Real Time ETL Improvement

Robert Halenar

***Abstract*—This paper describes the contribution to the Near real time ETL process, which shows way, how to maintain real time ETL process automated (without human – database administrator interference), in cost of reduced accuracy rendered by level of trust. First we described ETL as a part of KDD, what is Real time ETL and problem how to achieve real – time in real world. In next part we give several (near) real time approaches with their advantages and disadvantages and then we present our contribution to near real time ETL with new architecture containing equation for calculation the level of trust. Result is showed on real data of a small company engaged in the sale of computer components, which expands the market for Internet sales and there is need for fresh data using business data warehouse containing real trade data.**

***Index Terms*—Knowledge discovery in databases, extraction transformation loading, real time, business, data warehouse.**

I. Introduction

Our research is about finding new methods and tools using in different stages of Knowledge Discovery in Databases. In the early stages of the KDD it is necessary to collect and preprocess data [1]. Especially, we improved the Near Real Time ETL phase (Extraction Transformation Loading), using the new architecture.

Near real time ETL deviates from the traditional conception of data warehouse refreshment, which is performed off-line in a batch mode, and adopts the strategy of propagating changes that take place in the sources towards the data warehouse to the extent that both the sources and the warehouse can sustain the incurred workload.

The demand for fresh data in data warehouses has always been a strong desideratum from the part of the users. Traditionally, the refreshment of data warehouses has been performed in an off-line fashion. In such a data warehouse setting, data are extracted from the sources, transformed, cleaned, and eventually loaded to the warehouse. This set of activities takes place during a loading window, usually during the night, to avoid overloading the source production systems with the extra workload of this work flow. Interestingly, the workload incurred by this process has been one of the fundamental reasons for the establishment of data warehouses, since the immediate propagation of the changes that take place at the sources was technically impossible, either due to the legacy nature of the sources involved or simply due to the overhead incurred, mainly for the operational source systems but also for the warehouse. In most cases, a data warehouse is typically updated every 24 hours.

Nowadays, new types of sources enter into the scene. In several applications, the Web is considered as a source. In such a case, the notion of transaction at source side becomes more flexible, as the data that appear at a source web site are not always available later; therefore, if instant reaction to a change is not taken, it is possible that important information will not be gathered later, by the off-line refreshment of the warehouse. At the same time, business necessities – e.g., increasing competition, need for bigger sales, better monitoring of a customer or a goal, precise monitoring of the stock market, and so on - result in a demand for accurate reports and results based on current data and not on their status as of yesterday.

The usual process of ETL-ing the data during the night in order to have updated reports in the morning is getting more complicated if we consider that an organization's branches may be spread in places with totally different time-zones. Based on such facts, data warehouses are evolving to "active" or "live" data producers for their users, as they are starting to resemble, operate, and react as independent operational systems. In this setting, different and advanced functionality that was previously unavailable (for example, on-demand requests for information) can be accessible to the end users. For now on, the freshness is determined on a scale of minutes of delay and not of hours or a whole day. As a result, the traditional ETL processes are changing and the notion of "real-time" or "near real-time" is getting into the game. Less data are moving from the source towards the data warehouse, more frequently, and at a faster rate. [2]

II. Near Real Time ETL

The ETL market has already made efforts to react to those new requirements. The major ETL vendors have already shipped "real time" ETL solutions with their traditional platforms. In practice, such solutions involve software packages that allow the application of light-weight transformations on-the-fly in order to minimize the time needed for the creation of specific reports. Frequently, the delay between the moment a transaction occurs at the operational site and the time the change is propagated to the target site is a few minutes, usually, five to fifteen. Such a response should be characterized more as "near real time" reaction, rather than "real time", despite how appealing and promising can the latter be in business terms.

Traditionally, ETL processes have been responsible for populating the data warehouse both for the bulk load at the initiation of the warehouse and incrementally, throughout the operation of the warehouse in an off-line mode. Still, it appears that data warehouses have fallen victims of their

Manuscript received April 13, 2012; revised May 14, 2012.

Robert Halenar is with the University of SS Cyril and Methodius in Trnava, Nam. J. Herdu 2, 917 01 Trnava, Slovak Republic (e-mail: Robert.halenar@ucm.sk).

success: users are no more satisfied with data that are one day old and press for fresh data – if possible, with instant reporting. This kind of request is technically challenging for various reasons. First, the source systems cannot be overloaded with the extra task of propagating data towards the warehouse. Second, it is not obvious how the active propagation of data can be implemented, especially in the presence of legacy production systems. The problem becomes worse since it is rather improbable that the software configuration of the source systems can be significantly modified to cope with the new task, due to (a) the down-time for deployment and testing, and, (b) the cost to administrate, maintain, and monitor the execution of the new environment.

The long term vision for near real time warehousing is to have a self-tuning architecture, where user requirements for freshness are met to the highest possible degree without disturbing the administrators' requirements for throughput and availability of their systems. Clearly, since this vision is founded over completely controversial goals, reconciliation has to be made:

A more pragmatic approach involves a semi-automated environment, where user requests for freshness and completeness are balanced against the workload of all the involved sub-systems of the warehouse (sources, data staging area, warehouse, data marts) and a tunable, regulated flow of data is enabled to meet resource and workload thresholds set by the administrators of the involved systems [2].

The general ETL architecture of a near real time data warehouse consists of database sources including extraction tool, which pushes extracted data into temporary store. Then it prepares data for transformation process into transformation function – ready data format. Transformation runs in DPA (Data processing area) where data are transformed and cleaned and after then are data exported by transformation function. Loader then loads data into data warehouse fact and dimension tables. Whole process shows Fig. 1.

On this architecture is based traditional ETL. In case of Near real time ETL there are built in compensation structures, which alleviates impact of high frequently refreshment. In a real world this cannot be performed, due to many possible reasons like high number of users, high rate refreshment or too expansive software and hardware parts, and this situation is solved by several technical and structural accessories.

Practically it leads to compensated schema which contains complementary parts.

The difficulty of such a simple Transform function is to keep it simple. As though they can exert a vacuum of complexity, simple Transform functions attract additional functions and complexity to them. Resist this temptation at all costs. A simple Transform function is another beautiful and elegant design, and should be allowed to remain that way [3].

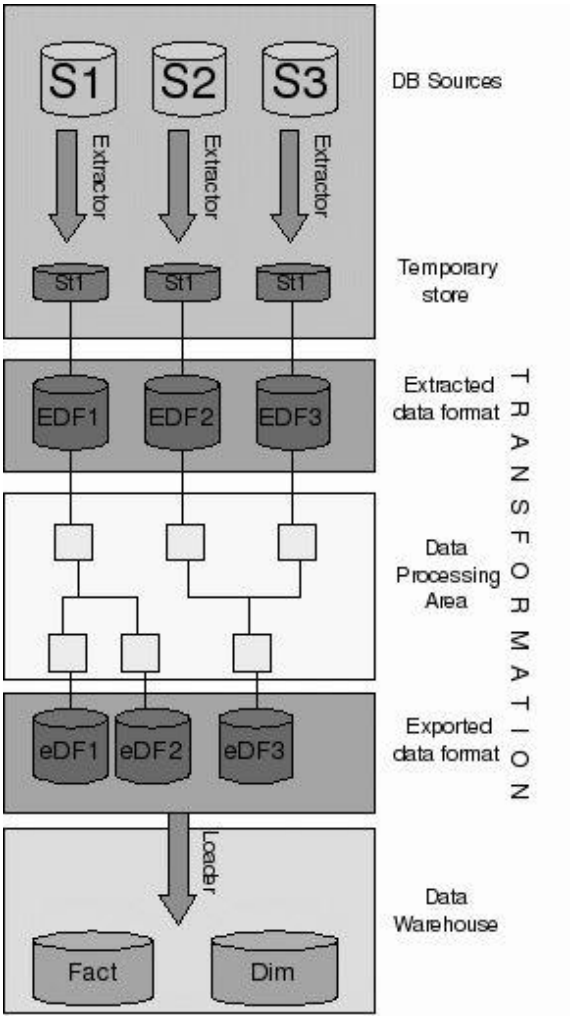

Fig. 1. Near real time ETL.

III. (Near) Real Time ETL Approaches

As usual, different alternative approaches have been proposed in the market to handle the need for freshness in a data warehouse.

Enterprise Application Integration, EAI. These approaches have the ability to link transactions across multiple systems through existing applications by using software and computer systems architectural principles to integrate a set of enterprise computer applications. An EAI system is a push system, not appropriate for batch transformations, whose functionality entails a set of adapter and broker components that move business transactions - in the form of messages - across the various systems in the integration network. An adapter creates and executes the messages, while a broker routes messages, based on publications and subscription rules.

The main benefit from an EAI system is fast extraction of relevant data that must be pushed towards the data warehouse. In general, an EAI solution offers great real time information access among systems, streamlines business processes, helps raise organizational efficiency, and maintains information integrity across multiple systems. Usually, it is considered as a good solution for applications demanding low latency reporting and bidirectional synchronization of dimensional data between the operational sources and the data warehouse.

However, as nothing comes without a cost, they constitute extremely complex software tools, with prohibitively high development costs, especially for small and mid-sized businesses. Also, EAI implementations are time consuming, and need a lot of resources. Often, many EAI projects usually start off as point-to-point efforts, but very soon they become unmanageable as the number of applications increase.

Fast transformations via Capture - Transform - Flow (CTF) processes. This solution resembles a traditional ETL process too. CTF approaches simplify the real time transportation of data across different heterogeneous databases. CTF solutions move operational data from the sources, apply light-weight transformations, and then, stage the data in a staging area. After that, more complex transformations are applied (triggered by the insertions of data in the staging area) by micro batch ETL and the data are moved to a real time partition and from there, to static data stores in the data warehouse. CTF is a good choice for near real time reporting, with light integration needs and for those cases where core operations may share periods of low activity and due to that, they allow the realization of data synchronization with a minimal impact to the system.

Fast loading via micro batch ETL. This approach uses the idea of real time partitioning (described in section 2.3.3.1) and resembles traditional ETL processes, as the whole process is executed in batches. The substantial difference is that the frequency of batches is increased, and sometimes it gets as frequent as hourly. Several methods can be used for the extraction of data - e.g., timestamps, ETL log tables,

DBMS scrapers, network sniffers, and so on. After their extraction the data are propagated to the real time partition in small batches and this process continuously runs. When the system is idle or once a day, the real time partitions populate the static parts of the data warehouse. The micro batch ETL approach is a simple approach for real-time ETL and it is appropriate for moderate volumes of data and for data warehouse systems tolerant of hourly latency. The main message it conveys, though, is mainly that dealing with new data on a record-by-record basis is not too practical and the realistic solution resolves to finding the right granule for the batch of records that must be processed each time.

On-demand reporting via Enterprise Information Integration (EII). EII is a technique for on-demand reporting. The user collects the data he needs on-demand via a virtual integration system that dispatches the appropriate queries to the underlying data provider systems and integrates the results. EII approaches use data abstraction methods to provide a single interface for viewing all the data within an organization, and a single set of structures and naming conventions to represent this data. In other words, EII applications represent a large set of heterogeneous data sources as a single homogenous data source. Specifically, they offer a virtual real time data warehouse as a logical view of the current status in the OLTP systems. This virtual warehouse is delivered on-the-fly through in-line transformations and it is appropriate for analysis purposes. It generates a series of (SQL) queries at the time requested, and then it applies all specified transformations to the resulting data and presents the result to the end user. EII applications are useful for near-zero latency in real time reporting, but mostly for systems and databases containing little or no historical data. [2]

IV. Our Contribution of Near Real Time ETL

Our research is focused on Near real time ETL improvement, using new compensation parts, showed on Fig.2.

Whole process in DPA runs automatically even there is no reason for excluding the data, like reference error or the system is overloaded due to high refreshment rate or high number of users. Each situation should be tested first. [4]

There is situated also a file of flags, containing file of conditions edited by administrator, that are used for additional inspection. Excluded and marked data are evaluated as well as reorganized, cleaned and transformed data. After then is calculated level of trust all outgoing data, which describes on how much valid the data are. It is showed on Equation (1).

$$[\%] = 100 - \frac{\left(\text{Data excluded from transformation process} + \text{Cleaned and transformed data marked with flag}\right) \cdot 100}{\text{Reorganized cleaned and transformed data}} \quad (1)$$

Of course, for each data row is also available the pertinent time, so level of trust should be calculated for a certain time interval.

This model of improved near real time ETL should be applied on the either ETL approach, and also should be applied on sequential, pipelining and partitioning execution of ETL process too. Improved architecture is showed on Fig. 3.

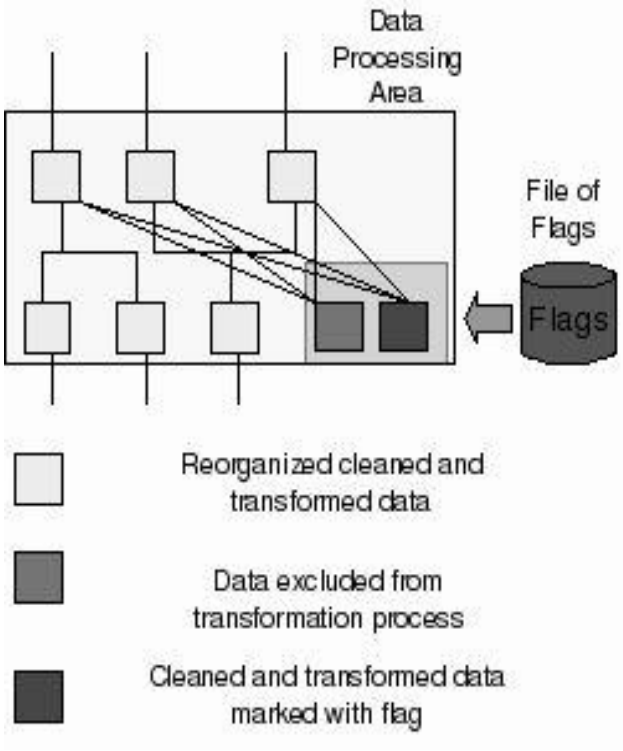

Fig. 2. Near real time ETL with compensation parts

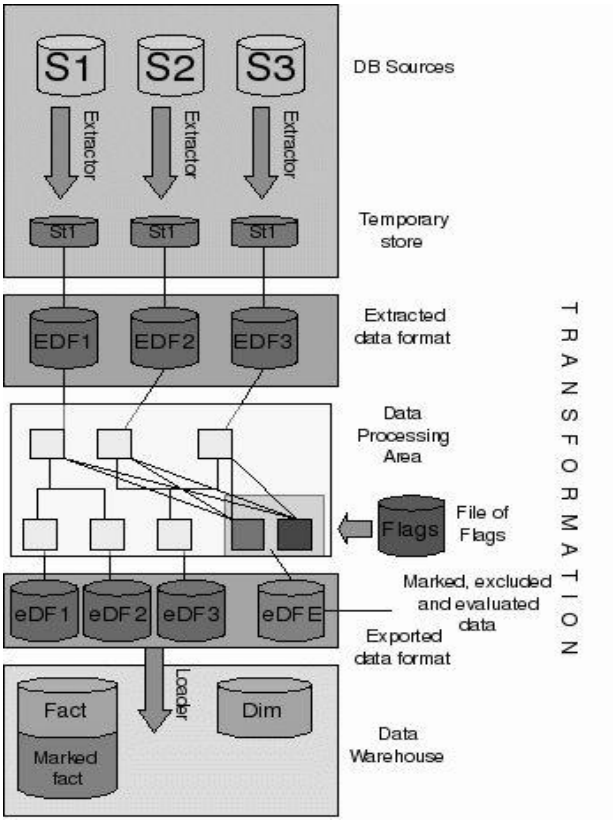

Fig. 3. Improved near real time ETL

V. Application in Business DWH

In our research we applied the new Near real time ETL architecture on Business data warehouse specified accord to Fig. 4.

The principle is subject oriented, so data will be grouped by subject, rather than author, department, or physical location. So, all manufacturing data goes together, and the sales data, and the promotions data, etc., regardless of where it came from [5], [6].

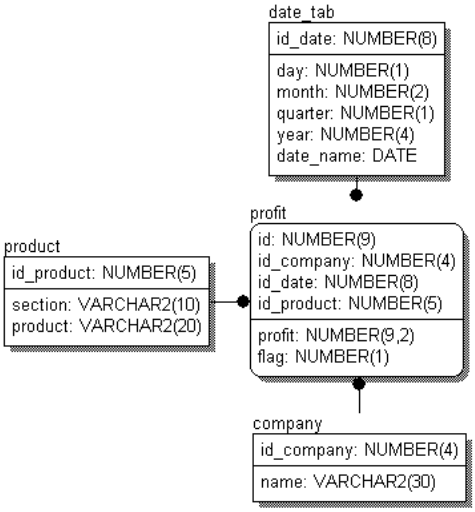

Fig. 4. Business data warehouse architecture

Our Business DWH is focused on sales data, concretely profit, depending on customer and product. It consists of three dimension tables and one fact table. Time dimension is represented by date table.

Source data are in Microsoft Database format, which are exported to XLS sheets entering into DPA. Transformation process is reorganizing data according to DWH needs, transforming measuring units, performing reference assignment, modifying key values, sorting products in sections and so on. File of flags contains exceptions for transformation process, like certain company (customer) assignment, some product exceptions, which are stored in "Exported data format for errors" (eDFE). Loading process then pushes exported data into DWH as well as exported "error" data, which are marked with flag in the fact table (Profit).

If any data are needed for analyzing, level of trust is calculated for specified data selection.

Table I and II show the appearance of Extracted data format, which already contains modified ID_Stock_Movement and prices, recalculated currency to euro. References ID_PVP and ID_Storage_card are pointed to another Extracted data formats, concretely to dial companies and dial stock cards. In the last row of Table 1 is ID_PVP pointing to not existing value in dial companies, because stock movement has been done indiscriminately Company (for example to my neighbor). Nevertheless, the transformation process continues and populates values into eDF (Exported data format), and then Loading process loads data into Business data warehouse, with flag marked on the given row of fact table Profit and set the appropriate id_company value to NULL. Dimension and fact tables show Table III, IV, V and VI. To provide anonymity of companies, we use common products label terms.

TABLE I: Modified Extracted Data Format – Part I

ID_Stock_movement	ID_PVP	ID_Srotage_Card	Amount_of_intake	Amount_of_expenditure	Amount_of_balance	EUR_Intake
7088	731	237	0	1	1	0.00
7504	753	237	0	1	0	0.00
5840	612	237	1	0	1	42.51
6015	614	237	1	0	2	37.28
6105	618	237	0	1	1	0.00
6399	669	237	1	0	2	33.26
8372	801	237	1	0	1	29.63
9188	-1	237	0	1	0	0.00

TABLE II: Modified Extracted Data Format – Part II

Euro_nubile	EUR_Balance	Unit_price	ID_Store	Sale_Price_per_unit_excluding_VAT	Other_costs	Date
37.28	33.26	37.28	1	39.83	0.00	22.7.2008
33.26	0.00	33.26	1	38.25	0.00	4.9.2008
0.00	42.51	42.51	1	0.00	0.00	1.1.2008
0.00	79.79	37.28	1	0.00	0.00	10.1.2008
42.51	37.28	42.51	1	48.13	0.00	14.1.2008
0.00	70.54	33.26	1	0.00	0.00	7.4.2008
0.00	29.63	29.63	1	0.00	0.00	21.10.2008
29.63	0.00	29.63	1	29.68	0.00	31.12.2008

TABLE III: Dimension Table Product

id_product	section	Product
237	Hard disc	IBM 160GB S - ATA 2, 8MB cache, 7200rpm

TABLE IV: DIMENSION TABLE DATE

id_date	day	month	quarter	year	date_name
834	22	July	3	2008	22.7.2008
634	4	September	3	2008	4.9.2008
561	14	January	1	2008	14.1.2008
810	31	December	4	2008	31.12.2008

TABLE V: DIMENSION TABLE COMPANY

id_company	name
34	Company A
165	Company B

TABLE VI: TABLE OF FACT PROFIT

Id	id_product	id_date	id_company	profit	Flag
7088	237	834	34	2.56	0
7504	237	634	165	4.99	0
6105	237	561	165	5.62	0
9188	237	810	NULL	0.05	1

VI. CONCLUSION

In table Profit, there is made a data selection of sold product (hard disc IBM 160GB S - ATA 2, 8MB cache, 7200rpm). This item was sold to Company A on July 22 2008 with profit 2.56 EUR and to Company B on January 14 and also in September 4 with profit 5.62 EUR and 4.99 EUR. There was made one more sold on December 31 2008, but without name of the company because of sale for my neighbor. So there is missing record due to my neighbor is not a company to do business. Original data source does not contain appropriate record in table Company, so this absence is transferred to DWH. Normally transfer function discarded record and human assistance is needed, but in new architecture of near real time ETL is this record transferred with mark, for later administrator assistance and meanwhile we operate with profit records with permissible error expressed by Level of trust. This level we calculate according to Equation (1) as follows in Equation (2).

$$\% = 100 - \frac{(0 + 0.05) * 100}{2.56 + 4.99 + 5.62} = 99.59 \qquad (2)$$

Value of the profit in the fact table Profit is 0.05, so for data selection showed in table Profit we can calculate Level of trust equal to 99.59%. This way we can determine the level of trust for analyzing data, while the (near) real time ETL process is maintained.

REFERENCES

[1] M. Kebisek, P. Schreiber, and P. Halenar, "Knowledge Discovery in Databases and its application in manufacturing," in *proc. International workshop Innovation Information Technologies – Theory and Practice*; 2010 September 06-10, Dresden, Germany, pp. 204-207.

[2] S. Kozielski and R. Wrembel, "New Trends in Data Warehousing and Data Analysis," Springer, 2009.

[3] L. Reeves, "A Manager's Guide to Data Warehousing," Published by *Wiley Publishing*, Inc., 2009.

[4] J. Zeman, P. Tanuska, and M. Kebisek, "The Utilization of Metrics Usability To Evaluate The Software Quality," in *proc. ICCTD 2009 International Conference on Computer Technology and Development*, 13-15 November 2009, Kota Kinabalu, Malaysia. IEEE Computer Society, 2009.

[5] F. Sivers, "Building and Maintaining a Data Warehouse," *CRC Press*, 2008.

[6] A. Trnka "Market basket analysis with data mining methods," in *proc. ICNIT 2010: International Conference on Networking and Information Technology*; 11-12 June 2010, Manila, Philippines. ISBN 978-1-4244-7578-0

Robert Halenar, Ph.D. is with University of SS Cyril and Methodius in Trnava, Slovak Republic. He is a member of IAENG and IACSIT. He received Ph.D. degree (2009) in area of informatics and automation from Slovak University of Technology, Faculty of Material Science and Technology Trnava, Slovak Republic. His research includes the field of Information Systems, Data warehouse, Health information and Data mining. He published papers in national and international conference proceedings and journals.