Departamento de Ciências e Tecnologias da Informação

Implementation of Business Intelligence Tools Using Open Source Approach

Carlos Alberto Monteiro Gameiro

Tese submetida como requisito parcial para obtenção do grau de
Mestre em Open Source Software

Orientador:
Doutor Carlos J. Costa, Professor Auxiliar,
ISCTE-IUL

Outubro, 2011

ABSTRACT

Discovering business intelligence is the modern organization's way of gaining competitive advantage in their market, supported by Decisions Support Systems or Business Intelligence Systems. The first step in any decision support system is to create the repository of data for the system to collect and display any information requested. This repository is the source of all business intelligence and implementing it requires the right software tools, essential for the data warehouse. Therefore, when choosing the software tool, the project size, budget constraints and risks should be kept in mind. Overall the right choice depends on the organization's needs and ambitions.
The essential work to be done here is to demonstrate that open source software can be an accurate and reliable tool to implement data warehouse projects. The two ETL solutions used were:
• Pentaho Kettle Data Integration Community Editions (Open Source Software)
• SQL Server 2005 Integrations Services (SSIS) Enterprise Edition (Proprietary Software)
The proprietary, commercial software in question (as well as others) is widely used. However, an open source solution has key features recognized by organizations worldwide and this work will show the different functionalities and benefits of this open source approach.

Keywords
Business Intelligence, Data Warehouse, Open Source Software, ETL, Kettle

RESUMO

Nas organizações a descoberta de conhecimento do negócio é o processo para alcançar vantagem competitiva sobre os seus concorrentes, e esta é apoiada por Sistemas de Suporte á decisão ou por Business Intelligence termo atualmente em voga.A primeira coisa a fazer em qualquer tipo de sistema de apoio à decisão é criar o repositório de dados de informação onde o sistema vai recolher e mostrar todas as informações solicitadas. Este repositório é a fonte de todo o conhecimento do negócio, e a sua construção exige as ferramentas de software corretas para o desenvolvimento do data warehouse. Deve-se por isso ao escolher a ferramenta de software pensar nos requisitos para a seleção do software do mercado, a escolha do software envolve o tamanho do projecto, orçamento, ou riscos a tomar em mente. Globalmente, a escolha certa depende das necessidades de organização e suas ambições.
O trabalho essencial a ser feito aqui é demonstrar que o software open source pode ser uma ferramenta fiavél e eficaz para implementar projetos de data warehouse. As duas soluções ETL utilizadas foram:

- Pentaho Data Integration Chaleira Editions Comunidade (Open Source Software)
- SQL Server 2005 Integration Services (SSIS) Enterprise Edition (Software Proprietário)

O software proprietario, comercial em questão (assim como outros) é amplamente utilizado. No entanto, uma solução de open source tem características fundamentais que são reconhecidas por organizações em todo o mundo e este trabalho irá mostrar as diferentes funcionalidades e benefícios desta abordagem de software open source.

Palavras-Chave
Business Intelligence, Data Warehouse, Open Source Software, ETL, Kettle

ACKNOWLEDGEMENTS

Starting any academic journey is a challenge and achieving what we set out for is highly satisfying. For me, being able to accomplish this goal is a milestone achieved; it was sometimes difficult to balance both work and academic study but it was worth all the effort and time spent.

First of all I want to thank my family for being there when I needed them, and for the support and confidence invested in me.

I also wish to especially thank Professor Carlos J. Costa who was always available to offer his guidance and support for this dissertation.

Finally I would like to offer thanks to João Alves, IT Director, for contributing his known-how of Business Intelligence.

List of Abbreviations and Acronyms

DW - Data Warehouse

BI - Business Intelligence

IT – Information Technology

CTC - Accouting

DEC - Fleet Management

OSS - Open Source Software

ETL - Extracts, Transforms and Load

SA - Staging Area

OLAP- On-line Analytical Processing

DSA – Data Staging Area

INDEX

1. Introduction

This dissertation takes place in a multinational company which specializes in full service lease and fleet management contracts for automobile fleets, as well as other services that involve outsourcing solutions, sales and leaseback.

The company has reached a mature level in the last years, and growth and success in this core business are now more than ever necessary to achieve the objectives proposed for the future.

In order to understand and better evaluate the business process involved in this area, the decision makers need to have a reliable and faithful repository of information regarding their core business. To make the right decisions at the right time can result in the improvement of the overall performance of the organization.

The key to acquiring this knowledge is the concept of Business Intelligence (B.I.). Generally, organizations use commercial solutions that are extremely expensive both in the short and long term.

Information is most relevant when it means something to the decision makers, and is therefore a critical element in the future success of any organization.

Taking advantage and control of all the information available is the crucial step for changing the organization's future and having a competitive advantage over their market competitors.

This is a challenge as, most commonly, all an organization's data is dispersed and in different formats. Therefore, the primary objective is to concentrate the raw data into an information repository to be analysed and furthermore used by its end users and its analysers/developers.

The purpose of this dissertation is to show:

- The correct practises of using this approach;
- That the implementation of one of the BI tools in open source allows for a clear reduction of costs and development effort, as well as efficient time usage and task effectiveness when compared to a commercial solution.

I have also dedicated some chapters to discuss the implementation and methodology to be used to allow successful use of the Open Source[1] tool in BI solution.

[1] Open Source software or OSS in generic terms is software that has the source code of the application open and available to developers, allowing customization of the source code for specific needs. This is the opposite of proprietary software, which has closed source code, not allowing access to the application source code.

1.1. Scope and Motivation

When implementing any proprietary IT solution, support and in-house experience is fundamental to overcoming problems and issues, but it cannot offer the flexibility and extensibility that an Open Source view and its solution can bring.

In general, problems emerge when proprietary software becomes outdated by its vendor and no-one can ask for support or request new functions.

In this situation, an organization has no other solution but to buy a new version of the software at any cost, or search for a better solution that can provide necessary future flexibility.

Open Source then becomes an alternative to the proprietary software mostly used. We must choose the right OSS to use, by selecting open source software that fulfils the organization's specific needs (and not the desires).

From this perspective, my motivation is in discovering which OSS is capable of exceeding the expectations in open source tools for ETL database integration.

Among all open source tools available on the market, is there a useful tool that can be suitable for the job of implementing a Data Warehouse?

In a time of a worldwide economic crisis, the need for cost reduction is clearer than ever. Choosing the right approach makes the difference, bearing in mind that when developing a Data Warehouse project, choosing the right tools can achieve the purpose by:

- Reducing the team effort;
- Reducing the time-consuming Task;
- Reducing hardware and software resources;
- Reducing Outsourcing services.

Ultimately all these points reduce costs if the organization believes that all software approaches are acceptable and are available for open source software choices.

This dissertation reflects on, and shows the use of, an open source approach by comparing open source software to a proprietary software application. It will demonstrate that when we choose open source software it can do exactly the same thing with less effort and in less time.

1.2. Research questions and objectives

The fundamental problem faced in this dissertation is to understand, by comparison, the use of two different software approaches when implementing a data warehouse. By using open source software tools and proprietary, commercial ones, we can determine that open source software, when properly applied, becomes a reliable tool for the organization.

Most organizations commonly use proprietary software to implement a data warehouse solution. This is because the support involved and years of matured proprietary software development are a key feature in a decision to choose which software tool to use[2].

Data warehouse operational processes are the focus of this work and the tools used in these routine procedures is specialized software, generally called Extraction-Transforms-Load (ETL) tools.

The essential work to be done here is to demonstrate that open source software can be an accurate and reliable tool to implement data warehouse projects. The two ETL solutions used were:

- Pentaho Kettle Data Integration Community Editions (Open Source Software)
- SQL Server 2005 Integrations Services (SSIS) Enterprise Edition (Proprietary Software)

The proprietary, commercial software in question (as well as others) is known to be superior over the open source software. However, an open source solution has key features recognized by organizations worldwide and this work will show the different functionalities and benefits of this open source approach.

This dissertation also hopes to be a tool which can help and guide other companies in the implementation of a data warehouse using open source software, by making suggestions and documenting issues that can emerge.

When working with a great volume of database data, one of the most critical issues is the efficiency of the transaction in database transformations. We do not want the process of any transaction to take a long time; what we do want is an efficient process that does the requested data transformations in the minimal time.

[2] Narayan (2009) shows the diverse aspects to bear in mind regarding BI software solutions, analysing both proprietary and open source software.

1.3. Methodology Approach

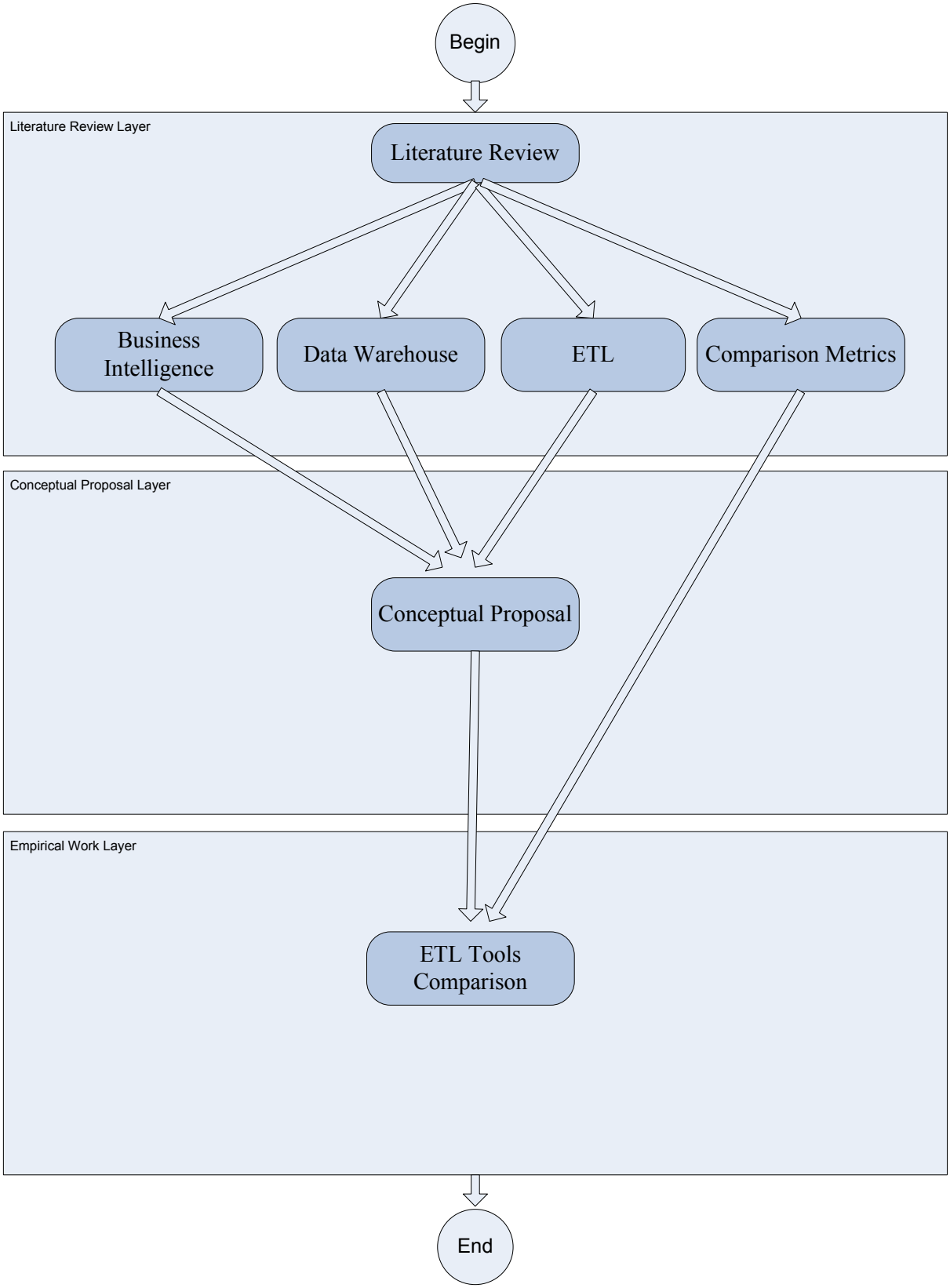

Figure 1 - Methodology approach framework.

The methodology approach represented in fig.1 shows the different steps of conducting this dissertation; first of all we must set the boundaries of our investigations and define the dissertation problem.

Under business intelligence systems' implementation, the data warehouse (DW) is the source of the decision-making knowledge, in other words the repository of information where all the critical organization data is kept.

Operational processes and workflows within the data warehouse are the focus of this dissertation. Managing them in an efficient manner, whilst reducing the effort and minimizing the time consumed in the task, is the core problem.

The first layer of this methodology approach is the literature review, where we search and investigate related works, retrieving clear ideas of the concepts at hand:

- Business intelligence;
- Data Warehouse;
- ETL;
- Comparison Metrics.

In the second layer, Business Intelligence, data warehouse and ETL are the key concepts for supporting our conceptual proposal, helping it by defining the proposal and tracing logical paths to implement the empirical work.

For the last methodology layer, the empirical work here was done by comparing two different ETL software tools in terms of analysis approach, one tool being Open Source software and the other being proprietary software. I would use the literature review as the basis for the empirical work, comparison metrics for support, and the concept proposal guidelines to perform and conclude the work.

Related works of the same subjects help us to understand which aspects we must be concerned with and emphasize the relevant points to undergo a software comparison.

2. Literature Review

2.1. Concept of BI

BI Systems support and assist decision-making in enterprises as part of the organization's strategic plan to achieve management effectiveness. It is a combination of both management and gathering of data (managerial work) from inside the enterprise ("getting the data in") with technological tools (Maira, P. & Marlei, P., 2003). It transforms, extracts, analyses and shows the information data in a practical and clear format view ("getting the data out") to the strategic planners.

(Solomon, 2004) describes BI concept in this way: "BI systems combine data gathering, data storage, and knowledge management with analytical tools to present complex internal and competitive information to planners and decision makers."

In this perspective, BI solutions generate relevant information for the enterprise executives and decision makers. The quality of the information is critical, however, so the objective of this system is to improve and manage the accurate information for the end user.

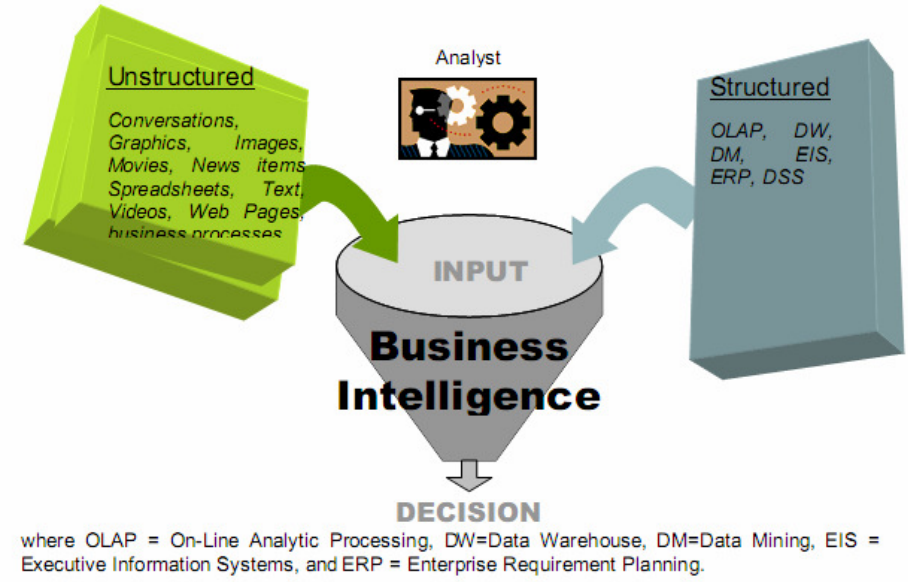

Figure 2 - Shows the variety of information needed to achieve decision making with intelligence (Solomon, 2004)

Over the years, the term Decision Support Systems (DSS) as been replaced by the new and recent term Business intelligence (BI). DSS was used, and first appeared 40 years ago (Power, 2004).

2.2. BI Process

BI systems combine gathering, acquiring and storing of data from different sources with analytic tools, presenting them in an ordered and logical form to the decision-makers throughout analytical reports, schemas or dashboards for a quick glance at the business state in the present, future or past.

From the user's point of view, Business Intelligence uses several methodologies and technologies to present the data that can be categorized (Jayanthi, 2005):

- OLAP (On-line analytical processing);
- Advanced Analytics;
- Corporate Performance Management (Portals, Scorecards, Dashboards);
- Real time BI;
- Data Warehouse and data marts;
- Data Sources.

2.3. Data Warehouse

A data warehouse is an integrated and time-varying repository of data information, used primarily in decision-making systems and analysed by online analytical processes (OLAP).

It is a database that gathers multiple information from different and heterogeneous data sources from distributed systems. It integrates and aggregates historical information, which are non-volatile data and business subject oriented, being a data information system that intends to deliver a unique view of the business reality to the end users.

Furthermore, William H. Inmon, considered to be the father of data warehouse by many authors (Watson, 2001), contributed with the notion that the differences between operational databases and data warehouse databases are the basis for implementing a correct warehouse (Inmon, 2000).

Table 1- Operational Database versus Data Warehouse

Operational Database	**Data Warehouse**
Time Critical Reading/Writing Accesses Accesses few records at a time Data update in real-time Structured for OLTP (Online Transaction Processing)	Historical data Reading access Accesses many records at time Data Updating periodical Structured for OLAP (On-Line Analytical Processing)

Source: (Maribel & Ramos 2006)

2.3.1. Implementing the Data Warehouse

When an organization decides to implement a data warehouse, there are some questions involving this implementation that are defined by (Gardner, 1998) and related to:

- **Costs:** how much does the organization have to spend on hardware, software, tools and services from the vendors?
- **Time:** How long will the project take? How much time is there?
- **Users:** What do the end users and IT users need from the Data Warehouse?
- **People:** Who will maintain the data warehouse? Who will build it?
- **Hardware:** Software and Tools: Which will we use to build the warehouse?
- **Services:** Can we do it ourselves or do we need extra help?

These questions must be answered by regarding the project involved and its needs; if the project is ambitious and the risks are high, then the quest for a better solution in terms of costs must be suited to the best appropriate option in the market.

In a proprietary scenario, commercial software suites in BI for implementing the data warehouse may be a solid solution based on the years of matured development in this area of business intelligence.

However, these solutions have high costs in maintenance and licensing products. Depending on the project, there are better solutions when choosing the software tool, combining the search for cost reduction and reliability.

If an organization doesn't have the need for a BI software suite, then it's not necessary to purchase extra functions for a product when they will not be used.

Therefore, the project managers must choose the right tool suitable for the project, and in this case implementing the data warehouse by choosing the right approach.

Like open source, approaches using software tools free of licence fees and involving reduced costs can be the right solution when the project involves the implementation of the Data Warehouse which only needs a proper ETL (Extract Transform and Load) tool.

2.3.2. Data Warehouse Design

When implementing a data warehouse, the design and architecture type is fundamental to the success of the project. That said, the organization must decide which architecture is more appropriate for its project specifications, a decision which depends on the issues and subjects related to the organization and the scope of the decision-making.

There are three types of data warehouse architecture (Gardner, 1998): Enterprise Data Warehouse, Dependent Data Marts and Independent Data Marts:

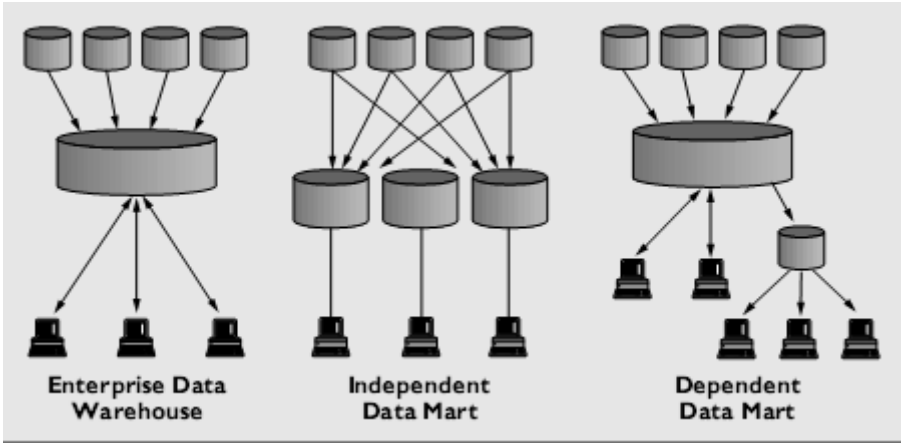

Figure 3 - Data Warehouse architecture (source: Gardner, 1998)

To understand these types of design, first we must define the concept of data mart.
A data mart is a simple form of a data warehouse that is focused on a single subject (or functional area), such as Sales, Accounting or Marketing (Cederberg, 2010).

Data marts are often built and controlled by a single department within an organization. Given their single-subject focus (Sharma & Gosain, 2010), data marts usually draw data from only a few sources. Those sources could be internal operational systems, central data warehouse, or external data.

An enterprise Data Warehouse is a centralized repository of information and a set of data sources that combine a variety of subject-oriented issues, all of which are accessible to all users.

In an Independent Data Mart design, all the external and operational data sources are transformed and loaded directly onto the data marts. This is useful for smaller organization units, which already have a well-defined subject oriented data mart.

For a dependent Data Mart, the main difference here is how we populate the data mart. In this case, it uses the central data warehouse for ETL procedures to load, filter and clean information into the various data marts available.

In this design, ETL processes are simpler simply because the information present in the data warehouse is summarized, organized and more detailed than in the operational sources. Its processes, therefore, simply involve selecting the subset of data we need for the data mart.

2.3.3. Constants and Influence Affecting Database Architecture

In the data warehouse architecture, there are influencing forces involving data, technology, and organizational requirements Lewis, G et al. (2001):

- **Data requirements** - Related to the operational and functional database needs for transactional applications, such as reports, queries, and ad hoc queries (created in a more flexible way using SQL management tools for mores structured queries). Persistent summaries and roll ups for consolidated information (these are reports designed to obtain a more organized view of the organization's complex transactions (an organization must have infrastructure support for a high volume of transactions), and data warehousing capabilities for decision-making systems.

- **Organizational requirements -** In today's organization, the branches of business are spread across geographical locations, and the headquarters and offices associated need to have full access to the integrated system, allowing them to access and include information distributed over the organization's structure, depending on the user credentials. The maintenance of the distributed data is therefore a challenge, in order to improve the performance of the operating database.

- **Technology requirements -** These requirements specify the need to acquire the appropriate hardware or software products to implement the ideal architecture for the organization, by identifying the right choices between the multiple products available in the market and regarding the price, quality and objective of the project developed.

These requirements are pertinent to identifying situations that we need to take into consideration. When we want to implement the data warehouse system, answering these issues is the best way to successfully conclude the project.

2.3.4. Multi-dimensional Models

The purpose of Multi-dimensional models is to produce a business schema to better understand the business itself. It is the database model structure of a data warehouse system.

These models are based in three concepts (Sharma & Gosain, 2010):

- Modelling Rules;
- Cube and measures;
- Dimensions.

The multi-dimensional database model in a DW must be easy to use and comprehend, and must be optimized for decision-making systems for a better performance in decision-making questions (Sharma & Gosain, 2010).

A relational database schema makes the representations of these data base models. There are three database implementation types: star schema, snow flake schema and constellation schema.

2.3.5. Multi-dimensional Design Decisions

Before starting to model our database schemas we must first decide on the schema design, which is essential for the project's sustainability.

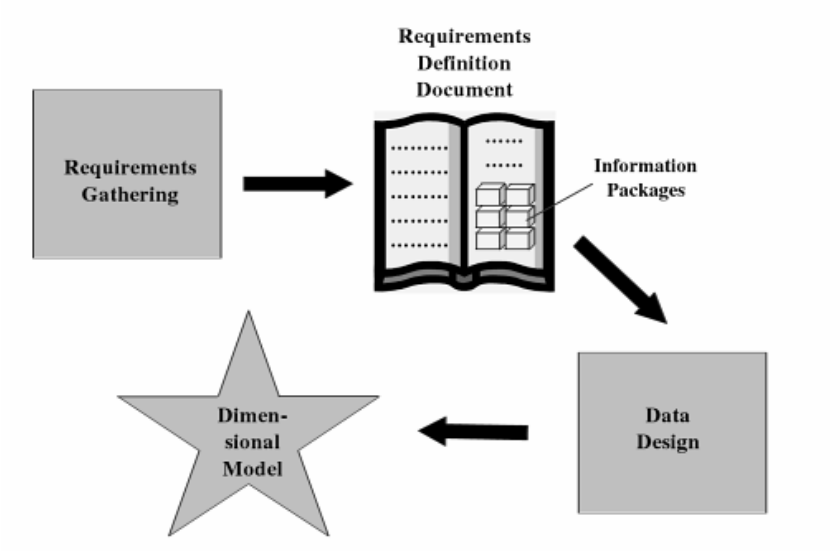

Figure 4 – Requirements for multi-dimensional modelling (Ponniah, 2001)

Aspects to bear in mind, before implementing any data warehouse (Ponniah, 2001):

- Selecting the Process - We must decide which specifications are needed for the first set of data structure, to define the business process for analysis.
- Choosing the granularity – Defining the level of detail present in the data.
- Identifying the dimensions – Choose which dimension will be used (Time, Client, Product, etc.) on the data structure.
- Identifying the Facts – Choosing the measurements or business facts (Total Sales, Total Quantities, etc.)
- Choosing the Historical time for Database – Decide how far back the data will store information (Years / Months).

2.3.6. Dimensional Model Schemas

2.3.6.1. Star Schema

This is the most common schema used in multi-dimensional model implementations because of its unique data relation schema. It is designed to be efficient and optimized for database query operations (Sharma & Gosain, 2010).

It contains one fact table and one or more dimension tables. The design resembles a star, where the facts table represents the centre and the multiple dimension tables are connected around it.

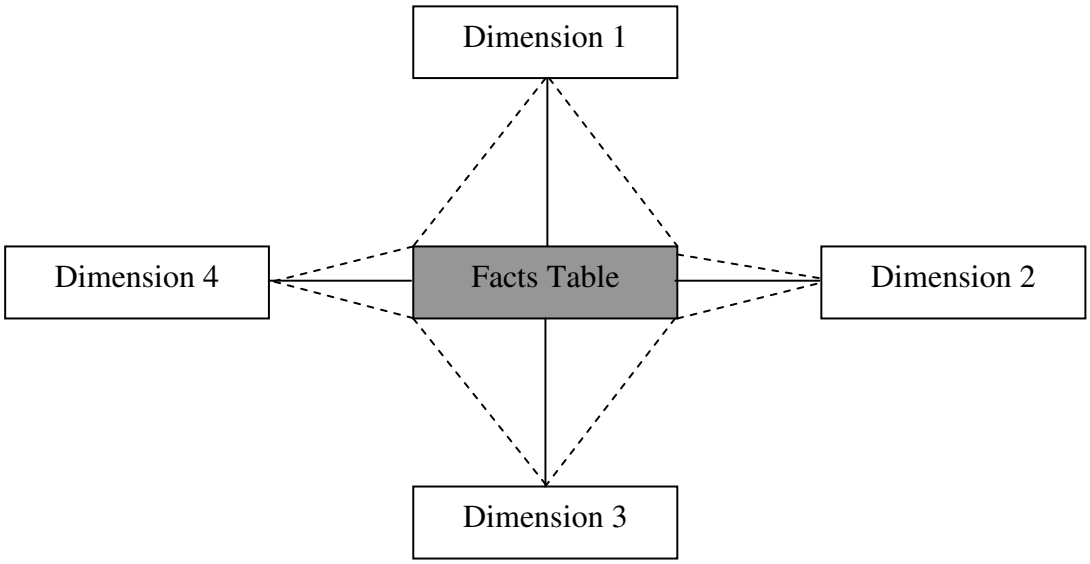

Figure 5 - Star Schema, source: (Maribel & Ramos 2006)

The facts table corresponds with the subject we are trying to analyse, such as sales or deliveries. This table includes attributes/measures for each fact (e.g. total value of sales, quantities, etc) and relation keys for each dimension table on the star schema.

In a dimension table, the relation with the fact table is made by the analysis over the facts through different perspectives. Each dimension offers a different view of the subject analysed.

Dimension tables help to answer questions present in the fact table (e.g. Who? When? Where? Why? How?) .

In a multi-dimensional model, these dimensions can be analyzed by visualizing them as cubes in which each face of the cube represents a dimension of the star schema.

Star Schema characteristics (Maribel & Ramos, 2006):

- Facts Tables present several attributes and one set of foreign keys belonging to each dimension associate;
- Facts tables are normalized;
- Fact tables contain a substantial amount of records and represent almost the whole size of the DW;
- Dimension tables can exist for as many perspectives as we wish to create;
- Dimension tables are not normalized;
- Dimension tables contain fewer records than the fact table.

2.3.6.2. Snowflake Schema

Snowflake Schema is similar to a star schema, but all the dimensions are in a hierarchy and normalized; one dimension can present one or more sub dimensions, representing another data structure (figure 6).

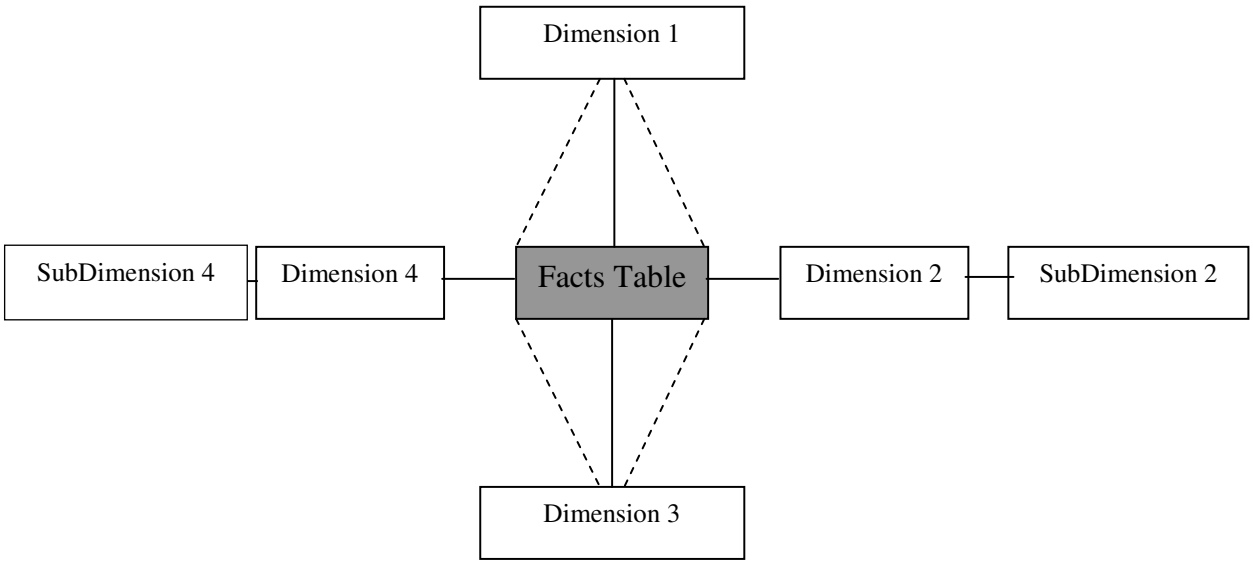

Figure 6 - Snowflake Schema, source: (Maribel & Ramos 2006)

This schema is more complex than the star schema, and it can be difficult for users to understand and maintain its structure. In its normalized dimension structure, the processing of decision questions has lower performance because when a table is normalized it cannot save redundant information.

2.3.6.3. Constellation Schema

Constellation schema is composed of one or more fact tables linked by one or more dimension table; essentially it is multiple star schema joined together (Figure 7).

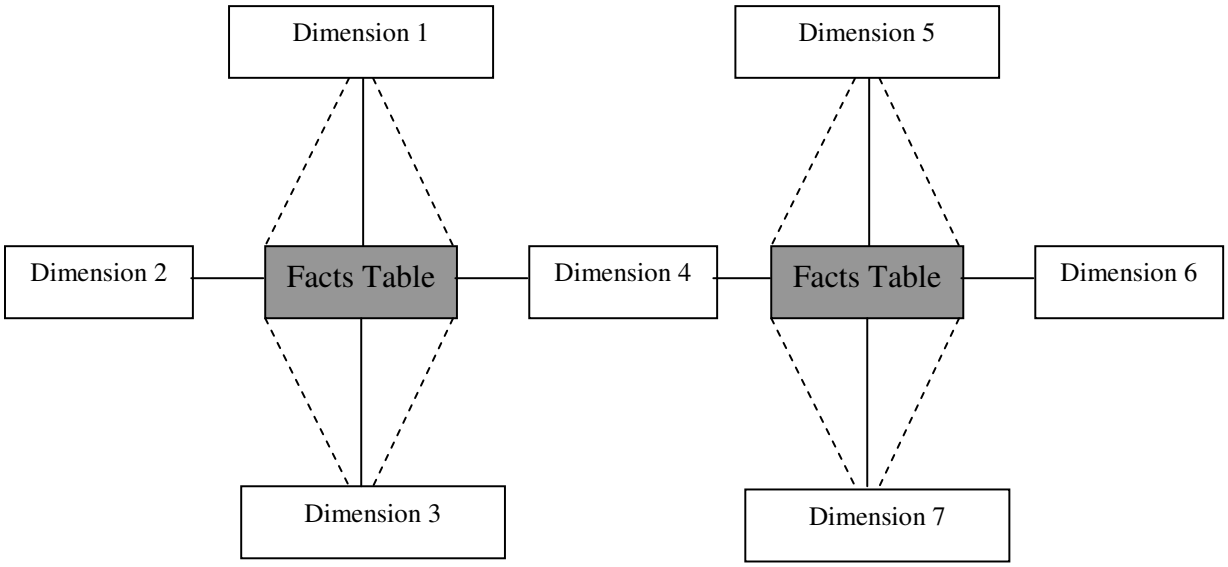

Figure 7 - Constellation Schema, source: (Maribel & Ramos 2006)

This structure is more complex and requires more effort with regard to maintenance and support. It is, however, more flexible than other schemas by the fact that a dimension can be shared among several facts tables.

It is useful when we need to drill down any information type, and because of its ramifications we can get more detailed information.

2.3.7. Facts Table

One of the first decisions when building database model schemas is choosing the business facts in order to reflect the analysis to be carried out.

The fact tables incorporate the business unit measurements that are the subjects associated to business processes; these measurements are quantitative or factual data associated to the subject (Maribel & Ramos, 2006).

The fact tables include keys and attributes related to the dimension tables. The records must reflect the subject in analysis depending on the data granularity defined; in table 2, we see that the granularity is defined by product by account by year.

Table 2. Facts Sales Example

Product	Account Number	Years	Client	Total Profit
1	000001	3	10001	50000
3	000003	1	20001	10000
6	000005	5	30001	40000
10	000012	4	40001	1000

Depending on the fact table's characteristics, we can analyse the facts in different ways, as usually a fact table's measurements are aggregated. However, we can drill down and deepen the search in each of the "cube's sides".

In order to analyze these facts, we must determine their different types (Maribel & Ramos, 2006):

- Additive Facts
- Non-Additive Facts
- Semi Additive Facts

Additive Facts – Additive Facts are those that can be aggregated through all dimensions.

Non-Additive Facts – Non-Additive Facts are those that cannot be aggregated through any dimension.

Semi Additive Facts – Semi-Additive Facts are those that can be aggregated only through specific dimension types.

2.3.8. Dimension Tables

Dimension tables contain all the attributes associated to the fact tables. These attributes can be descriptions or measurement units, used to summarize/aggregate useful information within the analysis at hand.

Table 3. Clients Dimension Table

Column	Description
Client ID	Specifies the Unique Client Key
Name	Specifies the Client Name
Company ID	Specifies the Client Company ID Group
Client Activity	Specifies the Client Business Activity

Each attribute corresponds only to one dimension table, and cannot be present in other tables; they are independent and therefore provide data consistency throughout the dimension table and in the modelling schema.

Because a data warehouse is a repository of information, the database operations are primarily restricted to reading operations. However, the information updates can be done on a periodical basis defined by the project manager and according to the subjects in analysis.

The dimension tables only keep the records' attributes until the next data warehouse update; this is not a common process because the historical data needs to be preserved over time.

There are three ways to proceed with the dimensions' updates (Maribel & Ramos, 2006):

- Overwrite Data
- New record Data
- Predict additional new attributes

Overwrite data is when we don't need to keep historical data from the dimension table records, and we just want to copy over with the updated record. This process is quicker than the others because it doesn't save data from the prior value now being updated.

Adding a new record to the dimension table is more complex than the previously updated type. When a new record is added, all the changes are kept in a chronological way, which allows us to see the traceability of the record by looking into the information history of the attribute.

Any attribute can have existing columns to preserve both current and past information, as well as being able to predict new columns for any future data information changes.

For example, in a specific vehicle contract we can predict, according to its maintenance history, a new column with its next due intervention (each column will show one intervention).

In this way we can create several columns for the vehicle's maintenance history, and another for the next intervention. We can therefore save historical information and changes made over time.

2.3.9. ETL Concept

Maintaining and managing the data warehouse operational processes is done by the Extraction-Transformation-Loading (ETL) concept which is a specialized tool for this purpose, used commonly by IT developers and data analysts.

An ETL tool gives a full set of data integration functionalities for developing data integration work flow tasks and routines, that are crucial ETL tasks used before the data loading phase into the data warehouse. Authors (Simitsis & Vassiliadis, 2002) summarize this task as:

- Identification of relevant information on the data sources
- Extraction of this information
- Consolidation and integration of information from heterogeneous data sources
- Data cleansing of the resulting data set
- Propagation of the data to the data warehouse and data marts

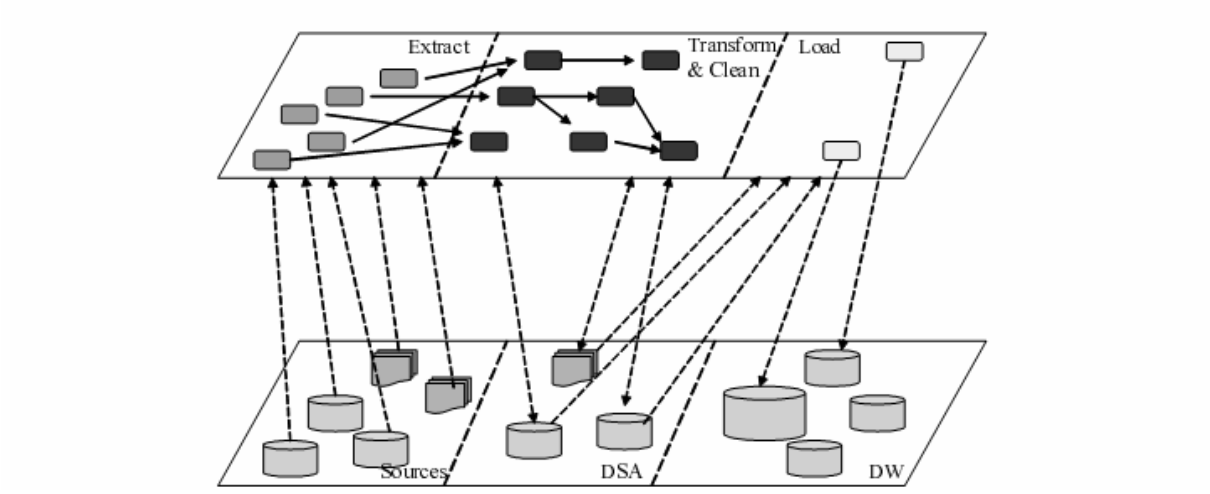

Figure 8 Data Warehouse Environment

In figure 8, we can see the ETL process' logical workflow. In the bottom layer we see all the data sources involved in the process, starting with the extraction of data sources, typical operational databases, and other external data sources of heterogeneous systems (Flat files, Excel files, Access Files, CSV files).

This extraction processes is performed taking in consideration the pre-requisites from the project, i.e., the extractions gather a complete a set of data from one specific source or from several sources.

After the extraction routines are complete, the data information is stored temporarily in a Data Staging Area (DSA). In this phase, the data information is transformed, cleaned, and treated to assure data quality [3] before loading into the data warehouse.

Through personal experience and in related works Pires et al. (2009) they suggest that the implementation of a data warehouse throughout ETL processes is a difficult and time-consuming task, consuming costs of both human and financial resources.

Depending on the data model complexity of the data warehouse, as well as the volume of data it generates, greater will be the effort for managing and understanding the data processes involved.

Therefore, ETL tools must have a functional way of providing the ETL developer with a simple and easy to use development environment, guaranteeing data performance process, scalability and extensibility for future data warehouse projects or for future project growths.

ETL management is one of the keys in integrating data into the data warehouse. The authors (Albrecht & Naumann, 2008) present a framework for efficiently managing the ETL procedures, focusing on the main idea "to reduce the amount of programming needed for developing or maintaining the ETL processes".

The authors suggest that for a more efficient use of the ETL processes within the organization, they must be kept in a central and common repository in order to allow for its use and benefit from prior processes which can be reused, saving time and team effort in new ETL process routines.

When choosing any ETL tool from the market, therefore, we must consider all functional aspects for better and efficient management of ETL processes.

A related work (by Golfarelli, 2009 and Levin, 2008) shows some comparison metrics for business intelligence platforms. The author (Golfarelli, 2009) compares three different Open Source BI platforms (Jaspersoft, Pentaho and Spago BI) evaluating comparatively their architecture, focusing on functionalities and usability.

(Golfarelli, 2009) demonstrated the various metrics used as evaluation criteria in the different types of open source business intelligence platforms, arguing and concluding that open source approaches can be considered a reliable alternative to commercial software business intelligence suites. As the author states in page 10: "Although OS BI platforms are still not as sophisticated as commercial ones, we can state that they have a sufficient level of reliability and must be considered a valid alternative to commercial suites."

[3]Data quality involves process of data cleaning, data validation, data testing, data filtering to assure data information consistency on the front-end of the data warehouse system, assuring always accurate and trustworthy information for the end users.

In the author's work, the comparison metrics used were the following (Golfarelli, 2009):

- Non-technical: platform philosophy, licensing type and availability of enterprise editions.
- Architectural: in terms of the global framework, modules and their relationships, programming languages and supported operational systems.
- Functional: in terms of functionalities provided natively by the platforms or made available to the users, using the integrated BI tools.
- Meta-data: in terms of expressiveness, completeness, standardization and level of re-usability.
- Security: in terms of functionalities provided for authentication and users' profiling, interfaces to external authentication systems, and secure data transmission.
- Usability: both from the user viewpoint in terms of level of transparency in using the different tools, and from the developers' and system administrators' viewpoint in terms of complexity of installation and administration, as well as development of applications, and quality of manuals and forums.

Where (Levin, 2008) details his work by comparing ETL software tools from the user's point of view, using both open source ETL software tools and proprietary software tools, the author shows several issues related to the use of this tools, comparing them by the use of following metrics :

- Total Cost of Ownership
- Risk
- Ease of Use
- Support
- Deployment
- Speed
- Data Quality
- Monitoring
- Connectivity

All the metrics from these related works can be extrapolated for a comparison between Open Source ETL software tools, and proprietary commercial software tools, supporting an empirical work for evaluating two different work approaches.

Applying some of the metrics used in this author's works, we can show by evaluation and reporting the different key aspects within these two ETL approaches, concluding which ETL tool better suits our project's needs.

2.3.10. Data Warehouse Multidimensional Models Exploration

The exploration of the data warehouse is commonly made by the use of OLAP (Online Analytical Processing) technology that does information processing rather than operational.

Using model views in the form of cubes allows basic OLAP operations to perform interactive data mining over the facts table with different levels of detail.

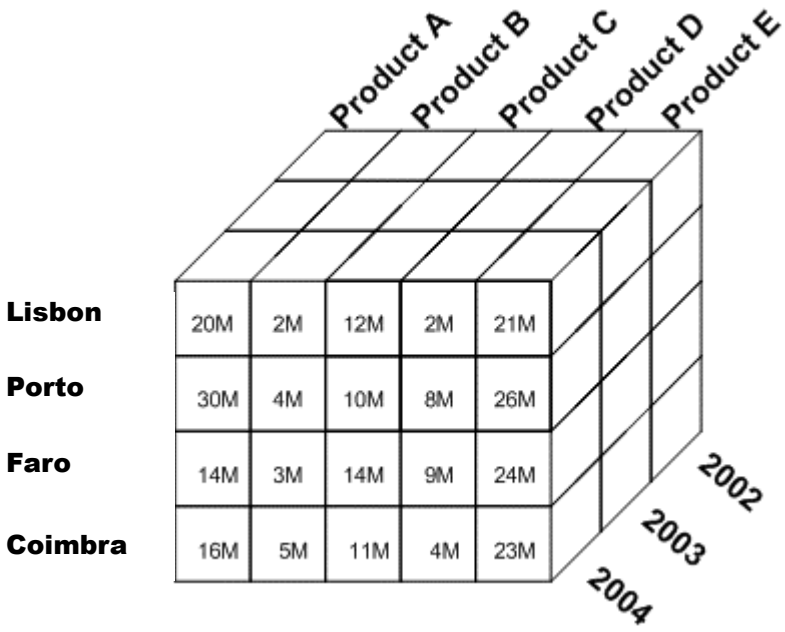

Figure 9 - Data Cube Example.

To better understand the exploration and the processing analysis of the data warehouse, first the concept of a data cube must be presented.

A data cube is a model view of multiple dimensions that can be viewed in different perspectives according to the implemented model‘s schema.

“A data cube allows data to be modelled and viewed in multiple dimensions. It is defined by dimensions and facts.” (J.Han and M. Kamber, 2000).

For the analysis of these cubes, OLAP has several types of servers depending on the project manager's needs and specifications (J. Han and M. Kamber, 2000)(Maribel & Ramos, 2006):

- ROLAP (Relational OLAP) - These are servers that use relational database systems to store and manage the data for analyses; it uses intermediate servers to bridge between front-end tools and relational databases.
- MOLAP (Multi-dimensional OLAP) - This server type uses multi-dimensional databases to store and manage the data and is much faster in processing database consults, benefiting from the fast index on summarized data because of the use of data cubes.
- HOLAP (Hybrid OLAP) - This type is a mix of the two above, combining ROLAP and MOLAP, profiting from the scalability of the ROLAP server and the better processing power of the MOLAP server.

OLAP servers allow the execution of different operations over data dimensional cubes, in an interactive and visual form; we can operate the data cube for analysis and query the different perspectives when examining any dimension.

Available OLAP operations (J.Han and M. Kamber, 2000)(Maribel & Ramos, 2006):

- **Roll-up** – Roll up Operation is the opposite of Drill-Down. It is also called Drill-Up. This operation aggregates data on a data cube by applying a hierarchical structure over the dimension, either climbing up or down one dimension.
  Figure 10 shows a roll-up operation by aggregating the location data, climbing up one dimension from cities to countries.

- **Drill-Down** – Drilling down in a data cube allows navigation from less detailed data to more detailed data, representing a view from a more detailed perspective for the data in analysis. This operation can be done by descending one dimension or by adding a new dimension for analysis. Looking into figure 10, the dimension time drills down from quarters to months, getting more detailed information in the cube view.

- **Slice and Dice** - Slice and dice operations allow reduction of the current cube view into a small portion of data or new sub-cube, by first making a dimension selection (Slice) and selecting a set of data filtered by a predetermined condition. The reduction (dice) defines a new cube and sub-cube from the selection of one or more dimensions, depending on the criteria defined.
  In figure 10, the slice operation is made by selecting the dimension time and applying the criteria for "time = Q2" and a new subset of data is shown for analysis. For the Dice operation in the figure, a selection is performed in three dimensions: Location, Time and Item, applying the criteria for each dimension and creating a new sub-cube with this information.

- **Pivot (Rotate)** – Pivot operations rotate the data axis of the cube in order to provide a different view of the data. In figure 10, the dimension axis Item and Location are rotated, providing an alternative view of the data.

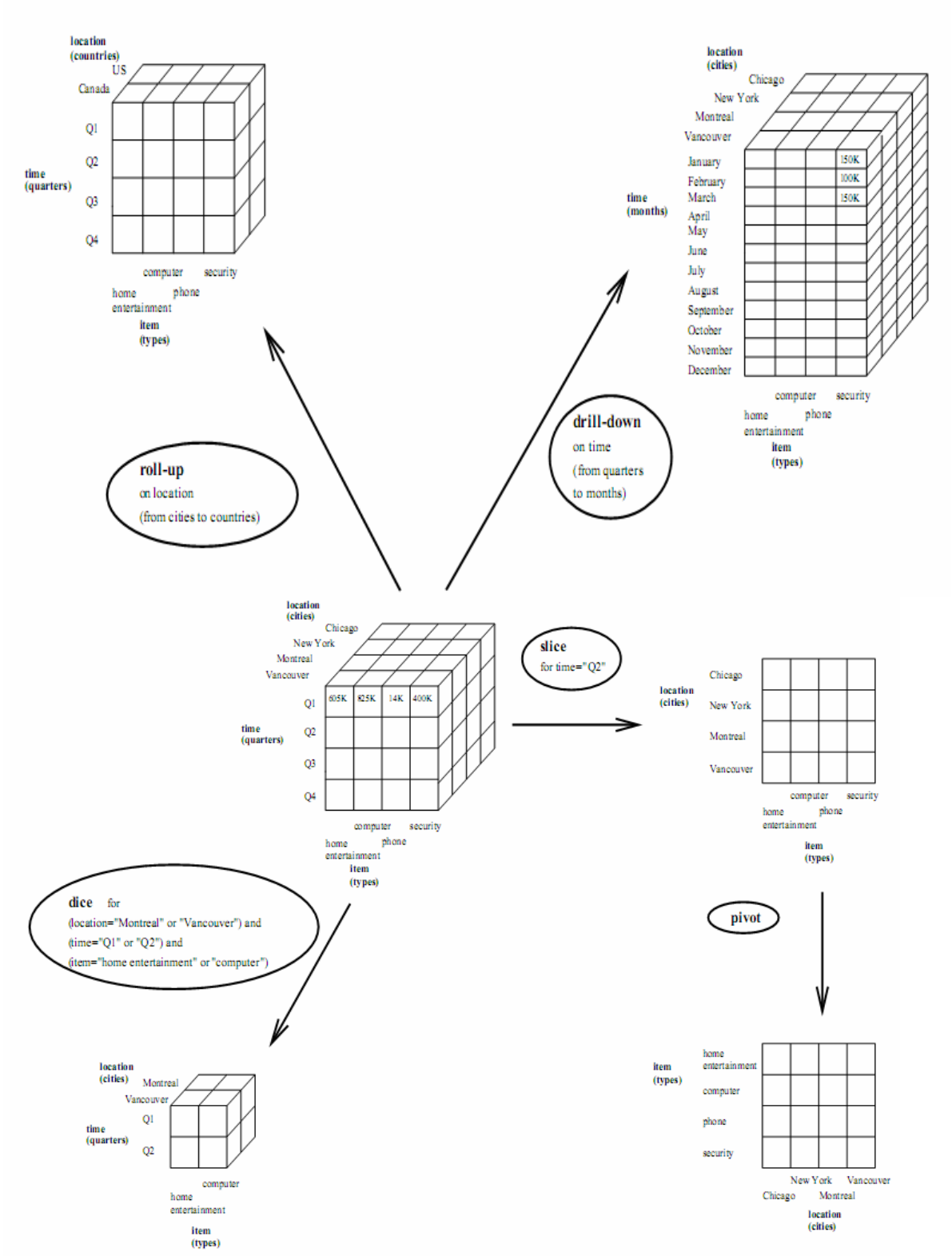

Figure 10 - OLAP data cube Operations example (Source: J.Han and M. Kamber, 2000)

2.4. Empirical Studies

Several empirical studies that used open source tools as an approach to business intelligence implementation are relatively successful.

Cramer (2006) made an analytical study of open source tools in OLAP environments, and tested and compared a number of open source tools.

After testing them and publishing the results, his work concluded that, for instance, a Pentaho BI tool for OLAP environments was not mature enough as a commercial solution because the tool doesn't support a drill-across OLAP cube functionality.

Another conclusion is that this open source tool doesn't offer the full solution to OLAP environments that their commercial counterparts do.

Open source tools nowadays are not perfect and still need a lot of work to be done. However, they are by far a better alternative to consider when the needs of a BI project match the open source tool capabilities.

Naimuzzaman (2009) concludes in is work:
"Eclipse BIRT fulfilled the customer's requirement and met the user satisfaction."

Imhoff (2008) shows the benefits of the Open Source model applied to Data Warehouse:

- Open Source Data Warehouses cost less up front, and in terms of maintenance and support.
- Open source DW employ skill sets that are widely available in the market.
- Open source DW promote greater standardization.
- Open source DW are far more flexible.
- Open source DW benefit from the community effect.
- Open source DW can be implemented in stages according to the project's needs.

Imhoff (2008) also recommend some practices when implementing an open source DW:

- Open source and proprietary need to co-exist.
- Look for large, active communities behind the product.
- The open source DW should be invisible to end-users.
- The open source DW should always support open standards.
- Look for rapid deployment ability and ease of use.
- Weigh the cost of transition.

3. Conceptual Proposal

3.1. Current Database Architecture

The current database architecture (figure 11) is based on an Alpha Server that contains all the raw data and the application DEC used for all the company.

On a daily basis, all the data present in the Alpha server is copied into the SQL SERVER database, making sure that on the next day all the information is present for daily use.

All the tables present in the current database use OLTP (Online Transaction Processing), and so this processing doesn't promote decision support systems[4].

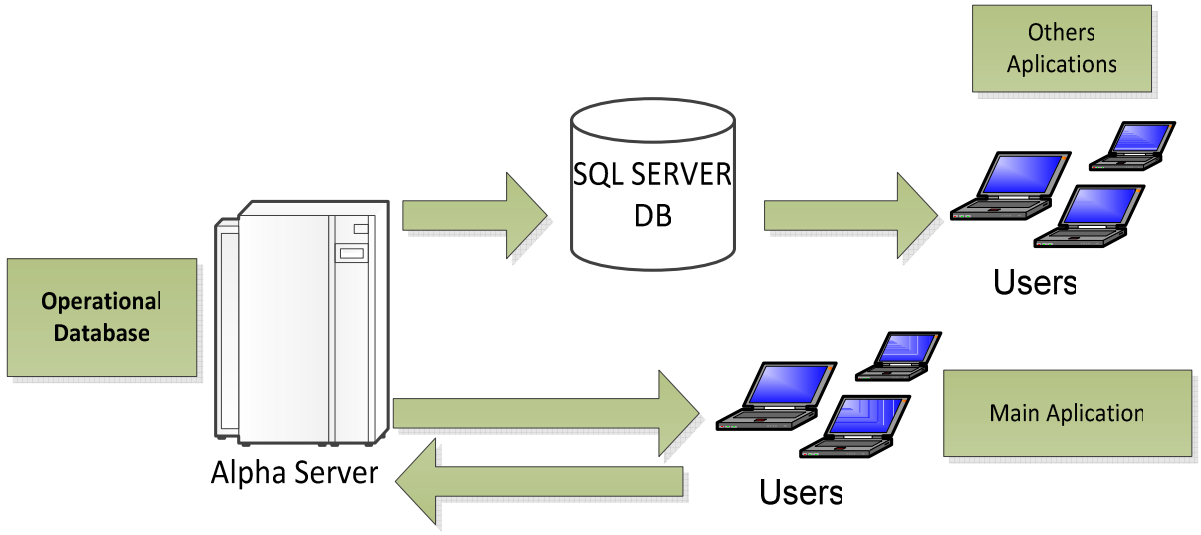

Figure 11 – Current Database Architecture

The present architecture is supported by proprietary systems and software, and is the main base used. All the internal and external information to the organizations reflect the use of this proprietary software (e.g. sending or receiving excel files for data information exchange between the organization and its partners).

The current database processes are linked directly to the organization's applications, but only for maintaining the flow of information available for the end-users, partners, customers, and suppliers. It is not intended to support decision-making processes, only operational ones.

[4] Decison support systems (DDS) are knowledge based systems that interact with the human judgement and decision making (Druzdzel & Flynn, 2010)

3.2. Organizations Needs

The current database architecture works well for the daily operations but doesn't give business feedback to the decisions makers, which is the type of information that the organization needs in order to achieve business management effectiveness.

The current organization's objective is to gain market margin over its competitors; therefore the necessity for more and better internal IT processes and systems is one of the milestones to accomplish.

In any organization, the business information is dispersed and difficult to find when needed. If any executive director wants to consult the total sales of the last 5 years for one determined product or group of customers, the information is not quickly accessible and requires time and effort from an IT team to gather and present it in a clear and simple way.

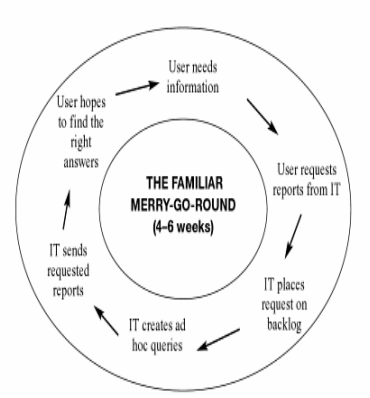

Fig.12 IT Attempts to provide strategic information (Source: Data Warehousing fundamentals: A comprehensive guide for IT Professionals, Ponniah 2001)

Bearing this in mind, the need for speeding up the internal processes is of high importance.
From the organization's perspective, therefore, one of the objectives is to have a data warehouse, a centralizing and integrating repository of data, which gathers information from different types of data sources and consults this information, considering one or more business subjects for analysis.

3.3. Identification and Preparation for the Data Warehouse

Before starting the implementation of the data warehouse project there is some planning to be done, mainly determining which requirements will be necessary for the initial preparation.

The data warehouse is based on the information gathered and extracted from the business operational sources and other heterogeneous sources. From the start, there must be a set of well-defined objectives for what the data warehouse will store and show to its users.

Data Warehouse Objectives/Requirements (Gameiro, 2011):

- Identify and characterize the business unit or departments (conducting a survey with the executive directors)
- What type of Operational Sources will be used, according to the departments identified
- Information type to be extracted (Database Criteria or Database filters)
- Which data marts will exist
- Defining the Staging Area
- Data warehouse Environment and Design
- Identification of tools and software

In the data sources from where the information will be extracted, the project manager must first define which are the sources to be extracted and loaded onto the data warehouse. Asking each business director to identify the business information they want to include in the data warehouse does this.

Having identified the various business units, we can then determine the most necessary and useful data marts, helping to define them.

The next step is to prepare our staging area. This is the area where all the raw data extracted will pass to be transformed/cleaned and loaded onto the data warehouse, or eventually onto the data marts.

Access is different depending on the information type. For example, if we are extracting data from an accounting database, the information present is related to business accounting subjects and the user's access permissions must be defined according to that department, ensuring the correct access to the right department.

After retrieving all the business units' information, we can focus on the data warehouse design, which is the crucial step when building and implementing this project. Depending on the project's dimension and the department's objectives, the data warehouse architecture must be chosen between organizational data warehouse, independent data warehouse and dependent data warehouse.

3.4. Selection of BI tools for ETL

On the software market there are various software ETL tools for selection, from proprietary software to open source ones. There are also many vendors, and it's up to us to choose the right tool for this type of project.

One of the organization's problems is that the costs are upfront and must be met prior to implementing the software. The benefits of the investment may appear late or never at all compared to the open source approach, which generates costs as it develops according to the project's needs. Keeping this in mind, the open source approach offers a better price/performance ratio than a proprietary choice. Figure 13 reflects this dynamic.

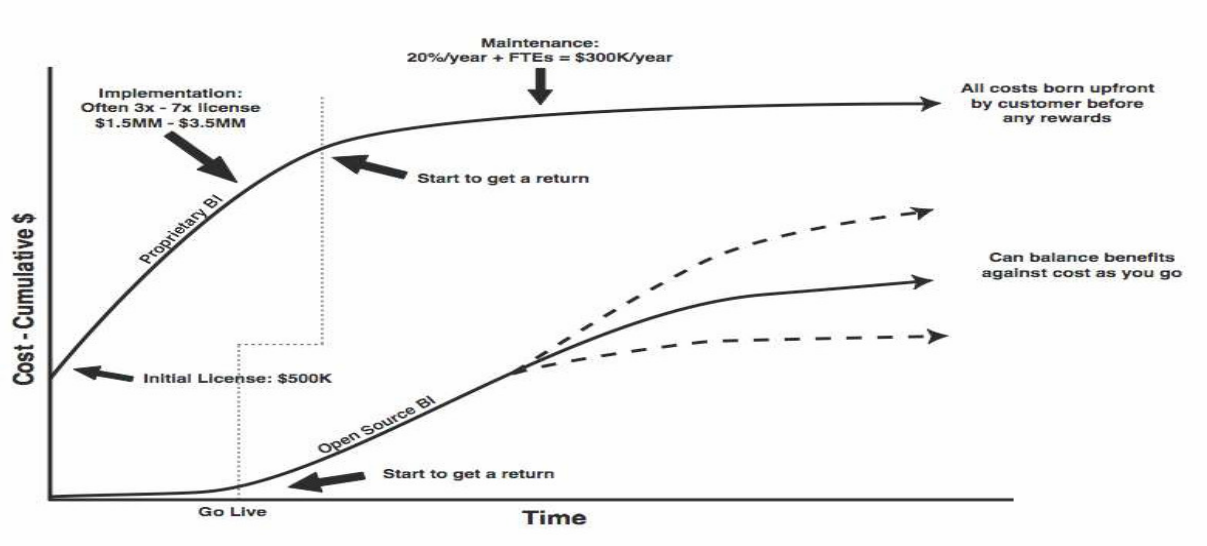

Fig.13. Software Cost over Time (Source: Business Intelligence Journal Saving Time and Money, Why Open-Source BI Makes Sense; Barry Klawans)

Depending on the project's dimension, a decision must be made between using either a proprietary or an open source approach. In my empirical work, I used the open source approach to reflect the benefits of open source tools and software. The emphasis is on the open source ETL software compared to the proprietary software, using for this purpose the Pentaho Data Integration (Open source) and SQL Server Integration Services (Proprietary).

3.5. Implementing the Data Warehouse

After planning and deciding the data warehouse's goals and objectives, we can pass onto the project's implementation. First we have to set up our development environment separate, if possible, to the production environment. We must also decide whether our data warehouse will be physically separate from the operational databases within the production and development environment, and put in a single machine to be used solely for data access.

In my empirical work, this approach was the chosen one because of the stress load that a data warehouse system can bring to the operational databases when combined in one server.

Thus, the query consultations made by the users will run much faster, as opposed to the case when the data warehouse was sharing the same hardware specifications with the operational database system.

More than half the work involved in the task of implementing the data warehouse is in the construction and configuration of the ETL process; this is the back-end system where we make the necessary data retrieval, transforming, cleaning and loading it onto the data warehouse system.

The ETL system and its tools, and the choice of this various tools depending on the business operational databases, are the focus of my dissertation.

In the specific case of my work, the selection of the ETL tools must support different databases types or sources, and if it is to be capable of integrating the information in the data warehouse system.

In order to fulfill all the heterogeneous operational database's requirements, the software tool must have the right capabilities and functionality to connect to the different types of data sources.

When implementing the data warehouse, it is possible for the IT team not to have direct access on the operational database. In this case, the task to be done is to make a copy of the tables and pass them onto a relational database system for easy access and maintenance.

This task can be done by creating an automated process which copies the raw data on a daily basis, making sure that the copy of the operational database is complete and equal to the real database, by using audit tables and logging processes.

The IT team must be able to create a reliable document describing the process of obtaining the copies from the operational database, and include also the dictionary tables to help in the implementation of ETL procedures.

After performing all the necessary database copies, we can start implementing our staging area (SA), the first step in the development environment; then creating the extraction, transformation and loading processes from the operational database. These ETL processes will be the focus of my empirical work, assuring data quality after performing the copy of the operational database.

In the staging area, we need to create mirror tables from the operational database to perform the necessary ETL procedures. The tables created must be without any indexes or primary or foreign keys; when making transformations row by row in the table records, if there are any indexes or keys then the process will take more time than necessary to check each one. What we want is to simply retrieve the raw data from the operational database and place it onto the staging area database without any waste of time.

.

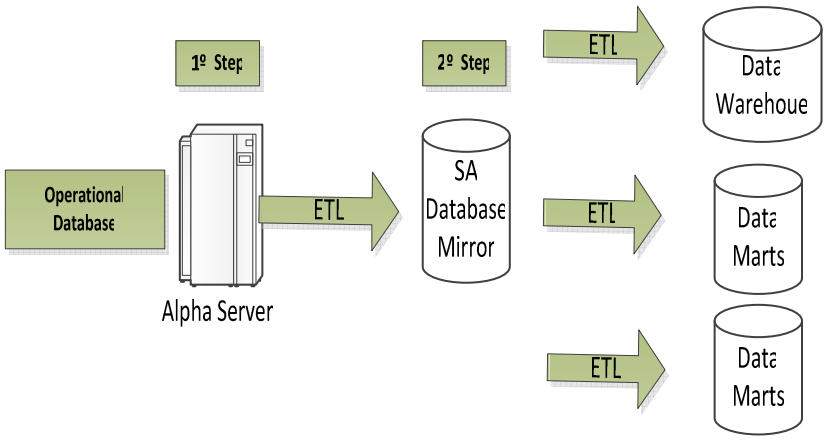

Figure 14 - Staging Area ETL procedures (Source: Gameiro, 2011).

Figure 14 explains this fact. In step one, we copy all the raw data using an ETL tool from the operational database onto our de-normalized database tables present in the staging area, applying the required database criteria or filters to the extraction process.

The staging area is just the temporary place where the data is copied to be available for integration on the data warehouse, making sure that this copy process doesn't affect the business cycle activities, and the business operational processes. This process, therefore, has to be configured to execute in an incremental and periodical way, depending on the business activity.

Step two shows our second stage of the ETL procedure, where we make the transformation and cleaning necessary from the staging area to the data warehouse and data marts. In this step, we need to have all the data free of inconsistencies and errors, according to the end user's search for subject oriented data, in order to select and deliver the correct information.

In ETL procedures, the IT team must assure the data quality present in the data warehouse and data marts. Some criteria can be followed to guaranty that quality Humphries et al. (1999):

- **Assess the current level of data quality** – Determining the level of data quality present in the data warehouse, setting up the importance of which data items are more valuable to subject orientated data warehouses and to the decisions makers.

- **Identify key data items** – In the data warehouse, we must ensure the best data quality in the valuable key data items first, before the remaining and less valuable data items.

- **Define cleaning tactics for key data items** – For each data item with poor data quality, there must be set cleaning tactics to remove all the unnecessary data information nested into the key data item.

- **Define error-prevention tactics for key data items** - The best way to eliminate error situations is to prevent them from happening, which can be achieved by introducing prevention tactics when we copy from the operational data source to the staging area, preventing in an early stage the data from being erroneously inserted into the data warehouse system.

- **Implement quality improvement and error-prevention processes** – By utilizing software tools, we can improve the quality of the key data items and prevent errors from happening, creating processes to execute quality improvements and error prevention.

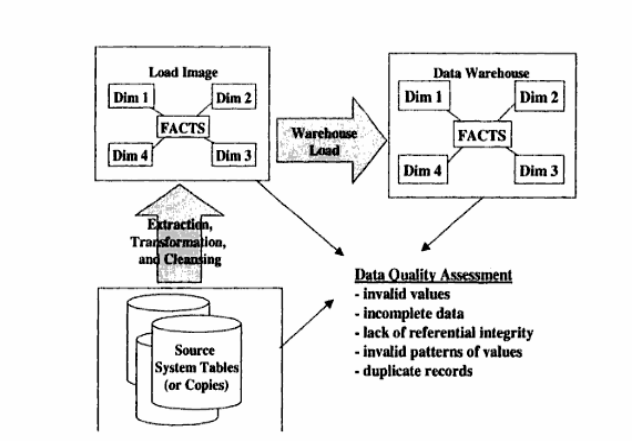

Figure 15 - Data Quality Assurance (Source: Humphries et al., 1999)

Some of the problems in figure 15, such as invalid values, incomplete data, duplicated values or lack of data integrity, can appear during ETL processes.

This must be solved before integrating the data into the data warehouse, which means an error-prevention and data integrity mechanism must be created (e.g. checking the number of rows for data completeness, the numeric columns and date columns for invalid values, and the key records for duplicated values).

To finalize our implementation of the data warehouse, the next task will be to perform test loads in the development environment. Before putting it into production, the data warehouse must be tested for performance and accuracy, as well as to discover what optimization may be needed or errors to be corrected.

The loading phase, the final stage of the data warehouse system, is a straight forward process and the simplest one amongst all the other data warehouse procedures. All that is required to do is to load and integrate the transformed data items onto the corresponding data sources.

4. Empirical Work

The focus of my empirical work was in one ETL process that copies the accounting information from the operational database to the staging area database; the idea is to create a mirror of the current operational database to a temporary database placed on the staging area and then execute the necessary ETL procedures over these mirror tables.

This is a necessary procedure because of the non-existence of direct access to the operational information. It is a secure way of utilizing the data without messing around the production operational database, allowing the IT team to work on the necessary operations.

In developing this ETL process, the approach applied in my work was to execute the same process using two different ETL software tools in order to compare and document which tool is more efficient.

One of the tools used was the SQL Server 2005 Integration Services (SISS), from the proprietary vendor Microsoft, which is the ETL application tool used in the organization. The other was the open source software Pentaho Data Integration Community Edition (Kettle).

The process involves the replication of the data integration on both ETL tools, using the same workflow logic reproduced with the SQL Integration Service and the Kettle ETL tool.

The process loads data onto a staging area and updates the data periodically depending on the user configuration.

The proprietary ETL tool SQL Server Integration Services (SISS) is an SQL Server component available just on “Standard” or “Enterprise” editions; it is a platform for data integration and workflow applications.

The SISS allows the user to create packages in which the data integration process is developed and workflows are created using the Microsoft Visual Studio or BIDS Business Intelligence Development Studio as IDE (Integrate Development Environment).

The package structure in the SISS is developed to carry out the ETL process task by task, using objects of different types such as data flow tasks, data connections (ADO, OLEDB, ODBC, Flat files, Files, Excel), event handlers for logging and also variables for package configuration.

Each package can be executed by using an exec program called “DTEXEC” that runs the ETL process.

There are different types of deployment, but the most common and easiest to use is setting up a batch file in Windows, and scheduling a task for our ETL process.

4.1. Open source ETL Tool

Amongst all the available open source tools for ETL data integration, two stand out: Jaspersoft ETL, and Pentaho Kettle Data Integration.

From my point of view, the first decision would be to go for Jaspersoft because of its well-known reputation in Reports. But the work focus is in the data transformation, and Pentaho Kettle Data Integration is more capable and has better documentation and data integration for the job.

Some key features of these two tools (Levin, 2008):

(http://www.jaspersoft.com/)

- Uses a code-generating approach and the GUI is based on the Eclipse RC;
- This ETL project began in 2006, which makes it a premature software (in terms of developmental years) when compared to Pentaho Kettle (begun in 2001);
- The community is much smaller when compared to Pentaho.

(http://www.pentaho.com/)

- Better data integration, ETL, and workflow automation;
- Uses a meta-driven approach and as a very intuitive GUI is very easy to use;
- Strong Community;
- Uses a stand-alone java engine to process all the data transformation.

Kettle is a set of tools and applications which allows data manipulations across multiple sources. It is easy to use and fast to learn, and documentation is easily available. The community is large enough for answering any question regarding this tool.

The user interface is so fast and remarkably easy to work with, which makes it ideal for developing ETL projects with deadlines, delivering an intuitive and graphic way of developing the workflow transformations and other data processes to its user.

Using a meta-driven approach in Kettle eliminates the need for writing, compiling or maintaining code making, thus offering a clear reduction of development effort. It provides the developer a rich set of graphical design components that without any training at all are easy to use and develop.

The main components of Pentaho Data Integration are:

- Spoon - a graphical tool which makes the design of ETL process transformations easy to create. It performs the typical data flow functions like reading, validating, refining, transforming, and writing data to a variety of different data sources and destinations. Transformations designed in Spoon can be run with Kettle Pan and Kitchen;

- Pan - is an application dedicated to running data transformations designed in Spoon;

- *Chef* - a tool to create jobs which automates the database update process in a complex way;

- *Kitchen* - an application which helps execute the jobs in a batch mode, commonly using a schedule which makes it easy to start and control the ETL processing;

- *Carte* - a web server that allows remote monitoring of the running Pentaho Data Integration ETL processes through a web browser.

4.2. ETL Process

The ETL process begins with copying the raw data from the operational database (Alpha Server) to the mirror database present in the staging area.

If this procedure succeeds, the next step is to delete the old values present in the destination database, which is the final database where the transformed and cleaned data items will be loaded.

After this, we can continue to the straightforward process of loading and copying the data items from the staging area to the mirror final tables, where the information will be accessible for posterior utilization by the data warehouse system.

Figure 16 shows the workflow in the ETL Kettle open source tool using three data transformations tasks for executing each procedure.

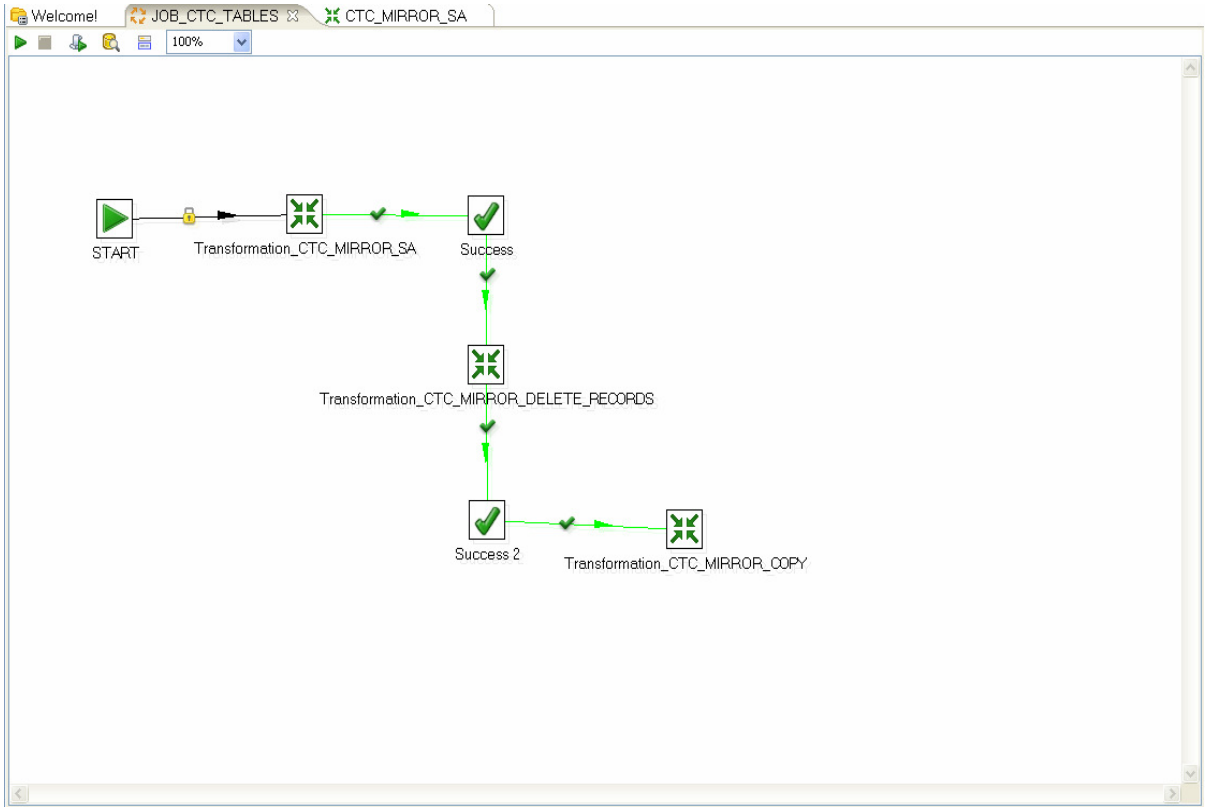

Figure 16 - Extraction and Loading data into a Staging Area Database (Kettle Open Source tool).

The present figure represents a Kettle Job, a file type that executes the workflow shown. Each link connection in the figure represents the success of the next procedure or task; if one process succeeds, the next process can be executed.

Each task in this figure is represented by a Kettle transformation, a file type that details the ETL process and structures the workflow row by row.

In each data transformation, the logic flow must be thought out in the same manner, line by line. This is why the division in distinct transformations simplifies the ETL process, making it easier to understand and work on than other processes.

For example, in SQL Server Integration Services (SSIS), the workflow is carried out task by task. It uses a package file that contains all the ETL procedures for designing/building/testing, for example, SQL execution queries or data flows.

Figure 17 shows the control flow followed in this process.

First of all the package truncates all tables present in the staging area database in order to clean and delete all the temporary tables renewing the data, restarting the copying process of the new and refreshed data from the operational database.

We then start the copying process from the predefined tables, which are considered in a configuration table created to insert and define each table's parameters for copying.

They are copied in two separate ways:

1. We start by copying all the data and establishing a date interval (week, month, year, etc.) to copy any data which suffered any type of change;
2. We copy the full data records from the operational database.

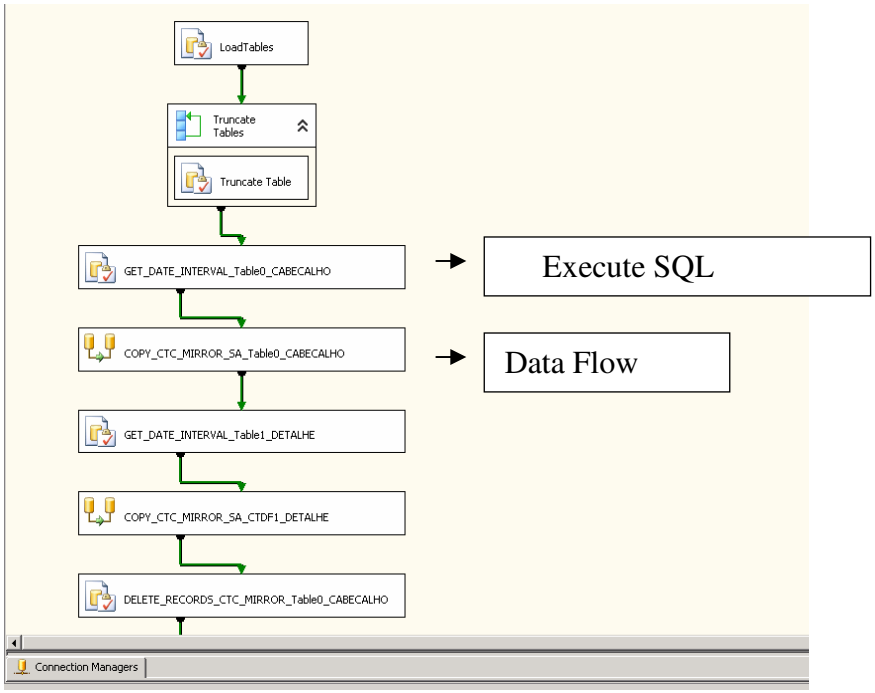

Figure 17 - SQL Integration Service Package for loading onto a database staging area.

Figure 17 show that this package includes an SQL Task, which, upon establishing a specific date, can select, insert, update or delete any value from the database. This figure specifically uses the SQL task to get a date for setting up the criteria filter for the next copying task of a table.

The dataflow tasks here represent the transformations in Kettle, where we define which data records will be copied, transformed or cleaned (figure 18), applying in this detailed data flow the transformation needed, such as:

- Merges;
- Lookups;
- Sorts;
- Aggregations;
- Copy Columns;
- Unions;
- SQL commands.

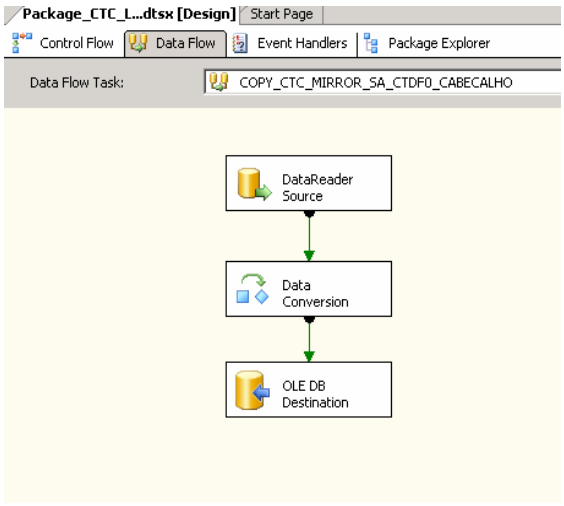

Figure 18 - SQL Integration Service Package Data flow.

In figure 18, we see a data conversion object, which is used to convert all the string fields from Unicode type to Non-Unicode type, a necessary step to insert the raw data into our staging area, as without it the copying process wouldn't work at all.

The way to circumvent this step resides in changing the type of columns of all staging area tables, which can cause problems with others operational applications that use these tables, making an unfeasible choice.

This workflow structure's task allows the developer to include all in one package: the transformations for copy, and deletion and loading from the staging area to the mirror databases for data warehouse usage.

In the open source approach, the workflow present in the figure 19 represents the copy from the Alfa Server (Operational Database) to the staging area databases, which is one of the three transformation steps present in figure 16.

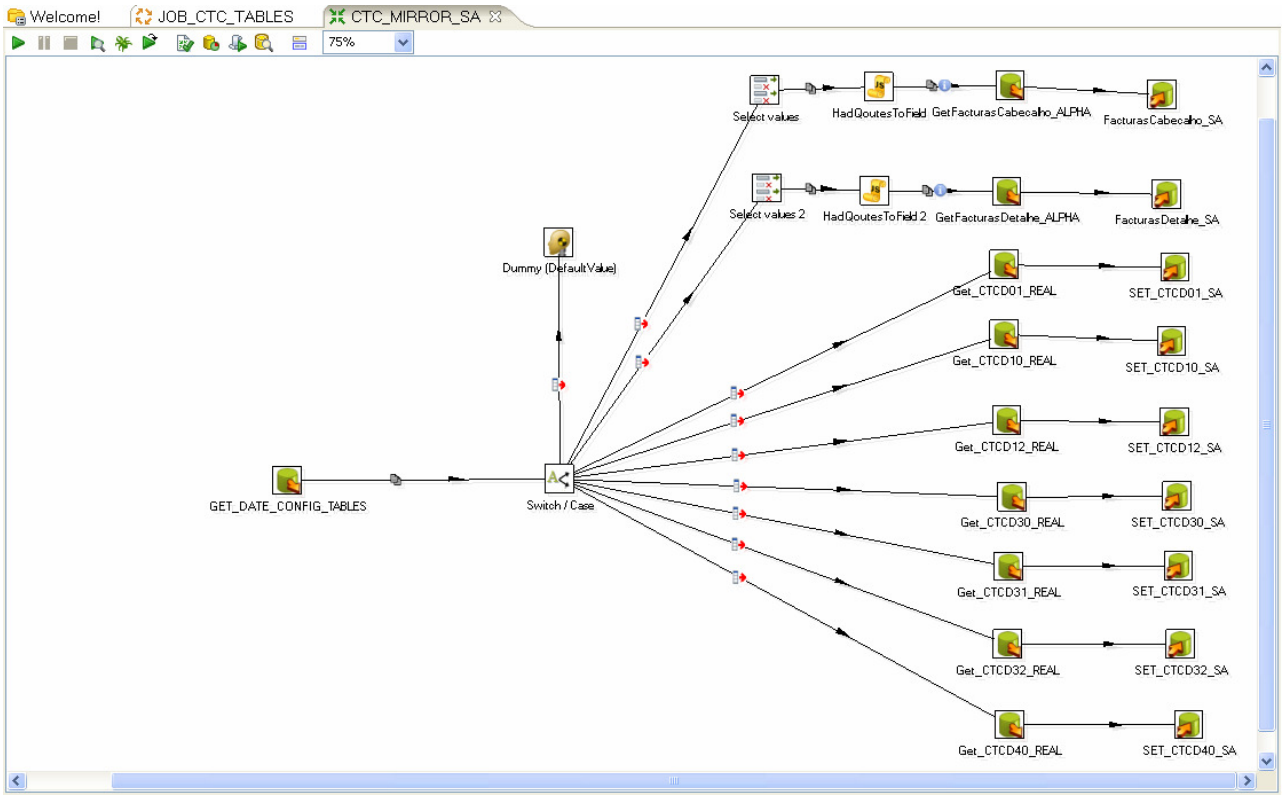

Figure19 - Extraction and Loading data onto a Staging Area Database (Kettle Open Source tool).

This workflow uses the same configuration table present in the SSIS package, which is a configuration table created to simplify the database copy tasks and maintain a well-structured process.

It has all the necessary tables present in the operational database to copy, and specifies whether a certain table is configured to be copied in an incremental or integral way, copying all the raw data records without applying a filter.

When a table is configured to be copied in an incremental way, it uses a date field present in one table view, created to associate the table name and the incremental flag of the operational database tables.

This table view calculates the date with two fields present in the configuration table (example Table 4).
One field determines the period type, e.g.:

- Year (yy)
- Month (m)
- Week (ww)

The other field determines the period quantity (1 or 2 weeks, for example) from which the date interval will be calculated, which is the filter criterion to extract the selected information. In table 4, we can see that, for instance, "Table 1" is configured to have to select data information of 3 months prior.

Table 4. Example of Configuration Table

Table Name	Period	Period Time	Incremental Flag
Table 1	M	3	True
Table 2	M	3	False

The incremental copy configuration of the several operational database tables is just for those tables that have a field date present and in the correct date format, so that this date filter is applied in the ETL process successfully.

In figure 19, the object "GET_DATE_CONFIG_TABLES" represents a table input step that contains an SQL selection; this table input returns, for each operational database table, the name of the table and the date field filled if the table is configured to be copied in the incremental way.

Kettle uses these two return fields from the table input and converts them into variables that can be used in the next transformation object.

These two variables, one for the table name and other for the date value, are used in the switch case object present in the workflow. A kettle object type that works as a control structure is particularly useful if we want to compare the same variable (or expression) with many different values, and execute a different workflow depending on the comparison result.

In this case, the value to compare is the variable table name, retrieving the correct date value from the correct table and then sending this date variable to the next transformation step.

The input database table shown in figure 20 named “Get_Facturas_Cabeçalho_Alpha”, gets the date variable value and uses it on SQL’s selection query, in which exists a variable that will be replaced by the date variable.

Figure 20 -Table Input Step.

After performing the query selection on the operational database table, we pass on to the next transformation step, that is the insertion and loading of data into the staging area database table seen in figure 21, which is an output table to store the raw data from the previous step.

Figure 21 - Table Output Step.

The referred workflow is just one of the three Kettle ETL processes created and, in opposition to the SQL Integration Services, we cannot put all the ETL transformation steps into just one file because of the workflow logic present in Kettle.

To call and execute the ETL transformation in Kettle we use the JOB file (Figure 16) which creates a workflow structure of Data integration. Scheduling and running this JOB file is like executing the package in the SQL Integration Services because we also use a batch file to create an executable file that runs the ETL process.

In Kettle, to create the batch file for scheduling the ETL task, we must explicitly refer the path to the Kitchen application which executes the job in a batch mode.

We also apply variables to change the batch file (e.g. in runtime, the ETL transformation server connections’ attributes), working in a similar way to the SQL Integration service batch mode.

To work and develop in Kettle is straightforward and it becomes easy to design our data integrations process because of the simple manipulation of several data components available in Kettle design mode, which makes designing the workflow process straight forward.

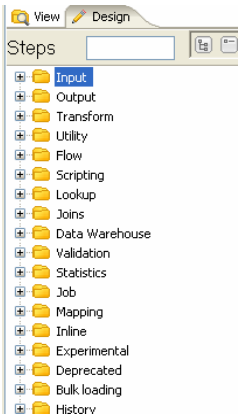

Figure 22 - Kettle Types of Transformation Steps

Figure 22 demonstrate the various types of transformation steps that help to develop our workflow process:

- Input Step – Data input, basic steps: Access Input, CSV Input, Excel Input, Generate random values, Generate rows, Json input, LDAP input, RSS input.

- Output Step – Output of Data, basic steps : Access Output, Delete, Excel Output, Json Output, LDAP Output, RSS Output, Table Output,SQL Output, XML Output)

- Transform – Data transformations, basic steps: Add constants, Replace strings, Unique rows, Value mapper, Split fields, Add sequence, Strings cut, Calculator.

- Flow – Control structures transformations, basic steps : Blocking Step, Filter rows, Abort, Dummy (do nothing), java filter, Switch/ Case.

- Utility – Utility transformations, basic steps: Change file encoding, Run SSH commands, Write to log, Execute a process, Send a message to syslog, Mail.

- Scripting – Execution of scripts, basic steps: Execute SQL script, Execute row SQL script, modified Java script value, Regex Evaluation, User Defined Java class.

- Lookup – Data consult, basic steps: Web service lookup, Check if ws is available, Call DB procedure, Database join, Database lookup, Table exists, File exists, HTTP Post.

- Joins – Joins between data sources, basic steps: Join rows, Merge Join, Merge Rows, Sorted Merge, XML join.

- Validation – Validation Data objects, basic steps: Credit card Validator, Data Validator, Mail Validator, XSD Validator.

- Data Warehouse – Data Warehouse Transformation, basic steps: Combination lookup, Dimension lookup.

The ETL processes here summarised are created and developed to copy crucial data information from the operational database into the staging area and then copy that data to the data warehouse database tables.

This ETL procedure is run every day, assuring data availability to the end users. At the same time, the process is executed using the package file developed in SQL Integration Services together with the Job created in the open source tool Kettle Data Integration.

With these two different execution approaches, we can perform tests to guarantee data consistence and see which process is more efficient. As for the execution time, both of the ETL tools used have a similar reading and writing data performance and it is not in the execution performance that the open source approach is more relevant.

The most fundamental key feature present in Kettle is the usability and development/design time, and effort of the developer.

This reason is present in figure 18: the example of data transformation in SQL Integration Services which uses a data conversion object, a step necessary in this ETL tool when the column type is different from the source operational database of the destination database.

This conversion requires a larger effort from the IT team and increases the development time in the project planning, because the need for selecting and changing columns' types one by one can be a hard task if in one table there exist four hundred columns to change.

In Kettle this issue doesn't exist because, in the ETL process created, we can directly retrieve and load the data without the need of data conversion in the middle. This has a significant advantage over the SQL Integration Service, as we can carry out the workflow process in half the time when doing it in Kettle using its design structure.

The main difference between these two tools is in the workflow logic; when creating and developing an ETL process in Kettle the workflow structure is made row by row.

In the SQL Server Integration Service, the workflow is made task by task. In this, the Kettle ETL tool is more intuitive than its counterpart, which makes us think more about the row-by-row flow for our integration process.

Other differences are relevant, and they include the database connectors for linking diverse operational databases (databases crucial for the daily business activity) that guarantee easy data extraction without the need for data conversion between transformations, saving extra process time.

Kettle database connectors surpass the SQL Server Integration Services which have few and not very specialized data connectors in comparison, where there are several specialized database connectors like SAP ERP system, ORACLE, Informix, Info Bright, AS/400, IBM DB2.

5. Summary and ETL Tool Comparisons

5.1. ETL Tools Comparisons

When deciding on the right tool it is also necessary to make a strong decision as to which software to search or purchase. Keeping this in mind, making a comparison of the two different software tools can help us decide on which tool is more appropriate for the project.

The metrics for such comparison used on the work that I was involved in during the development of the data warehouse using Pentaho Kettle and the SQL Server Integration Services were:

- Total Cost of Ownership
- Risk
- Ease of use
- Design Issues
- Logging
- Deployment

5.1.1. Total Cost of Ownership

Total cost of ownership is the overall cost of a product; this means that a product has initial costs involving services, licenses, consultation, support, training and maintenance, before the product can be in full use[5].

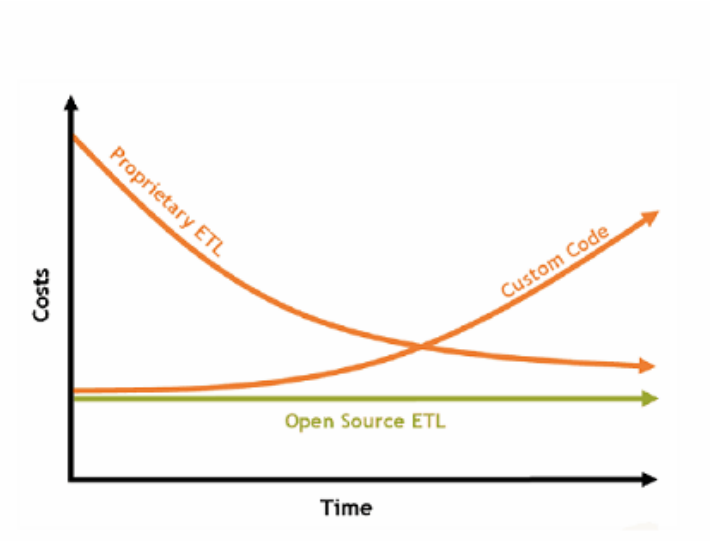

Figure 23 - ETL costs over time (Source: Levin, 2008))

[5] ETL comparisons by Jonathan Levin (2008)

For the open source ETL tools, the costs are lower or non-existent when compared to a commercial solution, but support or knowledge of the tool is what organizations need to pay if necessary.

5.1.2. Risk

When developing a data warehouse, project managers always have associated risks which they have to face and pay for, such as:

• Going over budget;
• Going over deadline;
• Non completion of the requirements or expectations of the end users.

At this point, the open source tool has a lower risk when compared with the commercial one because of the licensing fees used by commercial software.

5.1.3. Ease of use

Where the aspect is the focus on the GUI (Graphical User Interface), it simplifies user interface, and facilitates and reduces the time for the development process and also for the associated training time.

SQL Server Integration Services have a heavy GUI, which require training and know-how to use.

Pentaho Kettle has an easier GUI for development and the training is simple because of the documentation available online and in the community. It also has a repository functionality to save or re-use ETL transformations in a practical manner.

5.1.4. Design Workflow

Design Workflow is related to the workflow development type used to know if there is a cleaner and more practical design to use.

SQL Server Integration Services has a development approach thought to be done task by task. It uses one or more file packages to implement the workflow process, but if too many workflows are present in the same package the design of the workflow will become heavier.

Pentaho Kettle: the design workflow approach makes us think row by row; it uses transformation files that are designed for only one process type. For example, Transformation File 1: copy of the operational database onto the staging area; Transformation File 2: cleans old and deleted information from the staging area; etc. Because the transformation execution is done row by row, it makes it easier to perform data quality routines, and helps to maintain the workflow processes by separating the different ETL processes. We can have a clear view of what we want to modify or delete from the ETL process routines.

5.1.5. Logging

This is the functionality that discovers data errors and performs logging information along the ETL processes' activities.

SQL Server Integration Services uses different logging types. The logging type chosen in this work was the database table logging type, which uses stored procedures that are created and can be modified to customize our logging database table. This type of customizable logging works much better than Pentaho Kettle, just because the possibility of modifying our own logging functionality delivers to the ETL developer more logging possibilities.

Pentaho Kettle has one logging type that writes directly to a database table created by Kettle with different levels of detailed information, different from SQL Server Integration Services. It cannot, however, offer the customization that its counterpart has to alter the stored procedure according to our needs.

5.1.6. Deployment

SQL Server Integration Services is a Microsoft product-based system that only runs on Windows Operative systems and can only be deployed on windows machines.
In order to run, it can be deployed in a SQL server Data engine or schedule task.

Pentaho Kettle runs a stand-alone Java program that is capable of running on any operative system and any machine; for its reduced hardware specs it can also use "slave servers", different machines which help the ETL processing.

Table 5 - Metric Comparisons

	Kettle	SSIS (SQL Server)
Total cost of ownership	Low	High
Risk	Low	Medium
Ease of use	Intuitive	Heavy User Interface
Design Workflow	Row by Row	Task by Task
Logging	Detailed	Customizable
Deployment	Multiple Operative Systems	Windows only

I believe that the metrics applied contribute to the right choice for any software tool, simply because they contribute to a better judgement and decision in any project implementation.

Considering these software aspects, we can better understand the organization's needs; when choosing which software they want to use or purchase it is best practice to know, beforehand, the aspects to take into consideration as shown in these metrics.

6. Conclusions

After using this two ETL tools, it is clear that the Pentaho Kettle Data Integration product is capable of doing the same thing as its counterpart SQL Server Integration Services, but surpass it in its user friendly GUI which is really easy to use and develop.

When creating the data integration workflow, the process in Kettle doesn't consume much of the machine's resources. This is because it is lightweight in terms of hardware when compared to the commercial product and this indeed helps the project's development and execution.

In conclusion, choosing the right ETL tool depends on the project's dimension, budget, costs, and risks associated. Therefore, all software products are valid but not necessarily better; we must consider the quality/costs/needs ratio for better judgement and decision.

The decision on which software tool to acquire, in my opinion, must be related to the project's type and ambitions. For instance, an organization that just needs to have a Data Warehouse in a first stage of development, doesn't need a full set of software tools that do more that they require.

If the intent is to develop database processes and database transformations/integrations for a small to medium Data Warehouse, then Kettle is the correct choice.

If, however, the amount of data information in the project is intended to be by far greater than expected and the BI project is far more ambitious, then a more mature software tool is the ideal choice, and commercial versions in this case may be a better choice.

Similar literature review works show and help conclude that when using open source software we must be certain that the size of the project in hand can be sustained by the open source approach, allowing the project to terminate successfully and without taking any unnecessary risks.

Golfarelli's work (2009) reflects this dynamic with this statement: "Although OS BI platforms are still not as sophisticated as commercial ones, we can state that they have got a sufficient level of reliability and must be considered a valid alternative to commercial suites. This is particularly true in small and medium-sized enterprises where the quantity of data and the workload are not critical points."

In the future, open source software tools for ETL and BI suites will be in the same league as the reputable and top commercial versions. Let's wait for this and for the evolution in these open source software tools and their future developments.

Implementing a data warehouse is the first step in any business intelligence system. It is the base and the core of all business intelligence architectures, but is one small portion of what these systems can do and provide.

In business intelligence, several technologies are used along the life cycle of the system. Two of them distinguish themselves, providing for the end-users and decisions-makers a quick view of the general business situation and evolution. These are:

- Advanced Analytic (Statistical Analytic and Data Mining[6])
- Corporate Performance Management (Portals, Scorecards, Dashboards)

Future work will be done regarding the implementation of this data mining techniques using the same approach as in this dissertation, i.e. using open source software as front-end tools to discover and investigative if open source software in this area is useful and trustworthy enough to be used in all organizations' business intelligence projects.

[6] Data mining is a set of analytic techniques to discover relevant knowledge within the business organizations data warehouse.

7. References

Albrecht, Alexander and Naumann, Felix: Managing ETL Processes. Hasso-Plattner Institute at the University of Potsdam, Germany, 2008

Cederberg, Niclas: Master's Thesis in Business Administration MBA programme, 2010

Cramer, Renato: Estudo Analítico para ferramentas open source para ambientes OLAP, Agust 2006

Druzdze,l Marek J. and Flynn, Roger R.: Decision Support Systems, In Encyclopedia of Library and Information Science, Third Edition, Marcia J. Bates and Mary Niles Maack (eds.), Taylor & Francis, Inc., New York, 16 February 2010.

Imhoff, Claudia : Disruptor:
The Rise of the Open Source Data Warehouse, BeyeNETWORK, US Edition, December 2, 2 008 http://www.beyenetwork.com/view/9051.

Inmon, William H.: What is a Data Warehouse? Copyright 2000

Han, J. and Kamber, M.: Data Warehouse and OLAP Technology for Data Mining, January 16, 2000 (Chapter 2).

Humphries, Mark and Hawkins, Michael W. and Dy Michelle C.: Data warehousing: architecture and implementation, December 30, 1999.

Gardner, Stephen R.: Building the Data Warehouse, Communications of the ACM, September 52-60, 1998

Golfarelli, Matteo: Open Source BI Platforms: a Functional and
Architectural Comparison, DEIS, University of Bologna, Viale Risorgimento 2, Bologna, Italy, 2009

Jayanthi, Ranjan : Bussines Intelligence: Concepts, Components Techniques And Benefits, Journal of Theoretical and Applied Information Technology, 2005

Klawans, Barry: Business Intelligence Journal Saving Time and Money, Why Open-Source BI Makes Sense, October 2006

Levin, Jonathan: ETL comparisons, http://mysqlbarbeque.blogspot.com, 2008

Lewis, Grace Alexandra and Comella-Dorda, Santiago and Place, Pat and Plakosh, Daniel and Seacord, Robert C. : Enterprise Information System Data Architecture Guide Carnegie Mellon University, 2001

Maira, P. & Marlei, P.: The value of “business intelligence” in the context of developing countries. In Proceedings of the 11th European Conference on Information Systems, ECIS 2003, June 16-21
Naples, Italy. Retrieved April 6, 2008, http://is2.lse.ac.uk/asp/aspecis/20030119.pdf

Maribel, Y & Ramos, I.: Business Intelligence: Tecnologias Da Informação Na Gestão De Conhecimento. FCA – Editora de Informática, Lda, 2006

Naimuzzaman, MD. : Implementation of Business Intelligence and Reporting Tools, Master of Science Thesis in the Programme Software Engineering and Technology, June 2009

Narayan, Siddharth, Team EDSS: BI Reporting Tools Assessment, Kelly School of Business Indiana University, Spring 2009

Olszak, Celina M. and Ziemba, Ewa : Approach to Building and Implementing Business Intelligence Systems, Interdisciplinary Journal of Information, Knowledge, and Management Volume 2, 2007

Pequeno,Valéria and Abreu, Salvador and Pires, João Carlos Moura: Using a Contextual Logic Programming Language to Acess Data in Warehousing Systems, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, October 2009.

Ponniah, Paulraj : Data Warehousing fundamentals: A comprehensive guide for IT Professionals,John & Wiley Sons, Chichester, 2001.

Power, D.J.: A Brief History of Decision Support Systems. DSSResources.COM, World Wide Web, http://DSSResources. COM/history/dsshistory.html, version 4.0, March 10, 2007.

Simitsis, Alkis and Vassiliadis, Panos: Methodology for the Conceptual Modeling of ETL Processes, In Proc. of DSE’03, 2002.

Sharma, Umashanker and Gosain, Anjana: Dimensional Modeling for Data Warehouse GGS, Indraprastha University, Delhi, 2010

Solomon, Negash : Communications of the Association for Information Systems (Volume13, 2004) 177-195, (2004)

Watson, Hugh J. and Ariyachandra, Thilini and Matyska Jr, Robert J.: Data Warehousing Stages of Growth. IS Management, 2001: 42~50

MODELO EUROPEU DE CURRICULUM VITAE

INFORMAÇÃO PESSOAL

Nome	**GAMEIRO,CARLOS ALBERTO MONTEIRO**
Morada	Rua Nuno Alvares Pereira n.º 56 –A 2080 – 160 Almeirim
Telefone	243591056
Telemóvel	914101053
Correio electrónico	c_gameiro@hotmail.com
B.I.	12470281
Nacionalidade	Portuguesa
Estado Civil	Solteiro
Data de nascimento	06/08/1983

EMPREGO PRETENDIDO /ÁREA DE COMPETÊNCIA

Programação / Web design / Consultor

EXPERIÊNCIA PROFISSIONAL

• Datas (de – até)	De Agosto de 2010 até ao momento
• Nome e Endereço do Empregador	Zona Industrial Abrunheira Sintra Business Park-Edifício 4-Escritório 2-A,Abrunheira-São Pedro Penaferrim 2710-089 SINTRA; Website: www.aldautomotive.pt
• Área	Programação
• Função ou cargo ocupado	Programador Informático
• Principais actividades e responsabilidades	Desenvolvimento e manutenção de projectos internos da empresa ligados a gestão de frota automóvel e desenvolvimento de um data warehouse integrado com actividade da empresa.
• Datas (de – até)	De Janeiro de 2009 até Julho 2010
• Nome e Endereço do Empregador	Risa Informática, Lda., Rua 24 de Junho, Apartado 63, 2384-908 Vila Moreira, Telef.: 249 889 123; Fax: 249 889 127; WebSite: www.risa.pt
• Área	Programação
• Função ou cargo ocupado	Programador Informático
• Principais actividades e responsabilidades	Definição e desenvolvimento de projectos de sistemas de gestão integrados e de Sap Business One dentro das áreas Industrial, Financeira e Gestão de RH, bem como projectos dentro da área Web.

• Datas (de – até)	De Abril de 2008 até Janeiro de 2009
• Nome e Endereço do Empregador	Risa Informática, Lda., Rua 24 de Junho, Apartado 63, 2384-908 Vila Moreira, Telef.: 249 889 123; Fax: 249 889 127; WebSite: www.risa.pt
• Área	Estágio Profissional / Programação
• Função ou cargo ocupado	Programador Informático
• Principais actividades e responsabilidades	Desenvolvimento de projectos de sistemas integrados de gestão, de Sap Business One e alguns projectos dentro da área Web.
• Datas (de – até)	De Setembro de 2007 até Outubro de 2007
• Área	Desenvolvimento Web.
• Função ou cargo ocupado	Programador Freelancer.
• Principais actividades e responsabilidades	Desenvolvimento e manutenção do website: www.piquetedofuturo.com, utilizando a linguagem de programação PHP e AJAX no guestbook do website.
• Datas (de – até)	De Novembro de 2007 até Dezembro de 2007
• Área	Desenvolvimento Web.
• Função ou cargo ocupado	Programador Freelancer.
• Principais actividades e responsabilidades	Desenvolvimento e manutenção do website: www.dinatejohidro.com, suportado por um gestor de conteúdos.
• Datas (de – até)	De Janeiro de 2008 até Fevereiro de 2008
• Área	Desenvolvimento Web.
• Função ou cargo ocupado	Programador Freelancer.
• Principais actividades e responsabilidades	Desenvolvimento e manutenção do website: www.kicaplantas.com, suportado por um gestor de conteúdos.

FORMAÇÃO ACADÉMICA E PROFISSIONAL

• Datas (de – até)	De Setembro de 2010 até ao Momento
• Nome e tipo da organização de ensino ou formação	ISCTE (Instituto Superior de Ciências do Trabalho e da Empresa)
• Principais disciplinas/competências profissionais	Introdução ao Software de Código Aberto, Fundamentos de Sistemas Operativos de Código Aberto, Disseminação de Software de Código Aberto, Introdução ao desenvolvimento de Software de Código Aberto, Redes de Computadores em ambientes de Código Aberto , Bases de Dados em ambientes de Código Aberto, Desenvolvimento de Aplicações Web, Segurança em Redes de Computadores, Modelos de Negócio e Economia do Software de Código Aberto , Engenharia de Software em ambientes de Código Aberto
• Designação da qualificação atribuída	Mestrado em Open Source Software
• *Classificação obtida (se aplicável)*	
• Datas (de – até)	De Setembro de 2001 até ao Julho de 2007
• Nome e tipo da organização de ensino ou formação	Escola Superior de Gestão de Santarém
• Principais disciplinas/competências profissionais	Algorítmos e Estruturas de Dados,Análise e Concepção de Sistemas de Informação,Sistemas de Computação,Linguagens de Programação I e II,Sistemas de Gestão de Bases de Dados Ambientes de Exploração,Gestão de Projectos de Software,Sistemas Distribuídos,Seminarios Presenciais .
• Designação da qualificação atribuída	Licenciatura em Informática
• *Classificação obtida (se aplicável)*	13 Valores
• Datas (de – até)	De Setembro de 2001 até ao Julho de 2005
• Nome e tipo da organização de ensino ou formação	Escola Superior de Gestão de Santarém
• Principais disciplinas/competências profissionais	Algorítmos e Estruturas de Dados,Análise e Concepção de Sistemas de Informação,Sistemas de Computação,Linguagens de Programação I e II,Sistemas de Gestão de Bases de Dados, Ambientes de Exploração.
• Designação da qualificação atribuída	Bacharel em Licenciatura em Informática de Gestão
• *Classificação obtida (se aplicável)*	13 Valores

APTIDÕES E COMPETÊNCIAS PESSOAIS

PRIMEIRA LÍNGUA	LÍNGUA PORTUGUESA
OUTRAS LÍNGUAS	
	LÍNGUA INGLESA
• Compreensão escrita	Excelente
• Expressão escrita	Boa
• Expressão oral	Boa
	LÍNGUA FRANCESA
• Compreensão escrita	Boa
• Expressão escrita	Razoável
• Expressão oral	Razoável
APTIDÕES E COMPETÊNCIAS SOCIAIS	• Espírito de Equipa; • Boa capacidade de relacionamento; • Boa integração; • Sentido de responsabilidade.
APTIDÕES E COMPETÊNCIAS DE ORGANIZAÇÃO	• Capacidade de Liderança; • Sentido de organização; • Capacidade de Gestão de equipa; • Espírito de Iniciativa; Estas características foram adquiridas ao longo da vida académica, e em especial na organização de um Seminário com o tema " Inteligência Artificial e Data Mining" na Escola Superior De Gestão de Santarém.
APTIDÕES E COMPETÊNCIAS INFORMÁTICAS	• Conhecimentos de sistemas operativos: Windows (XP /2000/ Server 2003), Linux. • Conhecimentos de desenvolvimento Web: Bons em ASP.Net, ASP, HTML, Razoáveis em XML, Web Services, JavaScript, PHP e Perl. • Conhecimentos de linguagens de programação: Bons em VB e VB.Net, razoáveis em C, C#.Net e Java. Conhecimentos de bases de dados: MySQL, MS Access, Oracle9, SqlServer 2005. • Outros Conhecimentos: Photoshop, Ms Project, Ms Visio, ferramentas Case, linguagem de modelação UML e multimédia (Dreamweaver,Flash); • Facilidade de aprendizagem de novas linguagens; • Interesse por linguagens de Opensource ; • Conhecimentos de Crystal Reports ; Adquiridas no decurso da formação académica e/ou por motivação pessoal.
OUTRAS APTIDÕES E COMPETÊNCIAS	• Interesses por Artes como: Desenho, Musica, Cinema; • Praticante de basquetebol, e de desporto; • Interesse por novas tecnologias na área de informática.
Carta de Condução	SA-106360 1, Categoria Ligeiros B

INFORMAÇÃO ADICIONAL

Publicação ACM (Association for Computing Machinery)
Artigo :« "Implementation of business intelligence tools using open source approach"»