Delivering and Visualization of Data in a *Call Center* Data Warehouse

Extended Abstract

Edgar Alexandre Gertrudes Guerreiro[1]
Professor Orientador: Helena Galhardas[1]
Co-Orientador: Eng. João Damásio[2]

[1]Instituto Superior Técnico
[2]Business Intelligence Department, Link Consulting

September 29, 2008

Abstract

The growing demand of customers in their interactions with organizations, as well as the growing competition between these organizations, leads the search for solutions capable of optimizing the organizations business processes. One of these organizations where this fact is especially relevant is the *Call Center*. By daily dealing with a great number of clients, a huge amount of data is generated. This data as to be rapidly and thoroughly analyzed, so that operations can be improved, by reducing costs or by improving service quality.

In this master thesis is described the process of creating a system for decision support, or *Business Intelligence* (BI), that intends to facilitate the decision making at the *call centers*. This system consisted in the definition of the relevant information that had to be analyzed accordingly to the business requirements, the transformation, extraction and load (ETL processes) of this information. It were also created the components, *Dashboards*, responsible for the presentation of the information to the *call center* actors, in an easy and simple way.

With the BI system complete, the ET processes were evaluated, to determine their performance and adaptability to the business requirements. Due to the fact that *benchmark* tests, are still very limited to ETL processes, we opted for the use of generic frameworks, that evaluated the system created as a whole. With this evaluation it was still possible to study the adaptability of *Microsoft Integration Services* in creating these type of processes.

Keywords: *Call Center, Business Intelligence, ETL, Dashboards, Microsoft Integration Services*

1 Introduction

A call center [1] [2] [7] is formed by physical structures and people, that make and receive calls, respectively, outbound and inbound calls. These calls are aggregated in campaigns and each campaign as common goals, like selling a certain product. A call center besides call handling can also provide help desk services, handle faxes, e-mail and similar written correspondence.

The organizations defines with the *call center* a group of measures that have to be reached. These measures are usually called *Service Level Agreement* (SLA) and they allow organizations to control the *call center* performance. They correspond to groups of metrics important to the organization. In systems that help the decision support these metrics are called Key Performance Indicators (KPI), they are "significant predefined measures that provide individuals with the information they need to assess previous actions. KPIs define target performance and provide individuals with the ability to assess past performance" [14].

Normally the *call center* is the point of contact with the clients organization, in a communication that is made both ways: customers call to resolve problems, and the *call center* makes calls to sell products or inquire these costumers. In these communications a great amount of data is generated (like, customers preferences, call times, sales). These data must then be analyzed and processed accordingly to the KPIs defined. It is then imperative that the *call center* can be able to handle tese information in a fast and coherent way. It becomes obvious that a Business Intelligence (BI) system can greatly improve the *call center* in [1] [3]:

- Optimizing the daily operation and minimizing operational risks, by constant monitoring of the main KPIs;
- Improving service quality and customer relationships, by providing personalized and fast answers to the client;
- Reducing costs, by detecting and correcting certain problems in the operations;

Formally a BI system can be described as "a user-centered process of exploring data, data relationships and trends, thereby helping to improve overall decision making. This involves an interactive process of accessing data (ideally stored within the data warehouse) and analyzing it to draw conclusions, derive insights and communicate findings with the purpose of effecting positive change within an enterprise".

A typical BI architecture has four main stages:

- The data of the sources are submitted to *Extraction*, *Transformation* and *Loading* Processes (ETL);
- The data are stored in special structures, denominated *Data warehouse* or *data marts*;
- The data are manipulated trough the use os BI tools, in order to be visualized in *dashboards* or to be submitted to data mining processes;

[1] Although the term *call center*, refers mainly to call handling, having the term evolved to a new designation, *contact center*, that reflects the fact that there are more interactions, than just handling calls, in this article, we will use the term *call center*, indifferently, wether is a *call center* or a *contact center*.

[2] In literature sometimes the *call center* appears as *call centre* and the term *contact center* as *contact centre*.

- The data is shown to the end-users.

ETL processes are comprised of three stages. Extraction, extracts the data from their sources to a data staging area where they will suffer further modifications. Transformations is a stage were data is cleaned (example: correction of misspelling errors), purged (elimination of irrelevant data) and combined (creating unified views of data). Finally the data is loaded to multidimensional models (Data Warehouses, that are subject oriented, integrated and non-volatile collections of data) [10]. From the warehouses the data can suffer data mining processes (to extract further knowledge from data)[6] or be presented trough reports or dashboards. Dashboards are a special kinds of report that "consolidates aggregates and arranges measurements, metrics (measurements compared to a goal) on a single screen so information can be monitored at a glance" [12][2].

In this article it we developed a BI system for a *call center*, in cooperation with Link Consulting. We took special focus on the definition of the ETL processes, were we tried to summarize a generic methodology and the dashboards construction [8].

The main contributions of this work were:

- The resolution of an engineering project, with the objective to provide a *call center* with a BI system;
- Evaluation of the *Integration Services do Microsoft SQL Server 2005* module, mainly its operators and efficiency;
- Elaboration of a study with the state of art of BI tools for *call center*, as well as Benchmarks for ETL processes;
- Enabling *call center* users to generate their own reports in a simple manner using tools like *Microsoft Excel 2007*.

The rest of the paper is organized as follows. Section 2, presents an overview of BI tools for *call centers* and a study in the benchmarks and techniques to evaluate ETL processes. Section 3 presents the architecture of the BI system and the multidimensional model. Section 4 presents the ETL processes. Section 5 displays the dashboards constructed. Section 6 reports the results of the evaluations made to the BI system. Finally Section 7 presents the main conclusions.

2 Related Work

2.1 The BI tools market

In the market there are a great number of specific tools for *call centers*. There are also a considerable amount of tools that fit easily to *call center* needs, that is, tools that not being exclusive to the *call center* domain, have modules to this type of organizations. According to Gartner Research [11] and Forrester Wave Research [9], amongst the better tools for *call center* are *Genesys*, *Avaya*, *Business Objects* e *Oracle*. It is important to notice tough, that each *call center* is unique in terms of its necessities of a BI system, and so, it is important to study the *call center* needs to understand what tool fits better to it.

Although these tools are very helpful in constructing a BI system, sometimes the best approach is to build a new BI system from scratch, due to the price of such tools or specific *call center* requirements.

2.2 ETL Benchmarks

One of the main components of any BI system are the ETL processes. It is then important to understand if the current tools that provide ETL processes are able to handle the constant growing of data, that it is being observed in every organization [4]. It is then important to find ways to evaluate an ETL workflow. Unfortunately only recently, we started to see a preoccupation on creating frameworks that try to evaluate ETL workflows as a whole [5] [13], and the ETL industry as yet to set standard benchmarks for its many processes. One of this frameworks is the one suggested by *Vassiliadis*, *Karagiannis*, *Tziovara* e *Simitsis* [13], that describe, ETL as workflows, represented trough graphs. Their approach is generic to any ETL workflow, and then can be applied whatever the tool used to implement it. These authors consider two types of activities in a workflow: micro and macro activities. Micro activities are divided in Extraction, Transformation and Loading, and are operations that work on a set of rows. Macro activities describe the workflow of the system. They have invented the term "butterflies" to describe an workflow (see Figure 1). The left wing of the butterfly are the ETL activities, the center the data warehouses and the right wing the activities that use the data warehouses to reports and mining.

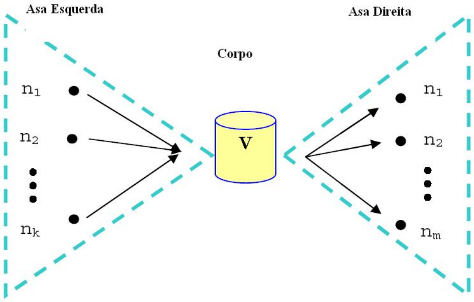

Figure 1: Generic butterfly of an ETL workflow.

These authors then came up with a generic set of tests that can be applied to any workflow. These tests are divided in two main types: effectiveness and efficiency. Effectiveness is the data completeness, consistency and freshness when it is used in decisions. In effectiveness we evaluate aspects like: percentage of data that violate business rules and resilience to failures. Efficiency pretends to assess how fast is the workflow executed and the resources it consumes.

3 Architecture

Figure 2 shows the architecture of the BI system constructed.

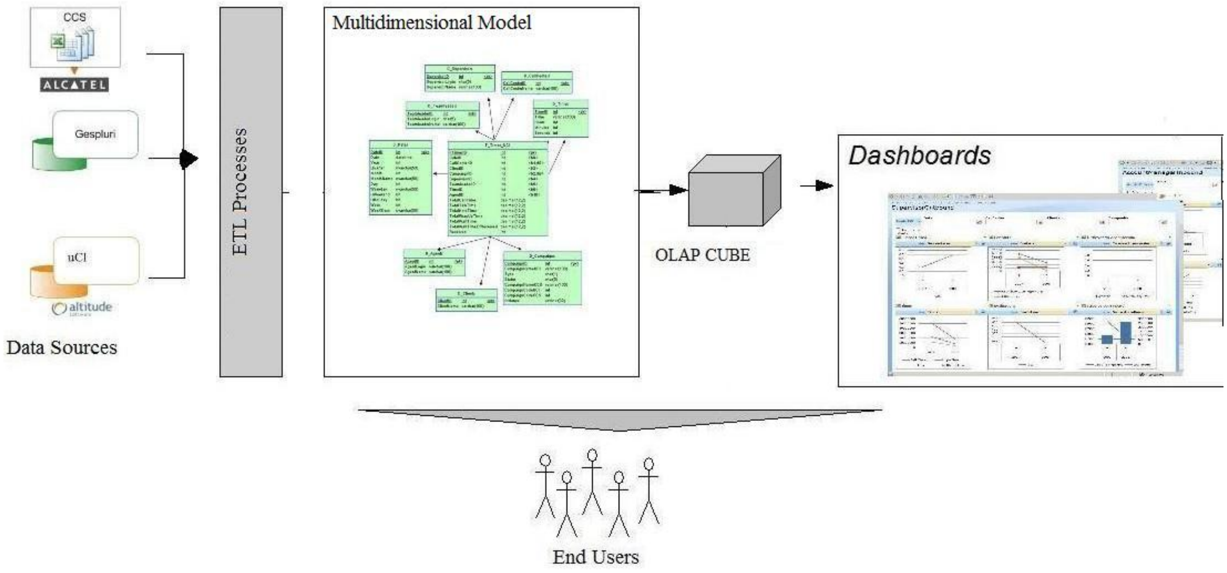

Figure 2: Architecture of the BI system constructed

The architecture constructed is based on a typical BI architecture. It has the data sources from the *call center* systems, a set of ETL processes that deploy the data onto multidimensional models. An OLAP Cube, that joins data from the warehouse systems and a set of dashboards that present the data to the end-users.

4 ETL Process

Figure 3 displays the methodology of the ETL processes.

There are two main stages in the ETL processe made: Diffusion and Integration.

In the *Diffusion* stage the data is extracted from the sources to the data staging area. Here they will be placed in tables that reflect the structure they had in the data sources with an additional field, *DiffusionID*, that will serve as a marker of the diffusion made.

The *Integration* stage consists of three main steps: Joining the data fields; grouping and calculating and loading the data in the multidimensional model.

In the first step we join the important fields we will use in later calculation in one single view. By joining these fields in one view, the calculations can then be made on a single place, and thus making the process easier. There are two main types of joins, *inner joins* and *outter joins*. Inner joins are used when the fields of the table being joined have a relation *one-to-one*, outter joins are used when the relation is *many-to-many*.

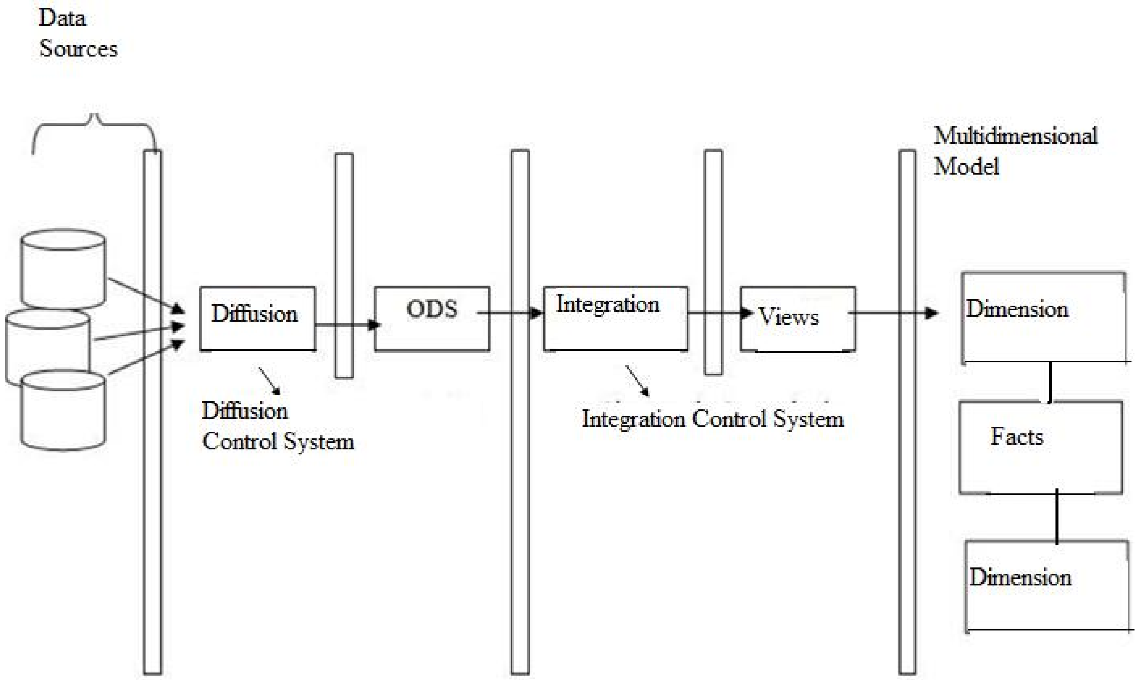

Figure 3: ETL methodology

The expression of join can be seen in Figure 4. Apart from joining the fields, we also make in this step minor formattings (like passing values in hours to minutes) and filter the irrelevant values (like values with no calls associated).

$$\pi_{B1,B2,C1,C2} (A \bowtie_{A1=A2} B)$$

Figure 4: Algebraic expression that represents an inner join

In the second step, we take the view from the first step and apply a set of calculations to it. These calculations usually follow the pattern over a set of views: if a given condition is verified then we make a calculation (like a sum or counting) for that row, if not, the value returned is zero. In the end of the second step the result is a view with all the business indicators calculated and ready to be loaded in the data warehouses.

Third step, just takes the view calculated in the second step, and deploys it on the data warehouse.

By dividing the ETL processes in these stages and steps we try to describe a methodology easy to follow and that can be applied to any ETL workflow.

5 Dashboards

The *dashboards* construction was an iterative processe, were the first prototypes were developed in paper. Only after the prototypes were validated we began making prototypes in *Microsoft Sharepoint 2005 Services*.

In the *dashboards* construction there was a precaution in designing them according the best standards, mainly those defined by Stephen Few [8]: not exceeding the boundaries of a single screen; make the *dashboards* accordant to the end-user needs; limitation of excessive detail; chose the right displays for the data; organizing the information correctly on the screen, highlighting the most important parts of information; do not introduce meaningless variety and not cluttering the display with useless decoration.

There was also an exhaustive research to find *dashboards* applied in the call center environment, so that the system created could be as close as possible as one expectable in a call center BI tool.

Although the dashboards suffered many iterations it is important to refer two. The first was the first prototype developed in *Microsoft Sharepoint 2005 Services*. These prototype showed data in a grid 3*2 (that is, three lines and two columns) and there was an effort in trying to maintain colour and graphics coherence. The main critics made were that the filters to select data, like the month, day, campaign, were not visible and there was not a clear separation from each graphic. In the second important iteration, the prototype was connected to the multidimensional model, so it could show real data. The main critics appointed before were corrected, and the layout due to the space occupied by the filters, was changed to a 2*3 grid. Figure 5 shows examples of these finals prototypes.

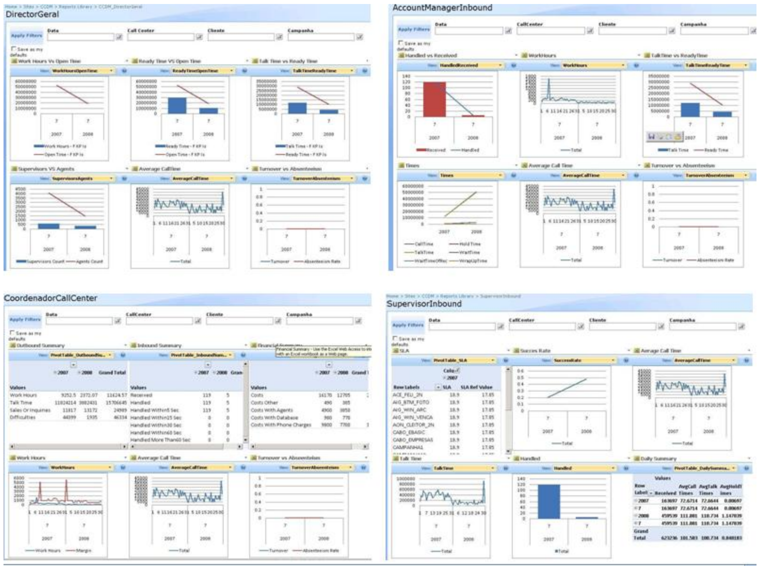

Figure 5: Examples of the dashboards from the final prototype.

6 Experimental Validation

In this section we will evaluate the ETL workflow developed using *Microsoft Integration Services 2005* to the specific domain of the *call center*. We evaluated this workflow under two parameters: effectiveness and efficiency. Effectiveness is the capacity to respect the business rules, mainly in providing useful and actual information when it is needed. Efficiency concerns the resources consumed by the solution, mainly time and memory. We will also make some observations concerning the *Microsoft Integration Services 2005* as a tool to construct ETL workflows to handle a big volume of data. In the tests made, some of the data had to be created in order to create certain conditions (like tests for failures and tests concerning a great amount of days). It is important to notice that all the testes were made in a virtual machine executing *Windows 2003 Server*, with 2500MB of RAM. This virtual machine is hosted in a *ACPI Multiprocessor X64-basedPC Intel(R) Xeon(R)* server with a *CPU* of 1,60Ghz.

6.1 Effectiveness

In order to respect the business rules, we have to understand what the minimum latency of decision making processes is. The most frequent decision processes are made in a daily basis. We assume then that from the moment that we extract data from their data sources to the moment this data is incorporated in a decision, cannot pass more than a day. In a typical day of operations the *call center* generates 1117700 rows of data to be analyzed. After tese data suffers the ETL processes it generates 58800 tuples to insert in the system. This amount of data takes about 56 minutes to be processed (an average of 0,057 seconds per tuple). This value clearly indicates that the solution respects the business rules. It is important tough to see exceptional situations, where for example, one day of operation cannot be processed, and then the next day has to process two days in a row. In a simulation made, were we loaded seven days of data, even there the solution only took seven and a half hours, which indicates that this process could be run at night and be available to be used in decision making processes in the morning. It is also important to know the amount of data that should be loaded onto the data warehouse but was not. In average only 0,073% of data that should be loaded in the warehouse is not. This data usually is not inserted due to lack of information to cross with this data.

6.2 Efficiency

We will evaluate the solution under two parameters: resilience to failures and resources consumed.

Regarding the resilience to failures, if the error is due to lack of files in one of the data sources, the solution can still load the rest of the data sources, and try to insert as many data as possible. If the error is on the data sources, because the solution was made in the assumption that no error will come from the data sources, in these situations the solution cannot recuperate from these errors. When the errors are from execution, like the system crashing, although the solution does not recover itself, it does provide a log, in which the user can consult to see what was the last module to run with success and then the user can retake the solution from these last module.

Regarding the resources consumed, we will focus mainly on time. As we said the solution from one day takes about 56 minutes. It is now important to understand which modules occupy the most time. Tests shown that about 75% of the time is consumed loading files from one of the sources used, this is mainly due to the fact that these sources are in an Excel format and *Microsoft Integration Services 2005* does not handle very well these type of files. Other big portion of time consumed is in the load of the data to the facts table, taking about 16% of the time. This time is mainly spent in the lookup operation, that is, seeing the references to that fact in the dimensions table. We decided to use the operator *Lookup* provided by the tool, and this operator revealed to be very slow (from all the loading time into the facts table, 88% was spent in the Lookup operation). Although we could not use this operator, making the solution faster, we decided for its use, because it provided a better control, by detailing in were the errors occurred, so the decision here was to sacrifice a bit of efficiency for more control.

6.3 Considerations about the ETL workflow developed

Regarding the limitation of (*Microsoft Integration Services 2005* we have to enunciate the lack of efficient operators, which leads to many times, having to code the operation using SQL. Also the poor ability to handle files like Excel is a downside of this tool.

In the design options taken, we have to talk about the decision to use views, instead of materialized tables. Although the use of views makes the solution more modular, its performance was clearly affected, because of the amount of operation the system had to do internally, consumed a lot of memory. In fact, if the data was near the millions, the time taken in computing the views could be huge, and sometimes the solution could simply not run.

7 Conclusions

This article described the development process to create a BI system for a *call center*. We were able to conclude the importance that such a system has in these type of organizations, in improving its processes (by reducing costs and augmenting service quality).

We took special care in the definition of the ETL methodology, mainly in taking an approach by modules that gradually would make all the ETL processes. This definition can be applied to any EL process, because it is generic and independent of the tool used.

After we searched ways to evaluate the performance of the ETL workflow constructed (a topic that is still poorly explored in the scientific literature). By applying a generic framework, that evaluates an ETL workflow as a whole, we were able to assess our own ETL workflow, and conclude it fairly met the requirements and environment it was meant to be used in.

Finally we made some considerations about the tool used in constructing the ETL workflow (*Microsoft Integration Services 2005*).

Besides the mentioned contributions, important aspects of future development should be pointed out:

- Realize more performance tests, mainly in the resources used by the system;
- Utilize tables instead of views, and assess the performance gains;
- Implement alarm mechanisms, so that the *dashboards* could be more pro-actively, by helping the user detecting and correcting problems;
- Test new tools to develop the ETL workflow and the *dashboards*;.

References

[1] Business intelligence in the contact center. *Viecore, Business Consulting Services*.

[2] Picturing performance: Dashboards and scorecards with cognos. *Cognos* (Novembro 2007).

[3] ABBOTT, J. C. The executive guide to call center metrics. *Robert Houston Smith Publishers* (2004).

[4] ALAN, J. High performance extract/transform/load benchmark. *RODIN Data Asset Management* (2002).

[5] BÖHM, M., HABICH, D., LEHNER, W., AND WLOKA, U. Dipbench toolsuite: A framework for benchmarking integration systems. *Database Technology Group, Dresden University of Techology* (2008).

[6] CHATERJEE, J. Using data mining for business intelligence. *MS SQL Server* (Janeiro 2005).

[7] DOWNEY, B. Call center vs. contact center. *Search CRM* (July 2001).

[8] FEW, S. *Information Dashboard Design*. O'Reilly, Janeiro 2006.

[9] GILE, K., MCNABB, K., MOORE, C., AND FOSSNER, L. The forrester wave: Bi reporting and analysis platforms, q1 2006. *Forrester* (Fevereiro 2006).

[10] KIMBALL, R. *The Data Warehouse Lifecycle Toolkit*. Wiley Publishing, Inc, 1998.

[11] KRAUS, D., AND ELLIOT, B. Magic quadrant for contact center infrastructure, north america. *Gartner RAS Core Research* (Agosto 2007).

[12] MORRISEY, R. Dashboards, analysys mind share, and kpi's. *Spectrum Magazine, September/October Edition* (2005).

[13] VASSILIADIS, P., KARAGIANNIS, A., TZIOVARA, V., AND SIMITSIS, A. Towards a benchmark for etl workflows. *ACM* (2007).

[14] WU, J. Visualization of key performance indicators. *DM Review Online* (Maio 2002).