Revised Framework for ETL Workflow Management for Efficient Business Decision-Making

Saifur Rehman Malik, Azra Shamim, Zanib Bibi, Sajid Ullah Khan, and Shabir Ahmad Gorsi

***Abstract*—Business decision-making is not a simple task. There are many reasons for that but the main reason is data comes from heterogeneous operational sources of an organization. Therefore, it is difficult to organize and maintain especially if a huge volume of data is involved. A data warehouse is helpful in this regard as it can assists in business decision-making. Data collection and loading it into a data warehouse is difficult job because data sources are not in consistent form. This job usually consists of three main processes that involve extraction, transformation and loading. To extract the data from different sources, then transform it into a unified format and consequently load it into the warehouse, ETL (Extract, transform and load) tools are required. Nowadays, the majority of ETL tools organize workflow. An ETL workflow can be considered as a group of ETL jobs with dependencies between them. In this research paper a revised ETL workflow management framework which is based upon different considerations is proposed. These considerations along with the addition of the components in the workflow scheduling layer would help in making more effective and quality business decisions.**

***Index Terms*—Data warehouse, ETL, ETL Tools, workflow management.**

I. Introduction

Sales orders, inventory control, accounts and customers information etc. are different business areas of an organization. Many operational systems are working separately to automate these business areas. These operational systems are capable of generating and analyzing data that corresponds to their own domain. Moreover, these operational systems can use different data sources like web services, OLTP (Online Transaction Processing), clients / server systems and other software systems at application layer. These sources continuously generate important data. To gain competitive advantage, organizations must utilize this data effectively and efficiently to support business decision-making.

As data come from different operational sources, a problem arises that data is in different formats because these operational systems designed specifically for a separate business area. The effective and efficient use of this data for business decision-making requires that the data must be in unified format. Data warehouse is the solution for this problem. This is because data warehouse is capable of integrating the data which is coming from various heterogeneous operational sources in a consistent form [1].

Manuscript received September 30, 2012; revised December 12, 2012.

Azra Shamim is with the Faculty of Computer Science and Information Technology University Malaya (e-mail: azra.majeed864@yahoo.com).

Integration of data from different source systems can be done through ETL process which includes extracting data from different heterogeneous sources, transforming it, and then loading it into the data warehouse. ETL operations can be performed by using ETL tools. These tools organize such operations as a workflow. An ETL workflow is used to capture the flow of data from the various sources to a data warehouse [2].

The rest of the paper is organized into different sections. Section 2 gives an overview of data warehouse, data mining, knowledge discovery and ETL process. Section 3 presents the proposed revised ETL workflow management framework and Section 4 concludes the research work.

II. Literature Review

A. Data Warehouse, Mining and Knowledge Discovery

"Data warehouse is a subject-oriented, integrated, time variant, non volatile collection of data in support of management decisions" [3]. "Data warehouse is a set of materialized views over data sources" [4], [5]. In data warehouse relevant data from different operational systems is extracted, transformed and integrated in a unified format into an enterprise data warehouse through ETL process. Raghu et. al. defined data mining as "Data mining is the exploration and analysis of large quantities of data in order to discover valid, novel, potentially useful, and ultimately understandable patterns in data" [6]. Data mining is the process through which information that is actionable and valid is extracted from large databases [7]. Knowledge discovery is defined as "the non-trivial extraction of implicit, unknown, and potentially useful information from data" [8], [9]. Data mining or knowledge discovery in databases used tools and techniques for exploration of databases to extract relevant and interesting hidden relationships between variables [8], [10]. Different data mining techniques are applied to extract valuable and hidden information. The result of data mining is further carefully analyzed in knowledge discovery process to provide the user valid, accurate and actionable information.

B. ETL Process

The integration of data that is coming from various sources is achieved through the use of an ETL process. This process is responsible for extraction of the data which is stored in heterogeneous data sources, the transformation of extracted data and loading it into a data warehouse. Transformation is the process of converting data into a unified form, and load is the process of loading data in to a target system. According to Simitsis et. al. [11] the backstage of the data warehouse

DOI: 10.7763/IJCTE.2013.V5.734

architecture consists of Extract- Transform-Load processes. These processes are discussed below:

1) Extract

Extraction is the process of extracting data from various heterogeneous data sources. The source systems are typically online transaction processing, web services and other software applications e.g. one source system for a sales analysis, data warehouse may be an order entry system that keeps the record all of the current orders.

2) Transform

Transformation process transforms the extracted data into a consistent form by applying a series of rules or functions to derive the data to be loaded to the end target system. If source data is already in a form that is compatible with the target system then it does not require any transformation. In some cases, only simple transformation is need, while in other cases, very complex transformations may be need to meet the business and technical needs of the target system [12].

3) Load

The load process loads the data into the end target system, which may be a data warehouse or a data mart. This phase is totally depends on the requirements of the organization. Some data warehouses might weekly overwrite existing information with cumulative, updated data, while other might add new data in a historized form, e.g. hourly. The timing and scope to replace or append are strategic design choices dependent on the time available and the business needs [12]. Appropriate design and maintenance of the ETL process are key factors in the success of a data warehouse project [13], [14]. Designing an ETL process is extremely complex, prone to failure and time consuming [15], [16]. Eckerson et. al. [17] reported that the cost of an ETL process and data cleaning tools are estimated to be at least one third of the total efforts and expenses of a data warehouse project. Simitsis et. al. [11] in their paper stated that ETL design and implementation constitutes 70% of the effort in data warehousing projects, and is a time-consuming, labor intensive service. Vassiliadis et. al. presented a conceptual model for the ETL process in [15]. Simitsis et.al in [18] enhanced this conceptual model by establishing a methodology for conceptual modeling. Li et al. in [19] provide a model that focuses on formal definitions of the ETL process. Munoz et. al. in [20] presented a model driven approach for the automatic code generation of the ETL process. In this approach the Model Driven Architecture (MDA) is introduced in order to reduce the design time and cost of ETL process. Trujillo et. al. in [21] proposed an approach based on the use of UML class diagrams, which allows the designer to decompose complex the ETL process into a set of simple processes. Simitsis et. al. proposed an approach for mapping conceptual design into logical design in [22]. Vassiliadis et. al. discussed a framework for the design of ETL scenarios in [23].

C. ETL Workflow

ETL workflows are usually data centric in nature. These work flows are responsible for transferring, cleaning, and then loading data from their respective sources into a data warehouse. In an ETL workflow, there are different jobs, their dependencies and time constraints. Each job runs in separate environment according to its requirement [24].

III. Revised Framework for ETL Workflow Management

In this section a revised framework is presented for effective workflow management that is the enhancement of authors work presented in [25] in which different important considerations about ETL workflow management was mentioned. These important considerations i.e. the availability of sources, availability of destination, dependencies among ETL jobs, duration of ETL jobs, upper bound, priority of a ETL job are the main point of focus in this paper. There are three main layers of our revised framework which are 1) application layer 2) workflow scheduling layer and 3) workflow execution layer shown below in Fig. 1. These layers make up the overall functionality of the system. In this paper, we are focusing on the Workflow Scheduling Layer. The functionality of this layer is divided into several components. These components work in parallel to augment the performance of this layer. It will ultimately have some effects on the overall framework. The aim is to increase the quality of the output so that effective decision making can be made. Further details about the framework and its layers are discussed below:

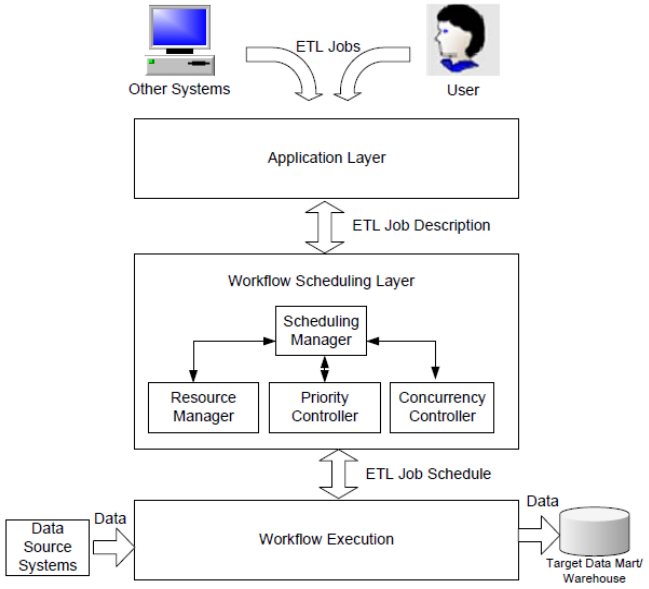

Fig. 1. Proposed framework for workflow management.

A. Application Layer

The top most layer of the framework is the application layer. Application layer is responsible for taking ETL jobs from the user or from other systems and passed them to the workflow scheduling layer. The proposed framework can take ETL jobs in two ways. First way is that the user inserts ETL jobs by itself and the second way is that query in the form of different ETL jobs are generated by another system connected to the framework. Every ETL job contains a job description which contains information like the required number of resources, target availability, priority, expected amount of time needed for the completion of the job, upper bound and prerequisite jobs. After taking job descriptions, these jobs are passed to the next layer which is workflow

scheduling.

B. Workflow Scheduling Layer

This layer is the main point of focus in this paper. The workflow scheduling layer is responsible for taking the ETL jobs and sends these jobs to the execution layer for the execution. The workflow scheduling layer decides how to commit resources among various different possible tasks in an effective and efficient way [25]. The workflow scheduling layer analyzes all the jobs according to their job description, parameters and dependencies. Then the layer passes higher priority job to the workflow execution layer to carry out execution. One of the examples of such an algorithm is "critical path". This algorithm is capable of analyzing and assigning job priorities and finding job concurrencies. In addition to it, many other scheduling algorithms can be applied based upon different considerations. One of these consideration in particular is "shortest job first" that can be identified by using the duration of an ETL job consideration. Software like business process management engine can be helpful in this regard. The scheduling layer consists of following components:

- *Resource Manager*: It checks the availability of resources which is an important consideration during the ETL process. It also assigns resources to different ETL job that are running concurrently.
- *Priority Controller:* It assigns the priorities to incoming ETL job based on the job description provide by the application layer.
- *Scheduling Manager:* It gets ETL jobs and their description from the application layer. It schedules ETL jobs on the basis of priorities, resources, dependencies provided by priority controller, resource manager and concurrency controller respectively. It provides ETL job schedule to the workflow execution layer for execution.
- *Concurrency Controller:* It checks the dependencies among different jobs and identifies which ETL jobs can be run concurrently.

C. Workflow Executions Layer

The Workflow Execution Layer receives jobs from the workflow scheduling layer to start the execution. When a job is received, its description is checked; then it is executed on the basis of its description. When the layer starts executing the job it changes its status as "In-progress". After the completion of job it updates the status of job as "Completed". Workflow execution can be implemented on more than one server. Normally, large volume of data results in greater number of jobs to be processed. For this reason, multiple servers and a distributed application of workflow management might be preferred. In this case the layer ensures that job is assigned to the server which conforms to the hardware and software requirements for that job.

D. Pseudo Code for the Proposed Architecture

```
BEGIN
Take ETL jobs through Application Layer from the User and/or other
Systems
Send the query to the Workflow Scheduling Layer
Workflow scheduling layer assigns the job to the Schedule Manager.
Schedule Manager checks the available resources through Resource
Manager
        IF the resources being used are less than the available resources
        THEN Schedule Manager checks the priority of the ETL job
being performed
                IF the priority of the job is higher
                THEN Schedule Manager checks IF the jobs can be
                performed in parallel, through Concurrency Controller
                        IF jobs can perform in parallel
                        THEN check if the resources are available
                for parallel execution
                                IF the resources are available
                                THEN send it to the workflow
                                execution layer to execute
                                ELSE wait for the availability
                        of resources
                        ELSE wait for the previous job to complete
                        IF the previous jobs is completed
                        THEN send the higher priority job to the
                Workflow Execution layer to execute
                        ELSE wait for the higher priority job to
                complete
                                IF the high priority jobs are
                        complete and no other job is executing
                                THEN send the job to
                        Workflow Execution layer to execute
                                ELSE wait for the other job to
                        complete and then send the job to
                        Workflow Execution Layer to execute.
                        ELSE wait for the resources availability
                and then send the job to the Workflow Execution
                Layer to execute
END
```

IV. Conclusion

This paper is the enhancement existing ETL work flow management. Some components have been added to augment the efficiency of the ETL workflow management. In this paper authors have given more emphasis on the workflow scheduling layer. This layer is further split up into four components i.e. Scheduling Manager, Resource Manager, Priority Controller and Concurrency Controller. These components work in parallel to each other which will increase the efficiency and the performance of the ETL workflow. Furthermore, this paper discussed a set of considerations which are required for effective workflow management. By adding different components in workflow scheduling layer ETL process may become more modular and efficient.

References

[1] T. Connolly and C. Begg, *Database Systems: A Practical Approach to Design, Implementation, and Management*, 4th ed. Addison-Wesley, 2004.

[2] V. Tziovara, P. Vassiliadis, and A. Simitsis, "Deciding the physical implementation of ETL," in *Proceedings of the ACM Tenth International Workshop on Data Warehousing and OLAP*, 2007, pp 49-56.
[3] W. Inmon, *Building the Data Warehouse*, 2nd Ed., New York: Wiley publisher. Inc, 1996.
[4] S. Chen, X. Zhang, and E. A. Rundensteiner, "A compensation-based approach for materialized view maintenance in distributed environments," in Computer Science Technical Report, Worcester Polytechnic Institute, Worcester, MA, USA, 2004.
[5] E. A. Rundensteiner, A. Koeller, and X. Zhang, "Maintaining data warehouses over changing information sources," *Communications of the ACM*, vol. 43, pp. 57-62, 2000.
[6] R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 3rd Ed., McGraw-Hill Professional, 2002
[7] H. Hussain, M. U. Shaikh, and S. U. Rehman, "Proposed text mining framework to explore issues from text in a certain domain," in *Proceedings of Second International IEEE Conference on Computer Engineering and Applications (ICCEA-2010)*, Bali Indonesia, vol. 1, pp. 16-21, March, 2010.
[8] W. J. Frawley, P. G. Shapiro, and C. J. Matheus, "Knowledge discovery in databases: An overview, AAAI press/MIT press," Cambridge, M.A., pp. 1-30, 1991
[9] A. Shamim, M. U. Shaikh, and S. U. Rehman, "Intelligent data mining in autonomous hetereogeneous distributed bio-databases," in *Proceedings of Second International IEEE Conference on Computer Engineering and Aplications (ICCEA-2010)*, Bali Indonesia, vol. 1, pp. 6-10, March, 2010.
[10] F. Zhang, B. Yang ,W. Song, and L. Li, "Intelligent decision support system based on data mining: Foreign trading case study," in *Proceeding of IEEE International Conference on Control and Automation*, Guangzhou, CHINA, May 30-June 1, 2007.
[11] A. Simitsis, K. Wilkinson, M. Castellanos, and U. Dayal, "QoX-driven ETL design: Reducing the cost of ETL consulting engagements," in *proceeding of SIGMOD'09*, Rhode Island, USA, June 29-July 2, 2009.
[12] Extract, transform, Load. Wikipedia, the free encyclopedia. [Online]. Available: http://en.wikipedia.org/wiki/Extract,_transform,_load
[13] S. March and A. Hevner, "Integrated decision support systems: A data warehousing perspective," *Decision Support Systems*, vol. 43 no. 3, pp. 1031-1043, 2007
[14] M. Solomon, "Ensuring a successful data warehouse initiative," *IS Management*, vol. 22, no. 1, pp 26-36, 2005.
[15] A. Simitsis and P. Vassiliadis, "A method for the mapping of conceptual designs to logical blueprints for ETL processes," *Decision Support System*, vol. 45, no. 1, pp. 22-40, 2008.
[16] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos, "Conceptual modeling for ETL processes," in *Proc. of the 5th ACM Int. Workshop on Data Warehousing and OLAP DOLAP*, McLean, Virginia, USA, no. 8, pp. 14-21, 2002.
[17] W. Eckerson and C. White. Evaluating ETL and Data Integration Platforms. [Online]. Available: http://www.dw-institute.com/etlreport/
[18] A. Simitsis and P. Vassiliadis, "A methodology for the conceptual modeling of ETL processes," in *CEUR Workshop Proceedings*, J. Eder, R. Mittermeir, and B. Pernici, editors, CAiSE Workshops, vol. 75, CEUR-WS.org, 2003.
[19] Z. Li, J. Sun, H. Yu, and J. Zhang, "Common cube-based conceptual modeling of ETL processes," in *proc. of International Conference on Control and Automation (ICCA2005)*, pp. 131-136, 2005.
[20] L. Muñoz, J. Mazón, and J. Trujillo, "Automatic generation of ETL processes from conceptual models," in *Proc. of the 5th ACM Int. Workshop on Data Warehousing and OLAP DOLAP'09*, Hong Kong, China, no. 6, 2009,
[21] J. Trujillo and S. L. Mora, "A UML based approach for modeling ETL processes in data warehouses," in *Lecture Notes in Computer Science*, I.-Y. Song, S. W. Liddle, T. W. Ling, and P. Scheuermann, editors, ER, Springer, vol. 2813, pp. 307-320, 2003.
[22] A. Simitsis, "Mapping conceptual to logical models for ETL processes," in *Proc. of the 8th ACM Int. Workshop on Data Warehousing and OLAP, DOLAP'05*, 2005, pp. 67-76.
[23] P. Vassiliadis, A. Simitsis, P. Georgantas, and M. Terrovitis, *A Framework for the Design of ETL Scenarios*, J. Eder and M. Missikoff (Eds.): CAiSE 2003, pp. 520–535
[24] A. Simitsis, P. Vassiliadis, and T. Sellis, "Optimizing ETL processes in data warehouses," in *Proc. of 21st International Conference on Data Engineering (ICDE'05)*, pp. 564-575, 2005
[25] M. U. Shaikh, S. U. Rehman, M. A. Qureshi, and S. Yaqoob, "Intelligent decision making based on data mining using differential evolution algorithms and framework for ETL workflow management," in *Proc. of Second International IEEE Conference on Computer Engineering and Applications (ICCEA- 2010)*, Bali Indonesia, vol. 1, pp. 22-16, March 19-21, 2010.