International Journal of Informatics and Applied Mathematics

International Journal of Informatics and Applied Mathematics
e-ISSN:2667-6990 Vol. 5, No. 2, 48-60

Integration of Business Process Data using Advanced ETL tools

Ali Khebizi and Mourad Hadjeris

Department of Computer Science, LabSTIC Laboratory, 8 May 1945 University, P.O. Box 401, 24000 Guelma, Algeria
ali.khebizi@gmail.com, m-hadjeris@yahoo.com

Abstract. The large proliferation of the **BPM** discipline has considerably propelled companies' information systems which become ubiquitous. Consequently, a huge volume of data is generated during **BP** execution and stored in adequate IT systems. However, such data are often heterogeneous and the problem of their integration is posed with acuity in order to allow data analysis and to coordinate companies' activities. In fact, different techniques and approaches have been suggested in the research literature to tackle such issue. In the business intelligence specific context, this challenge is addressed by deploying the conventional **ETL** software tools allowing to build an unique container supporing various data and which is perceived as a data warehouse. Nevertheless, current distributed information systems are spanning enterprises boundaries and deal with a large variety of data sources having various formats and produced in a continuous manner. Thus, actual ETL tools seems unsuitable to handle massive data and they remain limited for facing to issues imposed by related to business processes execution data stored in log files. In this paper we propose an enhancement of the structure and functionalities of standard ETL tools in order to handle heterogeneous data integration generated by business processes execution. The improved system, named **OLE-ST**, constitutes a fundamental enrichment of the existing ETL mechanism. The proposed approach has been implemented in a software tool that ensures best performances for exploiting the target data warehouse to be built.

Keywords: Business Processes · Business Process Management · Data Warehouse · ETL · Data Integration · Big Data

1 Introduction

The survival of today's businesses is closely tied to the efficient handling of business processes (**BP**) that are increasingly interconnected, dynamic and subject to very versatile environmental factors. From this perspective, several concepts, approaches and techniques have been proposed in the two last decades to meet the growing needs of organizations for managing business processes. We distinguish mainly the **B**usiness **P**rocess **M**anagement (**BPM**) technology which is a global managerial vision that puts business processes at the center of all organizations' activities and which offers different phases for easily managing BPs life-cycle. (*e.g., development, deployment, execution, management and monitoring*). Thus, nowadays **I**nformation **S**ystems (**IS**) are centered around BP and the concept of **P**rocess **A**ware **I**nformation **S**ystems (**PAIS**) [1] has recently emerged as a key concept to cope with challenges related to the integration of distributed and heterogeneous IS, for handling both data and business rules reflecting enterprises business goals.

A business process (***BP** for short*) consists of a set of activities undertaken in coordination by one ore more organizations, in an organizational and technical environment, to achieve a particular business goal [2]. Hence, the BP explicit representation must specify rigorously how the business logic is performed and how it should be conducted under real organizational constraints. In parallel, **BPM** is a comprehensive discipline that promote a process-centred approach to align organizations business processes with clients needs and to continuously improve business effectiveness and efficiency [2]. Recently, this technology is recognized as a key factor underpinning competitiveness and growth of modern companies. The great awareness of the importance of the BPM technology by the different stakeholders has lead to its large proliferation in socio-economic environments and its adoption by a huge number of moderns companies. As an immediate consequence, a large size of data is generated during BPs execution and is manipulated and maintained in current **B**usiness **P**rocess **M**anagement **S**ystems (BPMS).

While great advances have been achieved in the BP management field (*design, configuration, encactment and analysis*), however it has been observed that for over a long period of time, business processes practices have relegated data to a secondary citizens class [3]. In fact, the control flow (*e.g.; ordering of messages*) was the dominant aspect in BP models, while data are still "hidden" inside IT systems. This way of perceiving BPs has considerably reduced their importance and effectiveness. Thereby, data management is essential in BPM environments, on the one hand for allowing BPs description and on the other one to ensure instances' execution and resources monitoring. Hence, managing data requires more attention both from business and IT managers, because it constitutes a precious capital for each company.

To support data storage and access, techniques and technologies are more than necessary. Such target environments must handle both the persistent data in the underlying enterprise database(s) and the relationships of the managed processes with their abstract models. In this perspective, the BPs discipline has largely benefited from advantages of data and knowledge management advances, such as databases (*relational, NoSQL*) and integration tools (*ETL systems*)[4–8], data analysis (*OLAP systems and data mining techniques, ...*)[9] as well as data warehouses (**DW**) technologies [8, 10–13].

In the last decade, ETL tools have emerged as a promising technology for integrating interconnected, heterogeneous, scalable and flexible data originating from various sources. In fact, today the market for data integration tools includes vendors that offer software products to enable the construction and implementation of data access and data delivery infrastructure for a variety of data integration scenarios, such CRM, accounting and human resources management. Thus, these environments have brought significant progress to various fields of data science and knowledge management, such as business intelligence field, data analytics and data mining. Furthermore, it's observed that the large proliferation of automated business processes facilitates enterprises' collaboration and supports inter-organizational businesses that are spanning across enterprises boundaries. Despite the progress made by ETL tools, they remain limited in handling massive data originating from distributed systems characterized by a high degree of variation and a rapid evolution.

In this paper we tackle the issue of heterogeneous big data integration and we suggest an improvement of the basic existing ETL tool by extending its basic functionalities. The aim is to enrich the structure and functionalities of conventional ETL tools with advanced features allowing to extract and to load heterogeneous big data related to BP executions. The proposed approach has been implemented in a software tool that ensures best performances of the target DW to be built.

The remainder of the paper is organized as follows. In section 2 we introduce basic concepts and notions useful to make the paper self-contained by presenting BP data and ETL tools. In section 3 we present the improved ETL system and we discuss its architecture and functionalities. Section 4 is dedicated to the presentation of developed ETl tool and we conclude in section conc by summarizing our results and planning future works.

2 Background

In this section, we give the necessary material useful for understanding the remainder of the paper and that makes it self-contained. We start by presenting data properties related to BP execution and their categories, then we expose ETL technology and its underlying mechanism. Finally, basing on the nature of BP data and the drawbacks of current ETL tools, we discuss the problem statement and we motivate our work at the end of the section.

2.1 Big Data in BP context

The various issues behind the general question of data integration have been intensively investigated in different fields of the computer science discipline, such as databases integration [14, 15], software ingeneering [16–18], DWs and information systems. In fact, the recent literature is very rich in techniques and approaches that attempt to address the challenges inherent to heterogeneous data management. In what follows, we focus only on BP data.

The spectacular explosion of business applications supported by modern **I**nformation and **C**ommunication **T**echnologies (ICT), combined with the large proliferation of the Business Process Management (BPM) software suites, have profoundly changed and facilitated the way in which modern companies' information systems are designed and deployed. Consequently, massive data, known as big data, are continuously generated and stored in the underlying IT systems as event-logs. BP data constitute a valuable resource to be exploited in a business intelligence perspective. Hence, data management is essential in BPM environments, on the one hand to allow BPs description and on the other hand to ensure instances' execution. In a concrete scenario, a deployed BP of an organization may be invoked by a huge number of customers or stakeholders. Each execution of the BP generates a set of data reflecting the instance (or case) progression and the assigned resources. Such execution data are seen by organizations as an increasingly important capability that can complement the traditional data sources. Hence, information captured by IT systems, during business processes' execution, produces a value-added mine which requires a capital importance. These process data is especially relevant in the context of automated business processes, process controlling, and representation of organizations' core assets [19]. According to [3], four classes of persistent BP data are distinguished in the BP context.

– BP models data (*specification of the process schema; i.e, steps and activities*).
– Business data needed for achieving a BP instance (*e.g. a customer's delivery address*)
– Execution data (*instances states and history execution; i.e, execution traces*)
– Resource usage and their availability to ensure BP progression (*e.g. the available quantity of a product to satisfy the customer's order*)

During the execution of business processes', the generated data are stored in adequate databases. While, various data formats are supported by BPMS, the majority of existing BPM tools uses relational databases to store data.

Business processes data are characterized by the following intrinsic properties.

1. The volume of generated data is considerable, even massive.
2. Handled data are strongly connected and are related to various companies which may be working in coordination to achieve common goals.
3. They require flexible models to meet unpredictable requirements occurring in the environment.
4. Changes in law, regulation and policies affect BP specifications and require data adaptation to the evolution context.

5. For performance requirements, multiple join queries are deployed to extract useful data of interest.
6. The managed data are scalable.
7. Execution data are heterogeneous and need specific transformations to ensure their interoperability.

The previous BP data properties hinder their efficient exploitation and require specific software tools to enable their integration into an unified repository (*i.e.; database or DW*). Furthermore, such data satisfy the 3V fundamental characteristics of big data (Volume, Variety and Velocity) [13]. Hence, dealing with such specific data nature requires advanced technologies and techniques.

2.2 ETL Tools

Data integration refers to the activities used to combine data from multiple sources to provide an unified view of the data. It aims to consolidate data from disparate sources into a single data-set with the ultimate goal of providing users with consistent access and delivery of data across the spectrum of subjects and structure types, and to meet the information needs of all applications and business processes.

During its spectacular historical evolution, the data integration ecosystem has benefited from many advances materialized by various integration techniques and technologies. This phenomenon concerns both structured data from databases or DWs, as well as semi-structured or even unstructured data. In the context of relational databases, the Global As View (GAV) and Local As View (LAV) integration approaches have been widely used to conduct to a single schema. Similarly, the Enterprise Application Integration and Enterprise Serial Bus (ESB), as well as Enterprise Resources Planning (ERP) methods have been intensively used and were a subject of a great deal of work and have been implemented in operational software tools allowing to bring together different heterogeneous data in uniform formats.

In the field of DWs, Extract, Transform and Load (ETL) tools remain essential. Definition **1** bellow shows the basic ETL description.

Definition 1. *An ETL tool is a middle-ware for collecting and synchronizing data between different systems. It extracts the data, manipulates them (conversion, deletion of duplicates tuples, format transformation, ...) and integrates them into a common repository which is the data warehouse (**DW**) [20].*

According to this definition ETL processes are responsible for the operations taking place in the background of a DW architecture. As shown in the figure 1, an ETL tool ensures the junction between data sources and the target data wharehouse, by identifying the pertinent data sources, loading the concerned data and transforming them in a suitable format to be loaded in the target DW.

Conducting data integration using an ETL tool is an incremental process articulated around progressive steps for leading to an integrated data repository. Figure 2 bellows depicts the various steps underlying ETL processes.

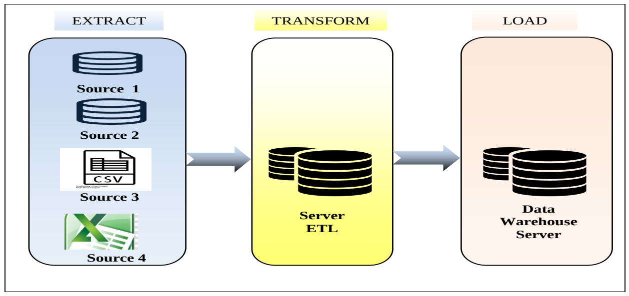

Fig. 1. Architecture and functionalities of basic ETL tools

- **a. Data Extraction**
  - Sources identification: As illustrated in the figure 2 (*sub-steps 1,2,3*), the ETL process starts by identifying the data sources that may be potentially be used to feed the target data warehouse. Thus, during this step metrics and attributes of the dimension tables are enumerated, then the matching rules between source and target attributes are elaborated. If different data sources are founded, the most relevant one is selected. Furthermore, if for a particular target attribute, various sources are eligible, its opportune to formalize the adequate consolidation rules. In the contrary, if an attribute source corresponds to more than a target attribute, in this case cutting rules must be specified.
  - Data extraction: Once data sources are identified, the extraction step is triggered. This action can be activated in two distinguished manners according to the context of the generation of the DW. If the last one is generated for the first time, the extraction is complete, otherwise it's conducted in an incremental fashion.

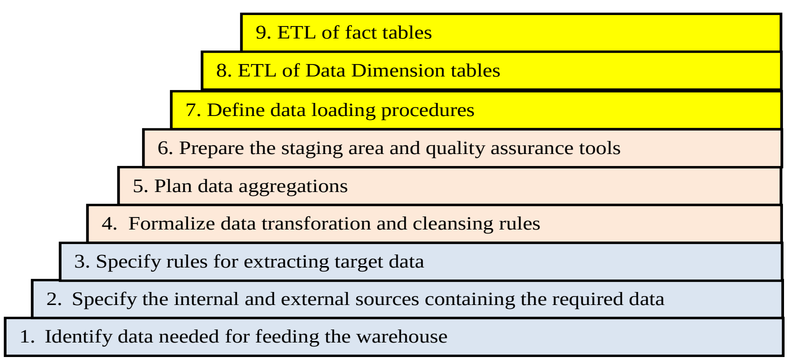

Fig. 2. Steps of an ETL tool

- **b. Data Transformation** (*see sub-steps 4,5,6 in the figure*). During this step all heterogeneous conflicts related to different data are handled and suitable rules allowing to convert data sources to corresponding target ones are specified without ambiguities. Hence, attributes mismatches, such as data format, the used encoding and the character sets, as well as measure units disparities are resolved. Further, pre-calculation of derived values and aggregated attributes are performed.
- **c. Loading data in the DW**(*see sub-steps 7,8,9 in the figure*). After having extracted and transformed data useful for the DW, the last step of an ETL tool consists to load the converted data in the target repository. To achieve this goal, three loading possibility are offered. The first one is the initial loading that is triggered when creating the DW. The second one is incremental and it handles the nature of changed values in the data sources. Thus, it allows integrating only data having be modified before the last loading operation. In the case where the incremental loading type is a resources consumer activity, (especially if the rate of data changes is consistent) a complete loading is preferable.

2.3 Problem statement

With the recent technological developments and the democratization of ICT, hardware performances have spectacularly evolved. Consequently, the internal and external data of organizations are becoming increasingly diverse, instantaneous and voluminous. As an immediate consequence, the conventional ETL technology seems insufficient to face to data integration requirements induced by new contexts. Indeed, they were not designed to handle the flow of remote data originating from the cloud. The problem of data flow management is particularly acute in real-time environments. In fact, in operational economic environments, modern enterprise software applications can't wait for hours or days to manage updated data sets and they must react to occurring new data instantly. Thus, contemporary organizations create and process data in a continuous stream in real time. The data characteristics of such management environments are the following.

- They have an ephemeral nature due to their versatility and rapid changing.
- They are originating from various types of sources including mobile users (*nomadic users*).
- They are very large in size and require dedicated supports for their storage and processing.

In this new context, traditional ETL tools can't cope with data scalability due to the large amount of data emanating from real-time environments and which interrupts the steps of ETL processes. Even worse, in some scenarios, data flow overruns the pipeline ETL. Thus, after the extraction phase the transformation procedure can be overwhelmed, causing bottlenecks induced by the mass of data extracted in the transit zone (*staging area*). The same phenomenon can occur between the two phases of transformation and loading. This phenomena is known in the literature as the **ETL pipeline overflow**. Further, when dealing with massive and instantaneous data data, the ETL mechanism takes time and consumes valuables resources to transform the data extracted from sources and which are saved before they become in turn obsolete.

To overcome the previous drawbacks, we propose in the next section an enhancement of the conventional ETL tools.

3 OLE-STL: an Enhanced ETL tool

Data flows originating from the Web are characterized by a significant degree of changes and handling their integration is a hard task for which we propose an improvement of the basic structure of conventional ETL tools. By doing that we operate a fundamental overhaul of its functioning in order to ensure an efficient data integration, while facing to performance issues.

This section is dedicated to presenting the different aspects of our approach intended to be used by data managers for integrating business processes data. It's structured as follows.

(i) We start by giving the fundamental properties of the novel ETL tool. *(i)* Then we expose the architecture of the conceived ETL tool and we describe the interaction between its components. *(iii)* Finally, we discuss the functionalities of the conceived solution.

In what follows, these steps are deeply discussed and illustrated.

3.1 Fundamental features of the solution

The conceived enhanced ETL tool, named **O**n-Line **E**xtract **S**elective **T**ransform and **L**oad (**OLE-STL**), is a framework which improves ETL systems. It extends their basic mechanism by refining the two first steps (*extraction and transformation*). Roughly speaking, our contribution consists in the three following aspects.

- A selective transformation technique that considers only the category(ies) of relevant transformations chosen by the data manager, instead of triggering a set of resources consuming actions and useless transformations.
- During the extraction step, an additive extraction possibility is proposed to be used for the specific context of DW schema evolution.
- A new technique for checking updates in the data sources which is based on comparing update dates of source files with the last upload dates. To achieve this goal, a table-type is deployed as a data structure to track continuous changes occurring in data sources.

3.2 Architecture of the enhanced ETL tool

As depicted in **figure 3**, the system OLE-STL performs the three basic functions (*Extract, transform and Load*) of conventional ETL tools. Furthermore, the two first steps (*Extract and Load*) are enriched with advanced functions to ensure integration of massive and evolving data flow.

The figure shows that the extraction step is more comprehensive and more consistent, in the sense that it allows three distinguished and complementary activities. First, it ensures the initial data extraction from data sources (*transition number 1 in the figure*). Further, it offers the possibility to extract additional data from new sources or files that haven't been considered during the previous extraction pass (*transition 1' in the figure*). Finally, in the case where some data sources have been modified since the last extraction process, the third loading option, i.e.; incremental extraction allows checking data sources and extracting updated information basing on their updates dates stored as properties in the system files repository (*transition 1" in the figure*).

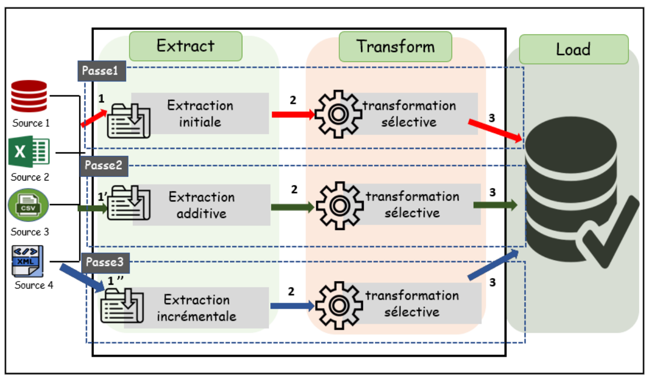

Fig. 3. Architecture of the system OLE-STL

The transformation step constitutes the core of the system OLE-STL (*transition 2 in the figure*) . In fact, regardless of the extraction type that was carried out previously, this step aims to convert heterogeneous data in an normalized format. The novelty incorporated in the OLE-STL system for the transformation phase consists in its customization according to the specific context of each user. Indeed, data transformation rules have been categorized into the following four distinct categories.

- Conversion rules: applied to data mismatches such as formats and lengths of various data.
- Coding rules: concern unification of codes and measurement units and transcription of values into codes.
- Granularity rules: focus on the structure of attributes by operating cutting/grouping actions for certain fields to make them coherent with the rest of data already stored in the DW.
- Pre-calculation rules: this rules sets concern the pre-calculations of derived values and other aggregated attributes that will be stored in the warehouse.

Such pre-calculations rules will allow a considerable improvement of performance when accessing and processing data in the DW.

In our approach, the data manager will be able to selectively apply one or more rules during the transformation process instead of triggering them holly. This latter approach will degrade considerably the overall performance of the ETL system. It's forth noting that all the transformations rules are triggered in a temporary work-space known as the *staging area*, so as not to alter the operation of the DW.

Once the transformation step is achieved, converted data are loaded into the DW (*transition 3 in the figure*). It's worth noting that three main distinguished fashions can be conducted for loading the transformed data into the DW; i.e.;initial, incremental and complete (*see [5] for more details*).

3.3 Functionalities of the system OLE-STL

After having exposed the main features of the system OLE-STL and its architecture, bellow the chronological functions to be activated for building the DW are described.

1. An initial **extraction** is operated on the different data sources. It allows feeding the target DW from scratch.
2. In our advanced system, **the transformation** step can be customized basing on the identified transformation categories: size based transformations, granularity transformations, coding and pre-calculation transformations. Thus, the user will have the option of selectively applying one or more rules during the transformation process instead of triggering all the various rules sequentially, as described in sub-section 2.2 (b).
3. Once data transformation rules have been identified and selected by the data manager, the **Transform** process is activated in order to convert original data to target ones that are adequate with the structure of the DW.
4. If the structure of either the target DW or the data sources have evolved (*i.e.; their schema has changed*) in order to face to laws and regulations as well as technological changes, then they require a complementary integration step. In this particular context, OLE-STL offers an additive extraction option that allows handling the concerned data source(s) for synchronizing data. Obviously, the transformation and loading of the evolved sources will be carried out accordingly.
5. Instead of focusing on data schema level, another interesting function of the conceived system consists to check variations and modifications at the data occurrences and values level. To this end the system allows identifying data sources having changed since the last data loading in the DW. This mechanism is based on the comparison of the descriptive attribute file date update which is a system property of data sources.
6. Finally, loading the new transformed data in the DW is performed to feel the DW with the converted data. As mentioned previously, this procedure depends on the usage and evolution context of the DW and it can be conducted in three distinct manners (*initial, incremental, complete*).

4 Implementation and Experiments

To illustrate the feasibility and the concrete exploitation of the proposed solution in a real world scenarios, this section briefly describes a prototype of the system OLE-STL which implements the proposed approach. The prototype has been developed with the language Python [1] [21] and we used complementary tools, such as PyCharm [2] and DB Browser(Sqlite) [3] for advanced programming needs and data sources access.

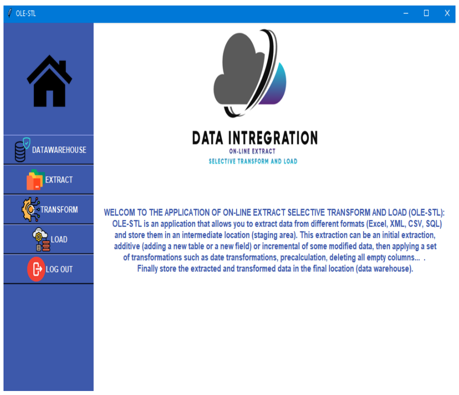

Fig. 4. OLE-STL main interface

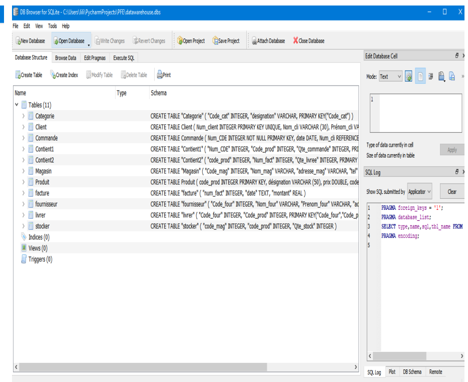

Fig. 5. Specification of the DW structure

As illustrated in **Fig. 4**, the developed prototype **OLE-STL** allows specifying the DW structure in an interactive manner. Furthermore, the basic functions of ETL tools (*Extract, Transform and Load*) are available for data management in the main interface. The screen-shot of **Fig. 5** illustrates the interactions during DW specification with the environment DB Browser.

Once the DW structure is designed and the data sources are identified, the data manager can select the desired extraction option. As shows in the screen-shot of **Fig. 6**, three extraction options, respectively, initial, incremental and additive are allowed by the developed system. Transforming the previously data sources into adequate format to be load in the target DW is an incremental process which is ensured by the system OLE-STL. Hence, a set of transformations rules can be activated basing on the data manager requirements and the historical loading, as well as the evolution context of the hole system. The implemented prototype was enhanced with an option to resolve manually data type mismatches that can occur between sources and target structures (see **Fig. 7**). In this case, according to the particularity of target data types and their

[1] https://python.org
[2] https://www.jetbrains.com/fr-fr/pycharm/
[3] https://sqlitebrowser.org/

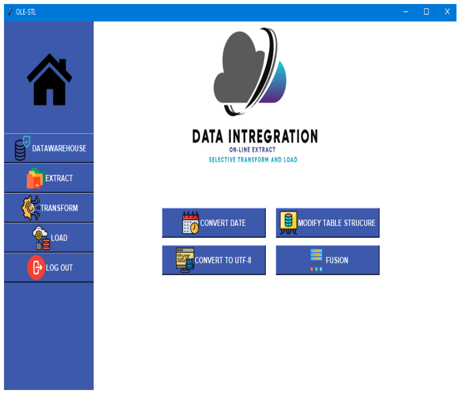

Fig. 6. Applying transformations rules

Fig. 7. Resolving data type mismatches

corresponding target ones, conversion rules are specified manually by the data manager.

5 Conclusion and Future Works

Research in the data integration area has mostly focused on ensuring safe continuation of enterprises applications, while little work has been done in the context of business processes data integration.

In this work we proposed an enhancement of the conventional ETL tools to face to massive data originating from the web and expressing BP execution. The proposed system OLE-STL allows extracting data source instantaneously, then it transforms them basing on a selective conversion rules allowing to resolve heterogeneity between data sources and the target DW structure. The implemented software constitutes an efficient tool useful for both data managers and data analytics, because it allows constructing data warehouses serving as a backbone for decision making systems.

Our future works will be devoted to extend the proposed formal framework to investigate the issue of data storage by considering the impact of DW technologies on the proposed transformation technique. Furthermore, we project to explore graph database paradigm for managing DWs. Finally, we plan to consider large-scale data-sets and the impact of new distributed and parallel computing paradigms (*e.g., MapReduce, Spark*) in our context study.

References

1. Wil M. P. van der Aalst. *Process-Aware Information Systems: Lessons to Be Learned from Process Mining*, pages 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
2. Mathias Weske. *Business Process Management - Concepts, Languages, Architectures, 2nd Edition.* 2012.

3. Richard Hull, Jianwen Su, and Roman Vaculin. Data management perspectives on business process management: Tutorial overview. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 943–948, New York, NY, USA, 2013. ACM.
4. Thomas Jörg and Stefan Deßloch. Towards generating etl processes for incremental loading. volume 299, pages 101–110, 01 2008.
5. Neepa Biswas, Anamitra Sarkar, and Dr-Kartick Mondal. Efficient incremental loading in etl processing for real-time data integration. *Innovations in Systems and Software Engineering*, 16, 03 2020.
6. Sreemathy J, Infant Joseph V, Nisha. S, Chaaru Prabha I, and Gokula Rm. Data integration in etl using talend. *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 1444–1448, 2020.
7. J Sreemathy, R Brindha, M Selva Nagalakshmi, N Suvekha, N Karthick Ragul, and M Praveennandha. Overview of etl tools and talend-data integration. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 1650–1654, 2021.
8. Agnas Michael and Purnima Ahirao. Improved use of etl tool for updation and creation of data warehouse from different rdbms. *SSRN Electronic Journal*, 01 2020.
9. Gergely Pintér, Henrique Madeira, Marco Vieira, István Majzik, and András Pataricza. Integration of olap and data mining for analysis of results from dependability evaluation experiments. *Int. J. Knowledge Management Studies*, 2, 01 2018.
10. Il-Yeol Song. *Data Warehousing Systems: Foundations and Architectures*, pages 684–692. Springer US, Boston, MA, 2009.
11. Leo Willyanto Santoso and Yulia. Data warehouse with big data technology for higher education. *Procedia Computer Science*, 124:93–99, 2017. 4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia.
12. Oras Baker and Chuong Nguyen Thien. A new approach to use big data tools to substitute unstructured data warehouse. In *2020 IEEE Conference on Big Data and Analytics (ICBDA)*, pages 26–31, 2020.
13. Syed Muhammad Fawad Ali. Next-generation etl framework to address the challenges posed by big data. In *DOLAP*, 2018.
14. M. Ahmed-Nacer and J. Estublier. Schema evolution in software engineering databases - a new approach in adele environment. *Computers and Artificial Intelligence*, 19(2):183–203, 2000.
15. J.Andany, Leonard M., and Palisser C. Management of schema evolution in databases. In *17th*, (VLDB), Spain, 1991.
16. Alexander Stuckenholz. Component evolution and versioning state of the art. *SIGSOFT Softw. Eng. Notes*, 30(1), January 2005.
17. Andreas RAUSCH. Software evolution in componentware using requirements/assurances contracts. ICSE '00, pages 147–156, USA, 2000.
18. Bennett P. Lientz and E. Burton Swanson. *Software Maintenance Management*. Boston, MA, USA, 1980.
19. Andreas Meyer, Sergey Smirnov, and Mathias Weske. Data in business processes. *EMISA Forum*, 31:5–31, 2011.
20. Panos Vassiliadis and Alkis Simitsis. *Extraction, Transformation, and Loading*, pages 1095–1101. Springer US, Boston, MA, 2009.
21. William F. Holmgren, Robert W. Andrews, Antonio T. Lorenzo, and Joshua S. Stein. Pvlib python 2015. In *2015 IEEE 42nd Photovoltaic Specialist Conference (PVSC)*, pages 1–5, 2015.