A Configurable Strategy for Extraction, Transformation and Load to Support Data Propagation on Active Data Warehouses

Carlos Roberto Valêncio, Paulo Scarpelini Neto,
Leandro Alves Neves, Geraldo Francisco Donegá
Zafalon, Rogéria Cristiane Gratão de Souza
Department of Computer Science and Statistics
São Paulo State University (UNESP)
São José do Rio Preto, São Paulo, Brazil
E-mail: valencio@ibilce.unesp.br,
pauloscarpelini@gmail.com,
leandro@ibilce.unesp.br,
zafalon@sjrp.unesp.br,
rogeria@ibilce.unesp.br

Angelo Cesar Colombini
Department of Computer Science and Statistics
Federal University of São Carlos (UFSCar)
São Carlos, São Paulo, Brazil
E-mail: accolombini@dc.ufscar.br

***Abstract*— This work consists of the construction of a strategy called ETL-PoCon to execute Extraction, Transformation and Load (ETL) processes in active Data Warehouses (DW) with a configurable policy. The original contribution of this work is to provide a strategy that considerably reduces the quantity of data transfers to active DW, besides maintaining a satisfactory level of data freshness. Said reduction is obtained by means of configurable policies of data propagation based on relevance of the data regarding to the information stored in the DW. The strategy was implemented in a database related to health worker that contains more than seventy thousand records of occupational accidents. Experiments have shown that the ETL-PoCon strategy significantly contributes towards a reduction of the overload on the systems involved in the active DW environment, since all results presented a reduction higher than 60% in the amount of DW refreshments.**

Keywords: data warehouse, ETL, active data warehouse, near real-time data warehouse.

I. Introduction

In Data Warehouse (DW) environments, data is extracted from sources and transferred to a central repository by means of the Extraction, Transformation and Load (ETL) processes. The vast majority of the DWs are refreshed with periodical ETL processes that, as they demand the deactivation of data source systems, are known as off-line executions [1-4].

The growing need of big corporations to reduce the time between the generation of data and an analysis of its information makes the off-line ETL process impracticable and demands the creation of approaches that are capable of transferring data in short intervals of time without the need to deactivate the systems [5-6].

Faced with that panorama, active DWs appeared with time intervals between the ETL tools executions reduced from days to hours, or even minutes, but this reduction directly affected the ETL process, that becomes performed in parallel to data sources systems and the DW. Thus, was added an extra consumption of computing resources arising from the execution of transactions in both data sources such as the repository, such transactions can be costly to the point of affecting the performance of the entire environment. Therefore, frequent refreshments made it necessary to create criteria to prioritize the data to be transferred, since a transfer of all and any data from the sources would overload the systems [7]. The adoption of strategies that do not have a pre-defined refreshment frequency become interesting, since they would diminish unnecessary transfers and reduce the use of source system resources.

This work presents a strategy for the execution of ETL process turned to active DWs, called ETL-PoCon. The strategy is distinguished from existing ones in that it has policies configurable of data propagation that allow the user to configure data transfers, based on relevance of the data regarding to the information stored in the repository (DW).

II. Theory and Related Works

A Data Warehouse can be understood as being a non-volatile repository of integrated data that has as its objective to support decision making in big corporations. DW architecture has a central database from where data from the sources is extracted, treated and stored. The ETL process may be considered as a key for the extraction of data from distinct sources and the transfer of them in a homogeneous way to a central repository [3]. Construction and maintenance of the tools that are responsible for the execution of the ETL process is fundamental for the success of a DW. In some cases, the development of these tools take up 80% of the resources applied in the DW project [10].

Works found in literature related to the construction of ADWs cover a wide scope. Some of them are proposals for DW architectures that support a high level of freshness [3], [15], [16], [21], while others focus on strategies for DW refreshment without any alterations in the architecture that merits highlighting [16], [21].

Nguyen [7] presents a survey of the state of the art related to a zero latency DW. The author assumes that although off-line refreshment still supports many organizations, the DWs are increasing their capacities to support not only strategic

978-1-5090-5081-9/16 $31.00 © 2016 IEEE
DOI 10.1109/PDCAT.2016.52

decisions but also the operational processes of the organizations. In this context, five phases of DW architecture evolution are described in which it is possible to verify that the emergence of new approaches for DW refreshment will permit a reduction of the latency between the events in the business and the actions taken as a consequence.

Vassiliadis [6] summarizes all the problems and challenges related to the development of tools for the execution of the ETL process for DW with real-time refreshments. The author starts from the problems present in conventional ETL tools to identify the technological challenges that are present in each one of the real-time ETL process stages.

Javed [3] presents a DW architecture proposal that joins the conventional approach to a real-time refreshment approach. With that approach, the system manages to reach a higher level of freshness, besides permitting that the ETL off-line process be executed in less time, since part of the data was already transferred in real-time.

Another architecture is proposed by Zhu [14], who contemplates the extraction process executed by a web-service and the transfer of data done with the use of XML files. The focus of the work is the creation of a structure made up of various cache levels, each one containing data that was altered during a certain period.

The work presented by Chen [16] consists of the development of a DW refreshment mechanism that does an analysis of some parameters to define the frequency of repository refreshments. Summarizing, the mechanism monitors the state of the data sources and the repository, analysing the impact should the repository be refreshed, as well as the number of registers that would be affected and the frequency in which data would be consulted. From those three factors, the mechanism is capable of defining when an update should be done in real-time or semi real-time (short intervals of time).

III. A Configurable Strategy for the Extraction, Transformation and Load on Active Data Warehouse

The ETL-PoCon strategy is described in this section. Initially, an analysis about the problem related to ADW refreshments is presented.

A. Definition of the problem

Works found in literature that are related to the development of ADWs are, mostly, turned to the construction of architectures to support a new level of DW freshness. Although some works mention the problem of overloads on the systems [3], [6], [14], few developed techniques and strategies to permit the execution of ETL processes only when relevant, that is, execute the ETL process only on data that directly affects the repository analyses [16].

Some strategies adopted for ADW refreshments are based only on the increase of the ETL tools execution frequency, and refreshments previously done daily would now be done in intervals of minutes or seconds. In that type of strategy, the DW refreshment frequency is pre-defined and may have the use of resources in operations about data that are not relevant to the analyses executed on the DW.

To exemplify, consider a department store with several subsidiaries that uses an ADW to analyse sales data. Suppose that company wants to analyse only the data of subsidiaries that reached a volume of 100 sales per hour. If the DW refreshment adopts an approach as previously described, the ETL process would be executed in pre-determined time intervals and without an analysis on the subsidiaries' sales vol-ume. Operations would be executed with non-relevant data (subsidiaries with sales lower than 100 per hour) and, consequently, resources would be used unnecessarily.

The adoption of strategies that do not have a pre-defined refreshments frequency is shown to be relevant, since they diminish unnecessary transfers and reduce the use of resources of the source systems. It is worth highlighting that the non-existence of a pre-established refreshment interval requires a strategy that has mechanisms capable of deciding when the ETL process must be triggered. To create a mechanism capable of defining the DW refreshment frequency and can analyse innumerable variables is not a trivial task and must rely on an analysis of innumerable variables. Some of those variables are described in the following.

- Refreshment frequency: trigger mechanisms of the ETL process must have structures that are capable of permitting different refreshments frequencies, that is, each mapping between a data source and the repository must have a specific refreshment interval. In short, data having a greater degree of importance must have a greater refreshment frequency.
- Volume: in an ETL process executed in an ADW, only data that was modified after the last refreshment must be transferred. Generally, that data set is named delta (Δ), and the quantity of tuples contained in that set is named volume of the delta (V(Δ)). The analysis of V(Δ) is important for the mechanism that defines the refreshment frequency, as the Δ represents the set of tuples that have not yet been refreshed in the DW. Therefore, the V(Δ) is directly proportional to the degree of outdates in the repository. The analysis of V(Δ) it is interesting to measure the lower the volume the less efficient the transfer, but the update level of DW is in-creased [5].
- Relevance: another factor to be considered is the relevance of the delta (R(Δ)). Tuples altered in the source can cause different impacts on the repository, as one tuple may be more or less relevant than another. A relevance analysis is quite complex and can involve semantic questions of the repository and sources.

B. *General vision of ETL-PoCon strategy*

The presented analysis evidences the need for the creation of a strategy to execute the ETL process that is capable of defining the refreshment frequency of the DW based on an analysis of the involved variables. The ETL-PoCon consists of a refreshment strategy focused on the control of the data transfer trigger.

The strategy is constituted of a mechanism that has income parameters configured by the user and does an analysis of the relevance of the delta to define the refreshment frequency of the DW. Moreover, it adopts an approach in which it is possible to determine specific refreshment intervals for each mapping between the data sources and the repository.

Figure. 1 (a) shows a classical DW scheme, while Fig. 1(b) shows a DW scheme which uses the ETL-PoCon, there it is possible to identify the presence of the controlling mechanism involving the deltas of each one of the sources, the Transformation and Load process, besides the repository itself. This representation is done with the intention of showing that the mechanism has direct interaction with the deltas (to ad-measure volume and relevance) and with the repository (to admeasure relevance). Next, some fundamental concepts of the strategy are defined.

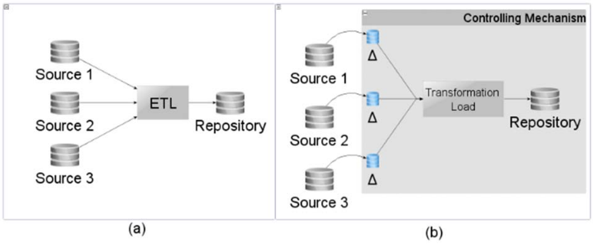


Figure. 1. (a) Structure of a classical DW (b) Structure of a DW with ETL-PoCon strategy

C. *Analysis of the relevance of delta - R(Δ)*

Relevance of the delta is a parameter to be considered when it is wished to diminish the execution of the ETL processes on non-impacting data. Relevance of a tuple is directly related to the impact caused when it is transferred to a repository. In the ETL-PoCon, the analysis considers the number of references made to the tuple at issue.

To exemplify, consider the scheme presented in Fig. 2. "Table 2" has a foreign key constraint in field "id_table 1" referent to the primary key of "Table 1". As may be observed, registers 1, 2 and 3 in "Table 2" refer to register 1 in "Table 1", while register 4 and 5 refer to register 2 in "Table 1" and so on successively.

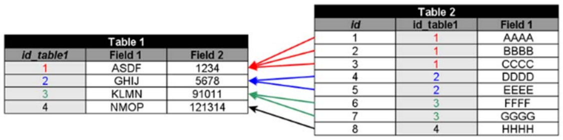


Table 1

id_table1	Field 1	Field 2
1	ASDF	1234
2	GHIJ	5678
3	KLMN	91011
4	NMOP	121314

Table 2

id	id_table1	Field 1
1	1	AAAA
2	1	BBBB
3	1	CCCC
4	2	DDDD
5	2	EEEE
6	3	FFFF
7	3	GGGG
8	4	HHHH

Figure. 2. Representation scheme of register's relevance

In that scheme, from the analysis of "Table 1", it can be said that register 1 is the most relevant since it is referenced three times in "Table 2". In that same "Table 1", registers 2 and 3 have the same relevance as both are referenced twice in "Table 2" and, lastly, register 4 has the lowest degree of relevancy as it was referenced only once.

The degree of relevancy is established based on the impact caused on the database should one of the tuples be updated or removed. An update of tuple 1 in "Table 1" would indirectly affect the information in three registers, while an alteration of tuple 2 would affect the information in only two registers. It is worth highlighting that, commonly, a same table may be referenced by innumerous other ones. Therefore, considering Tref as a set of tables that reference table T, the relevance of a tuple from table T could be defined by the expression:

$$R_t = \frac{Total\ of\ tuples\ in\ Tref\ that\ reference\ tuple\ A}{Total\ of\ tuples\ in\ Tref}$$

From this expression it is possible to calculate the relevance of each tuple in "Table 1" belonging to the scheme in Fig 2. In that case, Tref is constituted only by "Table 2" that has a total of eight tuples. In Table 1 is presented the degree of relevance of each tuple from "Table 1".

Therefore, R(Δi) is defined as the minimum relevance that Δ referent to the mapping i must be reached to trigger the transfer process. The R(Δi) can be calculated by adding the relevance of each tuple belonging to Δ. The R(Δi) value will always be between zero and one, as it is defined by the percentage of tuples that reference some tuple of Δi in relation to the total Tref tuples.

This way, R(Δi) with a value of 0.1 indicates that the mechanism must transfer the data to Δi only if these were referenced by 10% of the total Tref tuples. In other words, the mechanism must do the transfer only if the impact on the tables that reference the destination table be greater or equal to 10%. In the same way, R(Δi) with value 1 indicates that the mechanism must transfer Δi only if the impact were 100%, that is, all the tuples of the tables that reference the target table refer to a Δi tuple. The parameter R(Δi) is relative to the repository. For a definition of the relevance of Δ, the mechanism must admeasure the data sources, to identify the tuples that make up Δ, and the repository to identify the total number of tuples that reference Δ.

With the application of R(Δi) parameter is expected that there will be a reduction of unnecessary executions in the ETL process, since only the data that reaches the degree of relevancy defined by the user will trigger the process. Therefore, the relevance of Δi tends to grow until it reaches the limit defined by the user when the data are then transferred and the relevance of delta returns to zero.

IV. Experiments and Results

In this section the tests and results of the application of the ETL-PoCon strategy are presented.

A. Environment used

The database used stores more than 70 thousand work accidents and has a structure that enables a configuration of strategies and validation by means of the execution of simulations of data insertions and updates.

The definition of the test environment was conducted with a focus on validation of the strategy developed and allowed to set the parameter R(Δ) according to real requirements, besides to allow the measurement of all updates that occurred in each experiment. The implementation of the strategy in an environment with more data is unnecessary, since the tests and analyses are independent of the processing time of each transfer and maintain the focus on indexes updating of DW.

In this environment that was used for the tests. Both from the database source and the repository, only the "Company" and "Notification" tables were considered. The "Notification" table is used to store the work accident notifications and has information such as the name of the injured person, locality of the accident, date, etc. The "Company" table stores information about the company that employs the injured person and has information such as the name and branch of activity. It is worth highlighting that the cardinality between "Company" and "Notification" is many-to-one, that is, a company may employ various people who had accidents but a single accident can only involve one company.

B. Validation of the policy by relevance

The application of a policy based on defined relevance in the ETL-PoCon strategy can be quite useful in reducing total active DW refreshments. Bearing in mind that the principal objective is to permit a reduction of executions in the process of transferring data and to maintain a satisfactory degree of data refreshments in the repository, this section presents some experiments that analyse not only the total number of executed refreshments, but also the semantics contained in the transferred data. The objective is to verify if the application of the policy by relevance did not generate delays in the transference of sensitive data resulting from the reduction of DW refreshments.

For the analyses about transferred data was used a report that has as its objective to summarise the 10 companies that have the highest number of work accidents. Table 1 presents an example of that report extracted from the database used in the experiments. The table represents the percentage that the total number of accidents of each company represents in relation to the total number of accidents. Therefore, the company with the highest number of accidents is Company A that represent 3.195% of the registered total in the database.

TABLE I. Report of the 10 Companies Having the Highest Number of Accidents

Company	A	B	C	D	E	F	G	H	I	J
Percentage (%)	3.195	1.608	1.432	1.181	1.141	0.974	0.793	0.754	0.732	0.701

The companies that figure in the report constitute the focus of the vigilance work executed by public agencies responsible for worker's health. Any alteration in any of the companies must be transferred to the repository as quickly as possible. Therefore, in the environment used for the tests, the data of these companies is considered sensitive, that is, they directly alter the extracted results in the repository.

In the experiments described in the following, at each data transfer, an analysis of the transferred data is done to verify if some data of one the 10 principal companies has altered. That way, it is possible to verify if that data, which are really relevant, are being transferred with priority and without delays. Table 2 presents the parameters that were used in all four experiments, and Table 3 presents the configurations of the databases and executed transactions. The four experiments relied on the same configuration and the alterations were the only ones on the transacted data set.

TABLE II. Parameters Used in Experiments Referent to the Policy by Relevance

	Experiments I, II, III and IV
T	10 seconds
R(Δ2)	0.5%

TABLE III. Information of Experiments Referent to the Policy by Relevance

	Experiments I, II, III and IV
Transactions	1.000
Time	10 min
Transactions/second	1.66

1) Experiment I

In experiment I the DW was refreshed 12 times, while without applying the strategy 48 refreshments would be executed. Therefore, the percentage reduction was of 75%. Figure. 3 presents a R(Δ) graph of the execution of the experiment. An analysis of the migrated data permited identifying that among the 12 refreshments executed during the experiments, only one of them were from a company that is in the 10 leading companies. That alteration occurred in the data source between the instances "19" and "20" and was transferred in the instance "20". The graph shows the instance where there was an information migration from Company E.

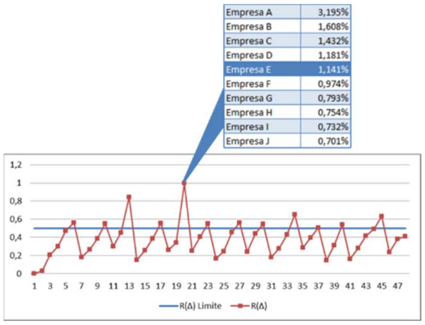


Figure. 3. Graph of the R(Δ) evolution during experiment I with sensitive data transferred in instance 20 highlighted

2) *Experiment II*

Figure. 4 presents a graph of R(Δ) evolution and highlights the data transferred in instances “22”, “34” and “40”. With the strategy, total refreshments were reduced by 62.7% as 15 refreshments were done.

In that experiment, of the 15 refreshments that were done, 3 directly affected the 10 principal companies. In all those cases, the refreshment of the altered company happened at intervals that were less than the controlling mechanism cycle, that is, in none of the cases was there a delay greater than 10 seconds for a sensitive data to be transferred to the repository. In instances “20” and “40” it is possible to verify that a relevancy of close to 1.5% was reached. This is due to the fact that the simulated transactions in the data source affected companies that represented 1.181% (Company D) and 1.432% (Company C).

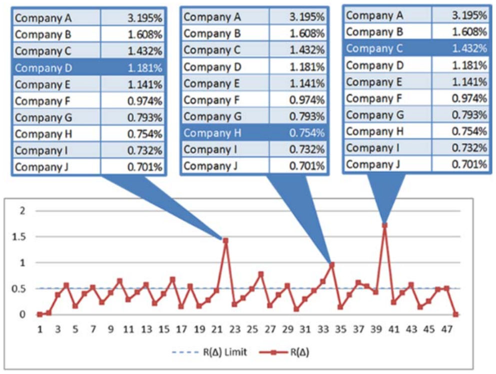


Figure. 4. Graph of the R(Δ) evolution during experiment II with sensitive data transferred in instances 22, 34 and 40 highlighted.

3) *Experiment III*

The graph of the R(Δ) evolution and the highlight of the sensitive data transfer are presented in Figure 5. Fourteen DW refreshments were executed which represent a reduction of 70%. During the experiment, there was only one alteration in the 10 principal companies and it happened between the instances “26” and “27” and was transferred in the instance “27”.

As with other experiments, the sensitive data transfer did not exceed 10 seconds referent to the controlling mechanism cycle. It is possible to observe that in instance “26” the average relevance was a little less than 0.5%. In instance “27” the relevancy took a leap to approximately 1.4% due to operations done on Company I that represent 0.732% of the total accidents.

4) *Experiment IV.*

In Figure 6 a graph of the R(Δ) evolution during experiment IV is presented. In this last experiment, the total DW refreshments were 16, which represented a reduction of 66%. In the experiment, there were transfers of sensitive data in the instances “13” and “20” and, same as with the previous ones, the delay between the transaction of companies’ information and the transfer of these to the repository did not exceed 10 seconds.

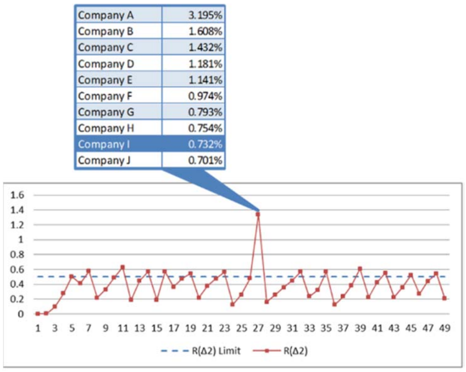


Figure. 5. Graph of the R(Δ) evolution during experiment III with sensitive data transferred in instance 27 highlighted

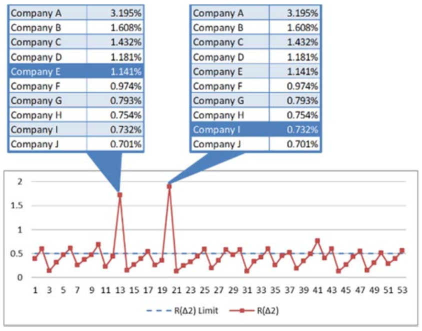


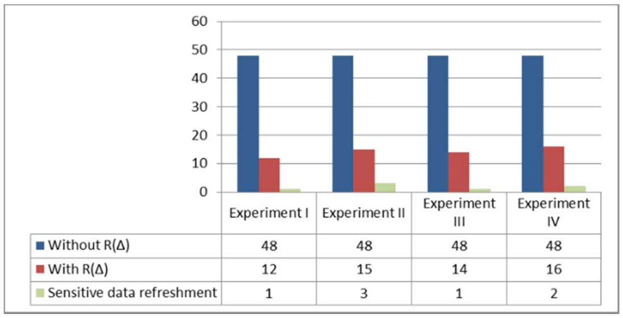
Figure. 6. Graph of the R(Δ) evolution during experiment IV with sensitive data transferred in instances 13 and 20 highlighted

5) *Discussion about results.*

In the four presented experiments there was a considerable reduction of the total refreshments when compared to the total that would be reached without the use of the R(Δ) parameter, as can be seen in the graph in Figure 7. Also shown by the experiments is that, even with a reduction of the number of refreshments, data considered sensitive were prioritised and not transferred with delays. The alterations of an executed operation on one of the 10 principal companies were transferred to the repository in a maximum of 10 seconds, an interval of time referent to the controlling mechanism cycle and defined by T parameter.

In some cases it is possible to verify that in the instances where there was a transference of sensitive data, the curve of the R(Δ) graph presents a higher peak than the others, that is, the relevance reached with an alteration in one of the 10 principal companies is relatively greater than the relevancy of the rest of the deltas. This characteristic shows that the value of R(Δ) could be increased and still maintain a satisfactory level of freshness.

On the other hand, the choice of an R(Δ) greater than 0.701% could cause delays with the transfer of information referent to Company J. This is because the chosen relevance would be superior to the percentage represented by that company, and, should the delta be made up of only that company, the R(Δ) would not be reached and the information would not be transferred.



	Experiment I	Experiment II	Experiment III	Experiment IV
🟦 Without R(Δ)	48	48	48	48
🟥 With R(Δ)	12	15	14	16
🟩 Sensitive data refreshment	1	3	1	2

Fig. 7. Total refreshments executed in the experiments I, II, III and IV

V. Conclusions

The ETL-PoCon strategy has as its central idea the definition of data propagation policies that will permit a reduction of the ADW freshness frequency to reduce the overload on the systems. In this context, policies based on the relevance of delta were defined, besides a definition that each mapping between the sources and the reposito-ry be treated in a different way with specific updating frequencies.

To define the degree of relevance of delta, a methodology based on the impact caused by each tuple of a data source to be transferred to the repository was defined. Admeasurements were done on the repository in the relevance calculation to verify the percentage of tuples that reference the delta.

The experiments that were done generated results which made it possible to verify that the proposed policies offered a reduction higher than 60% in the amount of DW refreshments without causing delay to the migration of sensitive data. Following, some of the principal contributions of the ETL-PoCon strategy are described. A poli-cy based on relevance is interesting for a reduction of refreshments. As R(Δ) is relative to the repository, a definition of that parameter can rely on analyses about the reports extracted from the DW.

Therefore, the developed work significantly contributed towards the construction of ETL tools for DW refreshments, since it offers an alternative to the treatment of the problem of overload in the systems.

References

[1] K. Sun, Y. Lan, SETL: A scalable and high performance ETL system, in: 2012 3rd International Conference on System Science, Engineering Design and Manufacturing Informatization, IEEE, 2012: pp. 6–9.

[2] L. Xu, J. Liao, R. Zhao, B. Wu, A PaaS based metadata-driven ETL framework, in: 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, IEEE, 2011: pp. 477–481.

[3] M.Y. Javed, A. Nawaz, Data Load Distribution by Semi Real Time Data Warehouse, in: 2010 Second International Conference on Computer and Network Technology, IEEE, 2010: pp. 556–560.

[4] A. Simitsis, C. Gupta, S. Wang, U. Dayal, Partitioning real-time ETL workflows, in: 2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010), IEEE, 2010: pp. 159–162.

[5] Song, Y. Bao, J. Shi, A Triggering and Scheduling Approach for ETL in a Real-time Data Warehouse, in: 2010 10th IEEE International Conference on Computer and Information Technology, IEEE, 2010: pp. 91–98.

[6] P. Vassiliadis, A. Simitsis, Near real time ETL, New Trends in Data Warehousing and Data Analysis. (2009) 1–31.

[7] T.M. Nguyen, A.M. Tjoa, Zero-latency data warehousing (ZLDWH): the state-of-the-art and experimental implementation approaches, in: 2006 International Conference onResearch, Innovation and Vision for the Future, IEEE, n.d. pp. 167–176.

[8] S. Luján-Mora, P. Vassiliadis, J. Trujillo, Data mapping diagrams for data warehouse design with UML, Conceptual Modeling-ER. (2004) 191–204.

[9] J. Shi, Y. Bao, F. Leng, G. Yu, Priority-Based Balance Scheduling in Real-Time Data Warehouse, in: 2009 Ninth International Conference on Hybrid Intelligent Systems, IEEE, 2009: pp. 301–306.

[10] R. Kimball, J. Caserta, The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data, Wiley, 2004.

[11] Y. Zhu, L. An, S. Liu, Data Updating and Query in Real-Time Data Warehouse System, in: 2008 International Conference on Computer Science and Software Engineering, IEEE, 2008: pp. 1295–1297.

[12] T. Jörg, S. Dessloch, Near Real-Time Data Warehousing Using State-of-the-Art ETL Tools, in: Enabling Real-Time Business Intelligence. Third International Workshop, BIRTE 2009, Held at the 35th International Conference on Very Large Databases, VLDB 2009, Lyon, France, August 24, 2009, Revised Selected Papers, Springer Berlin Heidelberg, 2010: pp. 100–117.

[13] L. Chen, W. Rahayu, D. Taniar, Towards Near Real-Time Data Warehousing, in: 2010 24th IEEE International Conference on Advanced Information Networking and Applications, IEEE, 2010: pp. 1150–1157.

[14] M.A. Bornea, A. Deligiannakis, Y. Kotidis, V. Vassalos, Semi-Streamed Index Join for near-real time execution of ETL transformations, in: 2011 IEEE 27th International Conference on Data Engineering, IEEE, 2011: pp. 159–170.

[15] Y. Fan, The Research of Active Data Warehouse Based on Multi-Agent, in: 2012 Spring Congress on Engineering and Technology, IEEE, 2012: pp. 1–4.

[16] R.J. Santos, J. Bernardino, Optimizing data warehouse loading procedures for enabling useful-time data warehousing, in: Proceedings of the 2009 International Database Engineering & Applications Symposium on - IDEAS '09, ACM Press, New York, New York, USA, 2009.