# A LITERATURE REVIEW ON DATA INTEGRATION CHALLENGES

Sagar Autade[1], Pravin S. Metkewar[2]
[1]MBA-IT, [2]Assoc. Professor,
SICSR, Affiliated to Symbiosis International University (SIU), Pune, Maharashtra, India

***Abstract: Data Integration can be said to be the process of bringing together historical, current or sometimes even real-time data, that is saved at different locations or formats (syntax or semantics), into a single structure so that the user is able to view it as a single entity irrespective of its location or syntax/semantics. This integrated data can help in Business Intelligence to make better decisions based on the reports generated from the data. The most prominent challenge that data integration faces is to make this happen in real-time. In this paper, we discuss about data integration and the challenges it has faced over time.***

***Index Terms: Data Warehouse, Data Integration***

## I. INTRODUCTION

Shortly after database systems were initially introduced in the business world, data integration emerged as one of the primary fields of research in the area. Data Integration is the central problem of combining data from various sources at a single place to give the users a virtually unified view of all the data. Integrating the data from various heterogeneous sources is critical from the business point of view, as it helps the businesses to analyze these data at a single place. The businesses can use the results of this analysis to make important business decisions. When data is passed from its source system to a data warehouse various inconsistencies, errors or redundancies may be introduced. These inconsistencies or redundancies need to be removed so that the DW is able to provide a reconciled view of the data to the users of the data warehouse. Integration of data is crucial in order to fulfill business and consumer needs. The data integration system provides a unified view of the data combined from various sources, called the global schema. The global schema provides an integrated, regulated and virtual view of the underlying sources. The primary intention of data integration design is how the global schema is defined. In mainly centers around which data model to use and what kinds of constraints can be expressed on the data. Also the relation between the data sources and the global schema needs to be defined. Basically, data needs to be integrated for two main reasons: firstly, for a given set of available information systems, an integrated view could be forged to expedite information access and reuse through a single avenue or access point. Secondly, for particular information need, data from different information systems is compiled to gain a more extensive basis towards the required need [1]. Answering the queries posed in terms of the global schema is an important service that the data integration systems provide to their users. Given the architecture of a system, a reformulation step is required for query processing in data integration: the query over the global schema needs to be reformulated in terms of the set of queries over the sources. Since the data sources are generally heterogeneous and autonomous, the problem of mutually inconsistent data sources persists in the real-world application scenario. This problem is usually solved by applying certain transformation and cleansing procedures on the data retrieved from the sources. This paper is structured as follows: In Section 2, we present the various data integration challenges faced by data warehouses. Section 3 will finally conclude the paper.

## II. DATA INTEGRATION

Irrespective of the location of the actual data or the number of information systems that the data come from, data integration systems integrate these data into a unified virtual view and present it to the users of the system as a homogeneous logical view of data spread across various heterogeneous data sources, by detecting and resolving the structural and semantic conflicts between the data and their schema. Generally, information systems are not designed to be integrated with other systems. Whenever an integrated access to heterogeneous source systems is coveted, the origins and their data that do not fit together need to be consolidated with added transformations and reconciliation functionality. An important point to be noted here is that there is no single problem in data integration. An integration task may usually depend on: the architectural view of the information system, the content and functionality of the component systems, the kind of information that is managed by component systems (alphanumeric data, multimedia data; structured, semi-structured, unstructured data), requirements concerning autonomy of component systems, intended use of the integrated information system (read-only or write access), performance requirements, and the available resources (time, money, human resources, know-how, etc.)[3]. Also different types of heterogeneity need to be considered including, hardware and operating systems, data management software, data models schemas, and data semantics, middleware, user interfaces, business rules and integrity constraints [3].

## III. INTRODUCTION TRADITIONAL DATA INTEGRATION APPROACH

Traditional data integration approaches include scripting, ETL, EAI, and real-time CDC. Scripts and ETL are batch-oriented in data delivery, whereas EAI and real-time, log-based CDC support continuous data capture [2]. Figure 1 shows the comparison of these different approaches with respect to different data architecture components.

## IV. CHALLENGES

Following are the challenges faced by data integration systems:

### *A. Manual Integration*

In this method, users directly interact with the information systems and integrate the data by selecting it from the individual, usually heterogeneous data sources. An important consideration in manual integration is the existence of different user interfaces and query languages. As we need specific data for integration, a detailed knowledge of the location of the data is necessary. In this method, the users select the data sources manually from the various sources available to them. They then integrate the data from these sources to generate a unified view of the data. A major issue with manual integration is that it is a very tedious and lengthy process. Also, being a manual process, it is very much prone to errors. The data generated by manual integration may also be inconsistent with respect to the actual data as some inconsistencies may be introduced in the data during the integration process.

### *B. Common User Interface*

Here, the users of the system are provided with a common user interface that gives a uniform look and feel for all the data. The data is still presented separately; hence, data homogenization and integration is still to be done. Thus, in this method the user is simply given the look and feel of the data being integrated and homogeneous, whereas, in reality it is actually still residing in various heterogeneous sources. Since the data is presented separately the users find it difficult to use the data for analysis purposes.

### *C. Integration by Application*

In this method, integration applications are used to access data from various sources and return an integrated view of the resulting data. This method, however, becomes impractical when the number of interfaces and data formats to be homogenized and integrated increases.

### *D. Integration by Middleware*

A middleware helps solve the dedicated problems of the integration problem by providing a reusable functionality. Even though middleware relieves the integration systems from implementing common functionalities individually, usually different middleware tools need to be combined to build integration systems.

### *E. Uniform Data Access*

This approach allows for logical integration of data at the data access level. Though only virtual view of data is available at this level, it provides global applications with unified global view of physically distributed data. However, data access, homogenization, and integration of data on a global scale, which are time-consuming, needs to be done at runtime.

### *F. Common Data Storage*

In this method, data from various sources are physically integrated by transferring the data to a new common storage; the local data sources may be withdrawn or may remain operational. This helps the data integration to provide faster data access. However, if the local data sources are withdrawn, the applications that accessed them need to be migrated to the new data storage as well. In case they remain operational, the common data storage needs to be refreshed periodically.

## V. CONCLUSION

In this paper, we gave a brief overview of the obstacles and challenges faced by data integration. There is no single solution to solve these problems. Every challenge faced by data integration is unique and needs a different and unique approach to solve it. The paper is an attempt to bring together as many challenges and issues together as possible, so that further work can be done on trying to resolve these issues.

## VI. ACKNOWLEDGMENT

I would like to thank Dr. Pravin Metkewar, Head of Department MBA-IT, SICSR for his extended support and encouragement for carrying out this work. I wish a special thanks to the department faculty members who help carry out this work.

## REFERENCES

[1] Gal. A, Managing uncertainty in schema matching with top-k schema, mappings, (2006) JoDS

[2] Data Integration Architectures for Operational Data Warehousing, An Oracle White Paper, September 2012

[3] Patrick Ziegler and Klaus R. Dittrich, Three Decades of Data Integration – All Problems Solved?, Database Technology Research Group, Department of Informatics, University of Zurich

[4] Data Integration Challenges: A Study, International Journal of Advanced Research in Computer Science and Software Engineering, July 2012