systematic way or, more ideally, look for more expressive logical models while preserving good query performance. [37]

The approaches to management of schema changes in DWs can be framed into two categories, namely evolution [36] and versioning [37]: while both categories support schema changes, only the latter keeps track of previous versions. If one is sure that previous schema information will never be useful again, schema evolution offers adequate functionality. Otherwise (e.g., to guarantee consistent re-execution of old reports), schema versioning offers the strictly more powerful approach. Actually, in some versioning approaches, besides “real” versions determined by changes in the application domain, also “alternative” versions to be used for what-if analysis are considered [3]. Overall, we believe that versioning is better suited to support the complex analysis requirements of DW users as well as the DW characteristic of non-volatility. Thus, the main research challenges in this field are to provide effective versioning and data migration mechanisms, capable of supporting flexible queries that span multiple versions [30].

Four, different views regarding a data warehouse design must be considered: the top-down view, the data source view, the data warehouse view, of the information system. The top - Down view allows the selection of the relevant information necessary for the data warehouse. This information matches current and future business needs. The Data source view exposes the information being captured, stored, and managed by operational system. This information may be documented at various levels of detail and accuracy, from individual data source tables to integrate at various levels of detail and accuracy, form individual data source tables to integrated data source tables. Data sources are often modeled by traditional data modeling techniques, such as the E-R model [6].

4- DYNAMIC EDSs (External Data Sources)

The data Heterogeneity indicates that the EDSs use different data models and different user and programming interfaces. Local autonomy and heterogeneity of EDSs mean that they were developed independently and are not aware of the integration issues [28]. An important result of the autonomy of EDSs is that they may evolve in time independently and that they change their data and schemas without being controlled from the global data warehouse level. The current data warehousing systems interact only with the first kind of changes, namely, with the content changes. The content changes of EDSs are detected and propagated to the data warehouse.

The current data warehousing systems deals only with the content changes. These changes of EDSs are spotted and propagated to the data warehouse in one of the following methods. For EDSs that are database systems the source can gather all updates that happened during a definite interval of time and send all updates regularly to the data warehouse. All updates can be shipped either as a collection of data (data shipment) or as a collection of transactions (transaction shipment). The size of the refresh period is based on the data warehouse usage type, types of data source, the balance of work, etc. Changes shipped to the data warehouse can be applied either all at once (batch update) or incrementally (incremental update). The entire process of, filtering, extracting, transmitting, transforming, cleaning, and loading updates into the data warehousing system is called data warehouse refreshment.. However, little research has examined the data warehouse refresh under schema changes and constraints modifications.[8]

Propagating all changes happening in the EDSs to the data warehouse can be also another solution. Changes should be merged into the metadata repository of the data warehouse. The metadata repository saves administrative data essential to manage the data warehouse, such as: descriptions of external data sources, their contents and schemas, view and derived data definitions, data warehouse schema, categories and hierarchies, data mart locations and contents, dimensions, data partitions, descriptions of predefined queries and reports, data extraction, cleansing, data refresh and purge rules, transformation rules, defaults, user profiles, user groups, etc. [8]

4.1. EDS SCHEMA CHANGE AND THE DATA WAREHOUSE CONTENT

All changes occurring in the EDSs schemas influences also the data warehouse content and must be properly treated. Another solution is to create a separate instance of the data warehouse content for each instance of the data warehouse schema. This brings in the idea of a multi-versioning of the data warehouse and creates new challenges with respect to the maintenance, evolution, and query processing in such multi-version data warehouse. This method is possible mostly for data warehouses that are exposed to dimension and hierarchy changes. For EDSs schema changes associated to creating, altering, or dropping attributes and tables more suitable is to regulate the data warehouse content to the change. This needs further knowledge that must be given by a user, e.g., when creating new attribute in each table the user must provide correct values of the newly created attribute for all tuples already saved in the data warehouse. Moreover, it also may be essential to update some aggregates and summaries included in the data warehouse. The adjustment of the data warehouse content can be done either in batch or incremental mode. The main issue is how to process a change in the EDS that is associated to changing the meaning of an attribute or an aggregate. This must be solved in order to provide the logical independence between the conceptual level of the data warehouse and the external level including of the front-

end applications. In other words, the problem is how to accommodate changes in the data warehouse in such a method that the current access and reporting tools are not influenced by the changes and keep giving correct results. To some extent, this problem can be treated by the appropriate data warehouse schema design and new data warehouse models that consider the changes and modifications in both data warehouse schema and the meaning of attributes, aggregates and summaries. [9]
The multi-dimensional view delivers long term data that can be analyzed along the time axis, whereas most EDSs only provide snapshots of data at specific time. Existing warehouse systems are therefore ready to deal with changing measures, e.g., changing profit or turnover. However, they are not able to treat modifications in dimensions, although from the EDS point of view there is nothing basically different between the measures data and the dimensions data saved in external data sources. One must consider that the dimensions data, i.e., the structure, the schema and the instances of the dimensions of a data warehouse are continuously changeable. This issue has lately became the interest of a lot of research [7].

There are two types of schemes to handle schema and content changes in DW. These are evolution and versioning schemes. The evolution schemes support a single version of DW schema for both schema and content changes. In evolution, the DW schema is updated and data is transferred from old to the new schema. The benefits of evolution schemes are as follows: querying mechanisms are not changed and the available commercial tools can be used. However, the schemes lack preserving history and maintenance cost is high. In contrast to that, the versioning schemes support multiple versions of DW schema for each schema change and the content change, which cannot be accommodated in the existing version [33]. The benefit of version schemes is that the history is preserved. However, retrieval of data requires writing cross-version queries which cannot be performed by using the available commercial tools.

As stated in [8], one of the existing solutions is multi-versioning of the data warehouse instances joined with temporal extensions to the existing data warehouse model. Herein the Temporal Multidimensional model are used that fulfills these assumptions and offers the following features: (1) representation of changes in data warehouses schema. (2) Identification of periods without a significant change. (3) Mapping and transformation functions between structure versions. (4) Processing queries reading data that is contained in several versions. Nevertheless, it is worth noticing that this approach allows the data warehouse content adjustment in response to changes occurring in the EDSs that modify the meaning of selected attributes, in particular changes that modify the content and structure of dimensions.

5- CONCLUSION

This paper discusses the effect of data warehousing technique in business intelligence. Two powerful tools determine the growth in business sector. The primary is data warehousing which is used to deal with large amount of data to find useful result, whereas the secondary is business intelligence which helps in making business related decisions. Furthermore, we discussed the problem of the data warehouse evolution resulting from the changes in the underlying external data sources, and suggesting to utilize temporal multidimensional model to handle this evolution. The paper shows business analytics with a wide application domain almost in every industry where the data is generated that's why data warehousing is considered one of the most important outwork in databases and information systems and business intelligence as an interface of the organization.

Based on the Synthesis we identify directions for future academic work: (a) the use of data mining techniques is a scalable approach to detect structural changes in DWs, (b) the exploration of self-adaptive methods to detect structural changes in DWs is an open research area, (c) a common framework for retrieving and presenting data in a temporally consistent manner, and (d) the support of cross version queries and their impact on interpretation of results needs further exploration. For the practitioner, the study established that no commercial DW provides support for cross-version queries. The querying support can be provided for monitoring different types of changes to support strategic decision-making.

Reference:

1. Abell´o, J. Samos, and F. Saltor. YAM2. (2006) A multidimensional conceptual model extending UML. Information Systems, 31(6):541–567, 2006.
2. Arti J. Ugale, P. S. Mohod, (2015) "Business Intelligence Using Data Mining Techniques on Very Large Datasets", International Journal of Science and Research (IJSR), Volume 4 Issue 6, June 2015 , pp2932-2937
3. B¸bel, J. Eder, C. Koncilia, T. Morzy, and R. Wrembel (2004) Creation and management of versions in multiversion data warehouse. In Proc. ACM SAC, pages 717–723, 2004.
4. Başaran, Beril P (2005), a Comparison of Data Warehouse Design Models, the Graduate School of Natural and Applied Sciences, Atilim University.
5. Cho V. and Ngai E.W.T. (2003) Data Mining for Selection of Insurance Sales Agents. Expert Systems. Vol. 20, No. 3
6. Dishek Mankad1, Preyash Dholakia. (2013) The Study on Data Warehouse Design and Usage 1 M.C.A., B.R.Patel Institute of Computer Application [MCA Program] 2M.C.A. K.S.K.V. Kachchh University MCA College, International Journal of Scientific and Research Publications, Volume 3, Issue 3, March 2013 1 ISSN 2250-3153.
7. Eder J., Koncilla C., (2001) Changes of Dimensions Data in Temporal Data Warehouses, Proceedings of the DaWak Conf. 2001, pp.85-94.

8. Eder J., Koncilla C., Morzy T. (2002) The COMET Metamodel for Temporal Data Warehouse, Proceedings of the CAISE 2002, pp.83-99.
9. Eder J., Koncilla C., Morzy T.(2001) A Model for a Temporal Data Warehouse, Proceedings of the Intl. OESSEO 2001 Conf. Rome, Italy, 2001, pp.86-97
10. E. Franconi and A. Kamble. (2004) A data warehouse conceptual data model. In Proc. SSDBM, pages 435–436, 2004.
11. George, S., (2012), Inmon vs. Kimball: Which approach is suitable for your data warehouse? ,Computer Weekly, http://www.computerweekly.com/tip/Inmonvs-Kimball-Which-approach-is-suitable-foryour-data-warehouse (Accessed 13-03-2017)
12. Giovinazzo, W. (2002) 'Internet-Enabled Business Intelligence', Prentice Hall.
13. H. Dresner and F. Buytendjik. (2003) Business Intelligence and Data Warehousing Scenario: Key Trends and Evolving Markets. Gartner Symposium ITXPO. France, 2003.
14. H¨usemann, J. Lechtenb¨orger, and G. Vossen. (2000) Conceptual data warehouse design. In Proc. DMDW, pages 3–9, 2000.
15. J. Maz´on, J. Trujillo, M. Serrano, and M. Piattini. (2005) Designing data warehouses: From business requirement analysis to multidimensional modeling. In Proc. Int. Work. On Requirements Engineering for Business Needs and IT Alignment, 2005.
16. J. Trujillo, S. Luj´an-Mora, and E. Medina. (2002) The Gold model case tool: An environment for designing OLAP applications. In Proc. ICEIS, pages 699–707, 2002.
17. Jayant Singh, Vijay Gupta. (2014) A Review of Data Warehousing and Business Intelligence in different perspective, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 8263-8268.

18. Koh, H.C. (2005) Data Mining Applications for Small and Medium Enterprises, Centre for Research on Small Enterprise Development, Singapore.
19. M. Golfarelli S. Rizzi and I. Cella. (2004) Beyond data warehousing: What's next in business intelligence? In Proc. DOLAP 2004, USA, 2004.
20. M. Golfarelli, J. Lechtenb¨orger, S. Rizzi, and G. Vossen. (2006) Schema versioning in data warehouses: enabling cross-version querying via schema augmentation. In Data & Knowledge Engineering, 2006. In press.
21. M. Poess and R. Othayoth. (2005) Large scale data warehouses on grid: Oracle database 10g and HP ProLiant systems. In Proc. VLDB, pages 1055–1066, 2005. (Data Warehouse Modeling)
22. Maniatis. (2003) Advanced visualization for OLAP. DOLAP. USA, 2003.
23. Multicom d.o.o. Vladimira Preloga 11 10000 Zagreb, Data Warehousing & Business Intelligence. multicom@multicom.hr www.multicom.hr
24. N. Prat, J. Akoka, and I. Comyn-Wattiau. (2006) A UML-based data warehouse design method. Decision Support Systems, 2006. In press.
25. P. Vassiliadis, A. Simitsis, and S. Skiadopoulos. (2002) Conceptual modeling for ETL processes. In Proc. DOLAP, pages 14–21, 2002.
26. Pui Mun Lee. (2013) Use of Data Mining In Business Analytics to Support Business Competitiveness Review of Business Information Systems – Second Quarter 2013 Volume 17, Number 2 Pp 53-58.
27. R. Winter and B. Strauch. (2003) A method for demand-driven information requirements analysis in data warehousing projects. In Proc. HICSS, pages 1359–1365, 2003.
28. Rudensteiner E. A., Koeller A., and Zhang X. (2000) Maintaining Data Warehouses over Changing Information Sources, Communications of the ACM, vol. 43, No. 6, 2000.

29. S. Abiteboul, I. Manolescu, and N. Preda. (2005) Constructing and querying peer-to-peer warehouses of XML resources. In Proc. ICDE, pages 1122–1123, 2005.
30. S. Bimonte, A. Tchounikine, and M. Miquel. (2005) Towards a spatial multidimensional model. In Proc. DOLAP, pages 39–46, 2005
31. S. Luj´an-Mora and J. Trujillo. (2003) A comprehensive method for data warehouse design. In Proc. DMDW, 2003.
32. S. Luj´an-Mora, J. Trujillo, and I. Song. (2006) A UML profile for multidimensional modeling in data warehouses. In Data & Knowledge Engineering, 2006. In press.
33. SAP. Business Planning and Simulations (SEM-BPS). 2003.
34. Spitzer, R. L., Kroenke, K., Williams, J. B., & Lowe, B. (2006). A brief measure for assessing generalized anxiety disorder: The GAD-7. Archives of Internal Medicine, 166, 1092-1097. doi:10.1001/archinte.166.10.1092
35. Vaisman, A. Mendelzon, W. Ruaro, and S. Cymerman. (2002) Supporting dimension updates in an OLAP server. In Proc. CAiSE, pages 67–82, 2002.
36. W. Martin and R. Nußdorfer. (2003) "Pulse Check": Operational, Tactical and Strategic BPM. CSA Consulting GmbH White Paper, 2003.
37. Y. Sismanis and N. Roussopoulos. (2004) The complexity of fully materialized coalesced cubes. In Proc. VLDB, pages 540–551, 2004
38. Y. Sismanis, A. Deligiannakis, Y. Kotidis, and N. Roussopoulos. (2003) Hierarchical dwarfs for the rollup cube. In Proc. DOLAP, pages 17–24, 2003.