Conceptual and Logical Models for Performing Extraction, Transformation and Loading Process in a Data Warehouse

Sonal Sharma[1],

Research Scholar

Uttarakhand Technical University, Dehradun

Asstt. Professor

Computer Application Department

Faculty of Management and Business Studies,

Uttaranchal University, Dehradun

Dr. Rajni Jain[2]

Principal Scientist

rajnijain67@gmail.com

NIAP, New Delhi

Abstract: Extraction, Transformation and Loading process is the important phase in building a data warehouse. This phase helps in integrating the heterogeneous data sources at a unified schema after transforming it into the desired format and at the same time handling various issues related to data quality. Researchers have proposed various conceptual models as well as logical models to perform this task. Through this research work, an attempt has been made to explore all the conceptual and logical models. Further, a qualitative comparative study of these models has been performed, which reveals the strengths and weaknesses of the entire model set. In addition to this, features that should be incorporated in conceptual and logical modeling of ETL process have also been highlighted through this study.

Keywords: DW (Data Warehouse), ETL (Extraction, Transformation and Loading), OLAP (Online Analytical Processing), OLTP (On-line Transaction Processing).

I. Introduction

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 the 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, as the different sources structure information, in completely different

schemata there is the need to transform the incoming source data to a common, "global" data warehouse schema. This will eventually be used by end user applications for querying.
Second, the data coming from the operational sources suffers from quality problems, ranging from simple misspellings in textual attributes to value inconsistencies, database constraint violations and conflicts or missing information; consequently, this kind of "noise" from the data must be removed, so that end-users are provided with 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. The software processes that facilitate the population of the data warehouse are commonly known as ETL processes. ETL processes are responsible for (i) The extraction of the appropriate data from the sources, (ii) Their transportation to a special purpose area of the data warehouse where they will be processed, (iii) The transformation of the source data and the computation of new values (and, possibly records) in order to obey the structure of the data warehouse relation, to which they are targeted, (iv) The isolation and cleansing of problematic tuples, in order to guarantee that business rules and database constraints are respected and (v) The loading of the cleansed, transformed data to the appropriate relation in the warehouse, along with the refreshment of its accompanying indexes and materialized views. ETL process consumes approximately 70% of the resources required to implement and maintain a data warehouse.
The data for a data warehouse comes from the transaction data generated over a period of time in OLTP (Online Transaction Processing) systems. But the OLTP systems cannot directly populate a data warehouse because the nature and purpose of a data warehouse is quite different from that of an OLTP system. This arises from the fact that the data elements of interest during a transaction will not be the same as the data elements of interest from a business analysis point of view. Also, the OLTP operations are critical for daily business, so it cannot be slowed down by querying and summarizing a large number of records. An ETL process acts as an interface between the transaction data from OLTP systems and the data warehouse. Data requirements for the data warehouse are identified and documented by a business analyst. The requirements are analyzed by ETL professionals and enriched with the technical details of the source and target data models. This source and target information and the operations that are to be performed on the source data to conform to the target requirement constitute the ETL logical model. Logical modeling and ETL programming are a continuous process as the source data model and the target data requirements are subject to change over a period of time. Various approaches have been proposed in literature for ETL modeling and designing based on different techniques. The underlying objective of all the approaches is to model the ETL process effectively and efficiently so as to ensure a good quality data warehouse. The Section II provides the conceptual modeling for ETL process whereas logical modeling for the ETL process has been explored in the section III. Section IV compares the conceptual models on various parameters. The comparative study of logical models haves been presented in section V. In section VI, the future dimension for conceptual and logical model has been listed. The section VII concludes the current research work.

II. Conceptual Modeling for ETL Process

This section talks about the various conceptual data models for ETL processes, as suggested in literature. It's very critical to design ETL process for building data warehouse as there is no standardized model available for ETL processing. The research community has focused on the conceptual part of the data warehouse modeling as this is the front end. As a result, most of the work conducted on conceptual modeling is concerned for grabbing the conceptual characteristics and features of the star schema and the following data marts. The related work in data warehouse design has been analysed and can be summarized as follows, without declaring any approach a clear winner:

The first attempt in the direction of conceptually modeling ETL process was by Bouzeghoub et al. [21], where an informal model for separating data warehouse refreshment from its traditional treatment, for maintaining separate view and bulk loading was proposed. It does help to prove that traditional approach was complex but could not formally model different processes.

Trujillo et al. [1] have proposed object oriented approach, in which they introduced a set of extensions to UML including set of constraints for showing multi dimensional modeling properties on multidimensional databases and OLAP applications using star schema. This work was further extended in year 2002 to incorporate the UML package diagram and design guidelines, which were explained by various examples.

Golfarelli et al. [2] have proposed a graphical conceptual model called Dimensional Fact model to give a methodology for building the system from pre-existing schemes using star schema. They extended the work to both logical and physical level. The user's requirements were gathered semi automatically from the operational database and after that the workload was prepared. They further implemented their methodology proposing a prototype case tool Warehouse Integrated Designer (WAND), which helped in structuring a data mart and semi automatically generating the conceptual design.

Luzan-Mora and Trijillo [3] have proposed the physical design of a DW consisting five different stages with three levels to reduce the overall time of DW development, from conceptual modeling to final implementation. Further, they extended their work in 2004 along with Vassiladis [4] to include the transformation rules at the attribute level and thus modeling the relationship between various sources and the target location with different levels of granularity.

Rifaieh and Benharkat [5] have defined a model based of various mapping expressions. In this approach, on the basis of queries the data from databases are stored into data warehouse. A mapping expression exists between source and the target data, which is represented in form of a query. Thus, now DBMS role is extended from just data store to data transformation engine. In order to implement this approach total interaction between data warehousing tool and the mapping metadata is required. They suggested mapping guidelines to implement this approach. To achieve mapping among attributes of different schema set of mapping information defined by the developer is to be stored in mapping guideline. Manually these guideline could be defined and saved as a paper document and can be referred as and when required to backtrack the evolution of target schema attribute from source schema. There are various drawbacks of this approach:

- Maintenance is tough.
- Evolution of the system is complex.
- Updations are tedious.

Above are some of the problems of implementing mapping guideline. This picture gets worsened when the mapping is done on the basis of various versions of schema. If any time new/updated mapping of an attribute is done then it become must to maintain the updates mapping guidelines also that too in paper format.

Madhavan et al. [6], Staudt et al. [7], Stonebraker et al. [8] applied mapping expressions in different application areas. Mapping expression of any attribute is the metadata needed to identify how any target attribute is created from [9].

Authors in [10 - 18], proposed the basic model for conceptual modeling of ETL process by using conceptual constructs. They introduce a framework, which represents model for ETL activities. Their framework had three layers, as shown in Fig. 1.

- **Schema layer:** It is the lowest most layers, which involve a specific ETL. Instances of the class data type, function type, elementary activity, record set and relationship are stored as entities in schema layer.
- **Template layer:** This the middle layer, which contains construct referred as meta-classes.
- **Metamodel layer:** It is higher layer that involves the above mentioned classes. The two layers are interlinked through instantiation relationship. Five classes are involved in the metamodel layer that is quite sufficient to generate a model for any ETL implementation.

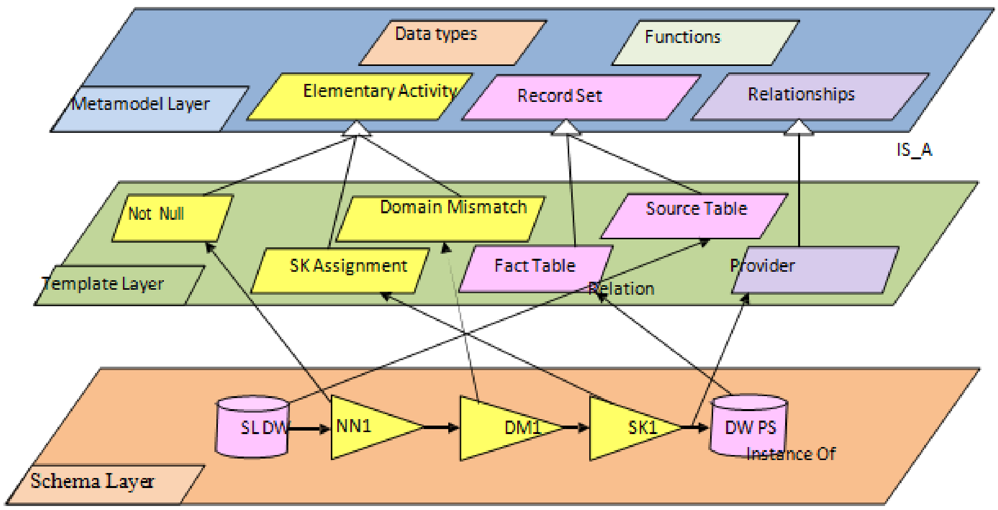

Fig. 1: The Metamodel for the Logical Entities of the ETL Environment. Source: Vassiliadis et al. [13].

It contains five different classes namely: Data types; Record sets; Functions; Elementary activities; Relationships.

- **Data types.** Every data type is identified by a name and has a domain so the values that could exist are countable. The values are also referred as constants.
- **Record sets.** Each record set is identified by its name, structure of record set (logical schema) and the finite set of values under that record set schema. If a data structure could be logically restructured then it could be treated as a record set. Relational tables and record files are the categories of record sets.
- **Functions.** Function consists of a unique name to identify it, a finite list of arguments data types and it specifies only single return type. It is the member of function type class.
- **Elementary activities.** A logical abstraction of source-code is referred as activity. It is employed in the form of logic-programming, declarative language statement. The purpose of this approach is to ignore peculiarities of any programming language [15].

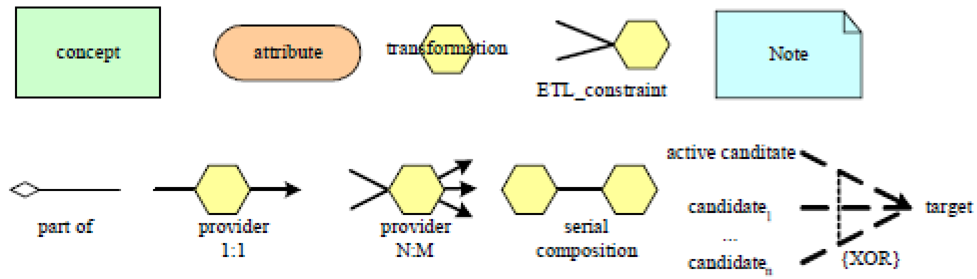

Fig. 2: Notations for the Conceptual Modeling of ETL Activities. Source: Vassiliadis et al. [15].

A model, which is based on the UML (Unified Modeling Language) notations has been introduced by various researchers, like Luzan-Mora et al., Nicolas P. et al., Stefanov V. and List B. [4], [19 - 20]. Luzan-Mora et al. [4] proposed the architecture of a data warehouse with several layers of data. In this model, authors established the relationship between classes itself. The authors extended the UML to model attributes as first-class citizens. The framework they proposed has various levels and it allowed zooming in and out the design of a scenario.

The output generated from one layer is passed on, as the input, to the other above layer. On the basis of this, they defined the entire architecture containing five stages and three levels. The five stages in their model can be understood using the Table 1.

Table 1: Stages Depicted in the Framework.

Stage	Function
Source	The data source from where the DW is to be fed could be OLTP or some external system.
Integration	To define the mapping between data source and the data warehouse.
Data warehouse	Define the structure of data warehouse.
Customization	Mapping between data warehouse and client's structure.
Client	Defines client interface to access the DW such as data marts or OLAP application.

They analyzed above mentioned stages with three different levels of perception: conceptual; logical and Physical. These stages were analyzed from conceptual point of view, logical aspects of the DW design like the definition of the ETL processes and from the perspective of physical store of data or the configuration required for the server to support DW.

In order to attain association between two attributes they proposed the representation of attributes as FCME (First-Class Modeling Element) in UML. They define attribute/class diagram along new stereotypes, Attribute and Contain. They also defined naming convention and some features that it must possess. They defined attribute/class diagram as an extended UML class diagram with attribute class and contain relationships.

There are three logical entities in relationship with respect to data warehouse:

a. **Provider entity,** which is responsible for generating the data to be further fed to higher level such as schema; table; attribute.
b. **Consumer entity,** which consumes the data generated from the provider.
c. **Intermediary entity,** which defines the mapping between data source and the data warehouse.

This approach allowed them to organize the mapping diagram into four levels using UML packages as shown in Fig. 3.

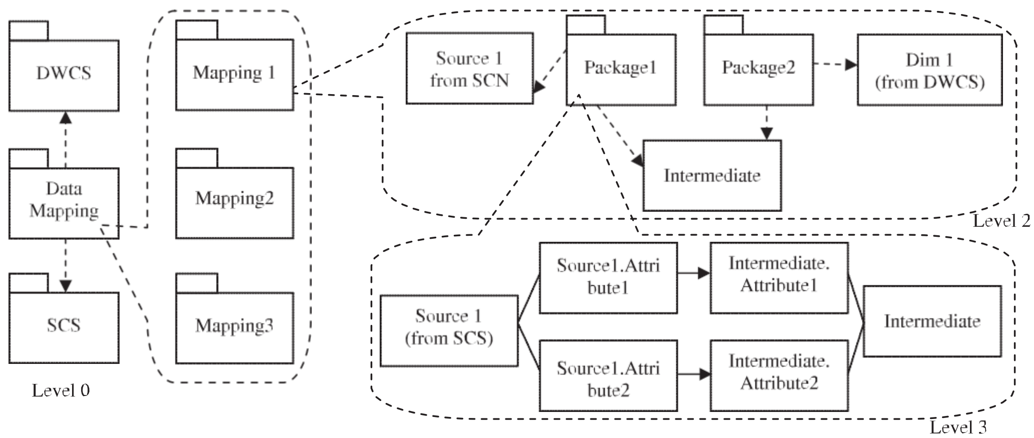

Fig. 3: Data Mapping Levels. Source: Luzan-Mora et al. [4].

a. **Database Level (or Level 0):** This level represents all the types of schema of DW (source conceptual schema; DW conceptual schema) along with their mapping through single mapping package. Single mapping package, encapsulates all the lower-level mappings among different schemata.
b. **Data Flow Level (or Level 1):** In this level the relationship between different source tables and the target DW is represented. In order to achieve this target the mapping diagram is expanded to provide more detailing and thus capturing the flow of data from source to target.
c. **Table Level (or Level 2):** All the intermediary transformation details of the flow of data from source to target is detailed at this level. The process of checking the contents also take place at this level. A set of packages is required to segment complex data mappings in sequential steps.
d. **Attribute Level (or Level 3):** Inter attribute mapping is maintained at this level. This implies that the mapping diagram has to be explored till the provider and consumer attribute mapping is not depicted. It could involve transformation and cleaning details too.

Few attempts, as discussed, are made in modeling ETL process conceptually. Muller and Rahm [22] presented conceptual modeling based on UML for ETL process using various UML based mechanism to execute the ETL operations like filtering, join, merge, wrapper, aggregation and conversion but these methods in turn inherits all the pros and cons of UML modeling techniques. The people from different domain specialization found it difficult to understand in early stages of data warehouse design because of the usage of encapsulation of various element of ETL process.

Kimball [23] has provided an informal documentation layout for the entire process of ETL where the methodology grouped as tips and guidelines have been included. Though inter attribute linkage, i.e., mapping of source attributes and warehouse attributes have been mentioned but there is no conceptual support presented for the entire process. More emphasis has been laid on the logical design of ETL process.

Cappiello et al. [24] have suggested that the quality of data is often explained as "fitness for use", i.e., its ability lies on the data collection to meet user requirements. The evaluation of data quality parameters must realize the intensity, at which data satisfy user's requirement. The service can possess multiple features on the basis of the user that access the data and the user's expectations are directly associated with the opted services. Both the attributes should be considered for assessing the quality of data and a reliable, effective assessment function has to be selected for inferring the results

Karakasidis et al. [25] have focused on the traditional practice of data refreshment in a data warehouse. In Active Data Warehousing, data warehouses are refreshed as soon as possible, to incorporate the need of most updated data. The authors further proposed a framework for the implementation of active data warehousing, with the various objectives as mentioned: (a) Software configuration of the source should acquire minimal changes, (b) Overhead incurred must be minimum, which arises due to the quick data propagation, (c) In a principled way, the entire configuration of the environment should be regulated.

Extraction-Transformation-Loading (ETL) tools are responsible for the extraction of data from multiple sources, their pre-processing, insertion and customization into a data warehouse [26 - 28]. The author has represented a conceptual model, which was later converted into an equivalent logical model for ETL processes. They have focused on the mapping of two models. Finally, they provided a methodology for the transition between the conceptual to the logical model.

None of the conceptual modeling technique, as proposed by various researchers from time-to-time, has been accepted as a standard. Hence, there lies lots of scope for the improvement in the methodology.

III. LOGICAL MODELING FOR ETL PROCESS

The literature review on logical modeling process has been presented in this section, keeping user interfaces, functionality, interoperability with ETL tools, internal representation of the data in main focus. As laid by the research community, there exist two different approaches to perform logical modeling of the ETL process.

Orchid: Dessloch et al. [29] proposed an ETL tool, named Orchid, which is modeled to integrate ETL processes and the schema mappings it as a three layer architecture as shown in the Fig. 4

The topmost layer is, External Layer. This layer specifies the ETL and Mapping System in the product specific manner. The second layer is, Intermediate Layer. The role of this layer is to capture the ETL process with focus on the various ETL activities and operation that are possible in the tool. This layer is also product specific as a result a separate intermediate representation is supported for each platform. The last layer is called as Abstraction Layer. This layer is based on relational algebra and is referred as Operator Hub Model (OHM). It is common to both ETL as well as Mapping System. The entire operator functionality is represented graphically using directed graph.

As the two layers out of three are platform dependent so the two separate implementations are required. The last layer OHM, can be customized as per the need. But Orchid is not available for the research community, as it is a part of Data Stage ETL System in IBM. It supports the logical modeling of ETL process of IBM's mining and ETL tool only. This tool is commercial and costly too. It is not available for the researchers.

Jorg and Dessloch [87] have proposed the method for incremental loading by rewriting the OHM instances of Orchid. They assumed that the change in the data would be pre identified at the source itself.

They proposed that the changed data would be:

$$D_{new} = D_{old} \uplus \Delta D \uplus \boxplus D \ominus \nabla D \ominus \boxminus D$$

where D_{old}, $D_{new,}$ ΔD, ∇D, $\boxminus D$ and $\boxplus D$ are old dataset, new data set, set of inserted tuples, set of deleted tuples, old data in the updated tuples and the new data in the updated tuples respectively. The $\uplus$ and $\ominus$ are multi-set union and difference operators. The assumption they made for execution was the pre-identification of changes, which practically is very complex to identify at the source. Basic operators, which are available in Orchid Hub Model are depicted in the Fig. 5.

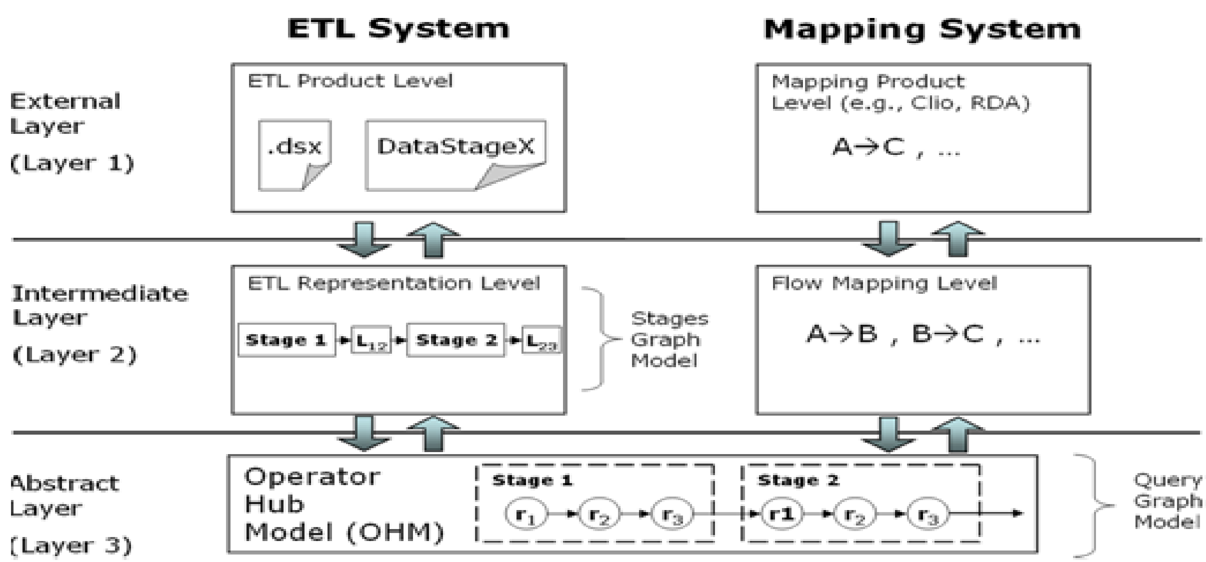

Fig. 4: Orchid Components and Abstraction Layer. Source: Dessloch et al. [29].

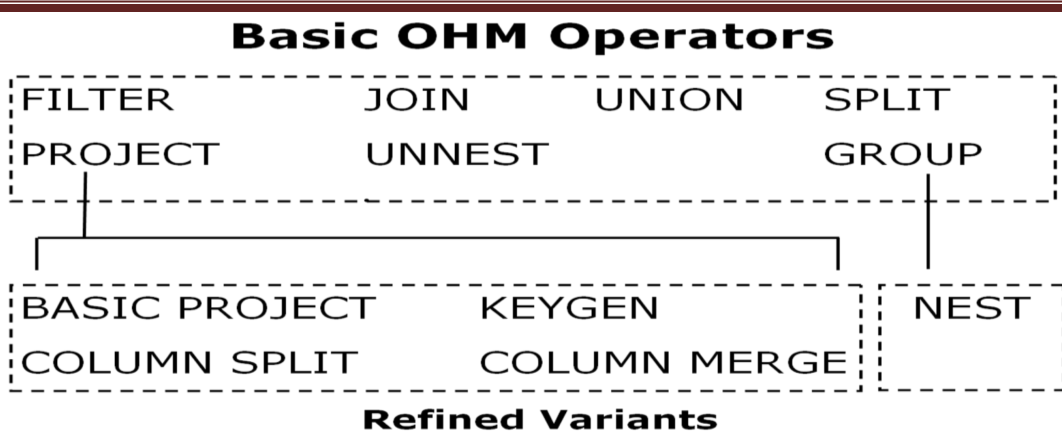

Fig. 5: OHM Operators and Instances. Source: Jorg and Dessloch [30].

Arktos II: Vassiliadis et al. [16] proposed another model for logical modeling of the ETL process, Arktos II, which has three layer architecture to describe the semantics of the processes, as shown in the Fig. 6.

- **The Metamodel Layer:** Provides meta classes for allt he ETL acitivities.
- **The Template Layer:** Contains specialization of the meta model classes.
- **The Schema Layer:** Captures the entire information.

This tool is a graphical representation of the ETL activities designed with the help of the template containing name, expression, mapping, parameter list for each activity.

- **Name:** Specifies the unique name of each activity.
- **Expression:** Specifies the statement mentioning operation that is to be performed by the ETL activity.
- **Mapping:** It is to show the linkage between input and output parameters.
- **Parameter List:** Contains the list of values to be assigned to the constants and also specifies the mapping condition.

The overall definition of ETL activities is represented by the tool in form of macros or stored procedures, the graphical representation also provides the ease of usability but the internal representation of the tool doesn't operates with any of the available ETL tool.

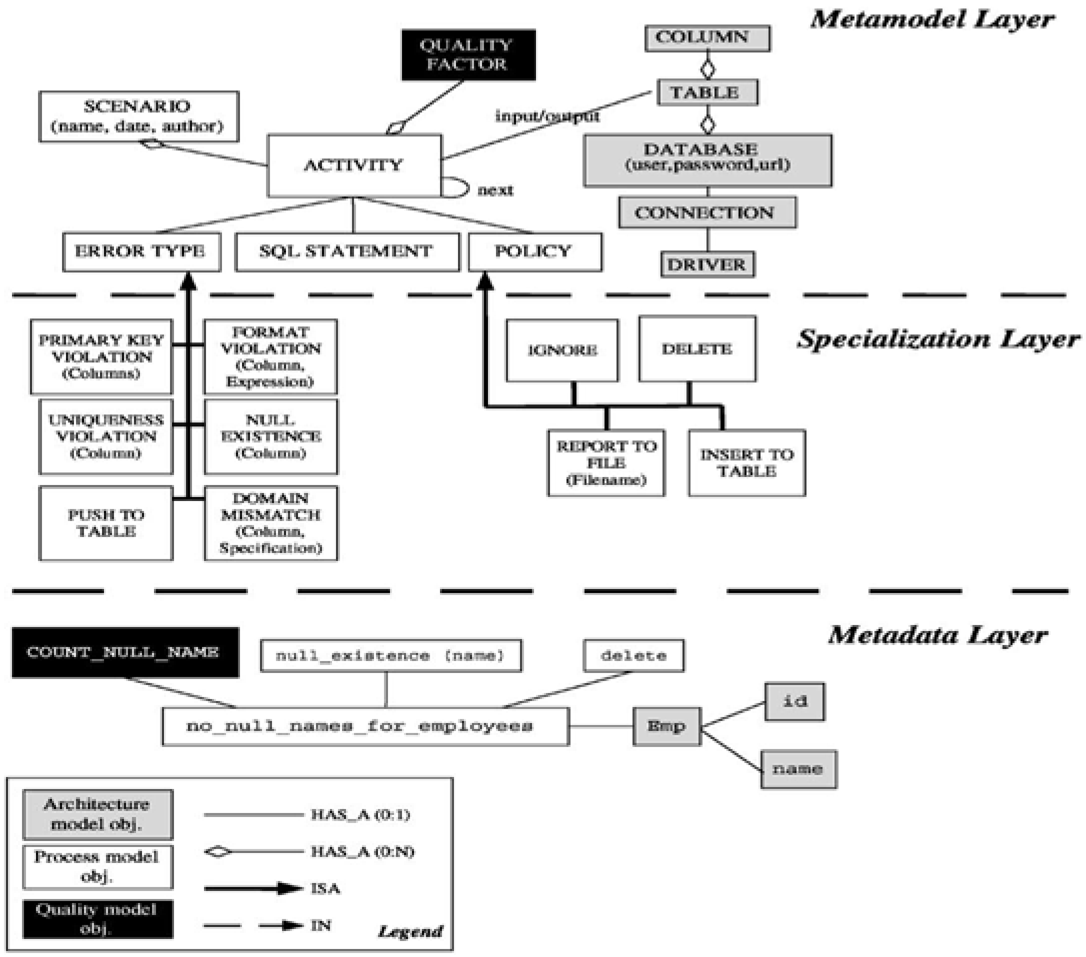

Fig. 6: Metamodel of Arktos II. Source: Vassiliadis et al. [16].

IV. Comparison between Conceptual Models

The section below, discuss various list of parameters, on which the various approaches of ETL modeling could be compared are:

- **SQL Query Based:** It describes whether the modeling approach has the support for querying from any data source on the basis of provided SQL query or not.
- **Interact with Data Warehousing Tool:** The interaction of ETL modeling with the data warehousing tool is also an important parameter for comparing the modeling approaches.
- **Maintenance is Easy:** Success of any system is highly dependent upon its maintenance, so it is expected from any ETL modeling process to be maintained easily.
- **Evolution of the System is Complex:** The complete process of ETL modeling is desirable to be simpler.
- **Easy Updations:** It's required that the system should be easy to update.
- **Lots of Paper work:** It is desirable that the evolved ETL modeling process should support in paper work reduction.
- **Framework for Activities:** The various ativities as discussed through literature review, should be facilitated by any of the modeling approach.
- **Graphical Model:** It becomes quite easy to handle a graphical user interface as compared to

the character user interface. As a result graphical ETL model are highly appreciated.

- **High Level Operators:** There are various operations that any ETL process needs to perform, but not all the operations are possible through each and every modeling approach. Therefore, existence of high level operators in any modeling approach out laws it from other approaches.
- **Reduces Complexity of ETL Processs:** The entire ETL process should be easy to design and develop. It is expected from a good modeling approach that it reduces the complexity of ETL process.
- **Layered Architecture:** The layered architecture allows to segregate the various components of ETL process. It provides data abstraction and at the same time allows to reuse the data processed at lower level.
- **Object Oriented Concepts:** The implementation of any modeling technique using object oriented approach allows the reusability of extracted data at various levels.
- **Mapping Operations and Relationships:** The ETL modeling approach should be capable enough to maintain the mapping between operations and relationships.
- **Interaction with the Metadata:** This feature ensures that the evolved system is capable to hand the metadata as provided by the user for the purpose of ETL modeling.

Table 2: Evaluation Matrix for Conceptual Models.

S.NO	Factor of Discrimination	Mapping Expression	Conceptual Modeling	UML Environment
1.	SQL query based	YES	NO	No
2.	Interact with data warehousing tool	YES	NO	NO
3.	Maintenance is easy	NO	YES	YES
4.	Evolution of the system is complex	YES	NO	NO
5.	Easy updations	NO	YES	YES
6.	Lots of paper work	YES	NO	NO
7.	Framework for activities	NO	YES	NO
8.	Graphical model	NO	YES	YES
9.	High level operators	NO	YES	NO
10	Reduces complexity of ETL process	NO	Partially	YES
11	Layered Architecture	NO	YES	YES
12	Object oriented concepts	YES	Partially	NO
13	Mapping Operations and relationships	YES	YES	YES
14.	Interaction with metadata	YES	NO	NO

V. Comparison between Logical Models

The various operators that are available in the discussed logical modeling tools are as summarized in the table given below. The notation '●' means that the operator has an implementation in the tool, and '-' means that the operator does not have an implementation.

Table 3: Summarized Operator Implementation Detail.

Operator / Approach	FILTER	JOIN	FORMAT	AGGREGATE	UNION	SORT	COPY
Orchid	●	●	●	●	●	-	●
Arktos II	●	●	●	●	●	●	-

There are various other ETL activities which would be required in an ETL model as to solve different types of issues that exists in any data set like handling missing values, detecting outliers, merging two data sources as well as providing the facility of performing unit conversion. Further, these models need to be implemented and then tested on some real time data set. Orchid is the product of IBM and it is not available to the research community whereas Arktos is the logical model representation in a form of a graph and hence no quantitative analysis could be performed. Hence, there is a strong requirement to develop a model which could cater the need of the domain system as per its requirement and further that model need to implemented and validated.

VI. Conclusion and Future Work

In this paper, authors have studied various conceptual and logical models as suggested in the literature. Further a comparative analysis has been done on various parameters. The logical models were compared on the basis of ETL activities available with them. It was found that none of the conceptual model has been declared as a standard model for ETL process. Through this study it was also revealed that there are various important ETL activities which are missing in both the logical models. Further these models have not been implemented; hence there exist a need of user interface to tackle the ETL process effectively and efficiently. The models as discussed provide very basic operations. In future, these models could be extended to include various operations like handling missing values, detecting outliers, merging two data sources as well as providing the facility of performing unit conversion whenever and wherever required. These models have not been implemented and the authors are working on the implementation part of these models. The quantitative analysis of these approaches would be possible then only. Though there exist various open source ETL tools but to achieve a customised ETL interface, it is recommended to hand-code the entire process.

VII. Bibliography

[1] Trujillo J., Gomez J., Palomar M., Song Y., "Designing Data warehouses with OO Conceptual Models", Computer, IEEE, pp. 66-76,2001.

[2] Golfarelli M., Maio D., Rizzi S., "The Dimensional Fact Model: A Conceptual Model for Data Warehouses", International Journal Cooperative Information Systems (IJCIS), Vol. 7, No. 2-3, pp. 215-247, 1998.

[3] Luzan-Mora, S., Trujillo, J., "A Comprehensive Method for Data Warehouse Design", in Proceedings of the Fifth International Workshop on Design and Management of Data Warehouses (DMDW'03), Berlin, Germany, 2003.

[4] Luzan-Mora S., Vassiliadis P., Trujillo J., "Data Mapping Diagrams for Data Warehouse Design with UML", Springer, LNCS, Vol. 3288, pp. 191-204, 2004.

[5] Rifaieh R., Benharkat N.A., "Query-Based Data Warehousing Tool", in Proceedings of Fifth ACM international Workshop on Data Warehousing and OLAP (DOLAP'02), pp. 35-42, 2002.
[6] Madhavan, J., Bernstein P.A., Rahm E., "Generic Schema Matching with Cupid", in Proceedings of the 27th International Conferences on Large Databases, pp. 49-58, 2001.
[7] Staudt M., Vaduva A., Vetterli T., "Metadata Management and Data Warehousing", Technical Report, The Department of Information Technology (IFI) at University of Zurich, 1999.
[8] Stonebraker M., Hellerstein J., "Content Integration for e-business", in Proceedings of the ACM SIGMOID/PODS 2001, CA, 2001.
[9] Jiang L., Cai H., Xu B., "A Domain Ontology Approach in the ETL Process of Data Warehousing", in the Proceedings of 10th IEEE International Conference on E-Business Engineering, pp. 30-35, 2013.

[10] Vassiliadis P., "Data Warehouse Modeling and Quality Issues", Ph.D. Thesis, National Technical University Of Athens, 2000.
[11] Vassiliadis P., Simitsis A., "A Generic and Customizable Framework for the Design of ETL Scenarios", in Proceedings of the 15th International Conference on Advanced Information Systems Engineering, Vol. 30, No. 7, pp. 492-525, 2005.
[12] Vassiliadis P., Simitis A., Skiadopoulus S., "On the Logical Modeling of ETL Process", CAise 2002, LNCS 2348, Springer, pp. 782-786, 2002.
[13] Vassiliadis P., Simitis A., Georgantas P., Terrovitis M., "A Framework for the Design of ETL.

[14] Vassiliadis P., Simitis A., Skiaopoulos S., " Modeling ETL Activities as Graphs", in Proceedings of the Fourth International workshop on the Design and Management of Data Warehouses (DMDW'02), Canada, pp. 52-61, 2002.
[15] Vassiliadis P., Simitis A., Skiaopoulos S., "Conceptual Modeling for ETL Processes", in Proceedings of the fifth ACM International Workshop on Data Warehousing and OLAP (DOLAP'02), pp 14-21, 2002.
[16] Vassiliadis P., Vagena Z., Skiaopoulos S., Karayannidis N., Sellis T., "Arktos: Towards the Modeling, Design, Control and Execution of ETL Processes", Information Systems, Vol. 26, pp. 537–561, 2001.

[17] Vassiliadis P., Bouzeghoub M., Quix C., "Towards Quality-Oriented Data Warehouse Usage and Evolution", Information Systems, Vol. 25, No. 2, pp. 89-l 15, 2000.
[18] Rizzi S., "Conceptual Modeling and Evolution in DWs.", Perspectives Workshop: Data Warehousing at the Crossroads, Dagstuhl, 2004.
[19] Nicolas P., Jacky A., Isabelle C.W., "A UML-Based Data Warehouse Design Method", Elsevier, Decision Support Systems, Vol. 42, pp. 1449–1473, 2006.
[20] Stefanov V., List B., "A UML Profile for Modeling Data Warehouse Usage", ER Workshops 2007, LNCS 4802, pp. 137–147, 2007.
[21] Bouzeghoub M., Fabret F., Matulovic M., "Modeling Data Warehouse Refreshment Process as a Workflow Application", in proceedings of DWDM, 1999.
[22] Muller, Rahm E., "An Integrative And Uniform Model for Metadata Management in Data Warehousing Environments", in Proceedings of The International Workshop on Design and Management of Data Warehouses (DMDW'99), Germany,1999.
[23] Kimball R., "The Data Warehouse Toolkit: Practical Technique for Building Dimensional Data Warehouses", John Wiley & Sons, New York, 2004.
[24] Cappiello C., Francalanci C., Pernici B., "Data Quality Assessment From The User's Perspective", in Proceedings of IQIS '04 , ACM, USA, pp. 68-73, 2004.

[25] Karakasidis A., Vassiliadis P., Pitoura E., "ETL Queues for Active Data Warehousing", in Proceedings of 2nd International Workshop on Information Quality in Information Systems (IQIS 2005), co-located with ACM SIGMOD/PODS 2005, USA, 2005.

[26] Simitsis A., Vassiliadis P., "A Method For The Mapping of Conceptual Design to Logical Blueprint for ETL Processes", Decision Support Systems, Data Warehousing And OLAP, Vol. 45, No.1, pp. 22-40, 2008.

[27] Simitsis A., Skoutas D., Castellanos M., "Natural Language Reporting for ETL Processes", in Proceeding of the ACM 11th International Workshop on Data Warehousing and OLAP (DOLAP'08), pp. 65-72, 2008.

[28] Simitis A., "Modeling and Managing ETL Process", Available at http://ftp.informatik.rwtaachen.de/Publications/CEUR-WS/Vol-76/simitsis.pdf

[29] Dessloch S., Hernández M.A., Wisnesky R., Radwan A., J. Zhou, "Orchid: Integrating Schema Mapping and ETL", in Proceedings of the 24th International Conferenceon Data Engineering(ICDE), Mexico, pp. 1307-1316, 2008.

[30] Jorg T., Dessloch S., "Towards Generating ETL Processes for Incremental Loading", in Proceedings of the 2008 International Symposium on database Engineering & Applications, Portugal, pp. 101-110, 2008.

[31] Sharma S., Jain R., "Comparison Between Various Architectural Approaches in Data Warehousing: A Discriminatory Analysis", in Proceedings of 9th National Conference on Smarter Approaches in Computing Technologies & Applications (SACTA-2014), India, 2014.

[32] Sharma S., Jain R., "Modeling ETL Process in Data warehouse: An Exploratory Study", in Proceedings of 4th International Conference on Advanced Computing & Communication Technologies (ACCT2014), India, 2014. Available at IEEE eXplore.

[33] Sharma S., Soni P., "Data warehousing: Challenges and Opportunities", Evolution Ideas: - Computing, Communication and Networking, pp. 83-89, ISBN: - 978-81-8220410, 2011.

[34] Sharma S., Jain R., "Enhancing Business Intelligence Using Data Warehousing: A Multi - Case Analysis", in International Journal of Advance Research in Computer Sciences and Management Studies, available at http://www.ijarcsms.com/December2013.html.