Two-ETL Phases for Data Warehouse Creation: Design and Implementation

Ahlem Nabli[1], Senda Bouaziz[1], Rania Yangui[2](✉), and Faiez Gargouri[2]

[1] MIRACL Laboratory, Faculty of Sciences, Sfax University, 1171 Sfax, Tunisia
ahlem.nabli@fsegs.rnu.tn, bouaziz.senda@hotmail.fr
[2] MIRACL Laboratory, Institute of Computer Science and Multimedia,
Sfax University, 1030 Sfax, Tunisia
yangui.rania@gmail.com, faiez.gargouri@isimsf.rnu.tn

Abstract. Building the ETL process is potentially one of the biggest tasks of building a warehouse. In fact, it is complex, time consuming, and consumes most of data warehouse projects implementation efforts, costs, and resources. Nevertheless, the difference on data structures imposes new requirements on the ETL process implementation and maintenance. What makes these tasks even more challenging is the fact that data continue to grow rapidly and business requirements change over time. In this paper, we propose a method that contains Two-ETL phases, one treats the pre-treatment phase and another deals with the actual ETL. Our method consists on determining the correspondence table, modeling new operations using the Business Process Modeling Notation (BPMN) and implementing these operations with Talend Open Source (TOS). In addition, our method allows the design of ETL process in an earlier stage, which enormously facilitates the implementation of this process. Another advantage of our proposal is the use of the BPMN which allows to cover a deficit of communication that often occurs between the design and implementation of business processes.

Keywords: Extract transform and load · Business process modeling notation · Data warehouse design · Transformation operations · Correspondence table

1 Introduction

Business Intelligence (BI) solutions are very important as they require the implementation and the design of complex ETL process. This process is a software which allows the alimentation of a DW and its periodic refreshment from different sources. It is often used to get back various information to feed regularly the DW. New applications, such as, real-time data warehousing, require agile and flexible tools that allow BI users to take suitable decisions based on extremely up-to-date data. This is the case of the BWEC[1] (Business for Women of Emerging Country)

[1] Towards a new Manner to use Affordable Technologies and Social Networks to Improve Business for Women in Emerging Countries.

T. Morzy et al. (Eds.): ADBIS 2015, LNCS 9282, pp. 138–150, 2015.
DOI: 10.1007/978-3-319-23135-8_10

project that aims at improving the social economic situation of Handicraft women in Algerian and Tunisian countries involved in this research works.

To feed the DW, data must be identified and extracted from their original locations. Consequently, the data must be transformed and verified before being loaded into the DW. The large amount of data from multiple sources causes a high probability of errors and anomalies. This increases the need of a new ETL tool which are able to be adapted with the constant changes, to produce and to modify executable code quickly.

Recall that in the literature [1,2], the main stages for the DW design methodologies can be summarized as follows: requirement analysis, conceptual design, logical design, ETL process design and physical design. In fact, it was recognised that ETL process is a very time-consuming step, it takes about 80 % of the total time of the decision-making implementation due to its difficulty and complexity [3]. The design and the implementation of an ETL process usually involve the development of very complex tasks imposing high levels of interaction with a vast majority of the components of a DW system architecture. The implementation and the maintenance of such processes face various design drawbacks, such as the change of business requirements, which consequently leads to adapt existing data structures and reuse existing parts of ETL system.

Several works [5,9,11,12] have dealt with ETL process modeling and they don't focus on incorporating pre-processing phase of ETL process since the conceptual modeling phase of the DW. Furthermore, it has been noticed that while trying to design the ETL process, people tend to overlook the work done in the conceptual phases and which contain a useful knowledge for the ETL process. In this paper, we propose a method called two-ETL phases for DW creation where the first phase is carried out since the conceptual design of DW. This phase handles the determination of the correspondence table and the modelling of the transformation operations. The second phase deals with the implementation of the ETL process.

The remainder of this paper is organized as follows. Section 2 reviews some related works concerning the ETL modeling process. Section 3 describes our proposed method to create a DW. Section 4 details the first ETL phase. Section 5 presents the second ETL phase. Finally, Sect. 6 gives a conclusion and some future research directions.

2 Related Works

Various approaches for designing and optimizing ETL process have been proposed in the last few years [5,9,11,12]. This approaches can be classified into three main groups. The first group uses UML (Unified Modeling Language) to model the ETL process. The second one uses MDA (Model Driven Architecture). The third group uses BPMN (Business Process Modeling Notation).

UML Based ETL Process Modeling: Trujillo and Luján-Mora [4] have proposed an extension of the UML language to model the ETL process. Also, Mallek et al. [5] proposed the use of the UML activity diagram for the modeling of the

ETL process named ETL-WEB. More recently, El-Sappagh et al. [6] proposed an entity-mapping diagram (EMD) framework, consisting in a new notation and a new set of constructs for ETL conceptual modelling.

MDA Based ETL Process Modeling: Munoz et al. [7,8] presented the modeling of the ETL process of DW with MDA by formally defining a set of transformation rules QVT (Query, View, and Transformation). The PIM is modeled using the UML activity diagram. Atigui et al. [9] have proposed an approach where the designer built his unified conceptual model PIM which describes the multidimensional structures and related ETL process.

BPMN Based ETL Process Modeling: El Akkaoui et al. [10] provide an independent platform for the conceptual modeling of an ETL process based on the BPMN. Using the same BPMN objects presented by [10,11] proposes a correspondence between the ETL process and the needs of decision-makers to easily identify which data are necessary and how include them in the DW. Oliveira et al. [12] proposed to extend a previous work [10] by defining specific conceptual models that takes into account of the capture of evolutionary data, the change of dimensions, the treatment of the substitutions keys and the data quality. Wilkinson et al. [13], for instance, presented a method to guide BPMN specifications in the definition of conceptual models of ETL systems. Table 1 highlights a summary of the literature review which is based on five criteria:

- C1-Modeling of ETL process. This criterion is relative to the level of abstraction adopted in the ETL modeling approach: **"C"** (Conceptuel modeling of ETL process) and **"L"** (Logical modeling of ETL process).
- C2-Modeling language: the language used for the modeling of the ETL process.
- C3-ETL operations such as:
  - **"D"** Operations on the predefined data: all types of transformations realized with the data such as aggregation, filter, join, concatenation, etc.
  - **"C"** Operations expressing the constraints: all types of transformations and declarations errors or constraints such as Incorrect, Log, etc.
  - **"U"** Operations defined by the user: where the designer can define new operations.
  - **"S"** Operations for structuring: to unify the structure of the inputs.
- C4-ETL Level: indicates the level of the ETL process versus the DW design methodologies.
- C5-Design approaches: take into account the needs in the ETL process.

Notice that, the majority of works proposed the conceptual or/and logical design for the ETL process for data driven approaches. Only the work of Jovanovic et al. [14] and El-Akkaoui et al. [11] take into consideration the business requirement in the ETL process.

For the modeling language of ETL process, BPMN notation seem to be a good choice since it can cover a deficit of communication that often occurs between the design and implementation of business process. For that, we adopt this modeling language in our method.

As general rules, the ETL process starts after the logical modeling of the DW schema like the works of [10–13]. Nevertheless, Jovanovic et al. [14] propose to start the ETL process at the logical level.

To facilitate and minimize the complexity of ETL process, we propose to start the ETL process from the conceptual design phase of the DW and to take into account the business requirements in the ETL process. In fact, starting an ETL modeling at an earlier stage allows to benefit from the knowledge generated during the conceptual modeling of data warehouses by saving the traceability of the data in a correspondence table and modeling the transformation operations with BPMN.

Table 1. Summary of the literature review.

Approaches	C1	C2	C3				C4	C5
			D	C	U	S		
Munoz et al., 2008	C/L	UML (Activity Diagram)	Yes	No	No	No	Fourth step	No
Wilkinson et al., 2010	C/L	BPMN	Yes	No	No	No	Fourth step	No
El-Sappagh et al., 2011	C	UML	Yes	No	Yes	Yes	Fourth step	No
Atigui et al., 2012	C/L	UML	Yes	Yes	No	No	Fourth step	No
El Akkaoui et al., 2012	C	BPMN	Yes	Yes	No	No	Fourth step	Yes
Oliveira and Belo, 2012	C	BPMN	Yes	Yes	No	No	Fourth step	No
Jovanovic et al., 2012	C	/	Yes	Yes	No	No	Third step	Yes
Mallek et al., 2014	C	UML (Activity Diagram)	Yes	Yes	No	Yes	Fourth step	No
Our approach	C	BPMN	Yes	Yes	Yes	Yes	Second step	Yes

3 Two-ETL Phases Method

In this paper, we propose a method called Two-ETL phases for DW creation from heterogeneous sources. This method is composed of two phases to accomplish the ETL process and overcome its complexity. The first phase is done in a parallel way with the conceptual DW design and the second phase is realized to ensure the implementation of the specified ETL. In this method we propose to advance the ETL that is on the fourth level of the DW design methodologies into the second level (*cf.* Fig. 1).

As input to our method, we have heterogeneous data sources with different schemes and business requirements. Our method contains three main steps (*cf.* Fig. 2.): (i) DW design process, (ii) first ETL phase and (iii) second ETL phase. Step (i) and the step (ii) are operating in parallel to allow the design of DW conceptual and logical shemes, the correspondence table and the transformation operations. Finally, the third step (iii) consists on the implementation of new operations or the use of predefined ones in order to create the DW.

The Two-ETL phases as defined greatly facilitates the ETL process by minimizing the complexity and the time allocated to the implementation based on the explicit knowledge stored in the CT and modeling step. In the folowing, we will detail the Two-ETL phases.

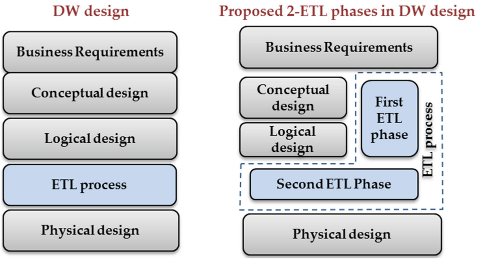


Fig. 1. Steps of DW design methodologie

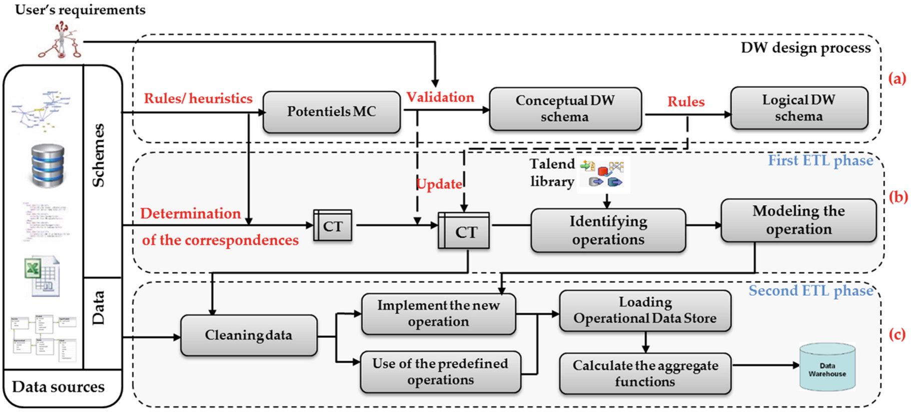


Fig. 2. Generic view of our proposed method

4 First ETL Phase

In the DW (mixed or data driven) design approaches, a set of rules/heuristics is used to identify potentials Multidimensional Concepts (MC) from the available data sources in order to obtain the DW conceptual schema. When starting this step, it is very important to save the data traceability of the used rules. For that, we propose to save this traceability on a table called Correspondence Table (CT).

Since the CT is well identified, we carry on the identification of the transformation operations.The last step of first ETL phase is the conceptual modeling of identified operations. As output of the first ETL phase we have the correspondence table with full documentation of all transformation operations. The explained process is modeled in BPMN language (*cf.* Fig. 3).

This flow uses pools of BPMN which provide a high expressivity for modeling. This pool encloses three lanes. This lanes focuses on the identification of the DW design shema, the determination of the correspondence, definition and modeling of the transformation operations, which allow to generate the correspondence table.

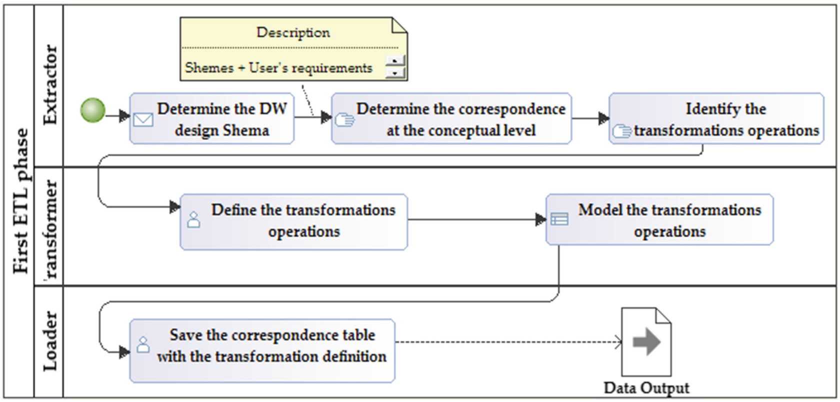

First ETL phase
Extractor
ransformer
Loader
Description
Shemes + User's requirements
Determine the DW design Shema
Determine the correspondence at the conceptual level
Identify the transformations operations
Define the transformations operations
Model the transformations operations
Save the correspondence table with the transformation definition
Data Output

Fig. 3. The first ETL phase

In the following, we will detail the preparation of the correspondence table, the identification of the transformation operations and the modeling of these operations.

4.1 Excerpt of the Correspondence Table

The identification of the Correspondence Table (CT) at the earliest phases of DW design is very important for the ETL process. This table stores the DS attributes names and the corresponding potential MC according to the used rule. At this stage, CT contains the first two columns of Table 2. The CT is then updated when the validation of potentials MC with business requirement is done. At this level, we obtain the valid DW conceptual schema and the valid CT. Finally, a set of rules is used to derive a logical schema. We recover the result of applied rules on our CT. Table 2 presents an excerpt of the correspondence table. The name of data sources (ONAT, Postal codes of Tunisia, ontology) will be presented in the Sect. 5.

4.2 Identification of Transformation Operations

After the determination of the CT, we propose to clarify the various types of transformation operations. In this context, we make a distinction between two types of operations: the ***defined*** transformation operations, which are supported by ETL tools (i.e. mapping, filtering, etc.) and the ***undefined*** transformation operations, which are unsupported by the ETL tools. In fact, the defined operations can not cover all the possible transformations because they depend on the data sources and the conceptual model of DW. Therefore, we need to apply some other transformation operations which depend on the context of work. These transformation operations are carried out through the addition of a new

Table 2. Excerpt of the correspondence table

Data sources		Target data	operations	
Name of the DS	Field	(data warehouse)	Operations names	Operations types
ONAT	Full_Name	Name First Name	Decomposition operation	Undefined
ONAT	BirthDate	Age_Group	Discrimination operation	Undefined
ONAT	Date	Day Month Year Week Quarter Semester	Explosion operation	Undefined
ONAT	Sex	Sex	Mapping operation	Defined
ONAT and postal codes of Tunisia	Postal_Code	Postal_Code Office_Desig Governorate Country	Join operation	Defined
ONAT	Address	Street_Number Street_Desig	Decomposition operation	Undefined
ONAT and ontology	Activity	Activity Activity_Desig	Mapping operation	Defined
ONAT and ontology	Activity_Group	Activity_Gr Activity_Gr_Desig Raw_Mat_Desig Raw_Mat_Price	Join operation	Defined

expressions that contain the composition of two or more predefined functions or calling a routine which contains a program according to the transformation operation.

Based on the operator library, we alter the CT by a new two columns (*cf.* Table 2): operation name and operation type. In the first one we indicate the name of the operation (i.e. join, split, etc.) and the second one if the operation is supported by ETL tool or not (defined or undefined).

4.3 Modeling of the Transformation Operations

In the ETL tools (Talend[2], Pentaho Data Integration[3], etc.) many transformation operations are available to transform data such as Mapping operation, Filtering operation, etc. But in the real case study we can needs others operations unsupported by those ETL tools for that we should add new operations.

This section is dedicated to present same proposed operations identified in the correspondence table such as: decomposition operation, explosion operation and discrimination operation.

Decomposition Operation: According to the correspondence table, the decomposition operation occurs when we need to decompose a field of the source into

[2] http://www.talend.com.

[3] http://www.pentaho.fr/explore/pentaho-data-integration/.

multiple target attributes in the DW. This operation is modeled in the workflow as a succession of tasks which are: select the field to decompose, define the criteria by which we will do the decomposition, execute the operation decomposition and save the resulted attribute into a temporary table.

Example 1. As mentioned in Table 2 the attribute ***Ful_ Name*** will be decomposed based on a regular expression to generate ***First_Name*** and ***Name*** as suggested in the DW (*cf.* Fig. 4).

Example 2. This type of operation is applied to the field ***Address*** to determine the parameter ***Street_Number*** and weak attribute ***Street_Desig*** of the dimension ***Artisan*** (*cf.* Fig. 5).

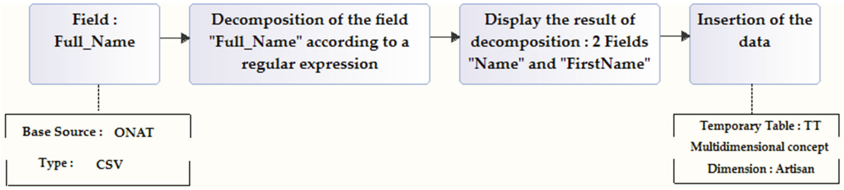

Field : Full_Name
Decomposition of the field "Full_Name" according to a regular expression
Display the result of decomposition : 2 Fields "Name" and "FirstName"
Insertion of the data
Base Source : ONAT
Type : CSV
Temporary Table : TT
Multidimensional concept
Dimension : Artisan

Fig. 4. BPMN modeling of decomposition operation for Full_Name attribute.

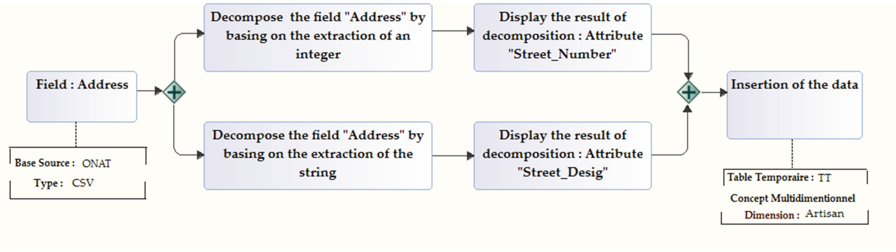

Field : Address
Base Source : ONAT
Type : CSV
Decompose the field "Address" by basing on the extraction of an integer
Decompose the field "Address" by basing on the extraction of the string
Display the result of decomposition : Attribute "Street_Number"
Display the result of decomposition : Attribute "Street_Desig"
Insertion of the data
Table Temporaire : TT
Concept Multidimentionnel
Dimension : Artisan

Fig. 5. BPMN modeling of decomposition operation for address attribute.

Discrimination Operation: This kind of operations occurs when we have to assign a categorical attribute from a set of numeric values. In our case of study, we have to define the parameter ***Age_Group*** from the ***Birth_date*** attribute. This operation must be preceded by a conversion operation. The conversion operation calculates the ***Age*** from the ***Birth_date***. Once the age is determined, we move to the discrimination operation. The workflow of this operation is as follows: select the attribute, define the conversion operation, execute this operation, define the operation of discrimination, execute this operation and save the result in the temporary table. An example of modeling is presented in Fig. 6.

Explosion Operation: The explosion operation aims at defining a multiple attributes from a single field. A concrete example of this operation is manifested

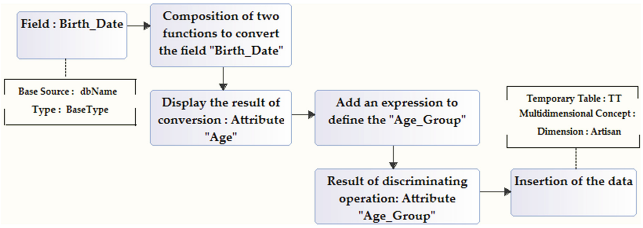

Field : Birth_Date
Base Source : dbName
Type : BaseType
Composition of two functions to convert the field "Birth_Date"
Display the result of conversion : Attribute "Age"
Add an expression to define the "Age_Group"
Result of discriminating operation: Attribute "Age_Group"
Temporary Table : TT
Multidimensional Concept :
Dimension : Artisan
Insertion of the data

Fig. 6. BPMN modeling of discrimination operation for Birth_Date attribute.

by the generation of several attributes from the field ***Date***. The workflow of this operation is: select the date, define the applied operation, execute the operation and save the attributes in the temporary table. Figure 7 illustrates the steps in the explosion operation of the field ***Date*** in ***Day***, ***Month***, ***Half_year***, ***Quarter***, ***Week***, and ***Year***.

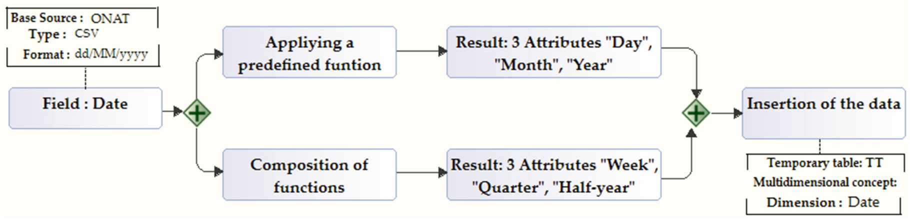

Base Source : ONAT
Type : CSV
Format : dd/MM/yyyy
Field : Date
Appliying a predefined funtion
Composition of functions
Result: 3 Attributes "Day", "Month", "Year"
Result: 3 Attributes "Week", "Quarter", "Half-year"
Insertion of the data
Temporary table: TT
Multidimensional concept:
Dimension : Date

Fig. 7. BPMN Modeling of explosion operation for Date attribute.

5 Second ETL Phase

This phase consists of the implementation of the ETL process based on the correspondence table and the modeled operations. So, we start by loading data from data sources into a temporal database based on CT then the cleaning (For example: treat the missing data and the null values) step is realized. After that, we use the supported operations and implement the unsupported ones. Finally, a set of aggregation functions used and then loaded into a DW shema. In Fig. 8, we present the sequence flow of this phase.

Our method is experimented in the real case study BWEC project. Many information are collected for this project about handicraft women such as those about profiles, productions and the ability to use new technologies. These information are represented through: the ONAT[4] data source which contains a list of artisans and their information, an ontology which contains the list of the raw materials of production and the data source of Postal Codes of Tunisia. Figure 9 presents the DW schema to be loaded (b), the excerpt of ontology (c) and the excerpt of ONAT (a).

[4] http://www.onat.nat.tn/accueil/.

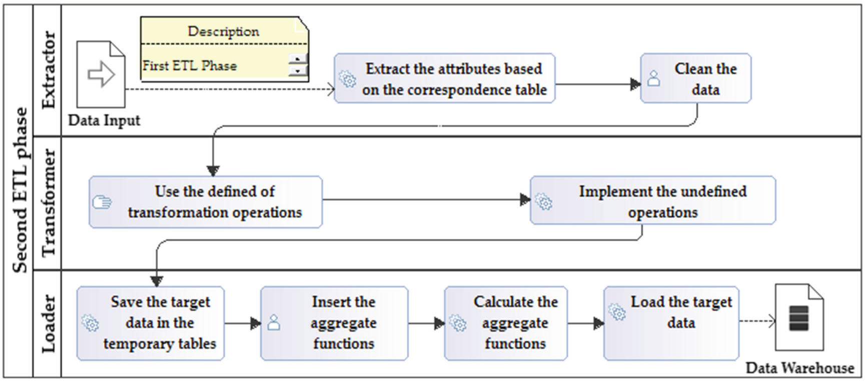
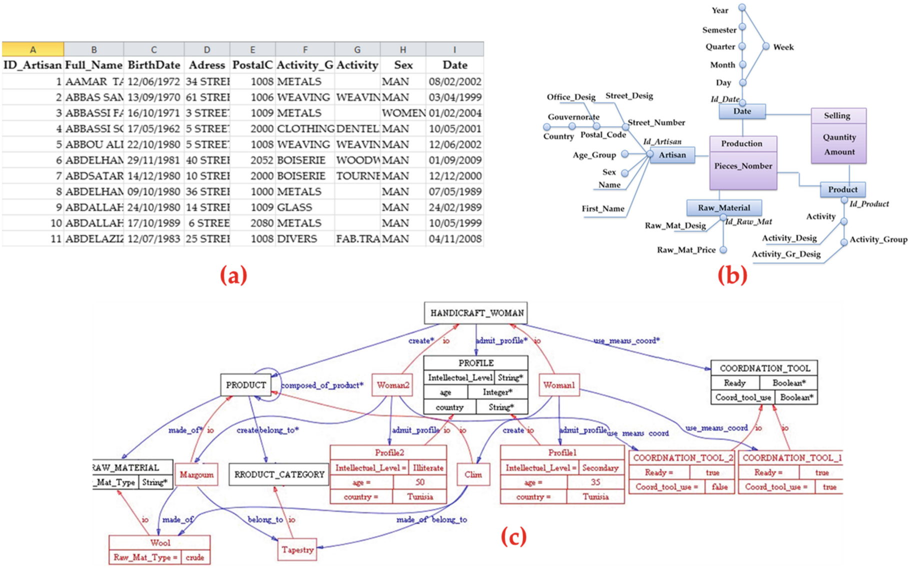

Second ETL phase
Extractor
Description
First ETL Phase
Data Input
Extract the attributes based on the correspondence table
Clean the data
Transformer
Use the defined of transformation operations
Implement the undefined operations
Loader
Save the target data in the temporary tables
Insert the aggregate functions
Calculate the aggregate functions
Load the target data
Data Warehouse

Fig. 8. The second ETL phase

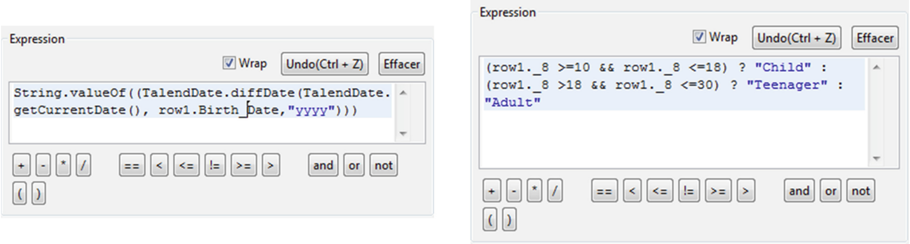


A	B	C	D	E	F	G	H	I
ID_Artisan	Full_Name	BirthDate	Adress	PostalC	Activity_G	Activity	Sex	Date
1	AAMAR TA	12/06/1972	34 STREE	1008	METALS		MAN	08/02/2002
2	ABBAS SAN	13/09/1970	61 STREE	1006	WEAVING	WEAVIN	MAN	03/04/1999
3	ABBASSI FA	16/10/1971	3 STREET	1009	METALS		WOMEN	01/02/2004
4	ABBASSI SC	17/05/1962	5 STREET	2000	CLOTHING	DENTEL	MAN	10/05/2001
5	ABBOU ALI	22/10/1980	5 STREET	1008	WEAVING	WEAVIN	MAN	12/06/2002
6	ABDELHAN	29/11/1981	40 STREE	2052	BOISERIE	WOODW	MAN	01/09/2009
7	ABDSATAR	14/12/1980	10 STREE	2000	BOISERIE	TOURNE	MAN	12/12/2000
8	ABDELHAN	09/10/1980	36 STREE	1000	METALS		MAN	07/05/1989
9	ABDALLAH	24/10/1980	14 STREE	1009	GLASS		MAN	24/02/1989
10	ABDALLAH	17/10/1989	6 STREE	2080	METALS		MAN	10/05/1999
11	ABDELAZIZ	12/07/1983	25 STREE	1008	DIVERS	FAB.TRA	MAN	04/11/2008

Fig. 9. Excerpt of the DS (a), (c) and the DW schema (b)

We have choisen Talend Open Studio (TOS) for the creation of our DW. TOS is based on the creation of a “job” to maintain the execution of the data process. The user can apply the various components of the palette to build the work on the design side and view the generated code.

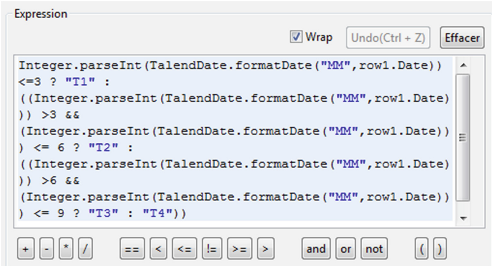
Available components in Talend realize some operations, but we notice the absence of some operations detected in the source analysis phase. TOS allows adding new operations. In the folowing, we detaied the realization of the new operations.



```
String.valueOf((TalendDate.diffDate(TalendDate.
getCurrentDate(), row1.Birth_Date,"yyyy")))
```

```
(row1._8 >=10 && row1._8 <=18) ? "Child" :
(row1._8 >18 && row1._8 <=30) ? "Teenager" :
"Adult"
```

Fig. 10. Expression of the discrimination operation



```
Integer.parseInt(TalendDate.formatDate("MM",row1.Date))
<=3 ? "T1" :
((Integer.parseInt(TalendDate.formatDate("MM",row1.Date)
)) >3 &&
(Integer.parseInt(TalendDate.formatDate("MM",row1.Date))
) <= 6 ? "T2" :
((Integer.parseInt(TalendDate.formatDate("MM",row1.Date)
)) >6 &&
(Integer.parseInt(TalendDate.formatDate("MM",row1.Date))
) <= 9 ? "T3" : "T4"))
```

Fig. 11. Expression which determines the parameter quarter

Realization of the Discrimination Operation: The realization of the discrimination operation requires two steps: the first step is the conversion of ***Birth_date*** in ***Age*** and the second step is the discrimination of ***Age_Group*** from the calculated ***Age***. Figure 10 describes the insertion of two expressions where we define the different age groups.

Realization of the Explosion Operation: The realization of the explosion operation requires the addition of expressions that we have to determine the parameters ***Day***, ***Month***, ***Half_year***, ***Quarter***, ***Week***, and ***Year***. To do this, we take the internal base ONAT input and output the Date table in Microsoft SQL Server. Figure 11 shows the insertion of the phase to generate a ***Quarter*** of each date of the source database.

Figure 12 present the execution of the ETL process (defined and implemented), the statistics appear on the graphical interface elements. These statistics indicate the success of our method.

6 Conclusion

In this paper, we have proposed two-ETL phases as part of an integrated and global approach for DW creating. Our method allows to paralyze the design of ETL process with the conceptual DW design, which facilitates the implementation of this process. In this method, we have used the BPMN that allows to cover

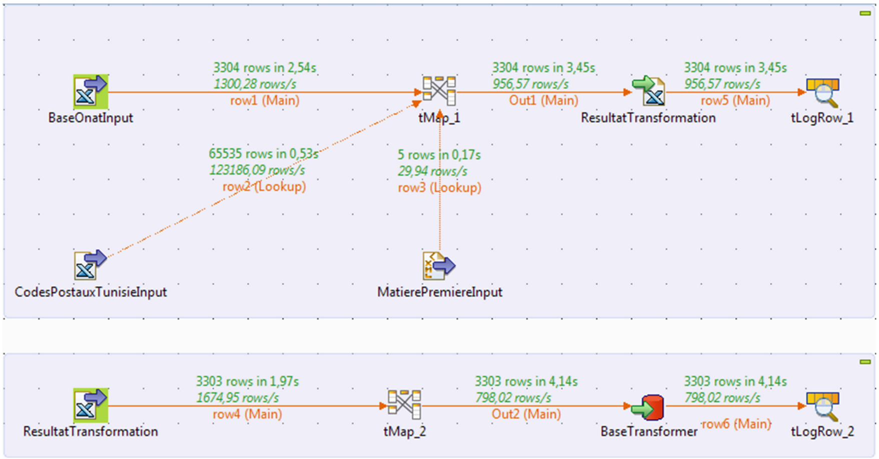


Fig. 12. General process of all operations

a deficit of communication that often occurs between the design and implementation of business processes. The first ETL phase allows the determination of correspondences and the identification of transformation operations. This step is performed with the DW design process. The result of this work provides the correspondence table that saves the traceability of data. From the CT, we have performed modeling of the operation to facilitate and minimize the complexity of the second ETL phase. Finally we implemented these operations and loaded them in the operational data store. Future works include developing a validation procedure for the produced models using this framework. This will allow to produce a rigorous comparison between the outcome of this methodology, and other ones, not only in terms of workflow structure, but also in terms of flexibility, adaptability to change, usability, and performance. Changes can occur during the lifecycle of the warehouse, not only in sources, but also within the warehouse.

References

1. Golfarelli, M.: From user requirements to conceptual design in data warehouse design-a survey. In: Data Warehousing Design and Advanced Engineering Applications: Methods for Complex Construction, pp. 6–11 (2010)
2. Nabli, A.: Approche d'aide à la conception automatisée d'entrepôt de données: Guide de modèlisation. Presses Acadmiques Francophones (2013)
3. Favre, C., Bentayeb, F., Boussaid, O., Darmont, J., Gavin, G., Harbi, N., Kabachi, N., Loudcher, S.: Les entrepôts de données pour les nuls. ou pas!. In: 2éme Atelier aide à la Décision à tous les Etages (EGC/AIDE), Janvier 2013
4. Trujillo, J., Luján-Mora, S.: A uml based approach for modeling ETL processes in data warehouses. In: Song, I.-Y., Liddle, S.W., Ling, T.-W., Scheuermann, P. (eds.) ER 2003. LNCS, vol. 2813, pp. 307–320. Springer, Heidelberg (2003)

5. Mallek, H., Walha, A., Ghozzi, F., Gargouri, F.: ETL-web process modeling. In: ASD Advances on Decisional Systems Conference (2014)
6. El-Sappagh, A., Hendawi, A., Bastawissy, H.: A proposed model for data warehouse ETL processes. J. King Saud Univ. Comput. Inf. Sci. **23**(2), 91–104 (2011)
7. Muñoz, L., Mazón, J.-N., Pardillo, J., Trujillo, J.: Modelling ETL processes of data warehouses with UML activity diagrams. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 44–53. Springer, Heidelberg (2008)
8. Munoz, L., Mazon, J., Trujillo, J.: Automatic generation of ETL processes from conceptual models. In: Data Warehousing and OLAP, pp. 33–40 (2009)
9. Atigui, F., Ravat, F., Teste, O., Zurfluh, G.: Using OCL for automatically producing multidimensional models and ETL processes. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2012. LNCS, vol. 7448, pp. 42–53. Springer, Heidelberg (2012)
10. El Akkaoui, Z., Zimanyi, E.: Defining ETL worfklows using BPMN and BPEL. In: Data Warehousing and OLAP, pp. 41–48 (2009)
11. El Akkaoui, Z., Mazón, J.-N., Vaisman, A., Zimányi, E.: BPMN-based conceptual modeling of ETL processes. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2012. LNCS, vol. 7448, pp. 1–14. Springer, Heidelberg (2012)
12. Oliveira, B., Belo, O.: BPMN patterns for ETL conceptual modelling and validation. In: Chen, L., Felfernig, A., Liu, J., Raś, Z.W. (eds.) ISMIS 2012. LNCS, vol. 7661, pp. 445–454. Springer, Heidelberg (2012)
13. Wilkinson, K., Simitsis, A., Castellanos, M., Dayal, U.: Leveraging business process models for ETL design. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 15–30. Springer, Heidelberg (2010)
14. Jovanovic, P., Romero, O., Simitsis, A., Abelló, A.: Requirement-driven creation and deployment of multidimensional and ETL designs. In: Castano, S., Vassiliadis, P., Lakshmanan, L.V.S., Lee, M.L. (eds.) ER 2012 Workshops 2012. LNCS, vol. 7518, pp. 391–395. Springer, Heidelberg (2012)