Volume 3, No. 10, October 2012
Journal of Global Research in Computer Science
ISSN-2229-371X
REVIEW ARTICLE
Available Online at www.jgrcs.info

AN OVERVIEW ON PHYSICAL IMPLEMENTATION OF SECURE ETL WORKFLOW

Nitin Anand
Research scholar,
Deptt. of Computer Science,
Ambedkar Institute of Advanced Communication Technologies & Research,
New Delhi, India
proudtobeanindiannitin@gmail.com

Abstract: Extraction-transformation-loading (ETL) tools are pieces of software responsible for the extraction of data from several sources, their cleansing, customization and insertion into a data warehouse. The main focus is on logical optimization of ETL processes. We consider each ETL workflow as a state and fabricate the state space through a set of correct state transition.

Keywords: Data warehousing; Extraction-transformation-loading (ETL), Data Extraction; Security

INTRODUCTION

The preparation of data before their actual loading in the warehouse for further querying is necessary due to quality problems, incompatible schemata, and unnecessary parts of source data not relevant for the purposes of the warehouse. The category of tools that are responsible for this task is generally called Extraction- Transformation- Loading (ETL) tools.
The functionality of these tools can be coarsely summarized in the following prominent tasks, which include:

1. The identification of relevant information at the source side.
2. The extraction of this information,
3. The customization and integration of the information and integration of the information coming from multiple sources [1].
4. The cleaning of the resulting data set on the basis of database and business rules, and
5. The propagation of the data to the data warehouse and/or data marts.

A RATIONALE FOR THE TAXONOMY

An ETL workflow can be seen as a directed graph as shown in Figure 1. The nodes of this graph are activities and recordsets. The edges of the graph are provider relationships that combine activities and recordsets

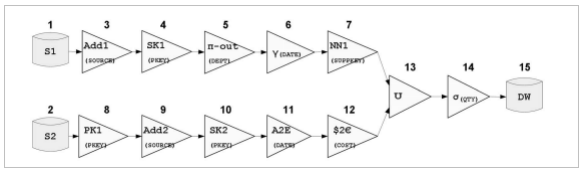

Fig 1 ETL workflow as a directed graph

The edges of the graph are provider relationships that combine activities and recordsets [2]. Following the common practice, we envisage ETL activities to be combined in a workflow.
Therefore, we do not assume that the output of a certain activity will be necessarily directed towards a recordset, but rather, that the recipient of this data can be either another activity or a recordset.
In Figure 2 [10]

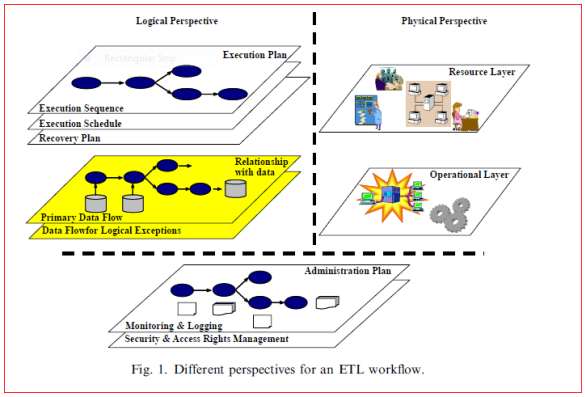

Fig. 2 Different perspectives for an ETL workflow

We follow a multi-perspective approach that enables to separate these parameters and study them in a principled approach. We are mainly interested in the design and administration parts of the lifecycle of the overall ETL process, and we depict them at the upper and lower part of Fig. 2, respectively. At the top of Fig. 2, we are mainly concerned with the static design artifacts for a workflow environment. We will follow a traditional approach and group the design artifacts into physical, with each category comprising its own perspective. We depict the logical perspective on the left-hand side of Fig. 2, and the physical perspective on the right-hand side. At the logical perspective, we classify the design artifacts that give an abstract description of the workflow environment. First, the designer is responsible for defining an execution plan for the scenario. The definition of an execution plan can be seen from various perspectives. The execution sequence involves

the specification of which activity runs first, second, and so on, which activities run in parallel, or when a semaphore is defined so that several activities are synchronized at a rendezvous point. ETL activities normally run in batch, so the designer needs to specify an execution schedule, i.e., the time points or events that trigger the execution of the scenario as a whole. Finally, due to system crashes, it is imperative that there exists a recovery plan, specifying the sequence of steps to be taken in the case of failure for a certain activity (e.g., retry to execute the activity, or undo any intermediate results produced so far). On the right-hand side of Fig. 2, we can also see the physical perspective, involving the registration of the actual entities that exist in the real world. We will reuse the terminology of [3] for the physical perspective. The resource layer comprises the definition of roles (human or software) that are responsible for executing the activities of the workflow. The operational layer, at the same time, comprises the software modules that implement the design

2.1 ETL activities

In this section, we discuss the different types of ETL activities based on the interrelationship of their input and output. We begin with a high-level classification with respect to input (e.g., unary, binary, n-ary) or output (e.g., routers, filters) schemata and within each such category, we discuss the mappings between input and output tuples.[4]

Unary activities. These activities take the data from the input schema, perform a transformation or cleaning operation to them and direct the processed data to the output. Unary activities have
exactly one input and one output schemata.[11]
Based on that, the most interesting values for the cardinality of mapping in unary activities are the following:
− ***1:1***, an input tuple is mapped to exactly one output tuple.
− ***1:M***, an input tuple is mapped to more than one output tuples.
− ***N:1***, more than one input tuples are combined to produce exactly one output tuple. Observe that this relationship introduces
a set of classes among input tuples: all tuples belonging to the same class correspond to the same output tuple. If each input tuple corresponds to at most one class, then these are equivalence classes.
− ***0:M,*** some functions or constant values are employed to produce one or more output tuples.
− ***N:M***, the relationship among a certain group of input tuples
and a certain group of output tuples cannot be simplified to one of the above categories.

N-ary activities. N-ary activities combine information from multiple inputs and populate one output scheme. Different tools provide different implementations regarding the input schemata. An n-ary activity (e.g., a multi-way join) may have *n* inputs or can be implemented as a series of binary activities. Although our analysis covers both, for the sake of presentation we discuss the case of binary activities, which involves the following configurations:
− ***Primary flow***. These are binary activities where one of their inputs is a part of a primary flow that probes a second input to test whether their values qualify for further propagation. [4,5]. An example primary flow – see Figure 3– may contain a series of the binary

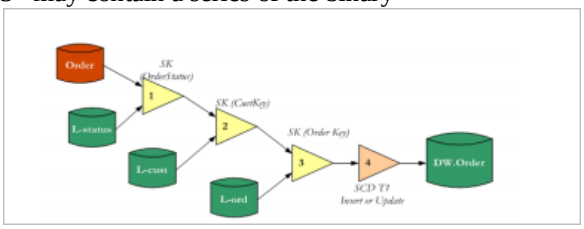

Fig 3 Primary flow in ETL

surrogate_key operators, which replace the production keys of the incoming data (this data
would be the first input) with surrogate keys found in lookup tables (these tables would constitute the second input). The primary flow may employ the vocabulary for unary activities (in fact, most of these binary activities can be classified under the 1:1 category.)
Here, we do not focus on how the input values are combined, but rather, how the tuples are related to each other.
In the case of primary flow, for each outgoing tuple there is exactly one input tuple in the primary flow that corresponds to it (but not vice versa). At the same time, typically there is at most one corresponding tuple in any non-primary input schema, for any tuple of the primary flow.

3. ALGORITHM FOR SIMULATION

The main procedure for secure data extraction process [6,7] is as follows.

// Identifying the sources and creating the source list.
This is done by the methods of Source Identifier class
1. Identify the list of clients attached to the server
2. Find the type of the databases by pinging to that client
3. Set the properties for the source
4. If it is a new source add to the data source list
// Establishing the connection and extracting data.
This is done by methods of Wrapper class
5. Check the type of the data source
6. Using appropriate drivers establish the connection
7. Map the data source and data staging area schemas
8. Extract the data
// Loading of extracted data into data staging area. Integrator class does this.
9. Establish connection with data staging area
10. Install the data into data staging area
// Modification / updation of Data Staging Area (DSA).
Integrator updates DSA with the help of Monitor.
11. Identify the changes in the data sources and Inform to the Integrator
12. Update DSA

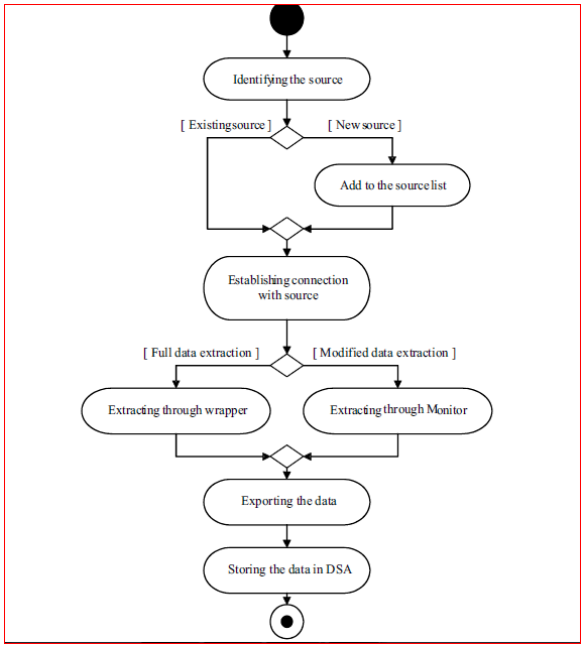

Figure 3: Activity diagram for data extraction

4. CONCLUSION AND FUTUREWORK

In a recent study [8], the authors report that
due to the diversity and heterogeneity of data sources, ETL is unlikely to become an open commodity market. This paper describes the simulation model of Secure Data Extraction in ETL processes.

This architecture gives us flexibility of adding various types of information sources, which ultimately helps in storing the data into the Data Staging Area. Since quality plays an important role in developing software products, I have presented functional requirements along with non-functional requirement i.e., security requirements.. This approach is better compared to existing systems. In [9], the authors report on their data warehouse population system. The architecture of the system is discussed in the paper, with particular interest (a) in a ‘‘shared data area’’, which is an in-memory area for data transformations, with a specialized area for rapid access to lookup tables and (b) the pipelining of the ETL processes.

The future work may include to dealing with other non-functional requirements like reliability, performance etc. In this paper, we have focused on the data-centric part of logical design of the ETL scenario of a data warehouse. First, we have defined a formal logical metamodel as a logical abstraction of ETL processes. The data stores, activities and their constituent parts, as well as the provider relationships that map data producers to data consumers have formally been defined.

Then, we have provided a reusability framework that complements the genericity of the aforementioned metamodel. Practically, this is achieved from an extensible set of specializations of the entities of the metamodel layer, specifically tailored for the most frequent elements of ETL scenarios, which we call template activities. In the context of template materialization, we have dealt with specific language issues, in terms of the mechanics of template instantiation to concrete activities.

REFERENCES

.

[1] J. Adzic and V. Fiore, “Data Warehouse Population Platform,” Proc. Fifth Int’l Workshop Design and Management of Data Warehouses,
2003.

[2] A. Simitsis, P. Vassiliadis, and T. Sellis, “Optimizing ETL Processes in Data Warehouses,” Proc. 21st IEEE Int’l Conf. Data Eng.,pp. 564-575, 2005.

[3] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, A.P. Barros. Workflow Patterns, BETA Working Paper Series, WP 47, Eindhoven University of Technology, Eindhoven, 2000, available at the Workflow Patterns website, at tmithttp://www.tm.tue.nl/research/patterns/documentation.htm.

[4]. A. Simitsis, P. Vassiliadis, U. Dayal, A. Karagiannis, V. Tziovara. Benchmarking ETL Workflows. In TPC-TC, 2009.

[5]. V. Tziovara, P. Vassiliadis, A. Simitsis. Deciding the physical imple-mentation of ETL workflows.In DOLAP, pp. 49-56, 2007.

[6] Guttorm Sindre, Andreas L. Opdahl, Eliciting Security Requirements with Misuse Cases, Proc. Requirements Engineering, 10(1), Springer-Verlag London Ltd, pp. 34–44, January 2005.

[7] G. Sindre and A.L. Opdahl, Templates for Misuse Case Description. In Proc. of the 7th International Workshop on Requirements Engineering, Foundation for Software Quality, June 2001.

[8] Giga Information Group. Market Overview Update:ETL. Technical Report RPA-032002-00021, March 2002.

[9] J. Adzic, V. Fiore, Data Warehouse Population Platform,in: Proceedings of the Fifth International Workshop on the Design and Management of Data Warehouses (DMDW’03), Berlin, Germany, September 2003.

[10] Panos Vassiliadis, Alkis Simitsis, Panos Georgantas, Manolis Terrovitis and Spiros Skiadopoulos”A Generic and customizable framework for the design of ETL scenarios”,2002.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.7971&rep=rep1&type=pdf

[11] Panos Vassiliadis Alkis Simitsis , Eftychia Baikousi “ A Taxonomy of ETL Activities” , Nov 2009 [http://cs.uoi.gr/~pvassil/publications/2009_DOLAP_ETL/DOLAP_2009_ETL.pdf

