A Study of Extract–Transform– Load (ETL) Processes

S.Sajida, M.C.A, M.Tech,M.Phil,
Research Scholar,
Dept. of Computer science,
Sri Venkateswara University,
Tirupati 1

Dr.S.Ramakrishna, M.Sc.,M.Phil.,Ph.D.,M.Tech(IT)2
Professor,
Dept. of Computer science,
Sri Venkateswara University
Tirupati 1

Abstract- **In Data Warehouse (DW) environment, *Extraction-Transformation-Loading (ETL)* processes constitute the integration layer which aims to pull data from data sources to targets, via a set of transformations. ETL is responsible for the extraction of data, their cleaning, conforming and loading into the target. ETL is a critical layer in DW setting. It is widely recognized that building ETL processes is expensive regarding time, money and effort. It consumes up to 70% of resources. By this work we intend to enrich the field of ETL processes, the backstage of data warehouse, by presenting a survey on these processes. Therefore, in current work, firstly (1) we review open source and commercial ETL tools, along with some ETL prototypes coming from academic world, secondly (2) we review the modeling and design works in ETL field. Also, (3) we approach ETL maintenance issue then Finally, (4) we present and outline challenges and research opportunities around ETL processes.**

Keywords: ETL, Data warehouse, Data warehouse Population, Data warehouse Refreshment, ETL Modeling, ETL Maintenance

I. INTRODUCTION

Data Warehouse (DW) defined by Inmon [1] as "collection of integrated, subject-oriented databases designated to support the decision making process" aims to improve decision process by supplying unique access to several sources. In real world, enterprises as organizations invest in DW projects in order to enhance their activity and for measuring their performance.

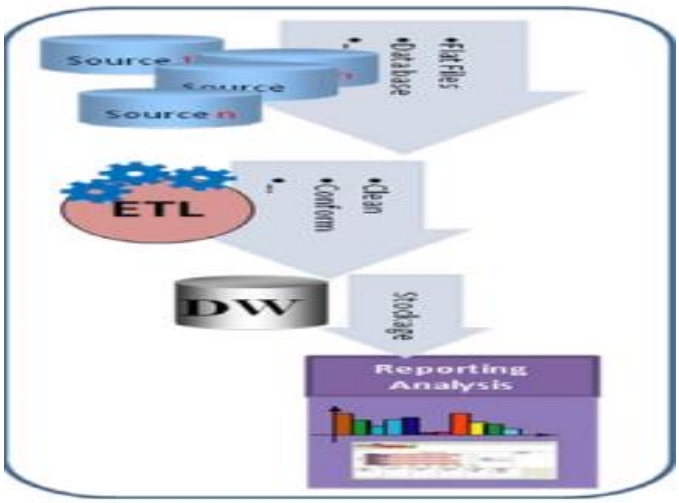
A data warehouse typically collects data from several operational or external systems (also known as the sources of the data warehouse) in order to provide its end-users with access to integrated and manageable information. In practice, this task of data collection (also known as data warehouse population) has to overcome several inherent problems, which can be shortly summarized as follows. First, since the different sources structure information in completely different schemata the need to transform the incoming source data to a common, "global" data warehouse schema that will eventually be used by end user applications for querying is imperative. Second, the data coming from the operational sources suffer from quality problems, ranging from simple misspellings in textual attributes to value inconsistencies, database constraint violations and conflicting or missing information; consequently, this kind of "noise" from the data must be removed so that end-users are provided data that are as clean, complete and truthful as possible. Third, since the information is constantly updated in the production systems that populate the warehouse, it is necessary to refresh the data warehouse
contents regularly, in order to provide the users with up-to-date information. All these problems require that the respective software processes are constructed by the data warehouse development team (either manually, or via specialized tools) and executed in appropriate time intervals Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited. for the correct and complete population of the data warehouse.

1.1DW Layers

Figure 1 illustrates the architecture of DW system. In this figure the one can note that DW system has four levels:

• **Sources**: They encompass all types of data sources. They are data provider. The two famous types are databases and flat files. Finally let note that sources are autonomous or semi autonomous.

• **ETL:** stands for Extraction Transformation and Loading. It is the integration layer in DW environment. ETL tools pull data from several sources (databases tables, flat files, ERP, internet, and so on), apply complex transformation to them. Finally in the end, data are loaded into the target which is data warehouse store in DW environment



1.1 . Figure 1: Data Warehouse Environment

• **DW:** is a central repository to save data produced by ETL layer. DW is a database including fact tables besides dimension tables. Together these tables are combined in a specific schema which may be star schema or snowflake schema. • Reporting and Analysis layer has the mission to catch end-user request and translate it to DW. Collected data are served to end-users in several formats. For example data is formatted into reports, histograms, dashboard...etc ETL is a critical component in DW environment. Indeed, it is widely recognized that building ETL processes, during DW project, are expensive regarding time and money. ETL consume up to 70% of resources [3], [5], [4], [2]. Interestingly [2] reports and analyses a set of studies proving this fact. In other side, it is well known too, that the accuracy and the correctness of data, which are parts of ETL responsibility, are key factors of the success or failure of DW projects. Given the fact expressed above, about ETL importance, the next section presents ETL missions and its responsibility.

1.2ETL Mission :As its name indicates, ETL performs three operations (called also steps) which are Extraction, Transformation and Loading.

Extraction step has the problem of acquiring data from a set of sources which may be local or distant. Logically, data sources come from operational applications, but there is an option to use external data sources for enrichment. External data source means data coming from external entities. Thus during extraction step, ETL tries to access available sources, pull out the relevant data, and reformat such data in a specified format. Finally, this step comes with the cost of sources instability besides their diversity. Finally, according to figure 2, this step is performed over source. Loading step conversely to previous step, has the problem of storing data to a set of targets. During this step, ETL loads data into targets which are fact tables and dimension in DW context. However, intermediate results will be written to temporary data stores. The main challenges of a loading step are to access available targets and to write the outcome data (transformed and integrated data) into the targets. This step can be a very time-consuming task due to indexing and partitioning techniques used in data warehouses [6]. Finally, according to figure 2, this step is performed over target.

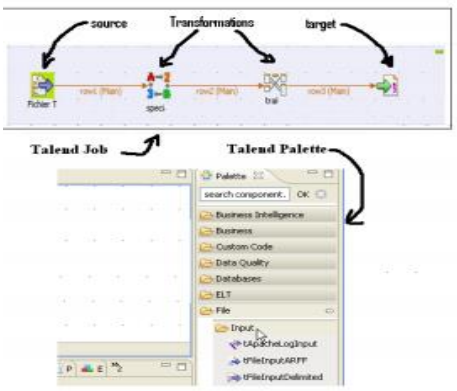


Figure 2: ETL Example and Environment with Talend Open source

Transformation step is the most laborious one where ETL adds value [3]. Indeed during this step, ETL carries out the logic of business process instanced as business rules (sometime called mapping rules) and deals with all types of conflicts (syntax, semantic conflicts … etc). This step is associated with two words: clean and conform. In one hand, cleaning data aims to fix erroneous data and to deliver clean data for end users (decisions makers). Dealing with missing data, rejecting bad data are examples of data cleaning operations. In other hand, conforming data aims to make data correct, in compatibility with other master data. Checking business rules, checking keys and lookup of referential data are example of conforming operations. At technical level and in order to perform this step, ETL should supplies a set of data transformations or operators like filter, sort, inner join, outer joins…etc. Finally this step involves flow schema management because the structure of processed data is changing and modified step by step, either by adding or removing attributes

Loading step conversely to previous step, has the problem of storing data to a set of targets. During this step, ETL loads data into targets which are fact tables and dimension in DW context. However, intermediate results will be written to temporary data stores. The main challenges of a loading step

II. ETL TOOLS AND RESEARCH PROTOTYPES

In Section 1, we present the technique of material views originally created to refresh the DW. In Section 2, we review ETL tools with examples of commercial tools as open source tools.

2.1 Commercial ETL Tools A variety of commercial tools overwhelms the ETL market which is a promising market. A study [7] conducted by TDWI, identifies a list of indicators and criteria for their comparison and evaluation. Each commercial ETL tool adopts its own notation and its won formalism. Consequently, metadata between these tools are not exchangeable. In contrast, among their commonalities, they all offer a graphical language for the implementation of ETL processes. We distinguish two subfamilies in the commercial ETL field. On the one hand, there is subfamily of payable ETL DataStage [8] and Informatica [9]. On the other hand, the second subfamily of commercial ETL comes with no charge. In fact, they are free under certain conditions. Indeed, despite the ETL licenses are expensive, major DBMS (Database Management System) editors like Oracle and Microsoft, offer there ETL solution freely for each DBMS license purchased. In other words, ETL solution is included in DBMS package license. In the following we present an example of each subfamily. DataStage [8] is the ETL solution of IBM editor. Its basic element for data manipulation is called "stage." Thus, for this tool an ETL process is a combination of "stages." Thus we speak about transformation stages and stages for extracting and loading data (called connectors since release 8) which are interconnected via links. The IBM solution DataStage

provides two other tools: Manager and Administrator. They are designed, respectively, for supervising the execution of ETL processes and for dealing with ETL project configuration. It should also be noted that IBM offers two versions in its ETL solution: DataSatge SERVER version and DataSatge PX version. This last version has the advantage to manage the partitioning in data processing. Finally, DataStage generates OSH code from ETL job built with stages placement. SSIS stands for Sql Server Integration Services. This is the ETL solution of Microsoft editor [10]. As mentioned above, SSIS is free with any DBMS SQL SERVER license which includes two extra tools that are SSRS and SSAS (respectively for reporting and OLAP analysis). The atomic element or the basic element in SSIS is called a "task". Thus, for this tool an ETL process is a combination of tasks. More precisely, SSIS imposes two levels of tasks combination. The first level is called "Flow Control" and the second level controlled by the first, is called "Data flow." Indeed, the first level is dedicated to prepare the execution environment (deletion, control, moving files, etc ...) and supplies tasks for this purpose. The second level (data flow) which is a particular task of the first level performs classical ETL mission. Thus, the Data-Flow task offers various tasks for data extraction, transformation and loading. In conclusion of this section, we have presented commercial ETL, along with some examples of theme. In next section, we present another category of ETL, open sources tools.

2.2 Open Source :ETL Some open source tools are leaders in their area of interest; for example, Linux in operating system area and Apache server in web servers' area. But the trend is not the same for open source business intelligence (BI) tools. They are less used in the industrial world

III. ETL MODELING AND DESIGN

Research in data warehouse field is dominated by the DW design and DW modeling. ETL field is not an exception to this rule. Indeed, in the literature one can note several research efforts that treat DW population performed by ETL processes. (ETL) are areas with high added value labeled costly and risky. In addition, software engineering requires that any project is doomed to switch to maintenance mode. For these reasons, it is essential to overcome the ETL modeling phase with elegance in order to produce simple models and understandable.

Finally, as noted by Booch et al in [19], we model a system in order to: • Express its structure and behavior. • Understand the system. • View and control the system. • Manage the risk.

. The authors based their argument on the following two points. At the beginning of a BI project, the designer needs to:

1. Analyze the structure and the content of sources.
2. Define mapping rules between sources and targets.

The proposed model, based on meta model, provides a graphical notation to meet this need. Also, a range of activities commonly used by the designer is introduced.

3.an extra node (element of the model).

Ignoring which source to prioritize to extract data, the model introduces candidate relationship to designate all sources likely to participate in DW population. The selected source is denoted active relationship. The authors complement their model via an extensive paper [21] by proposing a method for the design of ETL processes. Indeed, in this work, the authors expose a method to establish a mapping between the sources and targets of ETL process.

This method is spread over four steps:

1. Identification of sources
2. Distinction between candidates' sources and active sources.
3. Attributes mapping.
4. Annotation of diagram (conceptual model) with execution constraints.

Works around this model are reinforced by an attempt to transition from conceptual model to logical model. In addition, this work proposes an algorithm that groups transformations and controls (at conceptual level) into stages that are logical activities. Finally, a semi-automatic method determining the order execution of logical activities is defined too. Another work around ETL presents KANTARA, a framework for modeling and managing ETL processes [4]. This work exposes different participants that an ETL project involves particularly designer and developer. After the analysis of interaction between main participants, authors conclude that designer needs helpful tool, which will makes easy the design and maintenance of ETL processes. In response, authors introduce new graphical notation based on a meta-model. It consists mainly on three core components which are, Extract, Transform and Load components. These components are chosen according to ETL steps. Besides, each component manages a set of specific meta-data close to its associate step. This work is consolidated with another work presenting a method for modeling and organizing ETL processes [23]. Authors start by showing functional modules that should be distinguished in each ETL process. This leads to distinguish several modules, especially Reject Management module for which a set of metadata to manage are defined. The proposed approach takes five inputs (namely mapping rules, conforming rules, cleaning rules, and specific rules) and produces a conceptual model of an ETL processes using a graphical notation presented previously. 3.2 UML Based Works In 2003 Trujillo [24] proposes a new approach, UML based for the design of ETL processes. Finding that the model of Vassiliadis [20] is based on ad-hoc method, the authors attempt to simplify their model and to base it on a standard tool. In this model, an ETL process is considered as class diagram. More precisely, a basic ETL activity which can be seen as a component is associated with a UML class and the interconnection between classes is defined by UML dependencies. Finally, the authors have decided to restrict their model to ten types of popular ETL activities such as Aggregation, Conversion, Filter and

Join, which in turn are associated to graphical icons. In 2009 (DOLAP 2009), Munoz et al modernize this model through a new publication [25] dealing with the automatic generation of code for ETL processes from their conceptual models. In fact, the authors have presented a solution oriented MDA. It is structured as follows: • For PIM (Platform Independent Model) layer, which corresponds to conceptual level, ETL models are designed using UML formalism, more precisely the result of the previous effort. • For PSM (Platform Specific Model) layer which corresponds to the logical level, the platform chosen is Oracle platform. • For the Transformation layer that allows the transition from PIM model to PSM model is made with QVT (Query View Transformation) language. These transformations can bind PIM meta-model elements to PSM meta-model elements. Another research team presents in [26] another research effort about ETL and UML based. But this work is restricted to extraction phase omitting transformations and loading phases. Thus, this work presents an approach object-oriented for modeling extraction phase of an ETL process using UML 2. To this end, authors present and illustrate the mechanism of this phase as three diagrams. These diagrams are class diagram, sequence diagram and use case diagram of extraction phase. Finally, six classes which are data staging area, data sources, wrappers, monitors, integrator and source identifier are shown. These classes are used and in above diagrams.

IV. MAINTENANCE OF ETL PROCESSES

ETL process can be subject of changes for several reasons. For instance, data sources changes, new requirements, fixing bugs…etc. When changes happen, analyzing the impact of change is mandatory to avoid errors and mitigate the risk of breaking existent treatments. Generally, change is neglected although it is a fundamental aspect of information systems and database [31]. Often, the focus is on building and running systems. Less attention is paid to the way of making easy the management of change in systems [32]. As a consequence, without a helpful tool and an effective approach for change management, the cost of maintenance task will be high. Particularly for ETL processes, previously judged expensive and costly. Research community catches this need and supplies, in response, few solutions. The one can starts with a general entry point to change issue in [31] that is a report on evolutions and data changes management. Indeed, authors summarize the problems associated with this issue as well as a categorization of these problems. Finally, the authors discuss the change issue according to six aspects which are: What, Why, Where, When, Who and How. Regarding data warehouse layer, change can occur at two levels, either in schemas or in data stored from the first load of data warehouse. Managing data evolution, contrarily to schema evolution, over time is a basic mission of DW. Consequently, research efforts in DW evolution and changes are oriented to schema versioning. In this perspective, authors present in [33] an approach to schema versioning in DW. Based on graph formalism; they represent schema parts as a graph and define an algebra to derive new schemas of DW, given a change event. The formulation of queries invoking multiple schema versions is sketched. Same authors rework their proposal in [34] by investigating more data migration. Finally, X-Time [35] is a prototype resulting from these efforts. Using ETL terminology, above previous research efforts focus on the target unlike the proposal of [36] which focuses on changes in the sources. In this work, the authors consider the ETL process as a set of activities (a kind of component). Thus, they represent the ETL parts as graphs which are annotated (by the designer) with actions to perform when a change event occurs. An algorithm to rehabilitate the graph, given a change event in sources is provided too. However, this approach is difficult to implement, because of enormous amount of additional information required in nontrivial cases [37]. The authors extend their work in [38] by detailing the above algorithm for graph adaptation. The architecture of prototype solution has been introduced too. It has a modular architecture centralized around the component Evolution Manager. This prototype is called HECACTUS [39] and aims to enable the regulation of schema evolution of relational database. In other words, this approach is does not take in account other kinds of data stores or sources like flat files. Another approach dealing with change management in ETL is available in [32].In this paper, authors present a matrices based approach for handling impact of change analysis in ETL processes. Indeed, ETL parts are represented as matrices and a new matrix called K matrix is derived by applying multiplication operations. Also authors expose how this K matrix summarizes the relationship between the input fields and the output fields and how it synthesizes the attributes dependency. Particularly, the K matrix makes possible to know "which attributes are involved in the population of a certain attribute" and which attributes are the "customers" of a given one. In addition, an algorithm to detect affected part of ETL process when a change deletion event occurs, either in sources or targets or inside ETL, have been presented. Finally, this matrices based approach constitutes a sub module of whole solution that is a framework called KANTARA. These proposals dealing with change management in ETL are interesting and offer a solution to detect changes impact on ETL processes. However change absorption is not addressed. In next section, we present research works taking in account performance aspect.

V. OPTIMIZATION AND INCREMENTAL ETL

ETL feeds DW with data. In this mechanism and depending on the context, ETL performance will be critical fact. For example, [2] reports a study for mobile network traffic data where a tight time window is allowed for ETL to perform its missions (4 hours to processes about 2 TB of data where the main target fact table contains about 3 billion records). In situations like one described above, ETL optimization is much appreciated. Concerning this issue at research level, unfortunately, works and proposals are little. The first work dealing with this issue treats the

logical optimization of ETL processes [40]. In this proposal, authors model the problem of ETL optimization at logical level, as state space search problem. In particular an ETL process is conceived as a state and a state space is generated via a set of defined transitions. The approach is independent of any model cost. However the discrimination criterion for choosing the optimal state is based on total cost. The total cost of a state is obtained by summarizing the costs of all its activities. Another solution to achieve performance consists of extracting and processing only modified or new data. This is called incremental mode of ETL processes. More precisely, this style of ETL is suitable to contexts where the request of fresh data, from end users, is very pressing. By definition incremental ETL has two challenges. They are: 1. To detect modified data or new data at sources level. 2. Integrate data of previous step.

Incremental ETL is associated to near real time ETL. In one hand, the historical and business contexts of this category of ETL are sketched in [3] where authors motivate and explain why this new category of ETL. In other hand, authors of [41] formalize the problem of incremental ETL under the following assumption: for a given target, there are two types of ETL jobs feeding this target. Namely: initial ETL job, for first load and second ETL job for, recurrent load in incremental mode. Thus, this work presents an approach to derive incremental ETL jobs (second type) from the first one (initial load) based on equational reasoning. Transformation rules for this purpose are defined too. As input, they take an ETL job as an expression E described according to a defined grammar G. In output, four ETL expression Eins, Edel, Eun and Euo, are produced dealing with change events occurring at sources (insertion, deletion, update…etc). Another work dealing with incremental ETL is available in [42] where authors present an approach based on existing method of incremental maintenance of materialized views to get automatically incremental ETL processes from existing ones. Approaches presented above, both require the existence of the initial ETL processes to transform them into incremental mode. Therefore they are suitable for existing ETL projects wishing to migrate and take profit from incremental mode. Thus they are not suitable for new ETL projects starting from scratch. All previous sections review the literature in ETL field. In next section we present main research opportunities in ETL processes.

VI. RESEARCH OPPORTUNITIES

In our opinion, research opportunities and challenges around ETL exist and are promising. As in other research fields, design and modeling still dominate others research issues. Obviously all issues reviewed previously are open. Out of these challenges, we summarize in what follows main pressing issues: • Standardizing models: many conceptual models enrich the ETL design field. However no proposal becomes a standard neither widely accepted by research community like multi dimensional modeling in data warehouse area. Therefore proposals and works in this perspective are desirable and hopeful. • Mapping Language: mapping rules are an important delivery in ETL design. We consider that in order to use efficiently this delivery, it is desirable to express these rules in a standard language compatible with ETL constrains and aspects.

• Big Data and ETL:
Big data technologies arrive with exciting research opportunities. Particularly, performance issue seems solvable with this novelty. Thus, works and proposals using these technologies and taking in account ETL specificities; like partitioning, data transformation operations, orchestration…etc; are desirable. • Testing: Tests are fundamentals aspects of software engineering. In spite of this importance, and regarding ETL, they are neglected. Thus, an automatic or even a semi automatic approach for validating or getting data for tests is very hopeful.

• Unstructured data and Meta data:
theses two topics are not specific to ETL processes. They are common issues to data integration area. Thus, they are open challenges which can be addressed in ETL context.

• Change absorption:
As we said in previous sections, only few approaches handling changes impacts on ETL exist. But it is more challenging to automatically or semi automatically absorbing changes once they are detected. In other words, an approach is needed to adapt running ETL jobs according to changes occurring either in sources, targets or in business rule (transformation rule).

VII. CONCLUSION

ETL(Extraction-Transformation-Loading) is the integration layer in data warehouse (DW) system. ETL is known with two tags: complexity and cost. Indeed, it is widely recognized that building ETL processes is expensive regarding time, money and effort. It consumes up to 70% of resources. Due its importance, this paper focused on ETL, the backstage of DW, and presents the research efforts and opportunities in connection with these processes. Therefore, in current survey, firstly we give a review on open source and commercial ETL tools, along with some ETL prototypes coming from academic world. Namely SIRIUS, ARKTOS, PYGMATEL, DWPP. Also, Talend Open Studio and Microsoft ETL solution (SSIS), respectively an open source and commercial tool, were taken as examples for explanation and illustration. Secondly we cover modeling and design works in ETL field. Thus several works using different formalism or technologies like UML and web technologies, are reviewed. Then this survey continues by approaching ETL maintenance issue. Namely, after problem definition, we review works dealing with changes in ETL processes using either graph formalism or matrices formalism. Before conclusion, we have given an illustration of performance issue along review of some works dealing with this issue, particularly, ETL optimization and incremental ETL. Finally, this surveys ends with presentation of main challenges and research opportunities around ETL

processes. At the end of this survey and as a conclusion, we believe that research in ETL area is not dead but it is alive. Each issue addressed above is open to review and investigation.

REFRENCES

[1]D. Skoutas and A. Simitsis. “Designing ETL processes using semantic web technologies”. Proceedings of DOLAP06, 2006.

[2] D. Skoutas and A. Simitsis. “Ontology-based conceptual design of ETL processes for both structured and semi-structured data”, International Journal on Semantic Web and Information Systems, 2007.

[3] Z, ElAkkaoui and E.Zimanyi, “Defining ETL Workflows using BPMN and BPEL”. Proceedings of DOLAP09, 2009, pp 41-48.

[4] Marko Niinimaki, Tapio Niemi, "An ETL Process for OLAP Using RDF/OWL Ontologies", Journal on semantic web and information Systems, Vol 3, No 4, 2007.

[5] J.F. Roddick et al, "Evolution and Change in Data Management - Issues and Directions", SIGMOD Record 29, Vol. 29, 2000, pp 21-25.

[6] A. Kabiri, F. Wadjinny and D. Chiadmi, "Towards a Matrix Based Approach for Analyzing the Impact of Change on ETL Processes", IJCSI international journal, Volume 8, issue 4, July 2011.

[7] M. Golfarelli, J. Lechtenborger, S. Rizzi and G. Vossen, "Schema Versioning in Data Warehouses", ER Workshops, LNCS 3289, 2004, pp 415–428.

[8] M. Golfarelli, J. Lechtenborger, S. Rizzi and G. Vossen, "Schema versioning in data warehouses: Enabling cross version querying via schema augmentation", Data and Knowledge Engineering, 2006, pp 435–459.

[9] S. Rizzi and M Golfarelli, "X-time: Schema versioning and cross-version querying in data warehouses", International Conference on Data Engineering (ICDE), 2007, pp.1471–1472.

[10] G. Papastefanatos, P. Vassiliadis, A. Simitsis and Y. Vassiliou, "What-If Analysis for Data Warehouse Evolution", Proceedings of DaWaK conference, LNCS 4654, 2007, pp 23–33. [11] Dolnik A. (2009). “ETL evolution from data sources to data warehouse using mediator data storage”, MANAGING EVOLUTION OF DATA WAREHOUSES MEDWa Workshop, 2009

[12] G. Papastefanatos, P. Vassiliadis, A. Simitsis and Y. Vassiliou, "Policy-Regulated Management of ETL Evolution", Journal on Data Semantics XIII, LNCS 5530, 2009, pp 146–176. [13] G. Papastefanatos, P. Vassiliadis, A. Simitsis and Y. Vassiliou, " HECATAEUS: Regulating Schema Evolution. Data Engineering", International Conference on Data Engineering (ICDE), 2010, pp 1181-1184

[14] A. Simitsis, P. Vassiliadis, and T. Sellis. "Logical optimization of ETL workflows", Proceedings of International Conference on Data Engineering (ICDE), 2010, pp 1181-1184. [15] T. Jorg and S. Debloch. “Formalizing ETL Jobs for Incremental Loading of Data Warehouses”, Proceedings of der 12, 2009.

[16] X. Zhang, W. Sun, W. Wang, Y. Feng, and B. Shi, “Generating Incremental ETL Processes Automatically”, Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06), 2006