An **I**mproved **A**pproach in **D**ata **W**arehousing **ETLM P**rocess for **D**etection of **C**hanges in **D**ata **O**rigin

Rosana L. de B. A. Rocha
COPPE/UFRJ
PO BOX 68511
Rio de Janeiro, RJ - Brasil
(5521)2590-2552
rosana@cos.ufrj.br

Leonardo Figueiredo Cardoso
COPPE/UFRJ
PO BOX 68511
Rio de Janeiro, RJ - Brasil
(5521)2590-2552
cardoso@cos.ufrj.br

Jano M. de Souza
COPPE/UFRJ
PO BOX 68511
Rio de Janeiro, RJ - Brasil
(5521)2590-2552
jano@cos.ufrj.br

Abstract

In a data warehouse (DW) environment, the ETLM process (extraction, transformation, load, and materialization) comprises the basis for data acquisition and organization, changing data into information. In this environment, we have a well known issue, the detection changes on the data origin. When the operational environment does not posses or does not want to inform the information about the changes that occurred, controls have to be implemented to enable detection of these changes and to reflect them in the data warehouse environment. The main scenarios are: i) the impossibility to instrument the DBMS (triggers, transaction log, stored procedures, replication, materialized views, old and new versions of data, etc) due to security policies, data property or performance issues; ii) the lack of instrumentation resources on the DBMS; iii) the use of legacy technologies such file systems or semi-structured data; iv) application proprietary databases and ERP systems. In this article, we propose a framework for the organization of DW environment, and present some of the main issues involved in the ETLM process. Using this framework, we describe the main approaches to solve the detection of changes in data origin. The technique we developed and implemented was derived for the comparison of database snapshots, where we use signatures to mark and detect changes. The technique is simple and can be applied to all four scenarios above.

The paper also presents a the case study in the data warehouse project developed for Rio Sul Airlines, a regional aviation company belonging to the Brazil-based Varig group.

1. Introduction

The detection of changes in data origin issue is well known in the DW area. As mentioned by DO, DREW et al. (1998), most research work on DW update focuses on the problem that, given a differential relation, how do we refresh the DW efficiently, approach defined by ÖZSU & VALDURIEZ (1991). This approach capture the after and before images, for all lines affected by each operation. Some of these researches are based on the existence of an instrumentation resource of differential relation on the DBMS. The difference between them occurs in terms of DW capabilities, such as: convergent DW consistency (ZHUGE, GARCIA-MOLINA et al., 1995; ZHUGE, GARCIA-MOLINA et al., 1996), replication of some source relations (QUASS & WIDOM, 1997), full replication (HULL & ZHOU, 1996), versioning (QUASS, GUPTA et al., 1996), etc.

According INMON & KELLEY (1993), a DW is a repository of integrated information, available for queries and analysis (e.g., decision support, or data mining). The DW came to meet this demand for a possibility of fast analysis of business information. According to FINNEGAN, MURPHY et al. (1999), there is, in a dynamic and uncertain business environment, and with the growth of intense competition and vibrant globalization, a demand by the companies for information, both internal and external. This environment in being used by the companies as a basis to decision support systems for integrating and facilitating consulting a great number of data in different origins. The appearance of this environment occurred due to the evolution of the organizations in the last few decades. As described by MARAKAS (1999), the DW environment provides the facility for integrating the data generated in a world of non integrated information systems, allowing for this objective to be reached.

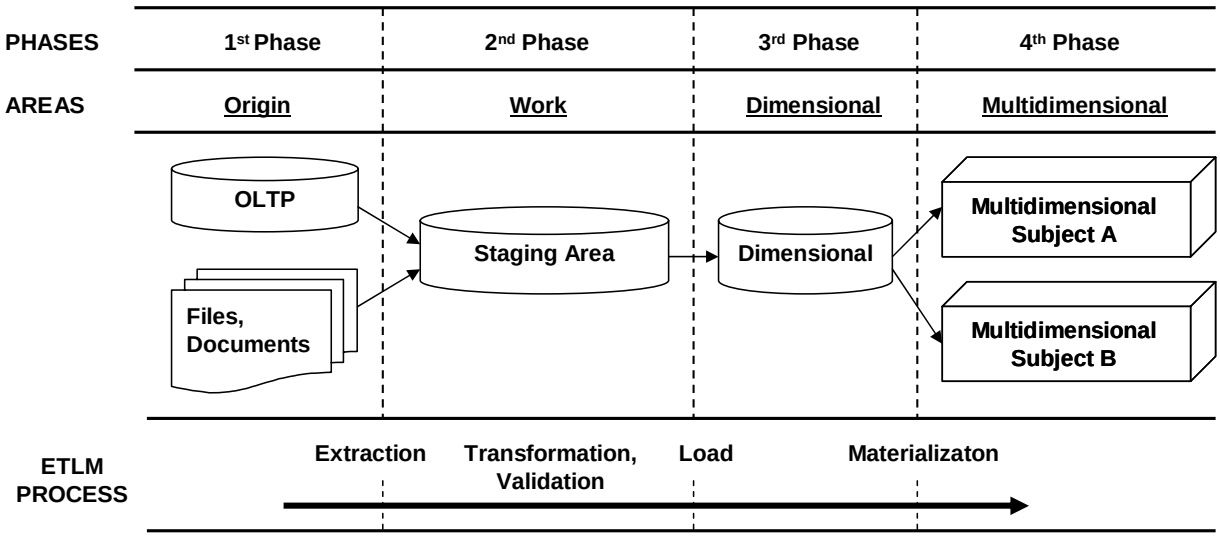

Figure 1 - The framework for the DW environment

According to KELLY (1996) the essence of DW concept it is the acknowledgment that the features an standards of use for the operational systems to automate the business processes and there decision support systems are fundamentally different but, despite all they are symbiotically linked.

In this paper, we developed an improved technique derivate from the comparison of database snapshots approach, in which we use signatures to mark of and detect the changes occurred in data origin.

The rest of the paper is organized as follows: In section 2, we present the framework for the DW environment and some of the main challenges involving the ETLM process. In section 3, we have described the main approaches which may be used for solving the problem of detecting change in data origin. In section 4, we have described the implementation of our technique. In the section 5 and last part, we present our final considerations and current works aiming at the improvement of this implementation.

2. The Framework

As a definition for the DW concept, we have mentioned INMON (2002), in which the DW is an integrated collection of subject-oriented databases, designed to assist the decision support function, in which each data unit is non volatile and relevant at a given moment in time.

According to MARAKAS (1999), there are two implicit premises in this DW concept, the first that it should be physically segregated of any other operational system, in the second that it keeps aggregate data and transactional data with management segregated from the OLTP systems.

The need for a segregated environment for the DW is an essential element for its concept in most cases, systems developed for operational environment are improper under several aspects to meet the analyses and decision making process whereas the DW environment was designed to meet this need.

The framework proposed for the DW environment follows the organization and the operation process as presented on Figure 1 and detailed as below. The organization of DW environment is normally subdivided into four phases, and has its main objective keeping the independence among them during their functioning. As we may observe, this is due to the fact that each one of the phases has its own characteristics and complexities, according to each environment. The logical segregation of each phase aims at the conceptual organization of the environment; and the physical segregation aims at meeting performance as each one of this phases demand time, space and processing.

The areas are segregated according to the processes which would be carried out in each phase. The set of all the process is known as ETLM (extraction, transformation, load and materialization).

In the first phase, we have the area of data origin and the extraction process. Depending on the knowledge of how and where the data origin is stored, we may have greater ease or difficulty in extraction process. In this process the data is obtained in the origin area and stored in the work area.

In the second phase, we have the work area and the processes for transformation and validation. At the end of the extraction process we will be able to carry out all the activities needed for the transformation and validation of this data as from the data stored in the work area. At the

end of this process, we obtain the valid data and the data which have been rejected with the information on which rules of validation have been violated.

In the third phase, we have the dimensional area and the load process. As from the valid data obtained at the end of the transformation and validation processes, we carried out the load process. In this process, we choose for each table, the implementation policy for changes which took place in the data, and we store it the dimensional area. Usually the chosen policy follows one of the slowly-changing dimensions (SCD) policy (KIMBALL, 1996; KIMBALL, 1998), and the data modeling standard as the one of dimensional modeling (star schema) (KIMBALL, 1996), in which fact and dimensional tables are used for representing data.

In the fourth and last phase, which is optional, we have the multidimensional area and the materialization process. In this process the relevant aggregations for best performance of queries, which are being made by the users, are carried out. In this area the data is stored in a multidimensional format. However as said before, this phase may or may not be implemented. This definition will depend on the complexities of the queries and on the amount of data to be handled by the users. In case the data stored in the third phase fully meet the query performance needs, this phase may not be implemented.

Several multidimensional databases may be defined and materialized, separating them according to the need of each company.

Some important functionalities should be taken into consideration at each phase of the ETLM process, such as (INMON, 2002; KIMBALL, 1996): detection of changes in data origin, the cyclicity of data, the complexity for data selection in the operational environment, the operational environment technology may be different from the DW environment, there may be several origins of data, efficiency in the choice of data to be extracted, understanding of the logic of the original data relationships, analyzing solutions for massive input data volumes, the operational input key generally need to be structured before they are recorded, formatting the data for define standard for the DW, carrying out the data cleaning process when possible, among others.

3. Detection of Changes

There are several challenges which would be overcome in the development of a DW environment. However, in this article, the main focus will gravitate around the detection of changes in data origin. According to LABIO, YERNENI et al. (1999), one of the main problems is updating derived data when the remote information sources change. For the cases in which the operational environment does not bear or does not want to inform the storage of changing history taking place on data, controls have to be implemented in order to meet this need. The main scenarios for these cases are: i) the impossibility to instrument the DBMS (triggers, transaction log, stored procedures, replication, materialized views, old and new versions of data, etc) due to security policies, data property or performance issues; ii) the lack of instrumentation resources on the DBMS; iii) the use of legacy technologies such file systems or semi-structured data; iv) application proprietary databases and ERP systems.

There are several approaches which may be implemented in detection of changes in data origin. In order to choose among them, we have to take into consideration the following factors: there are advantages, disadvantages, and the existence of the necessary features in the operational and DW environments. Generally, in the implementation of these approaches, we involve the processes of extraction, transformation / validation and load. According to KIMBALL (1996; 1998), the processes of extraction, transformation / validation are the slowest in the DW project, generally consume 60% of the entire development time.

The changes which have to be detected may be classified in tree types: insert of new data, update and delete of already existing data. Below, we will describe some of the most of used approaches for the solution of the change detection problem in data origin. We will use as a basis for this description, the framework for the DW environment in Figure 1.

A) New table with all the performed changes

Operational: for each table in the data origin area in which the mapping of changes is necessary, we have created a “daughter table” related to the originating table, identifying the following: the change operation which was performed, when it occurred, and in the case of alterations, which fields have been the altered and their values prior to the change. In this case, we generally take into consideration, in the mapping, only the changes performed in the relevant fields to the DW extraction process.

Necessary features: that the data from the origin area be stored in a DBMS; that the a DBMS bearing a trigger mechanism, with the approach of “old” and “new” versions implemented, for the operations of insert, update and delete. This approach was mentioned WIDOM & CERI (1996); that the using of triggers be permitted, as, in several cases, due to security policies, data property or performance issues, it is not possible.

Advantages: assurance the entire mapping of changes performed in the originating table have been stored; facility and greater speed in the extraction process, as only the lines changed from the last extraction performed would be consulted.

Disadvantages: overload on the operational environment, due to the need treatment in inclusion of mapping of each change taking place; increase in the need for space and storage of this new data and indexes; need for cleaning management, from time to time, of the mapping tables which beard a very large growth.

When to use: when the mapping of all changes performed in the data origin becomes really necessary.

B) Marking on originating table for storing the last change carried out

Operational: for each table in the data origin area in which the mapping of changes becomes necessary, we have created some columns identifying the change operation performed and when it took place. In the case of the change operation, we only mark when there has been a change in one of the fields relevant to the DW extraction process. Approach presented in (CRAIG, VIVONA et al., 1999).

Necessary features: as abovementioned "approach A".

Advantages: decrease in operational environment overloading this *à vis*, "approach A"; decrease in the need for storage space regarding "approach A"; is an greatest speed in the extraction process, as only the lines changed as from the last extraction performed will be consulted.

Disadvantages: the entire mapping of changes taking place in the originating table is not stored, only the last change; increase in the need for storage space of these new fields; the removals must be logical; the need for cleaning management, from time to time, on the originating table for the lines marked as logical removals which have already been extracted for the DW.

When to use: when the mapping of all changes performed in data origin is not necessary, only the last change.

C) Merge "approach A" and "approach B"

Operational: for each table in the data originating area, in which the mapping of inclusion and removal changes is necessary, we have created some columns identifying the operation of change performed and when it took place. In the case of a change operation, we have created a "daughter table" related to the originating table, identifying the change operation performed, when it took place in what have been the fields altered and their values prior to the change. We generally take into consideration, in the mapping, only the changes performed in the field relevant to the DW extraction process.

Necessary features: as abovementioned "approach A".

Advantages: assurance the entire mapping of changes performed by change operation in the originating table have been stored; facility and greater speed in the extraction process, as only the lines changed from the last extraction performed would be consulted; decrease in operational environment overloading this à vis, "approach A"; decrease in the need for storage space regarding "approach A";

Disadvantages: In case of update operation we have the same disadvantages as abovementioned "approach A". In case of insert and delete operations we have the same disadvantages as abovementioned "approach B".

When to use: when the mapping of all changes for the update operations is necessary and when the mapping of the all changes for the insert and delete operations, performed in the area of data origin is not necessary.

D) Interpretation of transaction log on the DBMS

Operational: as from the DBMS transactions log, we interpret all transactions carried out for each table in the data origin area, in which the mapping of changes performed is necessary. Approach presented in (WIDOM, 1995; DO, DREW et al., 1998; CRAIG, VIVONA et al., 1999).

Necessary features: that the data from the origin area be stored in a DBMS; this DBMS should have implemented the transactions log control; that the translation from the DBMS transactions log of SQL commands for insert, update and delete should be possible. This translation maybe performed through a specific tool from DBMS or through the development of a proprietary translator, but this last option should depend on the feasibility on understanding of transaction log control implemented by DMBS.

Advantages: there is no overload on the operational environment; decrease the necessity of space for storage, communication and processing; ease and greater speed in the extraction process, as only the lines changed since the last extraction performed would be consulted.

Disadvantages: need for controlling the transaction log area size by the database administrator (DBA), in order to prevent transaction lost. This may occur when the cycling of the transaction log takes place, that is, an area from the transaction log, with transactions already commit, maybe reused by the DBMS; the possibility of a physical or human failure in the transaction log file copy in a segregated area before the beginning of the backup process.

When to use: when the mapping of all or part of the changes occurred in data origin is necessary; possibility risk of lost of changes because of human, physical failure or by an unexpected operation by the DBA.

In approaches A, B and C, because they cause operational overload, its necessary that the operational environment beard slack in the processing of execution / recording and in space for storage.

E) Comparison of database snapshots

A number of methods may be used to perform a copy of data, from the origin area to the work area, necessary for the implementation of this approach. Depending on each one of these methods, the necessary features, advantages e disadvantages and when to use, would be complemented, when necessary in the detailing of each method. We present below the items common to all the methods of this approach. This approach was mentioned in (WIDOM, 1995; HAMMER, GARCIA-MOLINA et al., 1995; CHAWATHE & GARCIA-MOLINA, 1997; CRAIG, VIVONA et al., 1999).

Operational: for each table in the data origin area, in which the mapping of changes is necessary, we will perform a copy of the table of the origin area, for the work area. In this copy only the fields pertinent to the DW extraction process will be considered. We have identified this new table under "<name of table>_current". In case the extraction process is being carried out for the second time, there will be the analogous table to this new table, identified as "<name of table>_previous". These two tables will be used to verify the changes taking place. These changes will be found in the comparison between their data. In case the extraction process is being performed for first time it will not be necessary to carry out the comparisons.

Necessary features: that the data from the origin area be stored in a DBMS; a large storage space in the work area for the two versions the "current" and the "previous", for each table in which the mapping of changes is necessary.

Disadvantages: use of processing and recording time in data copy from the origin area to the work area and on data comparison, for the detection of changes an each of the tables; use of large storage space in the work area for the two versions the "current" and the "previous", of each table in which the mapping of changes is necessary.

When to use: in environments in which the DBMS in the origin area does not bear trigger mechanisms; or when the implementation of triggers is not allowed; or where there is no transaction log in DBMS; or where there is no possibility for translating the transaction log; or when there are problems and/or low performance access to the origin area data.

E.1) Copy method: Bulk copy

Necessary features: the DBMS in the work area should have bulk copy method.

Advantages: there is no overload on the daily routine of the operational environment in the origin area as all activities are to be performed in the work area; easiness in the extraction process as all table data will be load, with no need for any rule or control.

Disadvantages: we have no way to obtain the mapping of all changes performed in the originating table, only the last position at the moment of the extraction process.

E.2) Copy method: Replication

Necessary features: the DBMS at the originating and the work areas must have the necessary functionalities for the control and execution of replication process. Method mentioned in (CRAIG, VIVONA et al., 1999).

Advantages: ease in the extraction process as the entire work would be performed by the DBMS itself.

Disadvantages: the DBMS in the originating area must be the same as in the work area for the replication to be performed (or use some third-part heterogeneous replication tool); depending on how the replication process is being implemented by DBMS, there is the possibility of one not having how to obtain the mapping of all changes performed in the originating table; need for the administration and treatment of the replication conflicts by the DBA.

E.3) Copy method: Trigger (two phase commit protocol)

Necessary features: the DBMS at the originating and the work areas must have the trigger mechanism for the operations of insert, update and delete; that the DBMS in the originating area must be the same in the work area; that the DBMS should allowed for the performing of the operations of insert, update and delete of the triggers implemented accessing another database, local or remote.

Advantages: facility in the extraction process as the entire work would be performed by the DBMS.

Disadvantages: operational environment overloading; the DBMS in the originating area must be the same as in the work area; the DBMS in the originating and the work areas must be operating, as the stopped in the operation in the DBMS in the work area prevents the performing of the operations of insert, update and delete from the originating area for the tables using this approach; in order to obtain the mapping of all changes occurred in the originating table, we have to complement this approach using the "approach A" or "approach C", in the work area tables.

F) Refresh tables

Operational: for each table in the origin of data, in which the mapping of changers it is necessary, we have removed all data from the fact and dimension tables and performed the entire process of extraction and load again. This approach was mentioned in (WIDOM, 1995; KIMBALL, 1996; KIMBALL, 1998).

Necessary features: none in special.

Advantages: there is no need for implementing any control for change detection both in the origin and work areas.

Disadvantages: use of a large processing time of each new extraction, transformation / validation, load and materialization process as the process would be redone

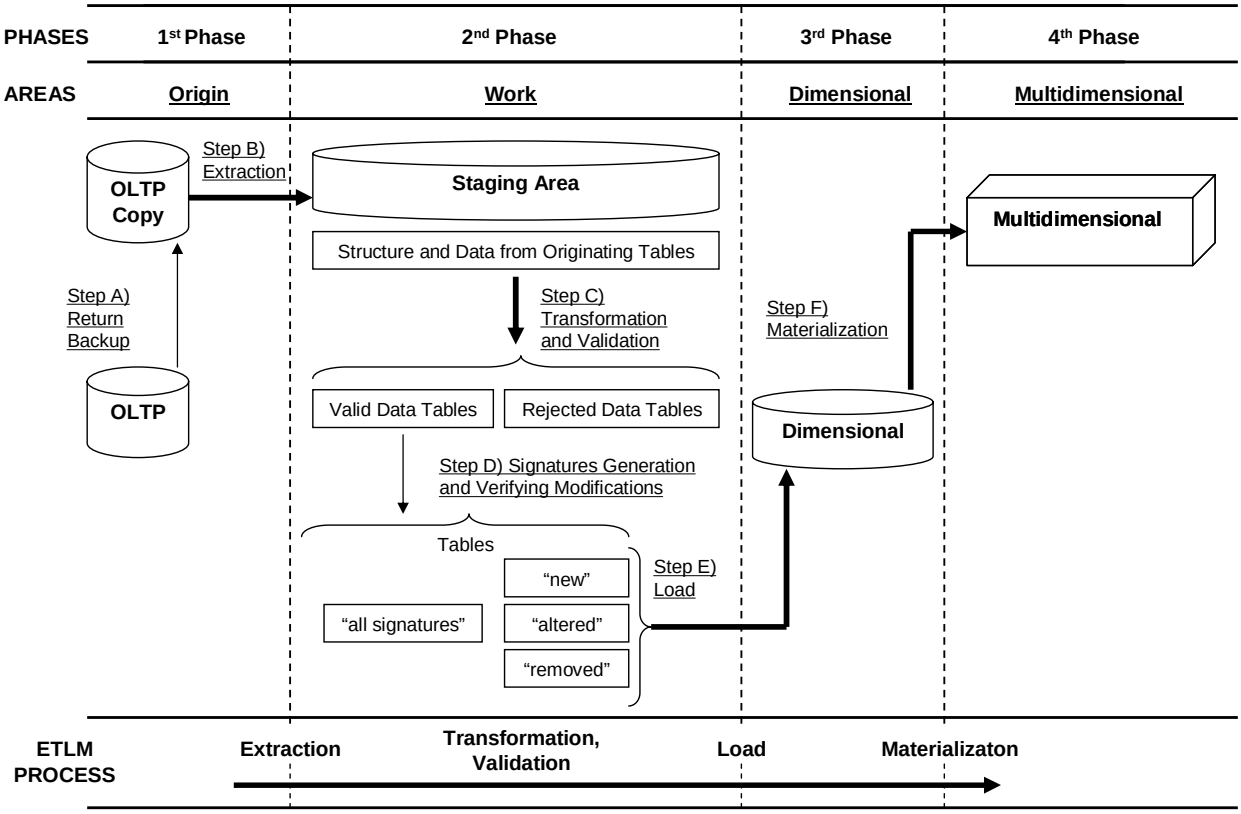

Figure 2 - The DW environment framework used at Rio-Sul Airlines

each time; the mapping of changes performed is not fully done as we only have the last position; lost of DW information history, as at each new ETLM process we would be recreating its data as from the last position in the operational environment.

When to use: when the mapping of all changes performed in data origin is not actually necessary; there is no need for keeping DW information history; there is no time restriction, as we will recreate entire DW environment at each new processing.

4. An Improved Approach

In our DW development and implementation at Rio-Sul Airlines, we have used as a basis the framework previously presented at Figure 2.

For the choice of the approach to be implemented in the solution of problem of change detection at data origin, we have taken into consideration the features found in the Rio-Sul Airlines operational and DW environment.

The main feature of this environment refers to DBMS in which the originating data was stored. The system has serious limitations both technical and policy-wise, such as: i) the inexistence of a instrumentation resource of triggers, replication and transaction log; ii) low performance in querying processing, leading to overload in the operational environment; iii) limitations in data access via ODBC; iv) concurrency problems; v) proprietary application database; vi) great difficulty for DBMS maintenance; vii) high data volatile; viii) an interrupted use of DBMS by users (24x7).

The technique we have developed and implemented derivate from the comparison of database snapshots ("approach E"), in which we use signatures to mark and detect changes. We describe below, in a step by step manner, how we have solved each of the problems found.

In this environment, we found our first problem: due to the low performance in query processing, the concurrency problems, the access difficulties and the uninterrupted use of the DBMS. How to obtain data from the operational environment without rendering unfeasible the utilization of these systems by the user?

For the solution of this problem we had two possibilities: the first would be, increase the CPU processing and the access / recording to the production machine disk; and the other option would be having another separate machine to enable us to copy production data and, from this data, perform the extraction process.

We use the second solution, with the smaller machine, in every weekend, we secured the backup from the previous night and we went up in this new machine (step A, Figure 3). As from this data, we started the

extraction process (step B, Figure 3) using the method of bulk copy (approach E, method E.1). At the end of this process, we had the tables and the data relevant to the DW project in the relational model, without foreign keys and with some indexes created, aiming at facilitating queries to be performed next.

As from this data, we carried out the data validation and transformation process (step C, Figure 3), having as a result, the information of valid lines and of rejected lines. For the valid line information, we marked the lines which had gone through all validations, using only a new filed created in the tables charged in the extraction process themselves. For the rejected line information, we created a new table with the relevant information from the problem table / line, identifying also the reason for rejection, aiming at facilitating the adjustment of incorrect data in the operational environment later.

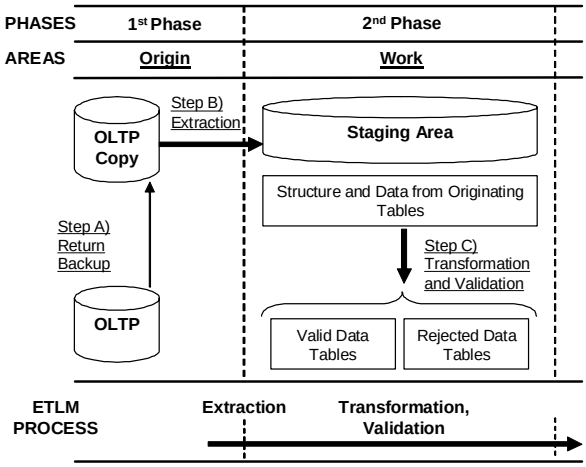

Figure 3 - Logical sequence of steps A, B and C

Our second problem was the follow: there were no resources in DBMS for trigger, replication and transaction log instrumentation. How to detect the changes through time which have occurred at data origin?

Due to these limitations of the DBMS, we had no other solution then use the approach of comparison of database snapshots ("approach E"). This approach could be used as there was no need to map all the changes which had occurred in data origin, only the last situation met the company needs.

With the purpose of improving the two main disadvantages of the approach of comparison of database snapshots ("approach E"), which are, namely, the processing time for comparisons in the verification of changes which have occurred and the space for storage of table copies; we implemented the following improvement: for each table which is the origin of data in fact and dimension tables and in which the detection of change in data origin was necessary, we created a new table in which we had the primary key of the data originating table and a signature field.

This signature is calculated using the CRC (Cyclic Redundancy Check) algorithm, taking into consideration the bringing together of all relevant fields on each table. In our implementation, the occurrence of collisions was very unlikely, as the queries performed on the tables with the suffix of "all signatures" were always carried out as from the primary key of the table, and not as from the signatures created. Therefore, the signatures were only used to detect whether there has been any change to the original data.

The implementation of signatures may also be used for cases in which the table of data origin does not bear a primary key or an identifier. This situation was mentioned by CHAWATHE, RAJARAMAN et al. (1996). In this case, we may use the signature, generated as from the relevant fields of the table under discussion, as the line identifier. Depending on the diversity of data, the use of a CRC algorithm for calculating the signature may have a greater probability of causing collisions. In this case, the use of a more complex algorithm is necessary to minimize this risk. In most cases, given the statistical nature of the use of warehouses, the missing of one operation will be of no harm.

In order to help in the identification and the query of the changes which took place, for each table in the process we have created four new tables: the three first are comprised by the primary key of the originating table plus the signature field of the line. And the fourth is comprised only by the primary key of the originating table. We describe the use of each one as follows.

The first table is used to keep all values of the last signatures. This new table have its name defined using the name of the table of data origin plus the suffix "all signatures".

The second table is used to full fill a situation of new which had lines which have been included. This new table have its name defined using the name of the table of data origin plus the suffix "new".

The third table is used to full fill the situation of lines which had any of there relevant fields changed. This new table have its name defined using the name of the table of data origin plus the suffix "altered".

The fourth table is used to full fill the situation of lines which have been removed. This new table have its name defined using the name of the table of data origin plus the suffix "removed".

We will describe, as follows, how the process of calculating and verifying signatures works (step D, Figure 4). For each table in which the detection of changes in data origin was necessary we obtained the lines marked as valid and carried out the calculation of the signature using the CRC algorithm. At each line calculated, we checked on the reference table with the suffix "all signatures",

using the primary key as a basis for queries. If the result of the query does not return any line, we include the data in the reference table with the suffix "new". If returns, we compare the signature calculated with the signature stored in the reference tables with the suffix "all signatures". In case the signature is different, we include a new line in the reference table with the suffix "altered" otherwise the line is the same as the last load. Finally, we move to the next line.

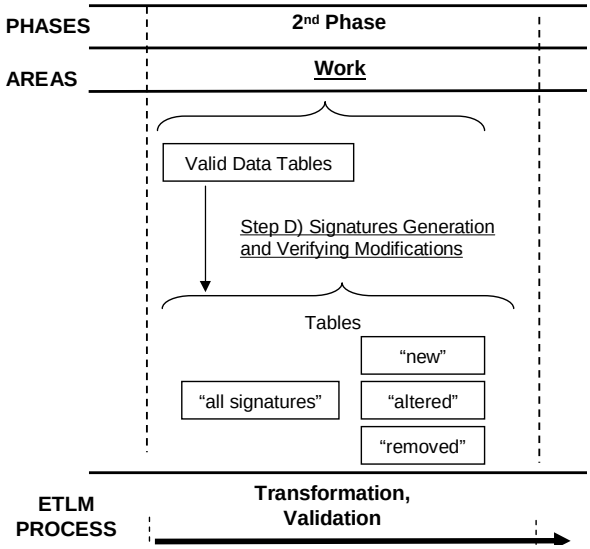

Figure 4 - Logical sequence of step D

At the end of the process for checking the inclusion and alteration modifications, we started the process to identifying the removals. For all tables with the suffix "all signatures", we queried as from the primary key, whether the line existed in the table of data origin referring to the table which is being checked. In case the query does not return any value, this means that the original line has being removed. In the end we include a new line in the reference table with the suffix "removed".

At the end of this process, as from data existing on tables with suffix "new", "altered" and "removed", we will start the load process (step E, Figure 5). In this process, we have follow the logical order, aiming at avoiding failures and the adjustments which will be performed on the dimensional model data, according to each case. As we describe below:

Step 1: Processing of new lines (dimensions)

- Query to the tables with the suffix "new", for the dimension origin tables;
- Inclusion of these new lines on the referenced tables with the suffix "all signatures";
- Inclusion of these new lines on the dimensional tables.

Step 2: Processing of altered lines (dimensions)

- Query to the tables with the suffix "altered", for the dimension origin tables;
- Alteration of the signature of these lines on the referenced tables with the suffix "all signatures";
- Alteration of data of these lines on the dimension tables, choosing one of the slowly change of dimension policies, defined in (KIMBALL, 1996; KIMBALL, 1998).

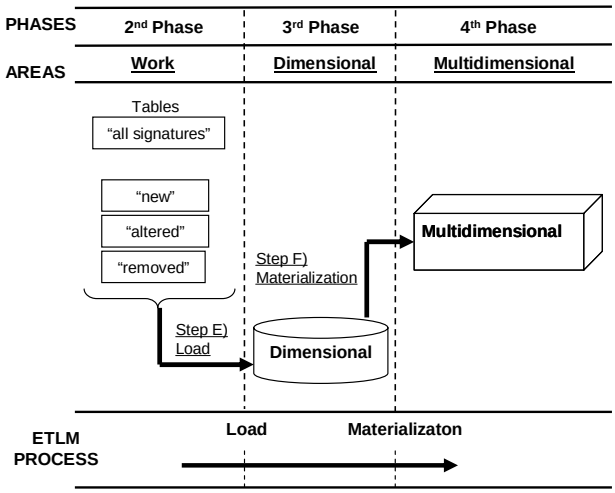

Figure 5 - Logical sequence of steps E and F

Step 3: Processing of new lines (facts)

- Query to the tables with the suffix "new", for the fact origin tables;
- Inclusion of these new lines on the referenced tables with the suffix "all signatures";
- Inclusion of these new lines on fact tables.

Step 4: Processing of altered lines (facts)

- Query to the tables with the suffix "altered", for the fact origin tables;
- Alteration of the signature of these lines on the referenced tables with the suffix "all signatures";
- Alteration of data of these lines on the fact tables, using the type one of the slowly change of dimension policy, as defined in (KIMBALL, 1996; KIMBALL, 1998).

Step 5: Processing of removed lines (facts)

- Query to the tables with the suffix "removed", for the fact origin tables;
- Removal of these lines from the referenced tables with the suffix "all signatures";
- Removal of these lines from fact tables.

Table 1 - Comparison of Main Approaches for Detection of Changes in Data Origin

Approach	Need OLTP DBMS Instrumentation	Space Requirements	OLTP Overload	Change History	Cleaning Management
"A"	Yes (triggers)	Medium	High	Complete	Yes
"B"	Yes (triggers)	Low	Medium	Some	Yes
"C"	Yes (triggers)	Medium (lower than "A")	High (lower than "A")	Almost Complete	Yes
"D"	Yes (transaction log)	None	None	Complete	No
"E1"	No	High	None	Some	No
"E2"	Yes (replication)	High	Low	Some	No
"E3"	Yes (triggers)	High	Medium	Some	No
"F"	No	None	None	None	No
Our technique	**No**	**Medium**	**None**	**Some**	**No**

Step 6: Processing of removed lines (dimensions)

- Removal of these lines from the dimension tables; in case there is any removal problem on account of violation of a reference validation rule, the line should not be removed and the following steps should not be carried out, and remove on towards the removal of the next line. This may occur, as the DW keeps historical data, and that, in this case, only operational environment references do not exist. Problem mentioned in (CRAIG, VIVONA et al., 1999).
- Query to the tables with the suffix "removed", for the dimension origin tables;
- Removal of these lines from the referenced tables with the suffix "all signatures".

At the end of the load process, we have the data stored on fact and dimension tables according to the standard defined in (KIMBALL, 1996; KIMBALL, 1998) for dimensional modeling. The next and final process is the materialization (step F, Figure 5), in which, as from the more relevant aggregate definitions, we process and the store the data in the multidimensional format.

We show in Table 1 a comparison of the main characteristics of all approaches.

5. Conclusion

According to LABIO, YERNENI et al. (1999), due to the constantly increasing size of warehouses and the rapid rates of change, there is increasing pressure to reduce the time taking for warehouse update. To meet this demand, this article had as its main purpose presenting our experience in the development of an improved approach derived from comparison of database snapshots, minimizing two main disadvantages with the use of signatures, calculated by CRC algorithm. These disadvantages are: the processing time for comparisons in the verification of changes which have occurred and the space for storage of table copies.

This development was done considering the company operational and DW environments features, existing at the Rio-Sul Airlines.

During this work, we followed the steps presented in this article in which we start by defining a framework for the DW environment. Following that, we study the main approaches for the detection of changes in data origin.

As current work, we aim to carry out improvements to our approach, with the purpose of increasing the performance through out the process. The first one is the use of parallelism for the signature calculation process. In the second improvement, we will create a signature structure organized as a tree. This will separate the lines from each table in groups, where the signature calculation will be carried out for this group of lines, thereby assembling a signature tree. The main objective is to expedite the signature comparison process. These comparisons will be started at the root node, which represents a group of signatures, and only if differences are found in the signatures of a higher level in the tree that we will go down in levels and carry out new comparisons. This process occurs continuously until we reach the comparison of signatures on leaves in which the modifications have been identified.

References

CHAWATHE, S. S.,GARCIA-MOLINA, H., 1997, "Meaningful Change Detection in Structured Data". In: *Proceedings of ACM SIGMOD International Conference on Management Data*, pp. 26-37, Arizona, USA, May.

CHAWATHE, S. S., RAJARAMAN, A., GARCIA-MOLINA, H., et al, 1996, "Change Detection in

Hierarchically Structured Information". In: *Proceedings of ACM SIGMOD International Conference on Management Data*, pp. 493-504, Montreal, Canada, June.

CRAIG, R. S., VIVONA, J. A., BERKOVITCH, D., 1999, *Microsoft data warehousing building distributed decision support systems*, New York, Wiley.

DO, L., DREW, P., JIN, W., et al, 1998, "Issues in Developing Very Large Databases". In: *Proceedings of the 24th VLDB Conference*, pp. 633-636, New York, USA, August.

FINNEGAN, P., MURPHY, C., O'RIORDAN, J., 1999, "Challenging the Hierarchical Perspective on Information Systems: Implications from External Information Analysis", *Journal of Information Technology*, v. 14, n. 1, pp. 23-37.

HAMMER, J., GARCIA-MOLINA, H., WIDOM, J., et al, 1995, "The Stanford Data Warehousing Project", *IEEE Quarterly Bulletin on Data Engineering; Special Issue on Materialized Views and Data Warehousing*, v. 18, n. 2, pp. 41-48.

HULL, R.,ZHOU, G., 1996, "Towards the Study of Performance Trade-offs Between Materialized and Virtual Integrated Views". In: *Proc. Workshop on Materialized Views: Techniques and Applications (VIEWS 96)*, pp. 91-102, Montreal, Canada, June.

INMON, W. H., 2002, *Building the data warehouse*. 3rd ed, New York, Wiley.

INMON, W. H.,KELLEY, C., 1993, *Rdb/VMS, developing the data warehouse*, Boston, QED Pub. Group.

KELLY, S., 1996, *Data Warehousing: The Route to Mass Customization*, New York, USA, John Wiley & Sons, Inc.

KIMBALL, R., 1996, *Data Warehouse Toolkit*, New York, USA, John Wiley & Sons, Inc.

KIMBALL, R., 1998, *The Data Warehouse Lifecycle Toolkit. Expert Methods for Designing, Developing, and Deploying Data Warehouses*, New York, USA, John Wiley & Sons, Inc.

LABIO, W. J., YERNENI, R., GARCIA-MOLINA, H., 1999, "Shrinking the Warehouse Update Window". In: *Proceedings of ACM SIGMOD International Conference on Management Data*, pp. 383-394, Philadelphia, USA, June.

MARAKAS, G. M., 1999, *Decision Support Systems in the 21st Century*, New Jersey, USA, Prentice Hall Inc.

ÖZSU, M. T.,VALDURIEZ, P., 1991, *Principles of Distributes Database Systems*. 1st Ed, New Jersey, USA, Prentice Hall Inc.

QUASS, D., GUPTA, A., MUMICK, I. S., et al, 1996, "Making Views Self-Maintainable for Data Warehousing". In: *Proceedings on Parallel and Distributed Information Systems*, pp. 158-169, Miami Beach, Florida, USA, December.

QUASS, D.,WIDOM, J., 1997, "On-Line Warehouse View Maintenance". In: *Proceedings of ACM SIGMOD International Conference on Management Data*, pp. 405-416, Tucson, Arizona, USA, May.

WIDOM, J., 1995, "Research Problems in Data Warehousing". In: *Proceedings of ACM CIKM International Conference on Management Data*, pp. 25-30, Baltimore, USA, November.

WIDOM, J.,CERI, S., 1996, "Active Databases Systems: Triggers and Rules for Advanced Database Processing.", San Francisco, California, USA.

ZHUGE, Y., GARCIA-MOLINA, H., HAMMER, J., et al, 1995, "View Maintenance in a Warehousing Environment". In: *Proceedings of ACM SIGMOD International Conference on Management Data*, pp. 316-327, San Jose, California, USA, June.

ZHUGE, Y., GARCIA-MOLINA, H., WIENER, J. L., 1996, "The Strobe Algorithms for Multi-Source Warehouse Consistency". In: *Proceedings on Parallel and Distributed Information Systems*, pp. 146-157, Miami Beach, Florida, USA, December.