Agile ETL*

Cristiano Xavier[1], Fernando Moreira[1,2]

1) Universidade Portucalense, Porto, Portugal
cristiano.xavier@outlook.com, fmoreira@upt.pt

2) CLEGI, Universidade Lusíada de Vila Nova de Famalicão, Vila Nova de Famalicão, Portugal

Resumo

Agile ETL is a tool for technicians working in the area of business intelligence, which facilitates consolidation of information in a central repository called data warehouse. Mechanisms have been established to create, control and monitoring processes of extraction transformation and loading of data, which are intend to give a faster response either in creating or monitoring processes. ETL Framework can quickly integrate and consolidate data with good performance indicators.

Palavras chave: extract; transform; load; data warehouse; business intelligence; agile; etl

1. Introdução

Companies to become more competitive have to respond faster to the needs because the constant fluctuations of markets. According to the study published by The Data Warehouse Institute, 510 companies, shows that with the arrival of the Business Intelligence there was a saving in time about 61% of companies and about 57% of companies had better strategic decisions while 56% had better tactical decisions, furthermore it has been inferred that 39% of companies generated a cost savings. [ECKERSON, 2010]

Creating a Data Warehouse is a complex process not only in the construction of the data model, but especially in the cataloging process that goes through three phases: Extraction, Transformation and Loading. [Kimball & Caserta, 2004]

ETL (Extract Transform and Load) system is much more than a tool for getting data from a source system to a central repository, it removes errors and corrects missing data, metrics provides confidence in data, data sets from multiple sources to be analyzed together and structure the data for use by end users tools. [Kimball & Caserta, 2004]

ETL is more than just data flow. It has the power to correct data errors and transform raw data into information that can be readily consumed by business users. In The Data Warehouse ETL Toolkit, Ralph Kimball and Joe Caserta state that the ETL portion consumes upwards of 70% of all resources required to build a data warehouse. [Kimball & Caserta, 2004]

Factors like Quality of Data, Complexity of the Source, Dependencies in the Data, Logging, In-House Expertise, Support, Disk Space and Scheduling, influence the approach to loading the data warehouse, which also affects the cost of the solution.

For minimize Data Warehouse implementation time, resources and cost, the proposed solution aims building a tool that automates and monitors processes of extraction, transformation and loading data. It is a framework that does not require large specialized expertise in Business Intelligence. The Framework allows create "Out Of The Box" extraction processes that put the information in a temporary area, component data transformation and loading of delete insert methodology or merge by primary key.

* Artigo publicado em Actas da CENTERIS 2013 – Conference on ENTERprise Information Systems / PRojMAN 2013 -International Conference on Project MANagement / HCIST 2013 – International Conference on Health and Social Care Information Systems and Technologies

There are two ways to operate Agile ETL, using PowerShell functions or through the Website in ASP.NET.

With Agile ETL, solutions paradigm like "Business Intelligence" tends to become "Data Intelligence" [O'Reilly, 2012], this change happens because solutions become more affordable, the ability to acquire tools of analysis will not be targeted just for big companies and the way they tend to get the final consumer, even for private purposes, analyze patterns and draw on their data to infer and make choices more assertive [Minelli, Chambers, & Dhiraj, 2012].

This paper is organized as follow. In section 2 we presented an overview (description of the implemented prototype and its components. In section 3 we presented and discussed a performance analysis the performance of the ETL processes, and finally, in section 4 the general conclusions of the work done and indication of future work are presented.

2. Overview

Through automation functions in the framework PowerShell scripts can respond in creating agile components that make up the entire flow from the creation of extractors that work as an interface with the data sources to integrate data already processed in accordance with the business model.

The architecture is based on components that can execute Store Procedures or SQL Server Integration Services packages that put information from multiple repositories into tables in a temporary area, known as Staging Area, this data is stored in raw, to minimize the impact in operation systems. After the data is temporarily stored in Staging Area, is time to work the information in order to consolidate according with the business rules into Data Warehouse. Finally, the data processing is performed on a multidimensional repository to provide an access method, visualization, and analysis with high performance and flexibility.

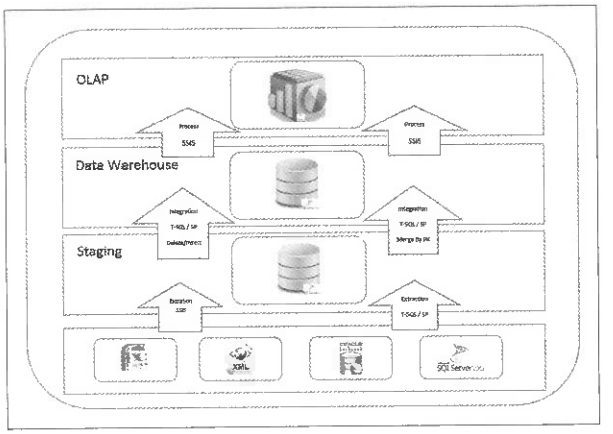


Figura 1 – Architecture Framework.

2.1. Major technologies

They were just Microsoft tools used because licensing issues and greater technical knowledge in this type of technology.

Microsoft Integration Services – platform for building enterprise-level data integration and data transformations solutions. You use Integration Services to solve complex business problems by copying or downloading files, sending e-mail messages in response to events, updating data warehouses, cleaning and mining data, and managing SQL Server objects and data. [Knight, Veerman, Dickinson, Herbold, & Hinson, 2008]

Microsoft Windows Service Applications – formerly known as NT services, enable to create long running executable applications that run in their own Windows sessions. These services can be automatically started when the computer boots, can be paused and restarted, and do not show any user interface. These features make services ideal for use on a server or whenever you need long-running functionality that does not interfere with other users who are working on the same computer. [Microsoft, 2010]

Microsoft SQL Server – is a relational database management system developed by Microsoft.

Windows PowerShell – is Microsoft's task automation framework, consisting of a command-line shell and associated scripting language built on top of .NET Framework. PowerShell provides full access to COM and WMI, enabling administrators to perform administrative tasks on both local and remote Windows systems.

ASP.NET – is a server-side Web application framework designed for Web development to produce dynamic Web pages.

Microsoft SQL Server Analysis Services, SSAS, is an Online Analytical Processing, OLAP, data mining and reporting tool in Microsoft SQL Server. SSAS is used as a tool by organizations to analyze and make sense of information possibly spread out across multiple databases, or in disparate tables. [Wikipedia, s.d.].

2.2. Agile ETL essential characteristic

Agile ETL framework created to perform the extraction, transformation and integration data in Data Warehouse, through orchestration process that includes a model of indicators, processes for data aggregation, scheduling ETL processes and OLAP processing.

2.2.1. Extractor

The extractors are the objects responsible for load data from data sources, OLTP, XML files, Excel files, etc., to a Staging Area, it is the responsibility of each extractor to know where and how to put the information in the tables of Staging. At this stage, typically there will be transformation data, for performance issues with connectivity to the operation systems should be minimized as much as possible to avoid negative impacts that may exist in the systems.

Currently there are two types of extractors: SSIS package or SQL store procedures. The framework provides a generic component SSIS, that by changing the variable values and the connectivity of the source may be reusable in more than one packet extraction.

2.2.2. Integrator

Integrators have the task of load data from Staging area to Data Warehouse and transform the data in a consolidated manner to respond to business model.

The integrations can be of two types, SSIS packages or SQL procedures, framework takes more advantage of SQL procedures (there are major limitations in SSIS packages in code reuse), thus, were constructed generic procedures through dynamic SQL designed in two distinct types of integrations. Strategy delete records before inserting, this strategy typically used for metrics makes a massive insertion, erasing records that include in their partial extraction earlier, or if the extraction is complete, erases the entire table and insert new records.

There is another strategy for loading data that never deletes records in data warehouse, only inserts new records or changes data who has been register before, this load requires a verification key that controls through the primary key if the record is new or not, it is used mainly in the analysis dimensions or metrics that have changes throughout its existence.

2.2.3. Indicator Model

The model is an agile architecture indicators storage metrics that enables the system to incorporate new indicators without having to create new tables or new groups of metrics in OLAP. With distinct structures and resources in multiple data sources, transform the consolidation goes through all the information collected and centralized in a table, with a generic scheme, where all the information will serve a single group of metrics in OLAP.

2.2.4. Scope

Scope is a feature in Agile ETL, to restrict the scope of extractions, and segmental information not only by partial extractions, that feature also reduces the scope for extraction, but by the start date and end date of registration. The concept of Scope exists in framework to respond in a manner responsive to small changes in the extraction processes. Example of a case study, an extraction of alarms on a server where it is being recurrent events of type warning, since the excessive information can cause disinterest as demand becomes more difficult, it was helpful to filter the alarms temporarily draw, ideally only extract critical alarms.

With Scope can easily add items restriction in SQL queries without having to go change the SSIS package and still want to disable when no code change

2.2.5. Aggregation data feature

As storage in enterprise systems is costly and analysis of historical data often do not have great need of detail, the model of indicators has native functionality aggregation of indicators, such as providing more disk space the servers where Data Warehouse is housed.

The timing has two distinct options of aggregation, aggregation at the hour, that adds records regardless of the level of detail at the time, in other words, the analysis of a performance indicator "Occupancy Rate of CPU", the agent system is collecting information from minute to minute and a server specifically etl process saves this information in data warehouse, with this detail. With aggregation process and specific timing model indicators in selected period replaces the data with a level of detail to another baseline, by aggregating their averages, sums and number of records inserted at the time. With this aggregation that was lost the down level detail, with decreasing the number of records in the Data Warehouse. With aggregation, the maximum level of detail of information that may be irrelevant to many of the cases, because knowing the "CPU Load Factor" in a particular minute to five years ago does not bring a significant added value compared to the cost of storage.

Currently there are two types of aggregation, bundling the hour and day time. Aggregation groups to all information relating to the level of time regardless of the level of detail available, the day that after aggregation at the hour, puts the information in a lower level of granularity, adding all hours of the day and related metrics.

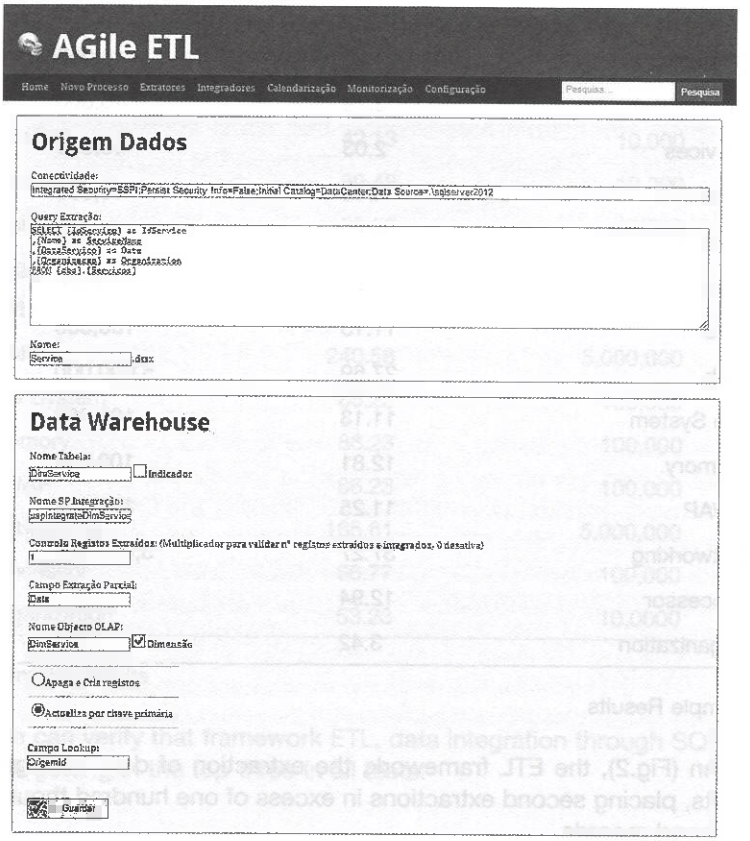
2.2.6. Logging

There is a centralized event table to add all the information and status of the Agile ETL framework. Log table should contain information on all processes like monitoring messages, warnings or errors in the application.

2.2.7. Web Site

Never losing the context of the framework, and always targeting the application for a specialized audience, as a tool to aid in the task of building and maintaining centralized repositories often called by Data Warehouse is available through a web interface that invocations PowerShell scripts creates the new ETL processes. In addition to creating new SSIS packages, new tables, views and procedures in SQL also

allows maintenance and configuration of extraction processes and integration. The timing, monitoring and process automation with portability, goals were achieved with this interface.



3. Performance

Knowing that the agility of construction or the maintainability of ETL processes have stronger visibility, it is necessary to demonstrate the gains also in execution performance. For that was held a performance analysis, for the integration and extraction tasks. We carried out a survey of a sample period between 2012/11/01 to 2012/12/01, and made up twenty-nine respective extractors and integrators, each with its timing.

The results were analyzed in comparison to other ETL technologies, based on a benchmark study [Infosphere, 2011]. ETL tools used in the performance comparison are:

- TOS – Talend Open Integration Solution;
- PDI – Pentaho Data Integration;
- DataStage – IBM InfoSphere DataStage;
- Informatica – Informatica PowerCenter.

3.1.1. Extraction

The extraction processes analyzed (Table 1) are all made by Agile ETL using SSIS packages. These packages are aimed at the transition of records from its origins to the staging area.

There are several distinct sources in the analysis, many servers that use the database in MS SQL Server 2005, MS SQL Server 2008, MS SQL Server 2008 R2, Oracle 10 and Oracle 11.

Extractor Name	Duration (seconds)	Number of Records
Backups	41.76	5,000,000
Configuration Items	8.97	100,000
Changes	4.29	10,000
Tickets	3.29	10,000
Services	2.03	10,000
Incidents	2.45	10,000
Workers	3.27	10,000
Log	3.06	10,000
CPU	11.13	100,000
Disk	27.69	5,000,000
File System	11.13	100,000
Memory	12.81	100,000
SWAP	11.25	100,000
Networking	37.27	5,000,000
Processor	12.94	100,000
Organization	3.42	10,0000

Table 1 – Extractions Sample Results

As can be seen (Fig.2), the ETL framework, the extraction of data through SSIS packages had quite satisfactory results, placing second extractions in excess of one hundred thousand records and third extractions in ten thousand records.

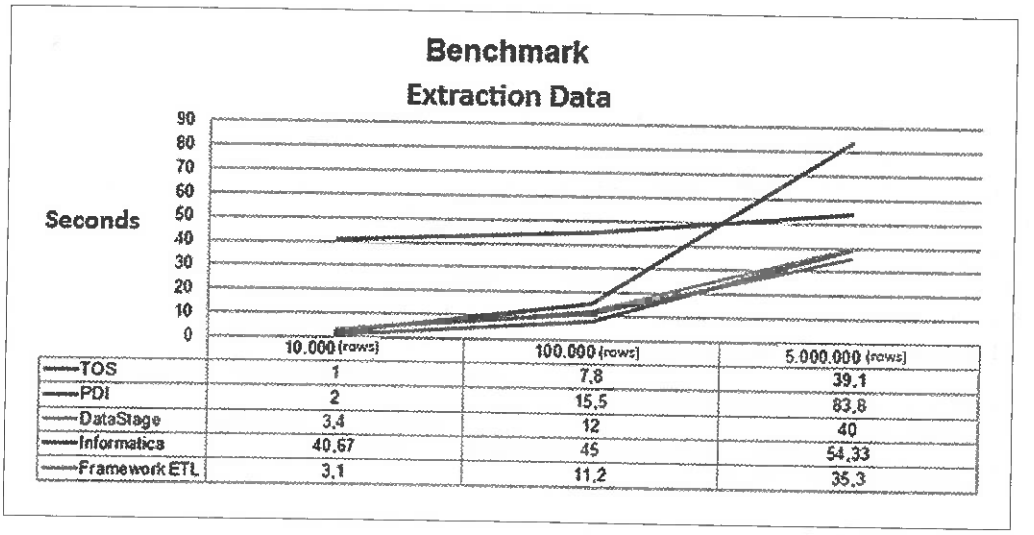


Figure 3 – Benchmark Extraction Data.

3.1.2. Integration

The integrations analyzed (Table 2) are all made by ETL framework, using procedures in SQL. These procedures aim to consolidate the information in the data warehouse, making changes to data stored in Staging area, aimed at responding to the schema designed and engineered in the Data Warehouse. Integration processes was made an adjustment performance by creating indexes, partitioning tables, ensuring better results in the transformation of data (Fig. 3).

Integration Name	Duration (seconds)	Number of Records
Backups	119.98	5,000,000
Configuration Items	63.9	100,000
Changes	46.13	10,000
Ticket	42.03	10,000
Service	42.13	10,000
Incidents	30.48	10,000
Planning	85.52	10,000
Log	61.23	10,000
CPU	63.03	100,000
Disk	240.58	5,000,000
File System	86.23	100,000
Memory	86.23	100,000
SWAP	86.23	100,000
Networking	165.61	5,000,000
Processor	66.77	100,000
Organization	53.23	10,0000

Table 2 – Integrations Sample Results

In Figure 3, we can verify that framework ETL, data integration through SQL procedures, had very positive results, always getting in the top three in all tests.

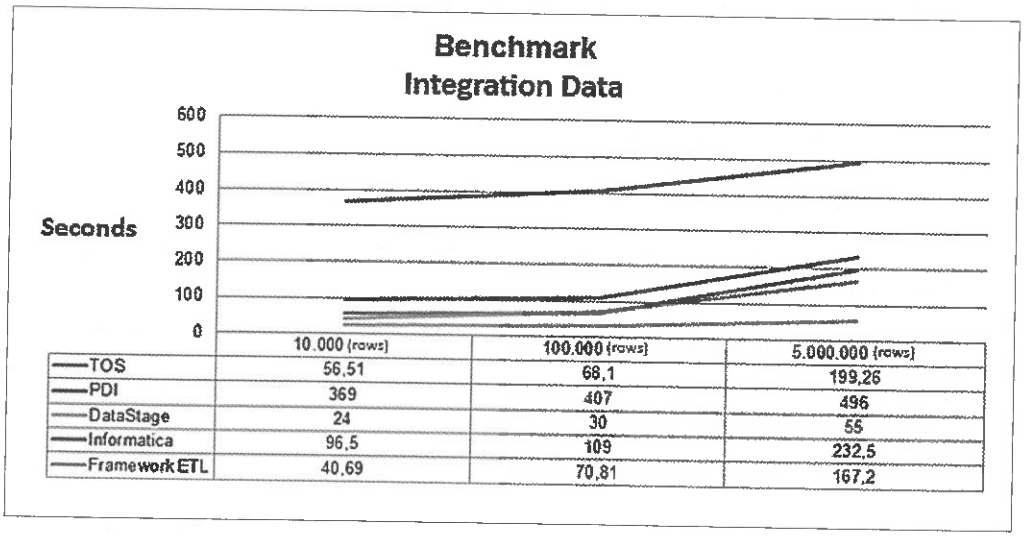


Figure 4 – Benchmark Integration Data

4. Conclusion

Building a framework that provides up functions with all creation mechanisms for extracting and integrating a Data Warehouse, was important; however, wished to reach a higher level, providing an interface layer that only through a wizard, you can create new ETL processes.

The main objective of the study was to get to propose a solution to streamline the process of loading data into a central repository, Data Warehouse. For such a prototype was created, implemented with

several distinct components, such as SQL tables, a Windows service, in PowerShell functions for process automation, web interface to respond more nimbly to change orders and monitoring of extractions or integrations and data packets SSIS.

The prototype implemented centralizes the creation and automation of ETL processes, enabling the construction of central repositories of data to companies with less financial capacity.

In the implemented prototype it was possible that some adjustments sensing further facilitate migration processes information. The idea would not only make the migration table to table almost from the origin to the Data Warehouse, but designing a new component like the "Data Source View" SQL Server Analyses Services, which defines all the desired structure through "drag and drop" the import of various objects, from different sources, and the model would be automatically adjusted in response to schema designed for the Data Warehouse.

Another future goal is to integrate the framework with a mobile interface that allows monitoring and create new components via a smartphone, tablet be it IOS, Android or Windows Phone.

In the long term there is the intention to add a new architecture framework for ETL, event-driven architecture uses a subscription system event, which unlike the current system, there is an architectural request and response between servers, whether of origin, Staging and Data Warehouse. The event-driven architecture, works through subscription of events, allowing the subscription, performing actions in response to events created. The one feature valid for the framework would be to remove schedule task in the extraction processes, making only subscriptions on the servers of operation systems, on the entries of new records or changes to the data, would be automatically routed to the Staging area and then processed and entered into the Data Warehouse.

5. Referências

Eckerson, W., "Smart Companies in the 21st Century", *Seattle: The Data Warehousing Institute*, 2010.

Infosphere, *ETL Benchmark Favours Datastage and Talend. ETL Benchmark*: http://it.com/blogs/infosphere/etl-benchmark--favours-datastage-and-talend-28695, 2011

Holmes, L., "Windows PowerShell Cookbook" *Sebastopol: O'Reilly Media*, 2010.

Kimball, R. e Caserta, J., "The Data Warehouse ETL Toolkit.", *Indianapolis: Wiley Publishing, Inc.,* 2004.

Microsoft., *Introduction to Windows Service Applications,* http://msdn.microsoft.com/en-us/library/d56de412(v=vs.80).aspx, (10 de Outubro de 2010), 2010.

Minelli, M., Chambers, M., e Dhiraj, A., "Big Data, Big Analytics", *New Jersey: John Wiley & Sons, Inc.*, 2012.

Ndlovu, S. W., *Programmatically Create Data Flow Task in SSIS Package Using C#,* http://www.selectsifiso.net/?p=288, (02 de Junho de 2011), 2011

T. Moss, L., e Atre, "S. Business Intelligence Roadmap", *Boston: Addison-Wesley*, 2012.

Sojo, E., *Creando paquetes de SSIS con .NET.,* http://eduardosojo.com/2012/01/03/creando-paquetes-ssis-con-net--creando-data-flow-task-y-elementos-internos/ (03 de Janeiro de 2012), 2011.

Thomsen, E., Spofford, G., e Chase, D., "Microsoft OLAP Solutions", *Indianapolis: Wiley Publishing, Inc.*, 1999.

Turban, E., Sharda, R., Delen, D., e King, D., "Business Intelligence", *London: Prentice Hall*. 2010.

Webb, C., Ferrari, A., e Russo, M. ,"Expert Cube Development with Microsoft SQL Server 2008 Analysis Services", *Birmingham: Packt Publishing Ltd,* 2009.

Wikipedia. (s.d.). Wikipedia Free Encyclopedia, http://en.wikipedia.org/wiki/Main_Page.