MODELING, ANALYSIS AND SIMULATION OF NEAR REAL-TIME E-T-L PROCESSES OF BIG DATA IN CLOUD

Thesis Submitted by

Neepa Biswas

For the award of the degree of
DOCTOR OF PHILOSOPHY (ENGINEERING)

Department of Information Technology
Faculty Council of Engineering & Technology
Jadavpur University
Kolkata, India

2022

Supervisor Details:

Name: Dr. Kartick Chandra Mondal
Designation: Assistant Professor
Institution of the Supervisor:
Department of Information Technology
Jadavpur University, Salt Lake Campus
E-mail kartick.mondal@jadavpuruniversity.in
Kolkata-700106
West Bengal
India

List of Publications

Journals

1. Biswas, N., Chattapadhyay, S., Mahapatra, G., Chatterjee, S., & Mondal, K. C. (2019). A new approach for conceptual extraction-transformation-loading process modeling. International Journal of Ambient Computing and Intelligence (IJACI), 10(1), 30-45.

2. Biswas, N., Sarkar, A., & Mondal, K. C. (2020). Efficient incremental loading in ETL processing for real-time data integration. Innovations in Systems and Software Engineering, 16(1), 53-61.

3. Biswas, N., Mondal, A. S., Kusumastuti, A., Saha, S., & Mondal, K. C. (2022). Automated credit assessment framework using ETL process and machine learning. Innovations in Systems and Software Engineering, 1-14.

Book Chapters

1. Biswas, N., Chattopadhyay, S., Mahapatra, G., Chatterjee, S., & Mondal, K. C. , Computational Intelligence, Communications, and Business Analytics, SysML Based Conceptual ETL Process Modeling, https://doi.org/10.1007/978-981-10-6430-2, Springer Singapore, Print ISBN: 978-981-10-6429-6, Electronic ISBN: 978-981-10-6430-2

2. Biswas N., Sarkar A., Mondal K.C. (2019) Empirical Analysis of Programmable ETL Tools. In: Mandal J., Mukhopadhyay S., Dutta P., Dasgupta K. (eds) Computational Intelligence, Communications, and Business Analytics. CICBA 2018. Communications in Computer and Information Science, vol 1031. Springer, Singapore.
https://doi.org/10.1007/978-981-13-8581-0_22 , Print ISBN: 978-981-13-8580-3, Online ISBN: 978-981-13-8581-0.

3. Mondal, K.C., Biswas, N. and Saha, S., (2020), "Role of Machine Learning in ETL Automation.", In: Proceedings of 21st International Conference on Distributed Computing and Networking (ICDCN 2020), Article No. - 57, Pages: 1-6, January 4–7, 2020, Kolkata, India. ACM, New York, NY, USA, https://doi.org/10.1145/3369740.3372778, ISBN: 9781450377515, (CORE Ranking - B)

4. Biswas N., Mondal K.C. (2022) Integration of ETL in Cloud Using Spark for Streaming Data. In: Mandal J.K., De D. (eds) Advanced Techniques for IoT Applications. EAIT 2021. Lecture Notes in Networks and Systems, vol 292. Springer, Singapore, https://doi.org/10.1007/978-981-16-4435-1_18, Print ISBN: 978-981-16-4434-4, Online ISBN: 978-981-16-4435-1

International Conferences

1. Biswas, N., Chattopadhyay, S., Mahapatra, G., Chatterjee, S., & Mondal, K. C. (2017, March). SysML Based Conceptual ETL Process Modeling. In International Conference on Computational Intelligence, Communications, and Business Analytics (pp. 242-255). Springer, Singapore.

2. Biswas, N., Sarkar, A., & Mondal, K. C. (2018, July). Empirical analysis of programmable ETL tools. In International Conference on Computational Intelligence, Communications, and Business Analytics (pp. 267-277). Springer, Singapore.

3. Mondal, K.C., Biswas, N. and Saha, S., 2020, January. Role of Machine Learning in ETL Automation. In Proceedings of the 21st International Conference on Distributed Computing and Networking (pp. 1-6).

4. Biswas, N., & Mondal, K. C. (2021, February). Integration of ETL in Cloud Using Spark for Streaming Data. In International Conference on Emerging Applications of Information Technology (pp. 172-182). Springer, Singapore.

List of Presentations in National/ International/ Conferences/ Workshops:

International Conference Presentation

1. Biswas, N., Chattopadhyay, S., Mahapatra, G., Chatterjee, S., & Mondal, K. C. (2017, March). SysML Based Conceptual ETL Process Modeling. In International Conference on Computational Intelligence, Communications, and Business Analytics (pp. 242-255). Springer, Singapore.

2. Biswas, N., Sarkar, A., & Mondal, K. C. (2018, July). Empirical analysis of programmable ETL tools. In International Conference on Computational Intelligence, Communications, and Business Analytics (pp. 267-277). Springer, Singapore.

3. Biswas, N., & Mondal, K. C. (2021, February). Integration of ETL in Cloud Using Spark for Streaming Data. In International Conference on Emerging Applications of Information Technology (pp. 172-182). Springer, Singapore.

PROFORMA – 1

“Statement of Originality”

I, Ms. Neepa Biswas, registered on 20/03/2017 do hereby declare that this thesis entitled **”MODELING, ANALYSIS AND SIMULATION OF NEAR REAL-TIME E-T-L PROCESSES OF BIG DATA IN CLOUD”** contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work.

I also declare that I have checked this thesis as per the “Policy on Anti Plagiarism, Jadavpur University, 2019”, and the level of similarity as checked by iThenticate software is **6**%.

Signature of Candidate:

Neepa Biswas

——————————

(Neepa Biswas)
Date :

Certified by Supervisor:

Kartick Chandra Mondal

——————————

(Dr. Kartick Chandra Mondal)

PROFORMA – 2

“CERTIFICATE FROM THE SUPERVISOR”

This is to certify that the thesis entitled **”MODELING, ANALYSIS AND SIMULATION OF NEAR REAL-TIME E-T-L PROCESSES OF BIG DATA IN CLOUD”** submitted by **Ms. Neepa Biswas**, who got her name registered on **20/03/2017** for the award of Ph.D. (Engineering) degree of Jadavpur University is absolutely based upon her own work under the supervision of **Dr. Kartick Chandra Mondal** and that neither her thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.

Kartick Chandra Mondal

——————————————————

(Dr. Kartick Chandra Mondal)
Signature of the Supervisor
and Date with Office Seal

Acknowledgements

At first, I express my profound regards to my supervisor, **Dr. Kartick Chandra Mondal**, Assistant Professor, Department of Information Technology, Jadavpur University, for his valuable guidance, scholarly input, and consistent encouragement I received throughout the research work. No word would be sufficient to express my gratitude towards him. This feat was possible only because of the unconditional support provided by him. A person with an amicable and positive disposition, he has always made himself available to clarify my doubts despite his busy schedule, and I consider it a great opportunity to do my doctoral program under his guidance and to learn from his research expertise.

I want to express my deepest appreciation towards **Dr. Parama Bhaumik**, Head of the Department of Information Technology for his valuable inputs, helpful reviews, and suggestions. Words are inadequate to express a deep sense of gratitude to members of the doctoral committee, Jadavpur University, Kolkata.

I am immensely thankful to all my teachers who taught me at various stages of my life starting from my childhood. Thank you for teaching me that my job in life was to learn, to be happy, and to know and understand myself.

I gratefully acknowledge the funding received for my Ph.D. for consideration of support under the DST PURSE Programme – Phase II, by the Government of India. I am also grateful to the funding authority received through the State Government Fellowship Research Scheme of Jadavpur University to undertake my Ph.D.

I would like to thank all my laboratory mates at Jadavpur University, Department of IT, especially Ryan Roy, Moumita Ghosh, Nilkantha Garain, Madhupa Sanyal, Anamitra Sarkar. All of them helped me directly or indirectly during this memorable journey. I also acknowledge all E-T-L researchers and Data scientists whose works inspired me to undertake this research. At last, I must express my sincere heartfelt gratitude to all the staff members of the Department of Information Technology at Jadavpur University . They helped me directly or indirectly during this course of work.

The thesis would not be possible without the moral support of my husband, Sudarsan Biswas who have shown confidence in me and supported me in completing my thesis work. My Parents and In-Laws have always been there to show their blessing and inspiration to complete this thesis work. My two little daughters, Srinika and Inika, are the sources of my strength to keep going when I feel like giving up.

Above all, I owe it all to Almighty God Lord Sri Krishna for granting me the wisdom, health, and strength to undertake this research task and enable me to its completion.

Neepa Biswas

Abstract

Timely and accurate decision-making provides a competitive advantage for each organization that needs to store and speedy access to the large volume of everyday transactional data. In the Data Warehouse (DW) environment, Extract, Transform, and Loading (ETL) plays a key technology that refines and integrates a large stream of heterogeneous operational and external data of any organization. The value of organizational data is significantly enhanced when content migration from various sources is done significantly by using the ETL process.

In the last few years, the use of ETL for constructing and managing Data warehouses has been gaining popularity in various real-life applications like e-commerce, banking, e-governance, etc. Moreover, many industry applications (Fraud detection, payment processing, IoT edge analytics, etc.) require real-time integration and reporting over data acquired from heterogeneous data sources. Many types of ETL solutions are coming to resolve these issues, like Batch versus Real-time and On-Premise versus Cloud.

ETL is a significant area of research for a well-established Data Warehouse environment. In this Thesis, I have discussed the main motivation behind the Ph.D. research work along with a brief literature survey and my research work accomplished in this domain. In this work, I have focused on planning and implementing a standard ETL process incorporating real-time data integration features in a cloud environment to handle Big data for performing data analytics efficiently. In this Thesis, I have worked towards the modeling, simulation, and empirical analysis of traditional and real-time ETL processes and advanced proposal of ETL workflows management by use of Machine learning and shifting the ETL workload in the Cloud environment.

Contents

List of Figures

List of Tables

Acronyms

API Application Programming Interface

ATM Air traffic management

BAM Business Activity Monitoring

BI Business Intelligence

BPMN Business Process Model and Notation

CDC Change Data Capture

CI Continuous Integration

CRM Customer Relationship Management

CST Cameo Simulation Toolkit

CSR cloud service resources

CWM Common warehouse metamodel

DSS Decision support system

DVO Data Validation Option

DW Data warehouse

EAI Enterprise Application Integration

EMD Entity mapping diagram

ERP Enterprise Resource Planning

ETL Extract Transform Load

fUML Foundational UML

G2B Government-to-business

G2C Government-to-citizen

G2E Government-to-employees

G2G Government-to-government

GUI Graphical User Interfaces

IaaS Infrastructure as a Service

ICT Information and communications technologies

INCOSE International Council on Systems Engineering

IoT Internet of Things

KM Knowledge Management

LOC Lines of code

M3C World Wide Web Consortium

MDA Model driven architecture

MDD Model-Driven Development

ML Machine-learning

ODS Operational Data Stores

ODI Oracle Data Integrator

OLAP Online analytical processing

OLTP On-line Transaction Processing

OMG Object Management Group

PaaS Platform as a Service

PIM Platform Independent Model

PSM Platform Specific Model

QoS quality of service

QVT Query View Transformation

RTDC Real time data cache

SaaS Software as a Services

SCD Slowly changing dimension

SSIS SQL Server Integration Service packages

SysML Systems Modeling Language

UML Unified Modeling Language

xUML Executable UML

Chapter 1

Introduction

A Data warehouse (DW) is a repository of historical data that holds transactional data in relational format [83]. In the warehouse, data is stored in a standard structure that is obtained by integrating data from different operational sources of an organization. Nowadays, data analysis has become an integral part of any organization to achieve optimized performance. The business analysts [18, 256] can access that data, perform analysis, incorporate Business Intelligence (BI) applications, make predictions, and make strategic decisions. For maintaining a DW, the main focus is to manage the large amounts of data generated from different types of systems (SAP, ERP, Oracle, mainframe, etc.) and store those data in a uniform structure [115].

For accessing and managing those data, ETL has a significant role. ETL is a widely used process in business organizations [281]. It identifies and extracts data from various sources, filtering and customizes those data according to the required format, and at last, integrates and updates it into the DW.

Traditional DW uses to store static data. Business data integrated from heterogeneous data sources in a DW are used to perform strategic analysis. The data is captured, aggregated, cleaned, and analyzed to derive better decisions. The analytical decision depends not only on data processing applications but also on the derived data. Therefore, the data should be accurate, relevant, and timely. More timely data ensure better decision-making.

In traditional batch processing ETL, DW refreshment is performed in an off-line mode on a daily, weekly, or monthly basis [282]. Data is extracted from different sources; then, it is cleaned and transformed and loaded into the data warehouse. These activities are generally performed at night during the warehouse downtime. Any interference is unwanted during the loading and query processing of the DW. These historical data is stored for future analysis purpose.

1.1 E-T-L (Extract Transform Load)

ETL is a standard paradigm where data is cleaned and integrated from multiple sources and later on stored in a Data warehouse or other system. ETL is a well-defined process for building and maintaining a DW. For many years, ETL has been a reliable process for organizations to get a consolidated view of their valuable data for driving better business decisions. In 1970, ETL started to gain popularity when organizations initiated using multiple data repositories for storing various kinds of business data such as payroll, sales, inventory, etc. The number of data types, sources, and systems is increasing day by day. ETL was one of several alternatives for managing those data coming from disparate sources and blending that data into a uniform format, and finally loading it into the target system. So, the vendor-made ETL tools have become a viable solution for data-empowered business organizations.

1.2 General ETL Architecture

Nowadays, many organizations are establishing their own enterprise Data Warehouses for historical data preservation, reporting, and analytical tasks. Creation of analytical report using BI, formation of decision support system (DSS), mining of data, and finally making correct decisions is possible over the consolidated data in DW. As per Gartner[1], in the era of the Internet of Things (IoT) globally, 20 Billion devices will be connected by 2022. A large number of more innovative and connected devices will trigger a massive inflow of data. This leads to continuous data preparation, integration, monitoring, and visualization. Nowadays, the high volume of data generated from each organization needs efficient management. Data Warehousing and ETL can play a crucial role in this scenario. ETL is a systematic process for shifting data originating from multiple sources, cleaning as per requirement, and loading it into the target DW. A general ETL workflow is represented in Figure 1.1. The increasing volume of data, expanding network, budget constraints, diverse background, and high-quality data demands new challenges for each data integration vendor. Let's get a brief idea about the Extract-Transform-Loading (ETL) process.

Extract: Firstly, data needs to be extracted from different types of data sources. The origin of data sources can be text files, spreadsheets, Application Programming Interface (API), websites, sensors, OLTP databases, ERP/CRM systems, etc., having a structured or semi-structured type. These heterogeneous data should be selected and combined by identifying logical relationships within them. Sources can produce variable amounts of data with variable incoming rates. This phase has some additional tasks with incoming data cleaning and validation tasks. Logically, either full extraction or incremental extraction tasks can be performed in this phase. Some basic cleaning tasks are done during this phase, like spam check-up, reconciling extracted data with the source data, data type check, duplicate removal, key check-up, etc.

Transform: The second phase handles data transformation jobs by reshaping the data into a uniform format as per business requirements. This transform task is performed on acquired data at the previous stage. For this task, data is placed in the staging area. A set of transformation rules should be applied for this purpose. The reformatted data should be readable by any analytical tool. At this phase, data is reshaped as per the suitable format of the target data warehouse. Some common validations done in this stage are filtering, unit conversion, lookup tables, rows and columns split, merging, transposing, etc.

Load: The last phase will load the reformatted data into the connected DW. There it can be used for query purposes in the future and for preserving historical purposes. This phase is responsible for the ingestion of data into DW as per the structure of fact and dimension tables. Two types of loading can be performed: initial load and incremental update. The initial type loads all data into the DW, whereas the incremental type loads only the updated contents into the DW. The rate and interval of loading depend upon the requirement of the system.

In the traditional ETL process, batch loading is performed with some fixed time window to populate the DW. But, at present many real-life applications are showing their interest

[1] https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf

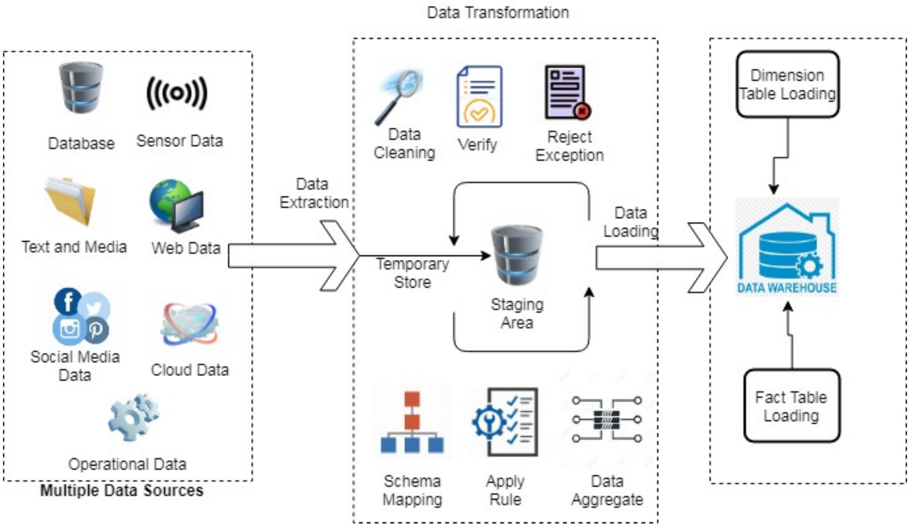


Figure 1.1: General ETL Workflow

in real-time ETL scenarios. It helps to gather timely data updating leading to data-driven BI, reporting, and decision-making. To address the application demanding low-latency and real-time operation, ETL has come with many new features, from batch to micro-batch to stream processing.

1.3 Three Steps of ETL

ETL workflow can be divided into three phases. They are Extract, Transform, and Loading. The overall procedure is shown in Figure 1.2. Some technical details of the three phases of ETL workflow are briefly described in this section. The below subsection discusses some technical issues, developments, and critical research work.

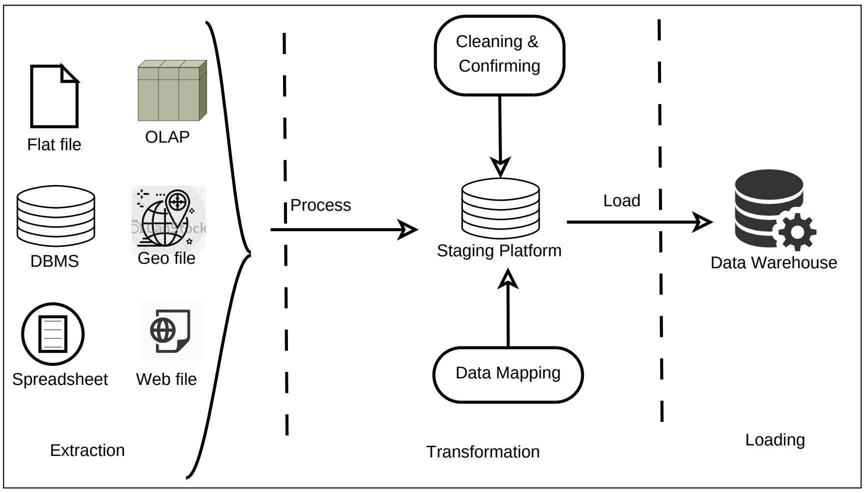


Figure 1.2: ETL Framework

1.3.1 Extract

During the ETL workflow, the first task is to identify the relevant data and extract it from source databases. Fetching raw data from different types of source systems is one of the critical tasks in ETL. Each source system can have a different type of data format. It can be flat files, XML, relational or non-relational database format or other types. After extracting, the data is converted into a unified format for further processing in the next stage.

During extraction operation, one of the main concerns should be minimum source overhead. So extraction of new data arriving at the source is a preferable approach with respect to regular interval extraction. Then, how do we identify any new data arrival on the source side? Applying **update notification** is a way to identify any changes on the source side. After getting a notification extraction process can be initiated. Otherwise, data extraction with continuous intervals can cause source overhead.

Logical Data Extraction Method

Before starting the extraction phase, at first, you need to decide how to logically or physically establish the process. Logically Two types of extraction methods can be performed.

- **Full Extraction:** In this method, all the data from the source side is extracted. This is applicable for those systems that are not able to identify which data has been changed. Because no record is maintained with respect to the previous load.

- **Incremental Extraction:** In this method, only changed data is extracted. The changed data can be identified by comparing it to the previous load. Each load record is maintained in the type of system where incremental extraction is performed. A piece of additional logical information (timestamp) is maintained on the source side. With respect to full extraction, incremental extraction has great performance benefit on ETL workflow [121].

- **Changed data capture (CDC)** A type of incremental extraction can be termed as CDC. This technique is suitable when we need to access only the new data that has been modified since the last extraction. With respect to bulk data loading, when new data can be captured, processed, and updated into the target Data Warehouse, the productivity of the overall ETL process becomes more efficient.

  For many decades CDC is a challenging research issue [13, 67, 269]. The overall procedure for detecting the changed data is shown in Figure 1.3. Several techniques have been proposed for detecting the changed data in the source system. Some common CDC solutions are the transactional log, Database trigger, Database log scraping & log sniffing, Snapshot differential, Timestamped index, etc. These mechanisms are further discussed in Section 3.1.1.

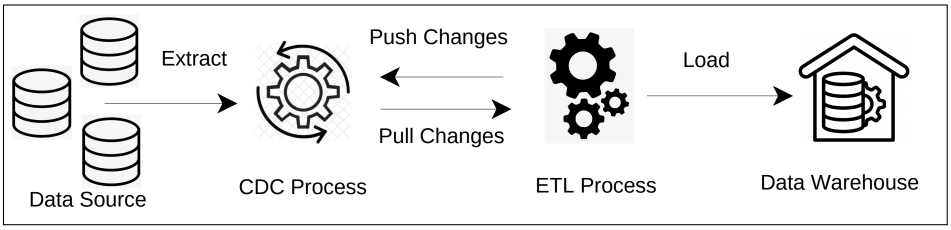


Figure 1.3: Changed Data Capture

Physical Data Extraction Methods

Physically, the extraction mechanism can be implemented in two ways. It depends on the selected logical extraction method and source-side constraints. Methods used for physical data extraction are:

- **Online Extraction** In this case, data is directly fetched from the source. The extraction process directly communicates with the source system for accessing required data. Extraction can be through an intermediate system that stores data in a pre-configured style.

- **Offline Extraction** In this case, data is staged outside the original source system. Data is indirectly fetched from that staging area. A predefined structure of data follows the total extraction routine.

1.3.2 Transform

At this phase, data is cleaned, conformed, and customized according to DW format. After extraction, data is temporarily kept in a place called *Staging Area*. It can be used as transit storage of data during the ETL process. Cleaning, transformation, and aggregation are performed on this data. This place can be treated as a manufacturing place where raw data are processed as a prerequisite demand of the data warehouse. Generally, the tables kept here are in relational database form. The user does not have any permission to access staging area data. Only the access, as well as read-write operation, is permitted for ETL processes. No query can be performed on it.

Data Cleaning Main task of this stage is to detect and remove the error from the extracted data. Generally, cleaning is performed in the data staging area. Cleaning activity deals with different conflicts. Various types of cleaning problems are identified and classified in article [282, 215].

- **Schema-level problems** mainly deals with *naming conflict* [134] where same name is used for different entity or different name can be used for same entity. *Structural conflict* [204] occur in different operational sources representing same object differently.

- **Instance-level problems** focus on removing duplicated records regarding the same attributes having a different type of representation in different sources. Suppose marital status, Sex, Date/time format, and Currency unit can have different types of representation in different sources.

To address this type of problem, there are many transformation procedures that need to apply, such as selecting, normalizing, de-normalizing, reformatting, joining, sorting, splitting, surrogate-key generation, etc. By applying the required procedures, clean data is produced.

Conforming process ensures that data has compatibility with the master data format and appropriate business logic application. Going through cleaning and confirming process data is finally become ready for loading.

Basic Transformation Types

When a set of data is extracted, they need to go through some basic transformation process. Transformation tasks assemble the data ready for analysis in the future. Some common transformation tasks are discussed here.

- **Decoding of Fields:** This is one of the common transformation tasks. For data coming from multiple source systems, you need to describe in same data item type. Coding *Male* to M and *Female* to F is one of the classic example of this task.

- **Conversion Units of Measurements:** Business organizations having global branches may require to convert various units of measurements.

- **Date/Time Conversion** This type converts Date and Time format as per the DW requirement. Suppose you need to convert from US date format (mm/dd/yyyy) to Europe date format(dd/mm/yyyy).

- **Character Set Conversion:** This technique requires the conversion of the character set as per the DW standard character set format. For example, the source data originating from a mainframe system having EBCDIC character set has to be converted into ASCII format if the DW has PC-based architecture.

- **Key Restructuring** After extraction, it is not sufficient to have a primary key. It needs to establish a key relationship within the tables. System-generated surrogated keys for fact tables and all dimension tables need to reconstruct.

- **Deduplication** The process of identifying as well as removing duplicate records is called Deduplication. For example, there can be an existence of many records for a single customer. However, DW will keep a single record for a single customer. This particular transformation type is applied in this case.

Advanced Transformation Types

- **Calculated and Derived Values:** This process performs any calculation task over the extracted data before storing it into DW. Calculating total sales, profit margin, average sales, etc., are examples of this type.

- **Splitting of Single Fields:** Separating a single column of data into multiple columns. For example, a candidate's total name needs to be split into first name, middle name, and last name.

- **Merging of Information:** This process combines different data fields into a single field. For example, product cost, description, and code come from different data sources. By merging different data fields, a single entity can be created.

- **Filtering:** Sometimes, it is required to select only certain rows or columns.

- **Aggregation:** Establish data aggregation over the data extracted from multiple sources and databases.

- **Data Validation** Some data validation rules can be applied in this segment. For example, if the first two columns of a selected row are empty, discard them for processing.

- **Summarization:** Values of the various field are calculated and summarized for loading in the DW. Suppose a marketing company wants to analyze its sales status. For this, it is required to find out different item-wise total sales. So, you need to store the summarized value.

1.3.3 Load

In this stage, processed data is finally updated to Data Warehouse. After loading, the user can access the data for further analysis. DW environment is designed by dimensional modeling technique supporting query data.

Loading can be done by updating previous data in a warehouse or adding new data to preserve historical information at synchronous intervals like daily, weekly, monthly, etc. Loading strategy [231, 116, 121] depends on the organization requirement. Different loading strategies are briefly discussed below.

- **Initial Loading** At the very first time, the data warehouse tables are loaded.

- **Incremental Loading** Periodically refreshing the data warehouse for updating the ongoing changes.

- **Full Refresh** Erasing whole data of one or more tables and updating with fresh data. Initial loading is one of a kind of initial loading.

During the loading procedure data warehouse should be in off-line mode. No OLAP queries can be applied. Either periodic loading in the batch window can be applied, or it can be in a continuous way. Standard optimization techniques should be applied to minimize the time window.

Dimensional modeling

Fact and Dimension are two essential concepts in dimensional modeling. During the loading period, the fact table is first loaded, then the corresponding dimension tables are loaded in the warehouse. Finally, dimension tables and fact tables are loaded according to the target database format, and key values logically relate to them. Before going to the next section, let us get a brief idea about dimensional modeling.

Fact table and Dimension table Fact table is the main data warehouse table containing real-world quantitative records. It resides in the central position of the star or snowflake schema of the warehouse. The fact table is correlated with the dimension table. Generally, the fact table has two types of entities. The summarized fact data that need to be analyzed and the foreign keys of different dimension tables. Dimension tables have the collection of reference information that is stored in a fact table.

Star Schema Figure 1.4 is a star schema based dimensional modeling design. The fact table is located in the center position. Four dimension tables surround it. Here *Fact_Sale* is the fact table and *Dim_Date*, *Dim_Store*, *Dim_Product*, *Dim_Customer* are its dimension tables.

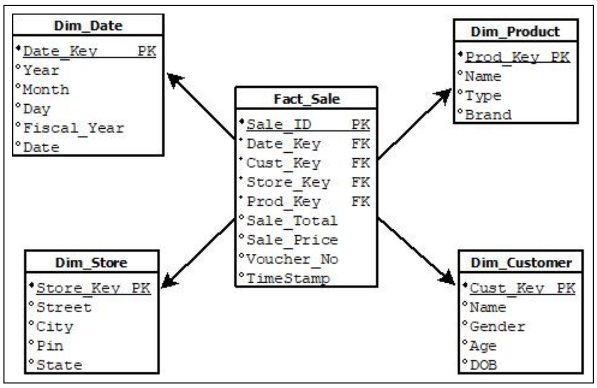


Figure 1.4: Star Schema Example

Snowflake Schema Figure 1.5 represents the arrangement of a snowflake schema. It consists of a fact table linked with multiple dimension tables. Moreover, these dimension tables are linked with other dimension tables by a many-to-one relationship.

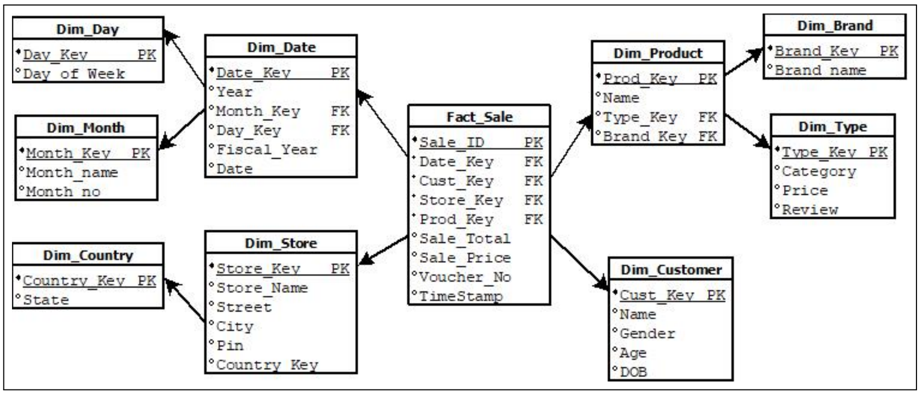


Figure 1.5: Snowflake Schema Example

Loading Mode

Before loading the file to the warehouse dimension tables, you need to determine how the loading process will be applied. There are different loading modes. Figure 1.6 shows all the loading modes using a suitable example.

- **Simple Loading** This process completely erases the data item in the target table and freshly updates the table with new data. If the target table is already empty, then this mode simply populates the new data.
- **Appending** Suppose the table's content already exists in the warehouse. Then, the new incoming data will be added as well as preserving the previous data in the warehouse.
- **Constructive Merge** In this mode, if the key value of the incoming data matches with the existing data, then the existing data is kept untouched, and the new data values are added. The new point over here is that the newly added data is marked as a supplement of the old data.
- **Destructive Merge** This mode act a little opposite of the previous mode. If the key value of the incoming data matches with the target key value, then the target data are updated. In the case when no key values are matched with the incoming record, then new values are simply added to the target table.

Loading Technique

There are different techniques available for data loading [201] into the reference DW. One of the main concerns of the loading mechanism is to minimize the warehouse's off-line time window. The developer has to decide which loading mechanism to use.

- **Bulk loading:** Without inserting row by row manner, this type executes bulk loading of rows. In this loading technique, data should be processed and saved in flat file

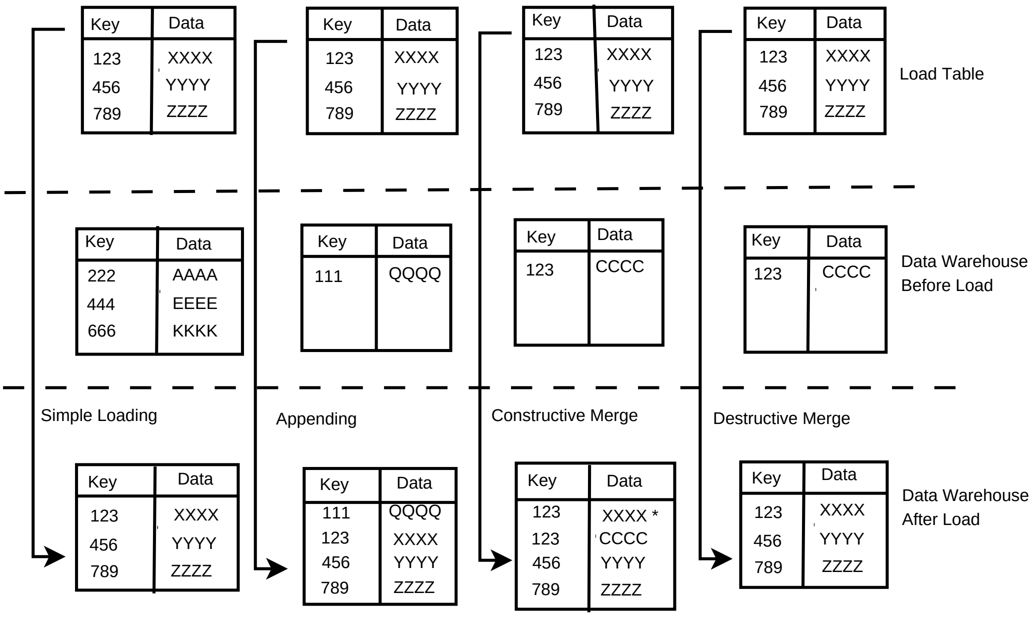


Figure 1.6: Different Loading Modes

format before loading. The loading process is shown in Figure 1.7. By using any bulk loader, popularly Oracle SQL* Loader, loading can be done to the target database. With loading Oracle SQL* loader provides some basic transformation operation also. It is ideal for the traditional batch loading technique dealing with a large scale of data.

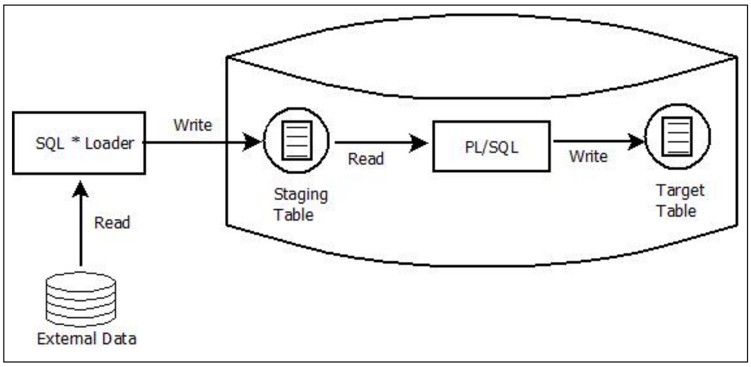


Figure 1.7: Bulk Loading

- **Load External Table:** Another loading approach for external data is implemented by Oracle's external table joining property. External tables can be joined as well as queries directly to the target database. There is no need to store the table in the staging area. Tables can be loaded directly to the target warehouse. This loading can be applied by pipelining technique also. The transformation phase can be integrated with the loading process. Unlike a normal table, the external tables are read-only. DML instructions (INSERT/UPDATE/DELETE) can not be given, and indexes cannot be added. The loading process is shown in Figure 1.8.

- **OCI and API:** Using Oracle Call Interface (OCI) application is another loading approach. It is applicable for such data tables having transformations outside the database.

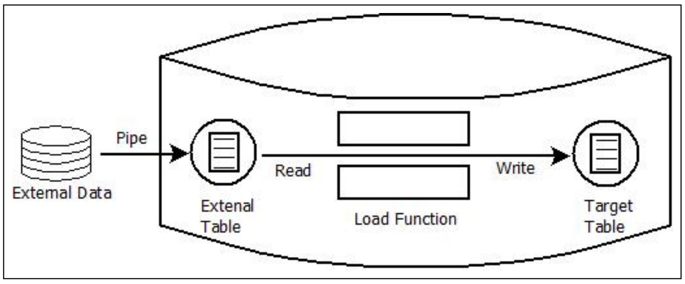


Figure 1.8: External table Loading

In this case, no flat file staging is done. Oracle has a standard API tool for loading in this way.

- **Export/Import:** This operation is done when no critical extraction is required. For loading to the target system, data is kept intact as it is coming from the source. Data can be loaded directly to the warehouse. Generally, a large volume of data is not handled in this way.

1.4 ETL Tools

ETL tools enable organizations by making the data integration task easier in hybrid environments and assembling their complex data in a presentable format. From the analysis of Gartner Inc. [2], 80 percent of organizations practice any vendor-made solution for their data integration use cases. However, hand-coded ETL is always a better option for any particular customization requirement. We have organized four categories of ETL tools.

- Batch ETL tools: These types of tools process a large amount of data at a fixed schedule of time. Most organizations employ batches of ETL jobs in off-hours. Some popular ETL tools are Informatica PowerCenter, IBM InfoSphere DataStage, Oracle Data Integrator, etc.

- Cloud native ETL tools: These types of tools are hosted in the cloud, and cloud-native data sources can be incorporated with this tool. Some of the vendors offering cloud-based ETL services are Alooma, Matillion, Snaplogic, Fivetran, etc.

- Open source ETL tools: Some dominant open source ETL tools developed by software infrastructure or researchers like Talend Open Studio, Scriptella, Apache Kafka, etc. They are publicly accessible and low-cost rather than commercial choices.

- Real-time ETL tools: Many organizations demand to access real-time data and can get modern ETL solutions from tools like Alooma, StreamSets, Confluent, Striim, etc. They are claiming to process the data stream in real-time.

[2] https://www.informatica.com/in/data-integration-magic-quadrant.html

1.5 ETL Application

When any company can integrate all its related data in a single place, it has the full potential to explore a deep insight within it. Those insights can give a competitive advantage to the companies. ETL has a great impact both on industry and academic perspectives for playing the role of data integrator. We have explored some of the most important real-life application areas of ETL technology. A comprehensive summary of the ETL process applied to various industrial sectors regarding data integration practice is presented here in Figure 1.9. Detailed analysis or up-to-date research trends in each identified sector can be found in the next section. Interested readers can get an overall visualization of ETL's usefulness in our modern life nowadays.

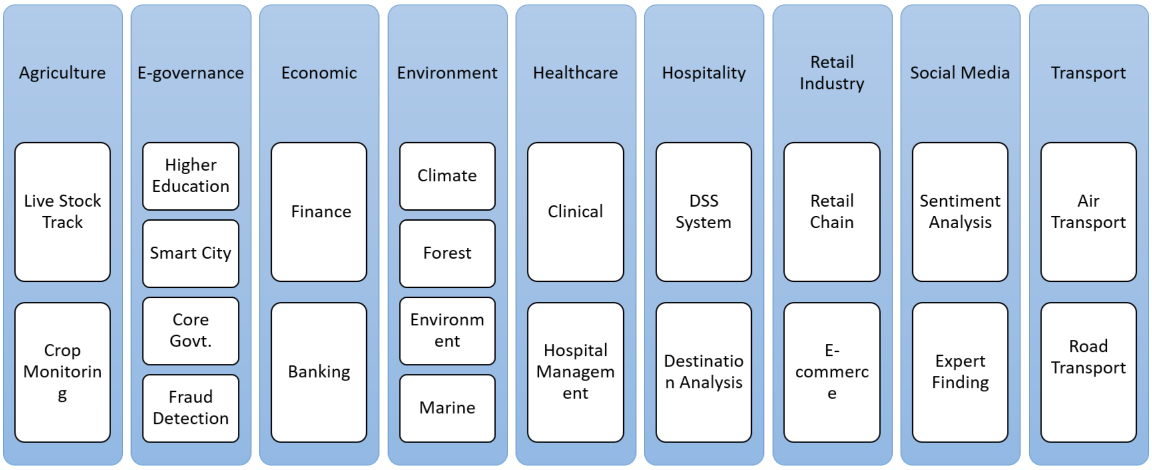


Figure 1.9: ETL Application Area

Considering the importance of current technological trends and the lack of factual research leads us to explore a complete scenario of research activities in ETL processing. In the next section, an in-depth literature study on the real-life application of the ETL process. A brief discussion about ETL processing stages as well as current industry data management trends is discussed here.

1.6 Research Problems and Challenges

The importance of ETL processing in any organization is directly proportional to the dependency of its data warehousing process. In accessing those data, ETL has a significant role in creating and maintaining a data warehouse. ETL is a widely used process that identifies and extracts data from various sources, filters and customizes according to the required format, and in the end, integrates and updates those data into a data warehouse. Incorrect data in the warehouse can mislead business analysis as well as decisions. So, a well-maintained ETL process is one of the key factors for a successful data warehouse implementation. Based on the report [68], designing a well-established ETL workflow consumes almost one-third of the cost and effort in a DW implementation. A well-designed ETL process is an essential aspect of accomplishing an effective DW. Researching the domain of ETL processing is a reasonable research goal.

The life cycle of an ETL process begins with the conceptual modeling task [247, 276]. At this stage, this model identifies the data sources and the involved processes. This model represents only the highest-level relationships among the entities. The second stage of model designing is the logical modeling task [287, 243]. This logical data model describes the data structure, rules, intermediate processes, and their relationship to the overall system. The database-specific implementation details are described in the third modeling stage, named physical modeling [159, 278, 28]. Execution parameters are defined in this model. Simulation is a widely used method to analyze system behavior. Modeling simulation can provide a clear insight into any complex system. Very few works have been found in the domain of ETL process modeling [178, 179].

ETL tools can be broadly categorized into two types, e.g., GUI-based tools or programmable ETL tools. There are many popular GUI-based ETL tools available in the market nowadays. These tools have very easy-to-use modules which are suitable to use by non-technical people. Still, many organizations believe in creating their ETL solution by making their own code. This code-based approach can offer much scalability, customization, and performance optimization. Some academic developments are done [273, 23, 189], but still, many open scopes exist.

Setup of ETL workload in any DW environment is one of the most time-consuming tasks. The automated ETL process can offer many benefits by reducing the time required for manual intervention and operation coordination within the organization. ETL automation is an appealing research goal [52, 265, 212]. The automated ETL solution can offer a data integration team to design, execute, and monitor the overall ETL workflow in an organized way.

The way organizations access data is rapidly changing. Nowadays, organizations want to access real-time transactional data to make an immediate decisions. Currently, many industries such as stock exchange, e-commerce, telecommunication, air traffic control, etc., have the requirements to correct reports based on fresh data in a Data warehouse as operational decisions can be made speedy. However, this cannot be performed on the status report of yesterday. The real-time ETL process ingests the data into the data warehouse very quickly as soon as they appear on the source side. So, how to define fresh data? Freshness is signifying from minutes to seconds or sub-seconds of data flow delay. The trends of “near real-time" [120, 54, 294] or “real-time" [126, 59, 130] is going to be the new challenges in technological solutions. Some commercial systems are working towards getting fresh data in the Data warehouse [291, 34].

To address the current technological demand, many organizations are switching from their traditional ETL set up to cloud-based ETL solutions [209]. Cloud-based ETL solutions can offer real-time data processing, scalability and smooth integration variety of data sources with increasing volume. From academic background some cloud-based ETL proposal are [155, 154, 203]. Migration existing workload in the cloud is still a challenging and open research issue.

1.7 Contributions and Outline of the Dissertation

Conceptual Modeling Data generated from various sources can be erroneous or incomplete, which can have a direct impact on business analysis. ETL (Extraction-Transformation-Loading) is a well-known process that extract data from different sources, transforms those data into the required format, and finally loads it into the target Data warehouse (DW). ETL performs an essential role in the Data warehouse environment. Configuring an ETL process is one of the key factors having a direct impact on cost, time, and effort for the establishment of a successful data warehouse. Conceptual modeling of ETL can give a high-level view of the system activities. It provides the advantage of pre-identification of system error, cost minimization, scope, risk assessment, etc. Some research development has been done for modeling the ETL process by applying UML, BPMN, and Semantic Web at the conceptual level. We have proposed a new approach for conceptual modeling of the ETL process by using a new standard Systems Modeling Language (SysML). SysML extends UML features with much more clear semantics from a System Engineering point of view. We have shown the usefulness of our approach by exemplifying using a use case scenario.

Model Simulation SysML language, standardized by OMG, is proposed to model and study any system. The OMG standards support the specification, design, analysis, verification, and validation of any system. Simulation is a common practice to estimate system performance. To handle the increasing complexity of any system model, it is preferable to go through the verification and validation process in the early stage of system development. Model-based systems engineering (MBSE) is one of the current system engineering methodologies which covers all of the key aspects of system modeling. It combines various aspects of the system model from requirements analysis, design, and simulation throughout the system development life cycle.

In this work, the proposed ETL model using SysML language is executed within SysML modeling tool (Magic Draw) uses some specific plugins. A SysML conceptual model of ETL is designed in our previous work. In this proposal, we are extending our previous work and presenting an MBSE-based tooled approach to automate the SysML models validation with the help of the No Magic simulator. Here The main objective is to overcome the gap between modeling and simulation and to examine the performance of the SysML model.

Empirical Analysis of ETL Tools ETL (Extract Transform Load) is the widely used standard process for creating and maintaining a Data Warehouse (DW). ETL is the most resource, cost, and time-demanding process in DW implementation and maintenance. Nowadays, many Graphical User Interfaces (GUI) based solutions are available to facilitate the ETL processes. In spite of the high popularity of GUI-based tools, there is still some downside to such an approach. This work focuses on the alternative ETL developmental approach taken by hand coding. In some contexts, it is appropriate to custom develop an ETL code that can be cheaper, faster, and maintainable. The contribution of this work is to highlight a new area by programmable ETL development technique. For this purpose, Some well-known code-based open source ETL tools (Pygrametl, Petl, Scriptella, R_etl Package) developed by the academic world has been studied in this proposal. Their architecture and implementation details are addressed here. An in-depth experimental evaluation is done on each tool. Subsequently, a feature-wise and performance-wise analysis report is provided. The aim of this work is to present a comparative evaluation of these code-based ETL tools. Not to acclaim that code-based ETL is superior to GUI-based approach. It depends on any organization's particular requirement, data strategy, and infrastructure to choose the path between code-based and GUI-based approach.

Near Real time ETL The focus of this previous work was to give an integrated analysis report in the research domain of a programmable ETL system. Afterward, a new solution model is proposed to meet the near real-time ETL demand. The target is achieved by implementing an incremental loading model with the assistance of the CDC (Change data capture) approach. The contribution of this work is to highlight a new area by programmable ETL development technique. The continuation of work is done through designing a new ETL-based data integration technique. The proposed solution makes the data integration more efficient by incrementally populating only the changed data in the DW at the right time.

Automated ETL process In the current business landscape, real-time analysis of enterprise data is very crucial for decision-makers of the organization to take strategic resolution and stay ahead of the competitors. Most of the time, it happens that data is outdated by the time it reaches the user. The organization needs reliable, up-to-minute information to make better proactive business decisions and improve the process and organizational efficiency. Availability of information and business-critical report in real-time can be achieved through an automated ETL process. Typically, running a data warehouse in an enterprise requires the coordination of many operations across many teams, including applications and database teams. Also, it requires a lot of manual intervention, which is error-prone. Executing all related steps in correct sequences under accurate conditions can be a challenge. The automated ETL process helps to address all these problems. Moreover, the pre-processing of data is a crucial step for making data ready to load in a data warehouse for analysis. Machine learning-based pre-processing can be used to ensure the quality of data. In this work, we have addressed the issues faced in traditional data warehouses related to availability and the quality of data. We have explained how to automate the ETL process and how machine learning can be leveraged in the ETL process so that the quality and availability of data have never been compromised and reach the user on a near real-time basis.

ETL in Cloud Extract-Transform-Load (ETL) consists of a series of process which collects raw transactional data and reshapes it into clean information which is actionable by Business Intelligence in the future. Presently most organizations are considering moving towards cloud-based implementation for their mission-critical applications. This trend is also affecting the management of ETL processes in the Data warehouse environment. The limitations of the traditional ETL process and the benefits of moving ETL into the cloud are discussed in this work. After that, challenges in cloud computing adoption regarding the ETL process are identified. Features offered by some leading cloud-enabled ETL solutions are incorporated herewith some brief analysis. This work will also cover the general issues in cloud ETL both from the perspective of cloud consumers and service providers. A novel framework is designed to process streaming data coming from a real-time data feed. It is an Apache Spark-based framework that enables processing, querying, and analyzing Big Data. This framework has the potential for near real-time data processing and in-memory data storage system results in multiple times faster than other Big Data-enabled technology. The solution facilitates the rapid development and deployment of near real-time ETL applications.

1.8 Report Layout

The overall thesis is divided into three parts as follows.

Part I discusses some terminology and state-of-the-art related to this work. In the second chapter, some terminologies are related to this research work. The third chapter presents a study on ETL modeling, Various types of ETL tools, and the real-life application area of ETL processing. The overall ETL modeling technique is divided into three parts: conceptual modeling, logical modeling and physical modeling. ETL tools are discussed in three sections: academic development approach in ETL tools, programmable ETL tools and cloud-based ETL tools. The ETL application section is classified into nine broad domains. They are Agriculture, E-governance, Economic, Healthcare, Hospitality, Retail, Environment, Social network, and Transport industry.

Part II focuses on the Research Proposal, which includes four chapters. Chapter four discusses a new Conceptual ETL Process Modeling approach. The modeling work is done by SysML language. Chapter five represents the simulation process of the designed Conceptual ETL Model. Chapter six presents some empirical analysis of programmable ETL tools. For the experimental purpose, the selected ETL tools are Pygrametl, Petl, Scriptella, and R_etl. Chapter seven proposes a new real-time data integration framework using an incremental loading approach.

Part III presents Research Proposal on Recent Trends in the ETL process. It includes two chapters. Chapter eight discuss some case study of ETL in retail, marketing, and financial service domain. A new solution is proposed for the automated data integration technique. Some discussion is there about the significance of machine learning in ETL automation. Chapter nine is about moving the ETL processing task to the cloud. A new Apache Spark-based ETL framework is designed here. The last chapter presents the conclusion of the overall work and some perspectives on the future scope.

Part I

Preliminaries

Chapter 2

Terminology

2.1 Related Terminology

In this section, we have discussed the essential terms related to the research work of the thesis.

Data Warehouse It is the central repository for a particular organization for storing historical data. The data record can be used for performing complex queries for analysis and decision support. Data warehouse consists of integrated multiple heterogeneous data sources like relational database, flat files, and online transaction records in a uniform format. Nowadays, active Data warehouse [130] are coming with real-time features associated with business intelligence applications. A suitable decision is triggered automatically for analyzing warehouse data based on a predefined active rule.

Fact table and Dimension table Fact table is the main table in a Data warehouse that contains real-world quantitative records. It resides in the central position of the star or snowflake schema of the warehouse. The fact table is correlated with the dimension table. Generally, the fact table has two types of entities. The summarized fact data that need to be analyzed and the foreign keys of different dimension tables. Dimension tables have the collection of reference information that is stored in a fact table. The definition is explained with an example 2.1 and 2.2. From 1st row of the fact table, the customer named A. Paul has customer key 6, and other details are listed in the dimension table has brought 3 units of product on day 5 from store 3. There will be five dimension tables corresponding to this fact table.

Date_id	Product_id	Cust_key	sell_quantity	Store_id
5	35	6	3	3
6	28	9	4	1
6	18	4	2	2

Table 2.1: Fact table

Cust._key	Cust_name	Cust_gender	Cust_city	Cust_contact
4	A. Banerjee	M	kolkata	9832238733
9	B. Mukherjee	F	Burdwan	9932238333
6	A. Paul	M	Darjeeling	9832233766

Table 2.2: Dimension table for Customer

Data Mart It can be called a subset of a data warehouse. The data mart is also a miniature repository of data that focuses on a particular functional area. It is customized for each department within an organization. For example, in an organization, stock, sales, finance, and HRM departments can access their own data mart explained in Figure 2.1.

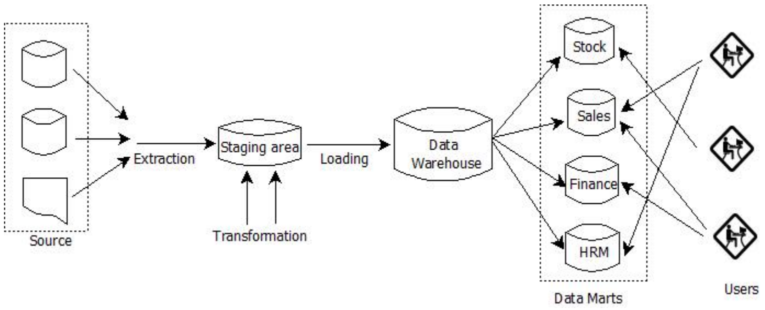


Figure 2.1: Data Mart

OLAP Online analytical processing (OLAP) is a data analysis tool mainly used to access the online live data and to analyze those data shown in 2.2. OLAP gives a user-friendly environment for interactive data analysis on a very large database[193]. Data warehouse primarily concentrates on historical data for further analysis, and OLAP analyzes those data in an application-oriented manner. The data are organized in multiple dimensions, and each dimension contains multiple levels of abstraction. Such an organization provides their user the flexibility to view data from different perspectives, and there exist several OLAP operations on data cubes that provide interactive querying and analysis of the data.

OLTP On-line Transaction Processing (OLTP) data modeling approach facilitated a system for processing a large amount of online transactional data. It supports fast query processing and ACID (Atomicity, Consistency, Isolation, Durability) property. By applying OLTP, the system is enabled to give an immediate response to user requests [57]. Automated teller machine (ATM), banking transaction management systems, and ticket reservation systems are an example of using OLTP. Here data resides in 3NF form. OLTP database always contains current date business transactions. Figure 2.2 explain the overall procedure.

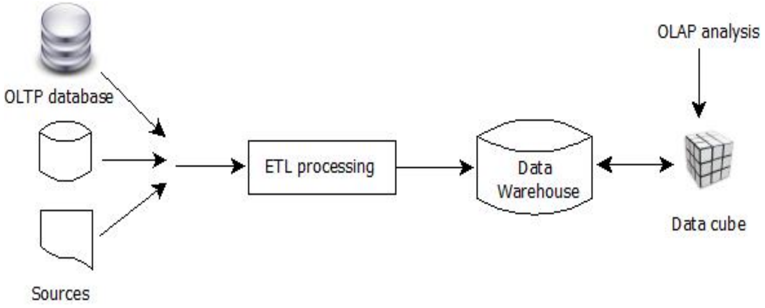


Figure 2.2: Data Mart

Staging Area It can be used as transit storage of data during the ETL process. It is a place where data is kept temporarily after extraction. Cleaning, transformation, and aggregation are performed on this data. After transformation, data is loaded to the data warehouse. This place can be treated as a manufacturing place where raw data are processed as a prerequisite demand of data warehouse. Generally, the tables kept here are in relational database form. The user does not have any permission to access staging area data. Only the access, as well as read-write operation, is permitted for ETL processes. No query can be performed on it.

Refreshment Cycle There are several stages in the ETL workflow. Data identification and extraction, placing data in the staging area for cleaning and transformation, and at last loading it in the Data warehouse periodically. Updating fresh data to the warehouse is termed a refreshment process. Furthermore, the total workflow is called the refreshment cycle.

Slowly changing dimension (SCD) SCD is a particular type of dimension that preserve and maintain both present and historical data over time. Basically, SCD keeps track of historical records of dimensions in the data warehouse. In the ETL process, keeping track of dimensions is one of the critical tasks. Types of different SCD techniques are discussed below.

- **SCD Type 0** This method is used to keep the original record. It is a passive method. Values remained unchanged.
- **SCD Type 1** This method replaces the existing record. No track of the old record is preserved. For example, the following table store the original record.

Vendor_Key	Name	State
1101	Harry	London

Table 2.3: Original table

If Harry moves from London to Glasgow, then the new Table 2.4 will be as follow.

Vendor_Key	Name	State
1101	Harry	Glasgow

Table 2.4: SCD Type 0

- **SCD Type 2** This method keeps the full history of the changed data. A new record is added to the dimension table. Both original and new records are kept. For example, of the original Table 2.5 below.

Vendor_Key	Name	State
1101	Harry	London

Table 2.5: Original table

After Harry moves from London to Glasgow, the new record is added as a new row. A new surrogate key is added with the new record. Example of this type is shown in Table 2.6.

Vendor_Key	**Name**	**State**
1101	Harry	London
1102	Harry	Glasgow

Table 2.6: SCD Type 2

Another way to implement this type of SCD is by adding ‘effective date ’columns. Example of this type is given in the Table 2.7 below. NULL in the End_date cell represent current tuple version.

Vendor_Key	**Name**	**State**	**Start_date**	**End_date**
1101	Harry	London	01-Jan-2016	31-Aug-2016
1102	Harry	Glasgow	1-Sep-2016	NULL

Table 2.7: SCD Type 2

- **SCD Type 3** This type store limited history. The original record is modified to preserve the changed data. A simple example of this type is given in Table 2.8 below. This method keeps history without increasing the size of the table.

Vendor_Key	**Name**	**Original State**	**Present State**	**Effective Date**
1101	Harry	London	Glasgow	31-Aug-2016

Table 2.8: SCD Type 3

- **SCD Type 4** This process is term as "history tables". One table store the present data, and the other one store all the changed information. A suitable example is given below, where *Vendor* Table is the original table, and the *Vendor_History* Table keeps the history of all changes.

Vendor_Key	**Name**	**State**
1101	Harry	Glasgow

Table 2.9: Vendor

- **SCD Type 6** This is a hybrid method which combines the previous three types of SCD 1, 3 and 3 $(1 + 2 + 3 = 6)$. an example of this type is explained in the Table no 2.11 below.

Here *Present State* store current value, *Historical State* store historical value, *Start Date* store effective starting date, *End Date* store effective ending date and *Flag* store the updated record.

Vendor_Key	Name	State	Create Date
1101	Harry	London	01-Jan-2016
1102	Harry	Glasgow	31-Aug-2016

Table 2.10: Vendor_History

Vendor _Key	Row _key	Name	Present State	Historical State	Start Date	End Date	Flag
1101	R1	Harry	London	London	01-Jan-2016	31-Aug-2016	N
1101	R2	Harry	Glasgow	London	01-Sep-2016	31-Dec-2016	N
1101	R3	Harry	Oxford	Glasgow	31-Dec-2016	20-Feb-2017	Y

Table 2.11: SCD Type 6

Change Data Capture (CDC) It is a new data integration approach that identifies and captures changes that occur to data sources and delivery only the changed data to the operational system [70]. This approach does not need data warehouse downtime or batch windows of ETL. Some CDC technologies operate in batch mode with a pulling technique. This means the ETL tool periodically receives a batch for all new changes made up to the last receive and executes them. The real-time CDC solutions apply a continuous streaming "push" approach for delivering data. The data changes are captured and delivered immediately to the target. The procedure is explained in Figure 2.3.

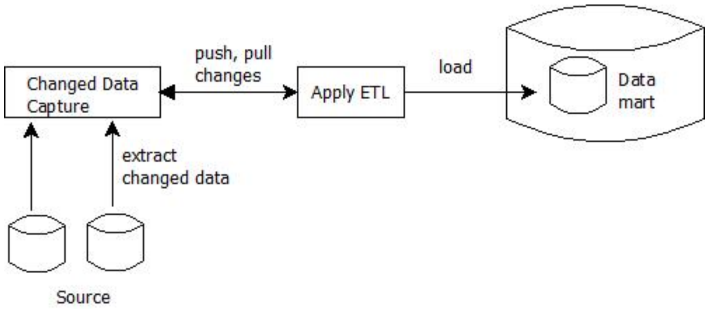


Figure 2.3: Change Data Capture

It is a cost-effective solution. The advantage of CDC is the latency can be cut down to minutes to even seconds which makes the data instantly available, eliminating the use of batch windows. Besides, it minimizes the amount of data flow; therefore, the resource requirement is minimized, and speed and efficiency are maximized. CDC addresses some business needs like building Operational Data Stores (ODS), Business Activity Monitoring (BAM), Application Integration, Real-time Dashboards, data quality improvement, etc. There are several techniques for detecting the change are addressed in the literature [13, 45].

RTDC In a real-time scenario, it is very difficult to implement simultaneous query processing and refreshing warehouse operation. To resolve this problem, an external database named real-time data cache (RTDC) is used to temporarily store real-time data outside the warehouse. It can work as another instance of the warehouse for storing real-time data. From RTDC, real-time data can be accessed, and loading can be done periodically to the actual warehouse.

Business Intelligence It is a tool that processes data and produces some meaningful reports which help to make better strategic planning of an organization [158, 61]. It is a Decision Support System (DSS) designed for giving a predictive analysis from large data from which effective action can be taken. BI has a collection of applications that extract important business data from an organization, apply some analysis operations like text or data mining, online analytical processing, and other statistical analysis methods then generate a simplified predictive report for the end user.

Data Stream It is a continuous real-time high speed transfer of data [272, 93]. There are many real-life applications like network traffic analysis, sensor data processing, web tracking, e-commerce, etc., where real-time data should be captured and processed. Here data stream technique is required rather than stored data. In data stream technology, data is continuously changing, and an uninterrupted data integration process should be applied to the data warehouse tolerant of zero latency. It is a new challenge for data stream management systems (DSMS) to handle rapidly moving data streams and apply non-stop query processing in real-time.

The definition of a data stream based on the literature[93] is given below:

A data stream S at t_i time is denoted by $S(t_i)$. $S(t_i)$ is defined as

$$S(t_i) = \{< (a_0, t_0), m_0 >, < (a_1, t_1), m_1 >, ..., < (a_i, t_i), m_i >\}$$

for $1 <= i <= n$. Where a_i is a set of attribute values of attributes $A_1, A_2, ..., A_n$ with domains $dom(A_i)$ and a stream starts at a time t_0 and t_i is a time-stamp of attribute value pair from a discrete, monotonic, infinite time domain T.

UML Unified Modeling Language (UML) is standardized (1997) by the Object Management Group (OMG) is a general purpose visual language comprised of a set of diagrams that helps to visualize, construct, specify and finally, documenting any software-based system. This diagram graphically represents any system model. UML diagram can present different views of a system model. Static view addresses the static structure of a system using class diagrams and composite structure diagrams. This type of diagram represents the objects, operations, attributes, and relationships. Dynamic view addresses the dynamic behavior of any system using activity diagrams, sequence diagrams, and state machine diagrams. These diagrams show collaborations among objects and the internal state changes within the objects.

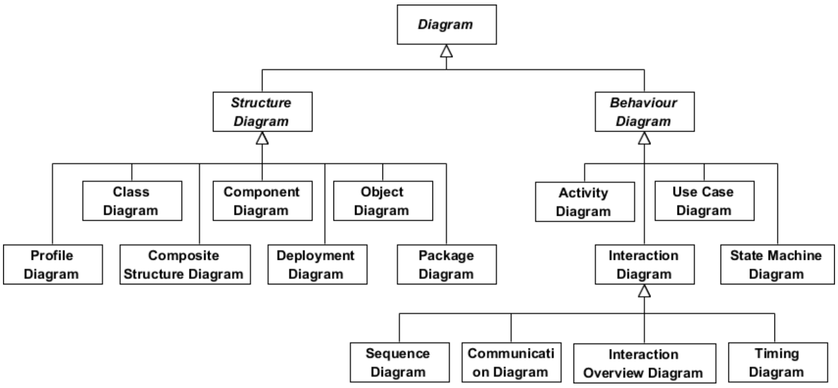


Figure 2.4: Types of UML diagrams

BPMN BPMN stands for Business Process Model Notation and consists of standard graphical notations which helps to understand business processes within an organization. It has been maintained by Object Management Group (OMG) since 2005. It is very similar to the activity diagram of the UML language. BPMN helps to support business process management both for the business users and the technical users. This language consists of simple notations easily understood by business users but still has the ability to present any complex business process semantics.

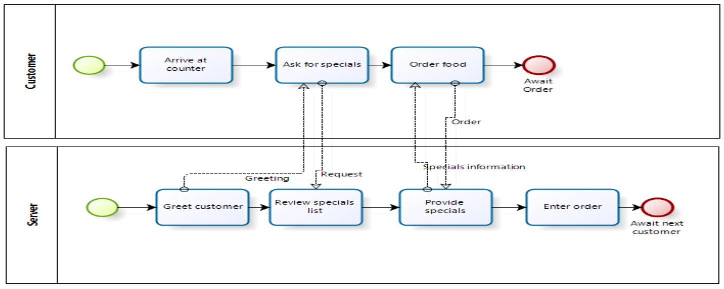


Figure 2.5: BPMN diagram example

SysML Systems Modeling Language (SysML) is a graphical modeling language jointly maintained by OMG and International Council on Systems Engineering (INCOSE) from 2003. It is an extension of the UML2 language for modeling any system structure, requirements, behavior, and parametric details. The purpose of the graphical language is to design, analysis, and verification of complex system design.

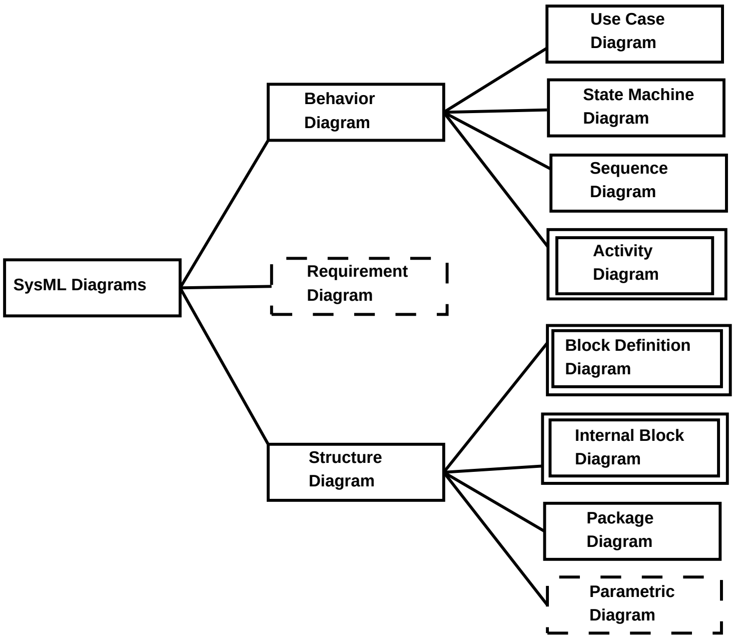


Figure 2.6: MBSE Features

Chapter 3

State of the Art

3.1 ETL Process Oriented Approach

In this section, we will organize research efforts on extraction, transformation and loading phase of ETL. Various development has been done in each phases of ETL from research community as well as industry experts. We have categories different challenges and solutions over each ETL phases in the following manner.

- Extraction Based Technique
- Transformation Based Technique
- Loading Based Technique

Gradually each approaches will be discussed with their significant research and development issues.

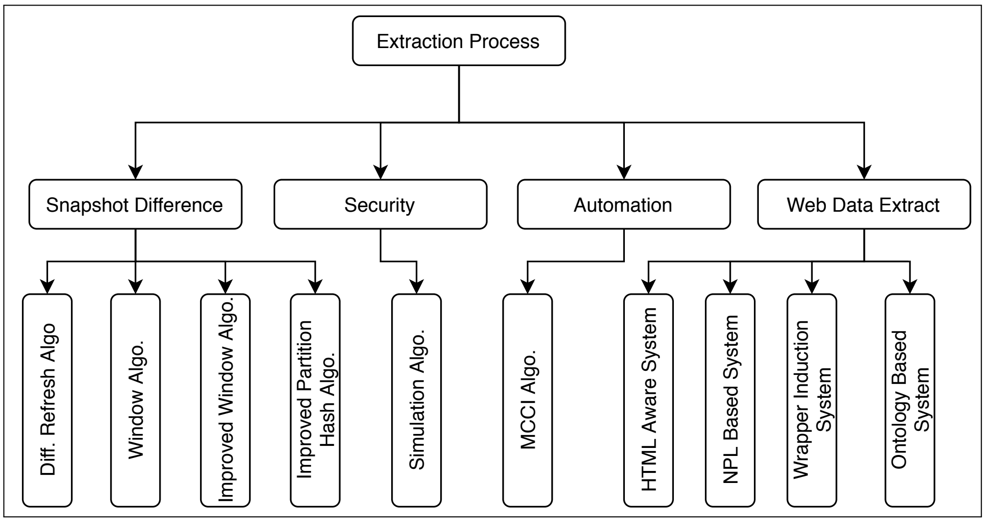


Figure 3.1: Extraction Techniques in ETL

3.1.1 Extraction Based Technique

The first task of the ETL workflow is to collect and manage relevant data situated in different data sources. Each source system can have different type of data format. It can be flat files, XML, relational or non-relational database format or other types. Moreover, the data source type can be heterogeneous. After extracting from the sources, the data is converted into a unified format for further processing in the next stage. We have categories different research flow in Figure 3.1 about ETL extraction phase. They are Snapshot Difference, Security issue, Automation and Web Data Extraction process.

Snapshot Difference Problem

In incremental extraction method only changed data is extracted compared to previous load. Formally change data capture (CDC) technique is implemented by comparing two snapshots of previous extraction and current extraction.

First research proposal towards identifying changed data was explained in literature [153] at 1986 by Bruce Lindsay et al. They have proposed an algorithm for refreshing snapshots.

After that significant research approach towards identifying changed data was explained in literature [145] in 1996 by Labio and Garcia-Molina. They have presented a snapshot differential algorithm which enable to identify insertion, deletion and updates between two snapshot. Different outer join algorithms are discussed in the essay including sort-merge join, partitioned hash join method. Application of compression technique for reducing the data and I/O memory size along with those algorithms are also mentioned.

By using sort-merge outer join algorithm snapshot difference can be identified comparing two snapshot input assuming first one is sorted. Finally, they have presented a *window* algorithm. In this algorithm each snapshot contains a *input* buffer and *aging* buffer. Two input buffer fetch input from two consecutive files and compare those data. Similar tuple are not considered. Unmatched tuples in input buffer are compared to the aging buffer of both snapshot. Again similar tuple are not considered. Finally, remain unmatched tuples in input buffer are the data identified for insertion or deletion. These tuples are pushed to the aging buffer. By keeping track the oldest tuple of the aging buffer is emptied if it is full. Window algorithm performs well if similar records are physically stored in same location of two snapshots.

An improved window algorithm [150] is proposed at 2010 for increasing the efficiency of snapshot differential algorithm. Additional cyclic redundancy code is applied with the previous algorithm which minimize I/O overhead. Finally, by showing a simulation result this algorithm efficiency is proved better compared to previous one.

An advanced Snapshot Differential Algorithm is proposed in article [303]. They have modified the traditional Partition Hash based algorithm for improving the time complexity. Window algorithm is an efficient one, but it can not handle data concurrency. For that reason they have chosen Partition Hash algorithm. For handling a bulky range of snapshots, their proposed work is implemented in Hadoop platform. As the working principle of Hadoop is based on "Divide and Conquer" rule. The huge data is split into several parts and each parts are processed separately and concurrently. After that the parts are merged. The final result is showing improved performance compared to traditional process.

Most recently at 2015 snapshot maintenance approach are proposed [211] for real-time environment. We can get a comparative view of research work on this domain in Table 3.1.

Proposed Work	Advantage	Disadvantage	Algo. Type	Platform	Complexity	Feature	Ref.	Year
Differentral refresh Algo.	Detects all changes and empty block	Message reduce & update cost	Time-stamp based Algo.	R*	Improved	Full refresh	[153]	1986
Sort merge outer join & Window algorithm	Identify difference	Well if records are closer	Hash Partition algorithms	Corba distributed object framework	$\|f_1\| + \|F_2\| + \|f_2\|$	Difference with Joins, compression	[145]	1996
Improved Window Algo.	minimize I/O overhead	Well if records are closer	Incremental Algo	Corba distributed object framework	Improved	Snapshot compress, CRC based	[150]	2010
Improved Partition Hash based Algo	Concurrency control, Efficient	in disk operation, high network traffic	Differential Algo	Hadoop platform	Improved	Map, Reduce	[303]	2011

Table 3.1: Overview on Snapshot difference work in ETL extraction

Security issue

At 2007 M. Mrunalini et al.[180] designed a conceptual level model for secured data extraction in ETL using UML language. They have validated their model by taking an banking application. After that in 2009 they [178] worked on implementing security aspect in the extraction process. By using UML 2.0 notations they have designed secure extraction method in ETL process in various UML behavioral diagrams. A password Regime is applied which can prevent attack like spoofing, flooding, unauthorized database access etc. Additionally records are kept encrypted. Any type of error is detected and prevented by Security monitoring system.

They have extended their work [179] by an automated security assessment process of ETL. Analyzing of the ETL process is done under two security aspect: vulnerability index and security index. A framework is developed for measuring the of the ETL system at the early phase. A new simulation tool SeQuanT, is developed for quantifying system security. For assessment of security metrics sensitivity analysis is done. It will show the security level of the system and desired security requirements for the system.

Automation

Article [305] by X. Zhang and W. Sun highlighted a new approach for automation of incremental process in ETL extraction. At begining the ETL process is canonicalized which only deals AUSPJ (Aggregation, Union, Selection, Projection, Join operator) and D (Difference operator) segments. By using previous maintenance methods [101, 42, 100] they have proposed an incremental maintaining algorithm formally named MCCI. This algorithm select incremental maintenance method with minimum cost for AUSPJ and D segments. Finally the implementation technique is briefly discussed.

Web Data Extraction

In todays scenario, websites contains a lots of information having various format. HTML pages can be three types: structured, semi-structured and unstructured. It is a difficult task for extracting different types of web data from various sources. The basic concept is to segment data records, extract those data and put it in a database table in a structured way. This task can be addressed as converting web data from unstructured to structured format.

Data extraction from websites is mainly done by using wrappers. Wrapper is a well known process which is used to access HTML data and further convert it to structured format (example XML format). Some of the commercial wrapper producer system for solving the web data extraction problem [25, 82, 79] discussed below.

- **HTML-aware System:** Some tools are based on formal structure of HTML pages for data extraction process. Some well-known commercial systems on this categories are Runner [58], Lixto [24] and W4F [229].

- **NLP-based System:** Natural Language Process based system [31, 162, 84] was applied for solving some certain problem like collecting facts resulting from speech transcriptions in email, resume, blog, newspaper etc. RAPIER, SRV and WHISK etc. are some well known tool.

- **Wrapper Induction System:** These type of tool automatically generates some rule based wrappers [14]. It is able to extract data from tree structured sources like XML, HTML.

- **Ontology-based System:** Main focus if this type of system is the data itself, not the page structure. Ontology can be used for certain domain applications like Bioinformatics, Social networking etc. Some ontology based approach for web data extraction is done in literature [104, 267].

The Table 3.2 describes various issues along with their solutions on web data extraction method of ETL.

Category	System Name	Advantage	Limitation	Platform	Feature	Auto mated	Ref.	Year
HTML Aware System	Lixto	GUI mode, interactive navigation	Client based version	Java-based HTML/ XML wrappers	Deep Web macro recording	Y	[24, 26]	2001, 2005
	Road Runner	No priori info need	Works on regular structured source	Java	For intensive Web sites, ACME algo	Y	[58]	2001
	W4F	light weight wrapper	Not flexible	Java	HTML to XML convert	Y	[229]	1999
NLP Based System	-	Maximum entropy modeling	Theoretical approach	context-sensitive modeling	-	Y	[31]	1996
	RAPIER	Learns unbounded patterns	Single-slot	-	Template based	Semi-auto	[175]	1999
	WHISK	Graphical interface	Doesn't support complex object	Text extraction rules	Multi-slot	Semi-auto	[257]	2000
Wrapper Induction System	Xpath	Continuous extraction,	Lack of robustness	Hand-craft wrappers	DOM tree	Semi-auto	[14]	2005
	WIEN	Customized, Fast to learn and extract	Cannot handle missing items	HLRT, OCLR and HOCLRT Algo	supervised learning process	Semi-auto	[143]	2000
	Soft-Mealy	Learns order of items	Can't generalize unseen separators	Java, finite-state transducers (FST)	Token based approach	Semi-auto	[112]	1998
Ontology Based System	-	Extract from Tables	Cannot extract other form of data	RDF graph	Generalized table structure	Y	[267]	2006
	BYU	Less labor-intensive	Specific domain based application	Ontology lexicons	Conceptual model	Y	[73]	1999

Table 3.2: Overview on ETL security and Web data extraction

Changed Data Capture (CDC)

Recently many research works are going on for efficient data extraction task in real-time ETL environment. CDC is a unique data integration approach [13, 242, 269] where only the new data is pull out from the enterprise source system since the last extraction. CDC mechanism can be implemented in number of ways. Multiple CDC solution [45, 67] can be set up for a single system. Some common type of CDC solutions are:

- **Transactional Log:** Most of the DBMS system maintain a change log which keeps record of all changes (add, delete and update) performed on it. This transactional log [138, 208, 242] keep track of date and time for last modification in the table. Generally the log columns are added to the end of each table. The extraction process verify the log file and select the new transaction. Analyzing those log file do not effect on operational database. Only point to remember that, the extraction should perform before the transaction log refreshing operation. Because, when the disk storage of log file became fill up, they are moved to another place. This process can be used to capture the changed data in real time. But it is not applicable for non database application files.

- **Database Trigger:** Trigger is a special type of activity in database system which is fired basis on some predefined function. Trigger can be set on every (add, delete and update) event for finding new data [280, 258, 262]. The output of the trigger program which is stored in another file can be used for extracting data. This type of application is suitable for source system having database application. But the trigger based system can have a performance impact on source system.

- **Database Log Scraping & Log Sniffing:** This technique [138, 160] takes a snapshot of the transaction logs maintained by the database system for backup and recovery at a scheduled time. Changes are identified from the transaction log. The log scraping approach has higher latency value as well as considered a miserable approach. The later techniques involves “pooling" of the active log file and identify changes on real time. Now a days some real time ETL service providers are using log sniffing application.

- **Snapshot Differential:** During the extraction phase a snapshot of complete source table is taken. Changed data can be identified by comparing consecutive snapshots. This method is termed as snapshot differential technique [145, 153, 211]. This method is bound to keep previous copies of all relevant source data. Sometimes comparing two rows in a large file is an unskillful option. If none of the other techniques are feasible then this can be the last option. It can be applied to any type of data source.

- **Timestamped Index** The operational system often maintain a timestamps column for keeping track of last update [201, 208]. This column is refereed as *audit column.* It generates a new time-stamp for any modification in tuple. These audit column can be used to identify new changes since last loading cycle. Some trigger-based techniques can be incorporated with timestamp method. The query written below can be used to extract todays data from an *sales* table.

```
SELECT * FROM sales WHERE TRUNC(CLIENT(order_date AS date),
'dd') = TO_DATE(SYSTMDATE,'dd-mm-yyyy');
```

It is a prerequisite that all the source system will have a time stamp column. This technique can be applied on any type of source file. If any source data is deleted in between two extraction process, then this process is unable to detect it. It can result to duplicate data extraction, in the case of any restarting after a mid-process failure.

Proposed Work	**Advantage**	**Disadvan tage**	**Algo Type**	**Feature**	**Ref.**	**Year**
Framework designed	Scheduling strategy	Only for Oracle DB	Log-based	Real-time update	[242]	2008
CDC using MapReduce	NoSQL databases support	Product specific	Log-based	Cell State Model	[160]	2015
CTL Framework	Minimize failure	Not dynamic modeling	Database Trigger	CDC on OLTP	[262]	2012
Real-time snapshot maintenance	High throughput	Applicable on specific tool	Snapshot Maintenance	Incremen tal recomputation pipeline	[211]	2015
Web Service Database Encapsulation (WSDE)	Real-time CDC	Web based application	Timestamped Index	Web based service	[67]	2010

Table 3.3: Overview on CDC work in ETL extraction

3.1.2 Transformation Based Technique

This is the most complex phase in ETL process. The data from sources are in different formats. It can be relational or non-relational [282]. So, it needs to convert in a common data warehouse format before loading. At this phase, data is cleaned, conformed and customized according to data warehouse format. Main task of this stage is to detect and clean error from the extracted data by applying a set of standard transformation rules. We have categories different research direction in ETL transformation phase. The overall categories are given in Figure 3.1.

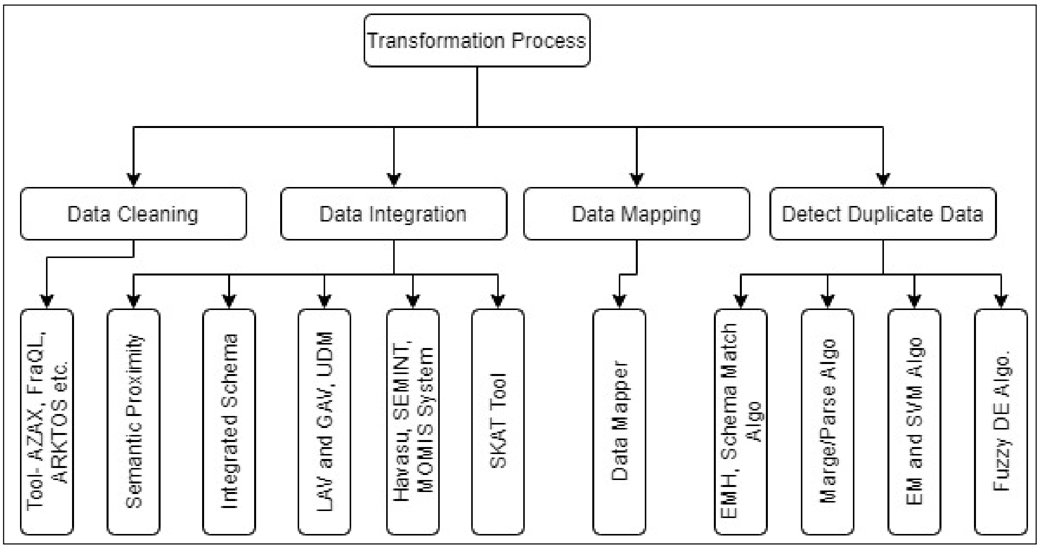


Figure 3.2: Transform Techniques in ETL

Data Quality Issues in ETL

Data quality is one of the most important technical concern in the data warehouse environment. This quality maintenance task is maintained by the ETL process. Since the data is extracted from various type of sources and containing different format. So it needs to be cleaned and represented in a standardized format. Quality maintenance is one of the important responsibility in the transformation phase. The common reasons of data quality problems are:

- Poor data handling processes
- Fault in data maintenance
- Wrongly data migration from one system to another
- Unfitted third party data format etc.

In simple way, we can tell that data quality is accomplished when data provided for an organization is comprehensive, unambiguous, uniform, relevant and timeliness. Data quality can be achieved on how data is acquired, processed, integrated and maintained throughout the ETL process. Standard quality criteria are described briefly in Table 3.4.

Factor	**Criteria of Data quality**
Completeness	It ensure that all the demanding informations are available. There is no missing values.
Consistency	It maintain the precise occurrence of a particular data instances. It handle data conflict efficiently. All the data values remain consistent.
Validity	It commits better data precision and reasonableness. Otherwise it can affect decision making process.
Conformity	It Corresponds to the assurance that data sets will be represented to a particular format. All the data will be reshaped to the required format.
Accuracy	It ensures that all the data objects are representing real world values with respect to the model. Misspelled information, un-timeliness data can have great impact over analytical operation.
Integrity	It ensure the trustworthiness of represented data. Linkage between data values should be transparent. Otherwise it can resulted towards duplicate data value.

Table 3.4: Data quality criteria

Research work covering data quality issues are discussed in this section. Data quality key issues are discussed in article [226]. A direction can be found on how to maintain data quality and consistency in an organization. It is identified the quality factors like: data not totally captured, heterogeneous system integration and lack of data management policy.

Literature [69] presented the way of high quality data maintenance in an organization. Which type of business impact can be resulted because of poor quality data is explored. Benefits of high quality data is also discussed. The reasons of data quality problems identified by this work is 1) late identification of error, 2) unreliable meta-data, 3) manual data profiling.

Various data quality rules have been proposed in the article [223]. according to this work meta-data tables are generated for storing information about the rule. These rules are further used to identify erroneous data. Rule violation informations are also stored for data analysis. The overall quality process are integrated with ETL process. And usefulness of this proposed work is examined. For interested readers many other articles covering data quality issues in data warehouse environment is in [252, 94, 12, 107].

Data Cleaning

Transformation and data cleaning are integrated part of this phase. Data transformation deal with the schema translation and integration with aggregating and filtering data to be stored in a data warehouse. When integrating data cleaning techniques should detect and remove all major errors and inconsistency in individual data sources. Transformation may include cleaning, filtering, joining, splitting, generating surrogate keys, sorting, and transposing row or column, deriving new calculated values, check data quality, applying advanced validation rules etc. various other processes.

Data cleaning [215, 55, 183] is a process of finding and correcting erroneous data with the target of achieving improved data quality. Low quality of data in warehouse can effect on the accuracy of data analysis. When data are integrated from multiple sources, the importance of dirty data cleaning grows highly. Because the data sources can have redundant, misplaced, duplicate, missing information and many other type of anomalies. The goal of data cleaning is to resolve these type of conflict. Cleaning process use to do some basic unification task:

- Making uniform identifiers. Like Male/Female, Man/Woman, M/F need to maintain a standard format Male/Female/Unknown.
- Standard format for phone number and ZIP code.
- Convert from null value into Not Given/ Not Available.
- Uniformity within address fields by proper naming. Like Street/St/St./Str./Str. will be converted as Street.
- Compare and delete duplicate data.
- Reorder Rows or Column as per destination DW.

Data cleaning is an essential process to maintain good data quality. Transformation and cleaning task broadly has to deal with *Schema-level* and *Instance-level* problem. Several data cleaning approaches has been developed. AJAX, FraQL, ARKTOS, Potter's Wheel etc. are some most popular data cleaning System developed by the research group.

Data Integration

Data integration (DI) is a critical process which takes a number of databases as input and produce an unified integrated schema as output and the corresponding mapping information. This unified view is formally mentioned as global schema. This global schema can response to query also. The mapping present the semantic relationships between input schemas and the single output schema.

For representing the data integration system in a formalized manner, I is the system with respect to triplet (G, S, M) where

- G represents *global schema*, comprise of language L_G using set of alphabet A_G. Alphabet consist of different symbols for each of the elements of G
- S denotes *source schema*, comprise of language L_S using set of alphabet A_S. Alphabet holds symbols for each elements of S.
- M presents the mapping between global schema G and source schema S. It is directed by *assertions* $q_S \rightarrow q_G$, and $q_G \rightarrow q_S$. Here q_S and q_G represent queries of same domain with respect to G and S.

Data integration itself is a big research domain. Some prominent research works are discussed below with a comparison in Table no 3.5.

Article [22] designed a conceptual model for schema integration technique. A framework is established with stepwise description of the integration process. At first schema integration methodology is discussed. Conflict are identified by comparing schema's and problems are resolved. After that an integrated conceptual schema is again created with describing various integration techniques. The authors also discussed schema comparison issues including naming and structural conflicts with examples.

At schema level integration the main task is to identify similar object in different databases which are semantically similar. In article [134] discuss on this issues by defining semantic proximity to explain the nature and measurement of semantic similarity. A linguistic and artificial intelligence, cognitive psychology based programming approach is applied to identify the similarities between objects. A brief semantic taxonomy is discussed to measure semantic characteristic between different object in multi-database system.

Another detailed DI approaches are discussed in literature [204]. The authors have attempt to take input of number of database and the resultant database is a integrated schema. At pre-integration stage, schemas are processed for making it semantically and syntactically equivalent. Next stage relationship in the schema's are established. A taxonomy of all conflicts are discussed. After resolving all conflicts a final database schema is developed.

Article [144] proposed an universal data model (UDM) to view the semantical aspect of relational, ER and XML data model. Main focus of this work is to resolve semantic heterogeneity problem. All the integration process is explained by relational algebra. Finally it conclude that it can combine several data models in a uniform format.

A new approach have been developed in article [30] for providing a Data integration framework. It is applicable for product classification of E-commerce system. It can produce a mapping within various product classification style of E-commerce system. They have experimented the methodology in the MOMIS system. The output XML mapping can be inserted into a E-commerce system. It will produce automatic rules for product classification. these rules can produce automatic data translation for showing the seller an unique code for the particular product which is classified by another way by the vendors.

Biomedical system is another new application area of Data integration process. literature [177] has worked in this domain. A general purpose data integration system is developed by the help of mediated schema. User can apply query to the schema. The system will automatically generate query plan with list of solution by which the query can be resolved. For modeling online genetic databases, at beginning phase a mediated schema (domain ontology) is build up. It contains the definition of sources. Using pair of geneticists, couple of queries is selected and perform the matching. The experimentation is done using online genetic databases.

A novel work is presented for query processing platform called *Havasu* is proposed in [127]. It is applicable for integrating web data sources. Total work is divided into two modules, StartMiner and Multi-R. In StartMiner midule association rule mining technique is applied here to find out statistics about the coverage and overlap within data sources. Collection of statistics are strictly based on threshold-based data mining process and hierarchical query classes. A system generated plan can be used to optimize both cost and query plan in the Multi-R module. These statistics are further used to process multi-objective query optimization technique.

To handle semantic heterogeneity a semantic knowledge articulation tool (SKAT) is developed by P. mitra et al. [166]. Their framework follows a semi-automatic rule-based approach for finding matches within two ontology. For ontology integration SKAT is used in ONOIN architecture [167]. Ontologies are represented into graph oriented model in ONION. Semantic relationship are denoted by articulation rules. Those rules can be graphically presented. *Articulation ontology* is created using matching rules between different ontologies. Articulation ontology can further be used to apply queries or add sources.

A new system SEMINT is developed for providing heterogeneous database integration in article [151, 152]. The main feature of this system is the use of neural network for finding the match. It utilizes the metadata of the database systems to judge the attribute relationship. For input their approach need similar attributes which will be clustered together. Automatic clustering is done by allocating all attributes by a distance less than a threshold value. training of the neural network is done by cluster center signatures. From second schema the attributes signature are given into the neural network to identify attribute cluster matching from first schema. According to their experiment they have shown that match approach using Euclidean distance is best suited to find out almost same attributes. Where as the neural network based approach is a superior way to find out less similar attributes. This system facilitates a hybrid matching technique through which numerous match criteria can be chosen and estimate simultaneously.

An ontology based integration methodology is proposed in [279] for geographic data Set. For certain real life application there can be a requirement for geographical data integration. It is a method to identify the relationship within variant geographical data sets of same geographic area which are corresponding object instances. Geo data sets can be developed by various agency with different point of view. So there is a requirement of domain ontology having same formulation of terrain object. An ontology-based conceptual framework has been developed for Geo data integration. The proposal is tested over two geographic data sets: GBKN and TOP10vector.

For last two decades data integration is a interesting topic for researchers. It has various types of application area along with different implementation approach. Regarding schema level integration various research issues [22, 204, 134] has been discussed for resolving *naming conflict* and *Structural conflict.* Interested readers can follow these article for further knowledge [214, 148, 103, 204, 149].

Proposed-Work	**Advantage**	**Automated**	**Application Area**	**Schema Type**	**Feature**	**Ref.**	**Year**
Schema integration Methodology	Unified schema	Y	Database schema integration	Relational	View integration	[22]	1986
Semantic proximity	Identify semantic similarities	N	Multidatabase system	Relational	Context-Based Approach	[134]	1996
Universal data model	Unifies data models	N	General	SQL, ER, XML	Uniform query interface	[144]	2006
MOMIS system	Products classified	Semi-auto	E-commerce	XML	Map diff. E-comm product	[30]	2002
DI system	Query formulation	Y	Biomedical	XML	Integrate Genetic databases	[177]	2001
Query processing framework *Havasu*	Optimized cost and Query plan	Y	Web sources	–	Web Data Integration	[127]	2002
SKAT Tool	Ontology composition for DI	Semi-auto	Ontology	XML, IDL, text	Graph-based OOP data model	[167, 166]	2000, 1999
SEMINT system	Multiple databases integration	Semi-auto	DI	Relational files	Neural network based	[151, 152]	1994, 2000
Conceptual framework for DI	reuse of data set	Y	Geographical info	Structured	Geographic Data Set Integration	[279]	1999

Table 3.5: Overview on Data Integration Processes

Detect Duplicate Data

Duplicate data detection is a process of finding out multiple data that represent a single real world entity. Detecting and eliminating duplicate data is one of the open problem in data cleaning stage [72]. The main goal is to identify duplicate instance of information about same entity in different database. Record linkage, semantic integration problem, object/instance identification, fuzzy match match, merge/purge, household matching etc. are another interpretation of this problem. This duplicate detection task improves the quality of data and makes it more presentable.

Data preparation is an essential stage before starting duplicate detection. Data preparation consist of three task: parsing, transformation and standardization. Parsing identifies the data elements, transformation refers to data conversion into specific type and finally standardization represents certain data fields into specific format. After data preparation phase, the data are stored in tables including comparable fields. Now the new task is to identify which fields should perform matching. For example, it is not justified to compare the *Name* field with *Designation* field. It required to locate similar fields with similar type values. After all these process, field matching techniques are applied. Different types of field matching techniques can be found based on various types of error. they are:

- **Character-based similarity metrics** This technique can identify typographical errors. It works on the basis of string-based representation of any record.
- **Token-based similarity metrics** It can give some additional facility to typographical errors in the case of rearrangement of words.
- **Phonetic similarity metrics** This approach works to find out phonetically similar data records. In this case data may be represented similarly in phonetics but dissimilar in character level.
- **Numeric similarity metrics** This process identify numbers with similar values.

The above mentioned points are used for matching individual fields for a particular record. In real-life application, records can have multiple fields. Duplicate detection process for multiple fields can be categories in two types.

- **Probabilistic Matching Models** This process strongly depends on training data for learning about how to match.
- **Supervised and Semi-supervised Learning** This process uses domain knowledge and generic distance metrics for matching records.

For last five decades many research work are going on this area. Some significant works in this area are discussed here. Duplicate detection technique was first formalized by Bayesian inference problem by Fellegi et al. [77]. It is commonly used notation in duplicate detection process under the category of probabilistic method using decision rule. Winkler [298, 299] further improved their work and proposes general unsupervised expectation maximization(EMH) algorithm. This algorithm works finely under certain conditions. They are data having large number of matches, matching pairs are isolated from other classes, minimum typographical errors etc. Winkler exhibit that when no training data is not available then EMH algorithm works well compared to the unsupervised algorithms.

In 1998 an effective algorithm formally named incremental Merge/Purge is proposed [109] by author M. A. Hernandez at el. to identify duplicate data and merge it properly. The author proposed a sorted-neighborhood method for basic merge/purge problem in a previous work. This approach minimize the number of comparison within the records. In first stage a key is determined for each record by taking part of field. Then the database records are sorted by using those keys. Finally to reduce the comparisons, a fixed size window is passed within the sorted list of records. By this way a cleaned data set is obtained. An incremental algorithm is used here which enable to merge new data with previously cleaned dataset. Finally the result is verified on real world data.

M. Bilenko at el. in 2003 designed a new duplicate detection [35] workflow for textual similarity measurement. A text distance function is defined related to each data field. They have improved two algorithm named Expectation maximization(EM) and Support Vector Machine (SVM) based on character and vector-space for measuring string similarity. Beside it is shown that the support vector machine can be used on dataset for both string and record similarity.

Article [36] in 2005 by A. Bilke and F. Naumann present an algorithm which finding duplicates comparing tuples of heterogeneous structure for which will be used for schema matching. The duplicate data are identified within asymmetrical schemas and for schema matching a set of fuzzy duplicates are used. Their approach can find out syntactically same but semantically dissimilar data values.

For past several years a number of tools for data cleaning as well as duplicate detection is introduced in the market. Febrl system, TAILOR, BigMatch, WHIRL etc. are some commonly used data cleaning tools that has data duplicate elimination feature. Duplicate detection is a large open area for research community. Many research work has been done in this domain. We have discussed few of them. Interested readers can also go through these articles [174, 147, 173, 90, 91, 53]. Moreover a comparative overview on research development is discussed in Table 3.6.

Proposed Work	Advantage	Problem type	Autom-ated	Approach	Feature	Ref.	Year
EMH Algorithm	Reduce computational paths	Bayesian problem	Y	unsupervised learning	Parameter estimation in Bayesian Networks	[298, 299]	1993, 2002
Incremental Merge/ Purge algorithm	Scalable and accurate o/p	Merge/ purge problem	N	Distance-based approaches	Sorted Neighborhood Method	[109]	1998
Duplicate records detection system	Detect approximate duplicate records	Record matching problem	N	Pair-wise record matching algorithm	Approxim ate duplicate records	[173]	2000
EM & SVM Algorithm, MARLIN system	Identify textual similarity	Character based, vector based similarity	N	Trainable similarity measures	Vector-space based measure	[35]	2003
Schema matching Algorithm	Identify corresponding attributes	Fuzzy or approximate duplicate	Y	Instance-based	Fuzzy duplicate detect	[36]	2005

Table 3.6: Overview on Duplicate Data Identification Processes

Data Mapping

Mapping of one to many relationship between tuples in the transformation phase are discussed by a series of work by P. Carreira at al. [48, 51, 49, 50]. They have proposed a new operator named *data mapper* [48] for implementing multiple tuple output from one input tuple extending features of relational algebra. In many situation one-to many transformation is required. Suppose for a organization database source tuple consist of yearly profit and final tuples consist of quarterly profit. In this situation each row need to transform from one-to-many formation.

In the long discussion of their work [51] properties of mapper are discussed. Algebraic rules are defined to optimize the execution. Firstly selection condition rule is applied to data mapper, second rule define how the selected attributes can be pushed through the mapper and finally projection rule can used to avoid extra computation for the mapper. Different level of cost estimation is done for optimization of the operator. Article [50] shows experimental result for using their proposed data mapper in RDBMS environment. A more detail discussion is given on their work in article [49].

3.1.3 Loading Based Technique

Loading is the last stage of ETL workflow. Cleaned and processed data are refreshed to the data warehouse. During the construction of data warehouse, bulk loading is performed.For daily maintenance incremental loading and bulk loading both are used. Maintaining of index and materialistic view are two other aspect of loading beside keeping track of dimension and track. Both of this two criteria can significantly contribute to increase the performance. Some popular commercial system has been developed for providing loading facility. Oracle V8 is designed for relational type, DBConnect, GemConnect, ROBIN, OpenODB are made for object oriented system. Figure 3.3 shows different loading strategies in research domain.

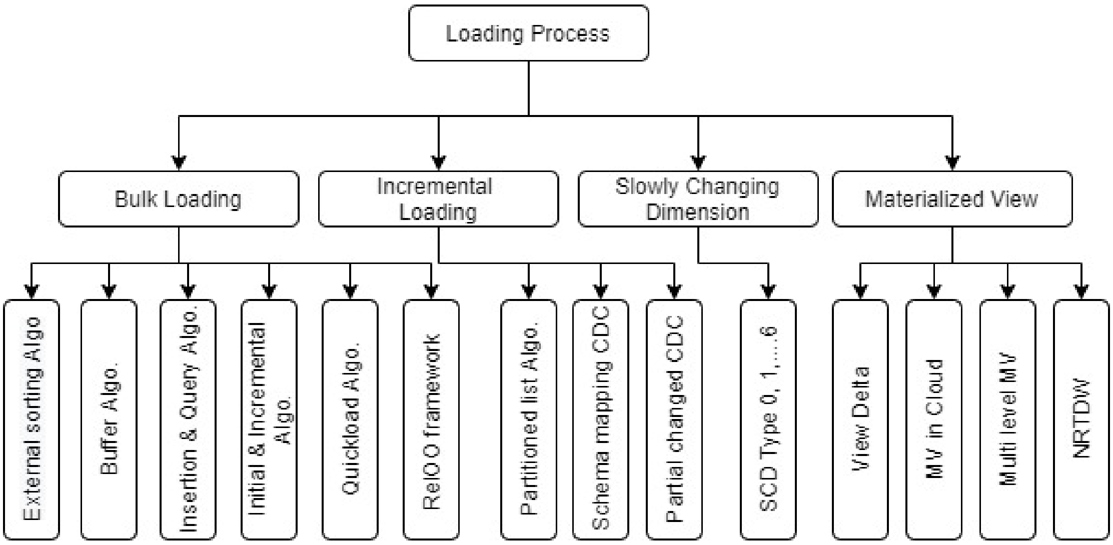


Figure 3.3: Different Loading Technique in ETL

Bulk Loading

Bulk loading is a technique for importing data into any database which are in the form of large chunks. For warehouse maintaining, it is required to load high volume of data regularly. This bulk loading technique can noticeably improve the performance for loading large amount of data. Formally bulk loading is a process for creating indexes from a data set. Some generic bulk loading techniques are discussed in this section below.

A superior bulk loading algorithm is presented in literature [44]. It takes a user defined splitting policy during the index creation. This algorithm works well when large data set is not fitted into main memory. They have proposed an external sorting algorithm for index construction in secondary storage. It results cost-effective query processing having $O(nlogn)$ runtime complexity.

Generally the bulk load operation create an index on the required data set from scratch. literature [15] uses lazy buffering techniques which uses available internal memory efficiently. This algorithm can perform over multidimensional index is a repeated insertion algorithm. A external queue is maintained for implementing the buffer. These are linked with the internal nodes of the tree except the root node. The new record is inserted into buffer without continuously traversing down towards the leaf node. When the total no of new recored exceed the buffer threshold value, then those are moved to next level. In some cases when update and query need to perform simultaneously then this approach can be applied. Insertion in the dynamic R-tree is discussed as well as complexity analysis is given.

Inspired form buffer based technique a new secondary indexing process is presented in [118] using Y-tree. Insertion became fast for single path insertion method. A fast insertion algorithm is developed. A unique point is that, at final phase the buffers are not emptied and keep link with the nodes. This process is capable of handling large node size.

In article [78], R. Fenk at el proposed two UB-tree based bulk loading techniques both for initial and incremental way. UB-tree is a cluster based multidimensional index featuring sequential data access. In UB-tree data is clustered by filling Z-curve pattern. Each point in multidimensional space is identified as Z-address which help to search dis-joined Z-region linked to disk pages. For the implementation of the bulk algorithm, at first key computation is done by taking the z-value and primary key for each tuple. Using those key external merge sort are applied on tuples of input data set. Finally the data set is loaded to the index. For initial loading, large page is filled by adding tuples exceeding twice its capacity. After that split is performed and two page is obtained. One of the page is added to UB-tree means data is saved to disk. And other left out page again add tuples iteratively until tuple ends. The incremental loading algorithm find out the right page for inserting tuples.

For bulk loading of an index for a particular data set is addressed in the article [29]. They have presented two sample based loading processes. At first they have proposed a new path based loading algorithm. Path based algorithm can describe as a top down algorithm, where data set is divided recursively until it fit into main memory. It can be applied upon Grow and Post-trees. Secondly they have proposed a Quickload algorithm which is applicable to OP trees. For this algorithm a sample is selected for input data partitioning purpose. The sample is as large as the main memory. The algorithm is recursively applied on each partition. using OP tree the samples are organized in memory.

In article [11] an optimized bulk loading framework is proposed by author S. Amer-Yahia and S. Cluet. Their approach is suitable for loading of relational, XML and object data into target. Another feature of their work is relational data can be converted into object database. User can place request in high-level-language and the system can generate an optimized loading program. At first, they have extended the algebraic formulation of [56] supporting creation as well as updation of new data. Beside they have designed a cost estimation model to calculate execution time, required bandwidth and memory. After selecting the optimal algebraic rule, automatic loading code is produced. The loading process is divided into several modules for efficient controlling of interruption and creating target database in incremental way.

A comparative view on various research work on bulk loading techniques are given in the Table 3.7 below.

Proposed Work	**Advantage**	**Disadvantage**	**Data model**	**Feature**	**Application Type**	**Complexity**	**Ref.**	**Year**
Fast algorithm	Indexes for high dimensional data	Split strategies needed	RDBMS	user-defined split strategy	X-tree,R-tree,C++	$O(nlogn)$	[44]	1999
Buffer Algorithm	Modular design	Low main memory uses	Spatial databases	lazy buffering	R-tree, C++	$O(\frac{N}{B}log_{\frac{M}{B}}\frac{N}{B})$	[15]	1999
Insertion & query Algorithm	single path insertion	Slower evaluation	RDBMS	Fast Insertion	Y-tree	Improved	[118]	1999
Bulk load Algo. for initial & incremental	Minimize IO and CPU cost	Suited for RDBMS	RDBMS	Multidimensional index structure	UB-Tree, C	$O(nlogn)$	[78]	2000
Quickload algo	Low overhead	High worst case bound	RDBMS	Path based algo	GP-tree, OP tree, C	$\theta(nlog_mn)$ $\theta(n^2/m)$	[29]	2001
Optimization Framework	Declarative approach	incapable multiple source data load	Relational-to-object	Cost model, search strategy	RelOO-language, Oracle, O_2	Not Given	[11]	2004

Table 3.7: ETL bulk loading processes

Incremental Loading

Incremental loading has much advantage over bulk loading if there is not any huge changes made in data source. Because this type of loading only transmit the new changed data made at the source side not total database at source side. Several research attempt has been done on bulk loading technique. Some primitive works are discussed below. After that the Table 3.8 will provide a comparative view on this section.

Object-relational databases (ORDB) and Object-oriented database (OODB) can load huge amount of data. For incrementally loading large set of data an algorithm named Partitioned list Algo. is discussed in article [296]. It is also applicable for smaller data set. They have implemented their proposal and evaluate its performance. Also it is pointed on how to handle after system crash by using checkpoint and resuming the loading process.

Further advancement of this work is presented in literature [295]. Here the author have pointed out the problem of relating between the existing object and the new object. A new loading algorithm is proposed for query evaluation. this loading algorithm can be easily incorporated to any OODB which supports value-based joins.

In [121], author T. Jörg proposes an algebraic incremental loading approach support incremental maintenance of materialized view. In their work relation between source and warehouse data are discussed by using high-level schema mapping techniques. An integrating schema mapping tool Orchid is used. Orchid creates an operator hub model (OHM) which represent various transformation operations in an instance graph from the schema mapping relation. From the OHM instance automatic SQL code is generated. This article extend the features of orchid by an incremental approach. Limitations of various CDC techniques are discussed and resolved. The output of CDC technique is send as an input to a suitable CDA technique, after that the data propagates through CDP process for warehouse refreshment.

They have enriched their work in article [119]. They have designed a ETL model for transformation operations using previous ETL technologies like dimension modeling, CDC and CDA. Main improvement of this article is derivation of new transformation rules for incremental loading task from initial load task.

Some research work covering both incremental and bulk loading are given in [225, 11]. In present time research work are mainly focused for implementation of loading in real-time ETL environment. For further reading article [116, 231, 268] will be a beneficial which is beyond the scope of this literature.

Proposed Work	**Advant age**	**Disadvan tage**	**Data model**	**Feature**	**Imple mented**	**Compl exity**	**Ref.**	**Year**
Partitioned list Algo	Handle large data set	New object load strategy need	Object Oriented	Handle large data sets	C++	$O(nlogn)$	[296]	1995
Partitioned list algorithm	Better query evaluation	No smart integrity checking	Object Oriented	deferred evaluation strategy	C++	$O(3 * C + 2 * N)$	[295]	1996
Loading from abstract schema mapping	Reduce complexity, Improved CDC	Algebraic approach	Relational	CDC techniques	Orchid OHM system	Upgrade	[121]	2008
Partial changed data load	Automat ed, Improved CDC	Algebraic approach	Relational	CDC techniques	Orchid OHM system	Upgrade	[119]	2009

Table 3.8: ETL Incremental Loading Processes

Slowly Changing Dimension (SCD)

One of the main task of transformation phase is to deliver dimension tables as well as fact table as per the required format. Dimensions can change asynchronously over time. So it is essential to keep track of changes made over time for preserving historical data. This can be achieved by implementing SCD technique [138, 60, 230]. SCD can manage both historical and present data on the basis of time in the data Warehouse. Various research development in SCD domain have been done. A comparative overview of some of these works are given in Table 3.9.

One of the primitive work over SCD is presented in technical report [43] by Bliujute et al. Some drawback have been identified over star schema architecture for the SCD process. For some cases in SCD type 2 their can be an overlap on the Fact table values. For example, reveling product information about old package is not possible after new package launching in the market. Because the date original date is not preserved in the Fact and dimension tables. They have presented a temporal star schema for storing both event-oriented and state-oriented data. Beside the schema store the dimensional reordering by adding time stamp record.

Proposed Work	**Advantage**	**IC**	**Data Type**	**Feature**	**Platform**	**SCD type**	**Ref.**	**Year**
Temporal star schema	Star schema problems identify	Y	State & Event-oriented data	Time stamp based	Oracle	2	[43]	1998
Comprehensive SCD	Active integration	Y	Event-oriented data	On-demand snapshot required	Oracle	2	[191]	2007
Relational Algebraic SCD	SCD Utility verify	–	Event-oriented data	Generic formal specification	Relational Algebra	1,2, 3,4	[230]	2011
Surrogate key-based temporal DW	Better query performance	Y	State oriented data	Surrogate key based	ETL	1,2,3	[76]	2014

^{IC}Improved Complexity

Table 3.9: Overview on SCD Processes

In literature [191] by Nguyen et al. a new approach is proposed for comprehensive SCD interface which refresh the warehouse by using a event based solution. It support queries for present state as well as historical state of dimension. For this they have modified the snapshot based approach. It doesn't require to keep all sequential snapshots. It only need the last refreshed event data snapshot which require minimum record processing. It provide the facility to preserve all the historical record with low latency. Finally prototype is implemented for their work and a case study is discussed.

A detailed analysis on SCD types are done in the article [230] by Santos and Belo. The authors have represented all of the SCD types by relational algebraic approach. They have pointed out some reason that there is no need to associate any types with SCD. keeping current and historical data separately will achieve the target. In article [259, 260] K.Srikanth et al. implemented SCD Type 1 and SCD Type 3 using commercial ETL tool *Informatica Power Center*. All process are discussed in step by step manner with a suitable case study.

Changing values of attributes is tracked by using SCD method. In article [76] three types of validity period are identified. They are disjoint, same and overlapping validity period. The second type of validity can not maintained by temporal star schema. They have experimented that SCD type 6 and temporal star schema can resolve this problem. A surrogate key-based temporal data warehouse (SKTDW) is proposed in this work. Various interesting development in the SCD implementation area are arranged in Table 3.9.

Materialized View

It is a useful process to pre-compute (join, aggregation) the query result is an intermediary stage to increase the overall query performance [101]. Here *view* is a derived relation from a base table and materialized view (MV) is created by storing the tuples (obtained from view) in the database. Index structures can be maintained over the views. Materialized views are useful for those type of application where query processing rate is high with complex views. It is not feasible to recompute the views for each query execution. Here materialized view works like a cache by providing fast access to required data. It is not a practical approach to recompute the view rather than only computing the changes of views for update or maintaining the materialization process. This is called incremental view maintenance. Data warehousing, lower CPU and disk loading, chronicle systems, mobile systems, integrity constraint checking, query optimization, data visualization etc. are some application area of materialized view techniques.

A novel approach is presented by Thomas Jorg et al. [122] for capturing partial deltas (changes) by using CDC techniques at the source side. It is studied that using CDC techniques it is not possible to capture complete delta. But the traditional view maintenance process uses complete delta. A discussion is done over maintainability of partial deltas and how the traditional way of maintaining materialized view can be generalized. For maintaining purpose, they have used dimension views as class of views. They have projected some generalized delta rules for deriving incremental expressions for maintaining views in the context of partial deltas.

In cloud environment, materialized views are generally placed in different data servers.

It is a challenging task to integrate the data that are located at multiple OLAP sources and generate the view. S. Sen et al. designed an architecture [239] for integrating multiple as well as heterogeneous data sources for building a virtual data warehouse deployed in cloud.

A novel approach is presented in article [302] for integrating ETL process and OLAP based data warehousing system by conditionally utilizing materialized views. The intermediate stages of ETL processing and the fact-dimension tables are stored in the materialized view objects. Various data transformation task and propagation of changes can be managed by a single refresh call. The system can get access over up-to-date information and as a result can achieve near real-time data analysis assistance.

A physical designing of ETL process is evaluated in the context of near real time data processing for any semantic data warehouse (NRTDW)[32]. Semantic aware data sources are main concern of this projected work. A dynamic materialized view selection method is presented for implementing ETL process. This system is designed to perform incremental maintenance of materialistic view. Effectiveness of the optimized are validated through a cost estimation model.

3.2 ETL Modeling

Data modeling gives an abstract view about how the data will be arranged in an organization and how they will be managed. By applying data modeling technique the relationship between different data item can be visualized. The modeling concept has a great benefit over organizational data to manage it in a structural way. At starting phase, it is highly recommended to make an efficient modeling and design of the total workflow.

ETL process modeling is a way to design the orientation of data and establish their relationship throughout the ETL processing activity. The ETL process modeling can be categories in three levels. They are

- Conceptual modeling
- Logical modeling
- Physical modeling

Conceptual modeling reflect high-level view of different entities and relationship among them. Logical modeling design implement a more detail level of view. It describes all attributes of entities indicated in former model, primary key, foreign key. No technical implementation detail is described. Physical modeling express the details such as tables and their relationship, column names and type etc. This model helps to understand database implementation details. Mapping between different level of model is possible. This Section will briefly describe ETL process modeling techniques at conceptual, logical and physical level and related research work on that domain.

3.2.1 Conceptual Modeling

During the ETL processing conceptual modeling reflect high-level view of entities and relationship among them. It only provide a abstract view of the workflow not implementation mechanism details. This model depict the the related data stores and the entity along with their attributes in the ETL task.

Different data modeling approaches are available for conceptual modeling of ETL. It is one of the main focused area for the researchers. Before starting any project it is preferable to make an efficient modeling of the total workflow. Because of the expensive nature of warehouse project, good modeling as well as documentation should be maintained. In this section, we will discuss different research development of conceptual modeling domain.

Previous research work in the domain of data warehouse conceptual modeling has been done in [137, 47, 114, 190, 95, 277, 235]. But modeling with respect to ETL was a new approach. Many explorations of ETL conceptual modeling has been done. We have grouped the modeling techniques in different sub categories which is represented in Figure 3.4. All of these modeling techniques are studied briefly in the following section.

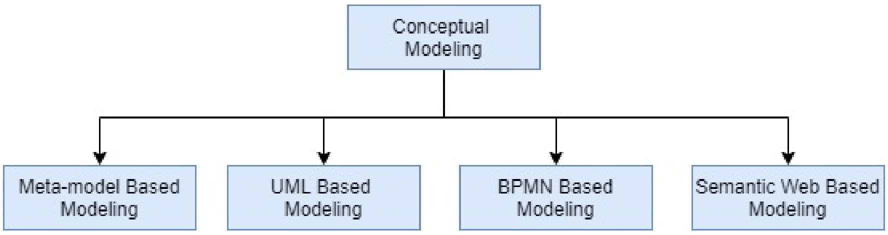


Figure 3.4: Conceptual Modeling Techniques

- **Meta-model Based Modeling**

  Meta-modeling is a process of developing an abstract model of any actual model for any predefined problem. *Model* is a representation of any real world system or problem. *Meta-model* is yet another representation of the model itself including only their important characteristics. Object Management Group (OMG) has a standard metamodeling architecture named Meta-Object Facility (MOF) for supporting UML language. Some meta-modeling based research approach for designing conceptual model are discussed here.

  First attempt for conceptual modeling of ETL was proposed by using meta-model based graphical model in literature [285]. Key terms in the article is *concept* and *attribute*. Concept indicates databases in both source and data warehouse and attributes are used for same purpose as in E/R dimensional models. Relationship among attributes of source and data warehouse is established through this model. This module support customizable templates of transformation like primary or foreign key checking, null value checking etc. Finally candidate relationship set is established for updating data in warehouse from multiple source database. The author further enriched their work [247] by proposing a methodology for the conceptual model. The methodology shows step by step procedure from source selection to warehouse population along with attributes relationship mapping and runtime obstacle handling issues.

A new ETL process modeling approaches is given in literature [71]. Their entity mapping diagram (EMD) represent the relationship with source databases and the data warehouse. A framework of EMD is proposed where a meta model is developed consist of two layers. One is abstraction layer and another is template layer. List of graphical symbol for EMD is provided with explanation. New layer can be added according to the user need. At last they have established their graphical model by an suitable example and compare their work with other approaches by using a evaluation matrix.

Another framework named KANTARA for modeling ETL processes are proposed in [125]. At beginning participants of an ETL process are discussed. Five essential modules for the KANTARA architecture have been identified. Moreover they have designed some new graphical symbols based on meta-model based notation having three core units extract, transform and load. The units are associated with list of meta-data. This work is enriched in article [123] by proposing a method for ETL process modeling design. At first the authors have depicted six modules in a ETL execution process. Among them *reject management* module is emphasized along with its meta-data details. By taking different types of input mainly consist of set of rules, their approach finally produce output of a conceptual model based on KANTARA notations.

Notation	Features	Advantage	Disadvantage	Year	Ref.
Concept & Attribute	Template based model	Customizable and reusable templates	General framework	2002	[285]
Concept & Attribute	Attribute interrelationship establish	handling runtime obstacle	Not practically implemented	2003	[247]
EMD framework	Abstraction & template layer	Customizable framework	Not validated	2011	[71]
KANTARA framework	Modular design	Reject management	Engine based	2011, 2012	[125, 123]

Table 3.10: Meta-model Based Conceptual modeling of ETL

- **UML Based Modeling**

  Unified Modeling Language (UML) is a standard general-purpose modeling language mainly used by system and software engineers to graphically design any system. The UML notations was developed by Rational Software in 1994–1995. Later in 1997, it was standardized by Object Management Group (OMG). In 2005, it has got approval for International Organization for Standardization (ISO) standard. Currently UML language support 13 different types of diagrams mainly belongs from two groups structural diagrams (Class, Object, Package, Deployment, Profile, Component, Composite structure) and behavioral diagrams (Use case, Activity, Sequence, Interaction overview, State, Communication, Timing).

J. Trujillo et al. designed [276] the workflow of ETL based on UML modeling approach. This was the first approach of conceptual model designing by using standard UML notations. The authors uses UML class diagram to establish database and their attributes relationship. Various transformation process (aggregation, conversion, filter, join etc.) is supported by their modeling with providing zooming in and zooming out facility for different level of design.

Another research effort using UML 2.0 was proposed in [181]. But they have only highlighted the extraction phase leaving transformation and loading phase. They have identified six classes and exhibit class diagram, use case diagram and sequence diagram for extraction phase using standard UML notation. Transformation and loading phase are not included in their work.

Article [187] come with the idea of modeling total ETL process by using UML activity diagrams. The activity involved in ETL process are expressed in diagram with control flow sequence supporting various transformation activity. Further they have enriched their work in [185] proposing automatically ETL processes code generation from conceptual models. They have used a new approach by using model driven architecture (MDA) for ETL process modeling. A conceptual model based on their previous work [187] is designed by using PIM (Platform Independent Model) supporting UML features. PIM can give a system functional view without thinking about the platform. Different PSM (Platform Specific Model) showing logical model view can be produced from the PIM. Automatic data structure creation code is generated form individual PSM. PIM model to PSM model transformation is done by QVT (Query View Transformation) language.

They have continued their work in [186, 184] by providing a list of measures to analyse the structural complexity of ETL conceptual model design which was created by using UML activity diagram. They have used a FMESP framework for providing the definition of the measures. A list of experiment is done for theoretically validating the measures of ETL conceptual model design.

- **BPMN Based Modeling** BPMN stands for Business Process Model and Notation consists of standard graphical notations which helps to understand business processes within an organization. This language became very popular for its easy and simple visual representation. It provides a common standard language for all members concerning any business organization like manager, stakeholder, technical and non-technical staff, business analyst etc. It was developed by Business Process Management Initiative (BPMI). Since 2005, the worldwide standards of this language is maintained by Object Management Group (OMG). BPMN 2.0 Version is released by OMG in 2011 and its new name is specified as Business Process Model and Notation. Latest version of BPMN 2.0.2 was published in 2014.

  First attempt of using BPMN notations in ETL conceptual modeling was done by

Notation	Features	Advantage	Disadvantage	Year	Ref.
UML notations	Class diagram	Flawless integration	Formal approach	2003	[276]
UML 2.0	class, use-case & sequence diagram	Detailed extraction process	Transform, load stage not included	2006	[181]
UML 2.0	Activity diagrams	Dynamic model	Entity inter-relationship not mentioned	2008	[187]
MDA, QVT	Platform independent model(PIM)	Automatic code generation	Lack quality issue and detail transform rules	2009	[185]
FMESP framework	ETL Processes Measures	Well maintenance	Structural complexity	2009, 2010	[186, 184]

Table 3.11: UML Based Conceptual modeling of ETL

Akkaoui and Zimànyi [6]. The BPMN notations and its advantages are briefly discussed. Conceptual model formation process is described and conversion from BPMN to BPEL (Business Process Execution Language) is done to execute the designed model as well as implementing relations with web services.

Later in [8] the authors have presented a Model-Driven Development (MDD) framework which describes a BPMN based conceptual model for overall ETL process which is not dependent of any vendor specific tool. From the designed model automatic code can be produced for any vendor specific platform. Further they have modified their work by proposing model-to-text and model-to-model transformation along with required maintenance factors [9, 7].

Inspiring from previous mentioned work Oliveira and Belo [195, 196] designed a set of generalized ETL meta model for some specific tasks by using BPMN notations. Finally they have validated their model by case study. Further they have continued their work for Conceptual to physical model auto-generation [197, 28] process.

- **Semantic Web Based Modeling** Semantic web is a standard maintained by the World Wide Web Consortium (W3C), is a new paradigm for the next generation technology on World Wide Web platform. The main usefulness of semantic web is to make the web data as machine-readable. It provides a vision of the linked data available on the web. RDF, SPARQL, OWL, SKOS are the enabling technologies for encoding of the data semantics.

  Semantic web technology is another new approach for constructing a data ware-

Notation	Features	Advantage	Disadvantage	Year	Ref.
BPMN and BPEL notations	Suitable for business processes	platform-independent approach	Not validated	2009	[6]
BPMN notations, MDD framework	Auto code generation	Platform-independent, Re-usability	Not optimized	2011	[8]
MDD framework	Automated maintenance	Automatic vendor-specific code	not validated	2012	[7]
BPMN 2.0 notation	Model validation	Effectiveness	Predefined pattern	2012	[195]
BPMN 2.0 extended	CDC design	standard ETL process mapping	Not validated	2013	[196]
BPMN, DSL	Physical implementation	Auto Generation of Physical model	Primary approach	2015	[28]

Table 3.12: BPMN Based Conceptual modeling of ETL

house. Skoutas and Simitsis modeled a High-level view of ETL process by using ontologies in literature [253, 254]. Use of ontology facilitates to identify the schema of the data source and data warehouse. OWL is used to construct the ontology of data source with proper annotation. Automatic transformation and data selection form source to warehouse population is established.

Extending their previous work in [255] a framework is proposed by using the feature of ontology with semantic specification of source and the target. A set of graph transformation rules are formulated to guide the flow of ETL operations. The transformation rules are flexible enough to determine the order of execution. The order is specified from the semantics of the domain ontology. They have concluded with better development of optimization technique.

Hoang Thi and Nguyen proposed a new semantic approach [271, 110] of ETL workflow using common warehouse metamodel (CWM) design standard. CWM support structured, non-structured and multidimensional metadata modeling of object in data warehouse. A ontology oriented framework is build with combining the feature of model driven approach. Data integration process is specified in details on metamodel level and instance level with claim of data quality improvement.

In 2008, Z. Zhang and S. Wang designed [306] a new framework for helping ETL processes automation and to resolve the structural as well as semantic heterogeneity problem with the presence of multiple data sources. The overall process continues in four phases: extracting meta data in the form of ontology, mapping between local and global ontology, reasoning of generated map and finally providing logic for ETL operations.

In 2010 Simitsis and Skoutas again come with a new idea [245] for non technical background people by converting the conceptual model into natural language explanation. At first stage, selection and extraction of semantics from appropriate data source is done. At next stage ontology is created according to requirement and proper data source are linked with it. Automation of source annotation and transformation rules are identified by applying a reasoner. At final stage a descriptive report in natural language is produced by translating some pre-designed templates with additional task of entity lexicalization.

The article in [270] proposed a new method RAMEP [266] by emphasizing on the requirement collection and analysis of ruling authority and stakeholder of company and ETL developers for establishing a data warehouse which is basically setting the goal at first. In next phase ontology is created for source and warehouse based on requirement and transformation rules are applied. Finally the ETL modeling is done by specifying ETL and warehouse schema.

In 2011 Romero and Simitsis presented [224] a system GEM featuring multidimensional ETL process modeling. The process starts with validation of source and business requirements from the ontology domain. Further details are added by annotating ontology. Next stage new fact and dimension are tagged and validated in multidimensional design space. Finally ETL operations are recognized and conceptual design is produced.

3.2.2 Logical Modeling

Logical modeling design implement a more detail level of view. It describes all attributes of each entity indicated in conceptual model including the order of execution. Their relationship is established by primary key and foreign key. Normalization is done in this model. More detail view than the conceptual model is possible at this level. But no technical implementation detail is described in this type of modeling. Physical execution in databases are not shown here. We have categories various ETL logical modeling process in three types (Figure 3.5), Meta Model Based Modeling, Graph Based Modeling and Grid Based Modeling. Different types of research contribution in logical ETL process modeling approaches have been categories and briefly discussed below.

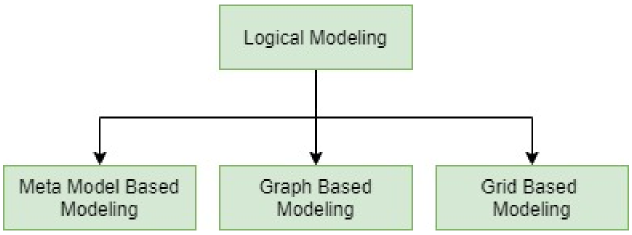


Figure 3.5: Logical Modeling Techniques

Notation	Features	Advantage	Disadvantage	Year	Ref.
OWL	High degree of automation	Specific schema semantics	Not applied on real-world data	2006	[253]
OWL-DL reasoner	Datastore graph creation	Heterogeneity resolved	Theoretical property not evaluated	2007	[254]
TPC-H schema, AGG	Graph transformation rule	Customizable & extensible design	Not optimized	2009	[255]
CWM-based model	Distributed & heterogeneous sources integration	Domain semantics management	Formal approach	2008	[271, 110]
OWL	Share & reuse of ontology	Resolve structural & semantic heterogeneity	No generalized mapping and reasoning methods	2008	[306]
OWL	Natural language model explanation	Easily understand	Improved text output required	2010	[245]
RAMEP method	Goal-oriented model	Satisfy all requirement	Time consuming	2010	[266]
GEM system	Multidimensional modeling	Satisfy all Requirement	Semi-automated	2011	[224]

Table 3.13: Semantic Web Based Conceptual modeling of ETL

- **Meta Model Based Modeling** In article [287, 288], Vassiliadis and Simitsis designed a metamodel based logical model of ETL by layered approach. In metamodel layer all elementary operation are executed. Template layer consist of a set of reusable templates for frequently used entities. At lower layer describe ETL activities at schema level. Metamodel and Schema layer are connected by instantiations. Moreover the model is enriched by zooming in and out facility applicable on its architecture graph.

- **Graph Based Modeling** Graph based logical modeling concept was first introduced in literature [286]. At beginning they have describes detail characteristics about logical modeling of ETL. After that mapping from the logical model to a graph (architecture graph) is established. In the graph entities are represented by nodes and relations are represented by edges. Standard graph transformations rule are proposed for eliminating complexities of the graph. Beside they have identified some *importance* metrics for identifying degree of dependency and responsibility of the entities.

  The authors have extended their work in article [248] by adding negations, aggregation and self-joins operations. Beside they have propose the rules for controlling insertion, deletion and update operations. Finally facility of multi level graphical view is provided by applying zooming in and zooming out features.

- **Grid Based Modeling** In 2014, Santos and Silva [99, 233, 232] tried to implement the ETL workflow in grid environment by a series of work. In this scenario the data warehouse has the characteristic to accept distribute data as well as queries. Firstly they have formulated distribution of warehouse task and logical model of ETL in the grid environment. The ETL work flow is successfully examined in grid environment. The main aim was to show an alternative way to implement ETL with minimum computational resource which helps to cut down the cost also. This solution is not suitable for near-real time ETL rather than medium ETL processes.

The Table 3.14 given below show the chart of logical model development work in various way.

3.2.3 Physical Modeling

This type of model explain about how the logical model is converted to physical model. Physical modeling express the details such as tables and their inter-relationship, column names, data type etc. Here entities and attributes are represented in tables and columns. This model helps to understand database implementation information with a details level of view. Various research effort are going on physical modeling implementation. We will briefly discuss these works in sequential way.

- **UML Based Modeling** This section is about research contribution in the domain of UML based physical modeling of ETL process. Luján-Mora, S., Vassiliadis, P., & Trujillo, J. has continued their previous work and proposed a novel approach in article [176]. In the proposed scheme for designing the ETL model extended UML version is used without using the generic functionality. The model is formulated in conceptual,

Model Name	Notation Used	Features	Advantage	Disadvantage	Year	Ref.
Meta-model Based modeling	ARKTOS II tool	Architecture graph, template based, zoom in/out	Generic & customizable	Not optimized, prototype	2002, 2003	[287, 288]
Graph Based modeling	Architecture Graph	Attribute, relationship, importance metrics	Graph transformation	Formal, not optimized	2002	[286]
	Architecture Graph	Multilevel graph transform	Zoom in/out	Not implemented	2005	[248]
Grid Based modeling	Vassiliadis notation	Resource sharing, distributed task	Low cost, fast execution	Scheduling needed	2009	[99, 233]
	Vassiliadis notation	Resource sharing, distributed task	Workload optimization	Not suitable for real time	2014	[232]

Table 3.14: Logical modeling of ETL

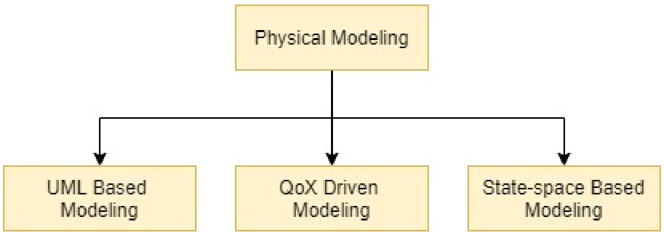


Figure 3.6: Physical Modeling Techniques

logical and physical level. It gives the facility to represent attributes relationship at different levels (database Level, data flow Level, table Level and attribute Level) of zooming. This is implemented by *data mapping diagram* where attributes are denoted as *first-class citizens* and presented by proxy class and relations are presented by classes.

The literature [17] has designed ETL framework by using Model Driven Architecture (MDA) which enable to design conceptual to physical level design. Transformation process are implemented by QVT language. At first the conceptual model is designed by using UML notations illustrating the MD schema and operations related to ETL. This models are platform independent (PIM). After that, logical as well as physical models are obtained automatically from former model by using QVT transformation rules. Derived logical models are platform independent but the physical models are platform specific.

- **QoX Driven Modeling**

  Most of the work in ETL model design point out the performance and implementation based issues. But maintenance of standard quality issues should be main concern. The approaches in article [251, 61] work towards the physical implementation of ETL process. Their proposal brings the focus on quality factors and its optimization issues. Set of criteria and their correlation has been identified and discussed for QoX Metrics. At beginning the QoX metrics are selected based on organizational requirement. Quality metrics are transferred from conceptual to physical level by different types of optimization techniques. Measurement of the QoX metrics are discussed in [61, 250].

  K Wilkinson et al. proposed a QoX oriented layered approach [297] with the continuation of their previous work. Conceptual model is designed based on BPMN notations. After that, it provides one model to another model transformation methodology with integrating QoX factors which also flow from each model to another.

  They are complementing their work in article [62, 249] by presenting optimal solution for analytical flow in modern BI applications. In modern BI embedded system the ETL operations are termed as analytic data flow which has to operate with multiple execution engines, source databases and targets. These articles discusses physical level implementation of analytic data flow along with QoX optimization in real time BI.

- **State-Space Based Modeling** Tziovara et al contributed a new techniques of implementing physical model designing of ETL from its logical level design in article [278]. The ETL activities are represented by a DAG. Logical to physical level transformation is supported by a list of predefined logical and physical templates. Each template consist its semantics and parameters. Additionally, sorter activity is developed for searching a cost effective physical model. They have used *butterfly* tool for validating their experiments.

The Table 3.15 given below show the chart of physical model development work in various way.

3.2.4 Mapping from Conceptual to Logical

After designing conceptual model with high-level view of the ETL workflow, the developer need to design its logical level model. This can be obtained by transforming from conceptual to logical Model. Logical model describe all attributes of each entities indicated in conceptual model including the order of execution. At first entities are mapped from conceptual to logical model. Transformations are mentioned as activity in logical model. Then the order of execution is determined. Some research effort has been done [246, 243, 244] for conceptual to logical Model mapping.

Model Name	Notation Used	Features	Advantage	Disadvantage	Year	Ref.
UML Based Modeling	FCME	Different level of view	Data Mapping Profile	Not validated	2004	[176]
	Model Driven Architecture (MDA)	Platform Independent Model	Automatic model generation	Requirement analysis needed	2011	[17]
QoX Driven Modeling	QoX metric	Quality issue, optimization	High performance	Not automated	2009	[251, 61]
	BPMN notations	Requirement based, QoX optimized	Business oriented view	Not automated	2010	[297]
	QoX Optimizer	Multiple analytical data flow optimization	Support modern BI, meets QoX objectives	Complex process	2011, 2012	[62, 249]
State-space Based Modeling	Butterfly method	Sorters added	Low cost physical model	Scheduling required	2007	[278]

Table 3.15: Physical modeling of ETL

At 2003, A. Simitsis has presented [244] the formal logical model of ETL. It was the initiation stage of mapping of conceptual to logical model design. The logical model was represented by using an architecture graph. Basic entities and notations are described. Optimization issues are also discussed in this article.

At 2005, A. Simitsis has implemented conceptual to logical model mapping of ETL processes in article [243]. At first conceptual to logical entity mapping techniques are described. After that a semi-auto method is proposed for execution of logical activities in a ordered way.

At 2008, A. Simitsis and P. Vassiliadis presented a complete view of conceptual to logical model mapping of ETL processes in article [246]. After describing the features of both conceptual and logical model, detailed mapping techniques are discussed. They have developed an algorithm named EOLW for selecting the execution order of logical workflow.

3.2.5 Mapping from Conceptual to Logical to Physical

This modeling express the physical implementation details such as tables and their interrelationship, column names, data type etc. The modeler need to design the conceptual model at first, then is is mapped to logical model and finally physical model is derived form logical model. Various research effort are going on physical modeling [176, 278, 17, 251, 249] implementation issue. The approaches towards physical model design can be categories in UML based, QoX based and state- space based approach. Research works in physical model design are already discussed in previous section.

This Section covers ETL modeling research issues and development at conceptual, logical and physical level. Different modeling techniques within each modeling level are briefly discussed with their research wise developments. The overall ETL modeling scenario can be visualized in Figure 3.7. Three level of modeling (Conceptual-Logical-Physical) and their inter relationship are visible here.

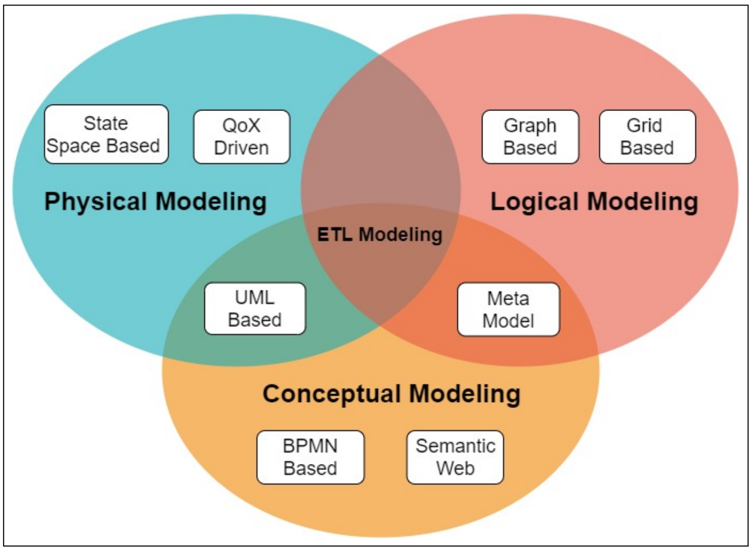


Figure 3.7: Overall Relationship of ETL Modeling Techniques

3.3 ETL tools

3.3.1 Academic development

Data transformation and data cleaning are integrated part of this phase. Data transformation deal with the schema translation and integration with aggregating and filtering data to be stored in a data warehouse. When integrating data cleaning techniques should detect and remove all major errors and inconsistency in individual data sources. Transformation may include cleaning, filtering, joining, splitting, generating surrogate keys, sorting, and transposing row or column, deriving new calculated values, check data quality, applying advanced validation rules etc. various other processes.

Data cleaning [215, 55, 183] is a process of finding and correcting erroneous data with the target of achieving improved data quality. Low quality of data in warehouse can effect on the accuracy of data analysis. When data are integrated from multiple sources, the importance of dirty data cleaning grows highly. Because the data sources can have redundant, misplaced, duplicate, missing information and many other type of anomalies. The goal of data cleaning is to resolve these type of conflict. Cleaning process use to do some basic unification task:

- Making uniform identifiers. Like Male/Female, Man/Woman, M/F need to maintain a standard format Male/Female/Unknown.
- Standard format for phone number and ZIP code.
- Convert from null value into Not Given/ Not Available.
- Uniformity within address fields by proper naming. Like Street/St/St./Str./Str. will be converted as Street.
- Compare and delete duplicate data.
- Reorder Rows or Column as per destination DW.

Data cleaning is an essential process to maintain good data quality. Transformation and cleaning task broadly has to deal with *Schema-level* and *Instance-level* problem. Several data cleaning approaches has been developed. AJAX, FraQL, ARKTOS, Potter's Wheel etc. are some most popular data cleaning System developed by the research group.
Their characteristic overview are presented in Table 3.16.

AJAX This system [90, 91] is developed by INRIA France which does some basic data cleaning task such that duplicate data detection, inconsistency between matching words, mistyping. Within the flexible system, the logical and physical level task for data cleaning are not interdependent. This extensible framework is designed such as the cleaning process is represented by a directed graph for its transformation task. Supported transformation task are mapping, matching, clustering, view and merging. The main task of this system is to transform data according to target schema, which are collected from single or multiple sources by removing the duplicate records.

ARKTOS This is an ETL tool [290, 289] capable to modeling, re-use and executing the ETL workflow. ETL process can be described by either graphical method or two declarative language XADL and SADL. Data cleaning, scheduling and transformation tasks are treated as an integrated part of this system. Targeting error that can be handled by this tool are primary key, reference along with uniqueness violation checking, Null value checking, domain mismatch and format mismatch checking.

Potter's Wheel This system [218] claim for its interactive data cleaning support. It integrates data transformation and error detection within a single spreadsheet-like interface. In this tool, user can select transformation types by selecting graphical operations or by any example. As soon as error are found, user can add transformation and get clean data without writing complex code. The effect of transformation is immediately visible on screen. This system functionalities are limited within flat files and tabular data.

FraQL This is another declarative language based data cleaning system [236]. Benefits of this system includes: effortless external as well as multiple source integration, renaming, conversion and mapping functions are available for correcting descriptive details, structural and semantic conflict. Advanced schema transformation, user-defined aggregation, duplicate elimination, missing value fill up are the extra features are presented in the proposed framework.

	AJAX	**ARKTOS**	**Potter's Wheel**	**FraQL**
Domain Format Error	Yes	Yes	No	Yes
Irregularity handling	Yes	No	No	Yes
Missing Values	Yes	Yes	No	Yes
Graphical Interface	No	Yes	Yes	No
Duplicates	Yes	No	No	Yes
Invalid Tuple	No	No	No	Yes
Interactivity	No	Yes	Yes	Yes
Constraint Violation	Yes	Yes	No	No

Table 3.16: Data quality criteria of different cleaning tools

3.3.2 Programmable ETL tools

A number of commercial and open source data integration tools are available in the market. Besides, some renowned DBMS vendors are integrating ETL functionalities with it. Every year Gartner Inc. publishes a market research reports [1] on these tools where Informatica, IBM, SAS, SAP, Oracle, Microsoft are suggested as leading commercial tools and Talend, Pentaho are the open source challengers in the market. All those tools offer GUI based ETL process design.

Thomsen & Pedersen [275] have done a survey on open source business intelligence tools. It includes some ETL tool outline also. An overview of ETL tools characteristics are discussed there without any performance comparison.

A detailed survey is done by Vassiliadis [281] which mainly addresses research work in

[1] https://www.informatica.com/in/data-integration-magic-quadrant.html

each stage of the ETL process with some academic ETL tools (Ajax, Arktos, Potter's Wheel, HumMer - Fusion). These tools mainly offer data cleaning or work-flow designing task. Following that work, Vassiliadis et al. addressed three commercial ETL tools (SQL SSIS, OWB, DataStage) and made a taxonomy of distinct ETL characteristics in article [284]. A detailed discussion on macro-level ETL flow generation process are studied in this article.

The focus of this section is programmable ETL tools. Some survey was conducted regarding comparative study on popular GUI based ETL tools which are available in market [124, 202, 161]. But, no such survey covering experimental analysis work is found in this area from where features and performance based overview of code based ETL tool can be evaluated. For this purpose, we have selected four code based ETL tools that are well accepted by the academic world. Two python based tool **Pygrametl** and **Petl**, one java based tool **Scriptella** and one R basec tool **R_etl** is evaluated. The architectural and characteristic overview of these tools are discussed in the Empirical Analysis section.

X. Liu et. al has extended *Pygrametl* [274] framework by using a Map-Reduce based approach in ETLMR [156]. It supports basic DW features like star as well as snowflake schema and slowly changing dimension (SCD). Use of Map-Reduce results in much scalability and fault tolerance for managing parallel ETL processing and data synchronization. A performance comparison with popular ETL tool Pentaho Data Integration (PDI) proves the efficiency of this approach [155].

Pygrametl Most remarkably, Pygrametl [274, 273] is an open source python based ETL framework first released in 2009. This software is licensed under BSD. Till now continuous up-gradation is done on this tool.

Without drawing any ETL process using GUI based tool, Pygrametl [2] suggest performing ETL tasks by writing python codes. It offers some commonly used ETL functionality to populate data in DW. The data flow can be achieved into three stages, namely extraction, cleaning and insert into DW. Data is represented using python dictionary having key and value pair. PostgreSQL, MySQL, Oracle are the supported databases. Seamless integration of any new kind of data source can be done using merge-join, hash-join, union-source functions. Both the batch or bulk load can be performed as per the requirement.

It is easy to populate fact and dimension tables from the source data through one iteration. It offers to insert data into star dimension or snowflake dimension which span into several tables. Besides it provides advancement on dimension support applying SCD type 1 and 2.

Petl Most notably, Petl [3] is a general purpose Python package which is able to perform conventional tasks of ETL. This package is supported under MIT License. Petl provides support both object-oriented and functional programming style. A well explained documentation is available to implement general ETL tasks. Petl can handle wide range of data sources with structured file like CSV, Text and semi-structured file like XML, JSON etc. PyMySQL, PostgreSQL, SQLite are three compatible databases with this package.

Petl support maximum transformation patterns required in any ETL process. Besides

[2] http://www.pygrametl.org/

[3] http://petl.readthedocs.io/en/latest/

timing, materialized view, lookup etc. utility function provide extra benefit to the developer. Addition of any third party package can be easily done within it. Efficient use of memory is implemented by the use of lazy evaluation and iterator. ETL data flow are synchronized using ETL pipelines. However it does not have SCD or parallelism handling mechanism.

Scriptella Scriptella [4] is another script based ETL tool written in Java [5]. It is licensed under Apache Version 2.0. Plain SQL queries are executed using JDBC bridge in this scripting language. In case of non-JDBC provider can be added using mixed SQL script. For describing various ETL task, XML script is used. SQL or other scripting language can be used for transformation purpose.

The main application is focused on executing script those are written in SQL, JEXL, Javascript and velocity for the purpose of ETL operations to/from various databases as well as file format like text, CSV, XML, LDAP etc. A thin wrapper created by XML script can give extra facility to make dynamic SQL script.

Multiple data sources can be added to an ETL program with additional support to some JDBC features like batching, escaping etc. No installation is required for deploying the tool or it can be worked as *Ant* task. Only JDK or JRE with version above 5.0 is required. Execution of this tool is also very simple. It is compatible with many popular databases having JDBC/ODBC compliant driver. For non-JDBC data sources a Service Provider Interface (SPI) is developed. It's integration provision cover Java EE, JMX, Spring framework, java mail, JNDI for easy scripting with enterprise standards.

Basic ETL task can be executed but with limited transformation support. Both batch load and bulk load can be implemented through this tool. It does not provides any support for parallelism as well as warehouse specific facility like SCD. Scriptella does not provide any GUI facility.

R_etl Now a days, R is a promising language which is gaining popularity in the field of Data Science. A newly developed package for R [23] named *etl* is selected for this piece of work [6]. It is licensed under CC0 with version 0.3.7 and available in CRAN [7]. It provides a pipeable framework to execute core ETL operations. It is suitable for working with medium size data.

This *etl* package can work as a basis for extending its dependent packages for managing any particular data sets. Seven open source and cross-platform dependent packages are available to easily access and analyze publicly accessible medium data sets (PAMDAS). This *etl* package can be extended to perform ETL operation for any data which is stored in an R package.

RPostgreSQL, RPostgreSQL, RSQLite are the DBI drivers for R is compatible with this package. It is suitable to handle data which can reside either in the local or remote database. Database creation or management can be done without having any expertise in SQL. Some utility functions like dbRunScript, smart_download, smart_upload, src_mysql_cnf etc. can

[4] https://github.com/scriptella/scriptella-etl/wiki

[5] http://scriptella.org/

[6] https://cran.r-project.org/web/packages/etl/README.html

[7] http://github.com/beanumber/etl

provide some additional benefits to the developers. Very few lines of code is required to implement this tool. But only some basic ETL functionalities are enabled here. It does not meet the requirements of current ETL technologies.

Bubble another code based ETL tool [8] is evaluated but could not be included in this work as this module is not properly maintained now. Bubbles [9] is an open-source Python based framework for the purpose of data processing and data analysis. Data processing pipeline [10] are used to depict any ETL task in the form of directed graphs. Metadata is used for expressing pipelines. It does not provide SCDs or parallelism facility. Till now it is a prototype and has limited transformation and source variety support.

ETLator [213] is another scripting language based ETL framework. It is implemented in python language and provide support for both slowly changing dimensions and parallel task execution. Parallelism is achieved by file as well as directory naming and nesting protocol. It facilitates with logging and documentation enable to produces data flow images.

SETL [188, 189] is a new proposal in python code based ETL framework which will construct a semantic warehouse. Data integration is achieved by utilizing semantic web technology. SETL performs well to handle both relational data and RDF data. A use case proves more productivity and better quality compared to traditional ETL solutions.

3.3.3 Cloud ETL Tools

This study reveals that still for many organizations moving confidential data in the cloud is the main concern due to privacy issues. It is observed that currently some side-line functionality like application program or IT management system is moved frequently in the cloud. Core activities are kept within the organizational access. As a result, IaaS is more accepted compared to SaaS for most organizations.

In earlier days, ETL processing was performed locally. Traditional ETL has some limitations as well. The expenditure of the ETL infrastructure establishing process and the huge data storage requirement was very high. Moreover, keeping all data in a single location has a high threat of catastrophic loss. Here, cloud comes with the privilege of cheap data storage, increasing processing speed, and many more additional benefits. At beginning, the pioneer ETL vendors Oracle Data Integrator (ODI) [11], Informatica [12], IBM InfoSphere [13] etc. used to provide traditional ETL solution. These type of tools [39] were mostly designed for extracting enterprise databases and loading it into Data warehouses. Executions were done in a multi-threaded fashion using SMP servers in batch mode. In the meantime, Apache Hadoop comes with many features to support the demands of the Big Data plat-

[8] https://github.com/stiivi/bubbles

[9] http://bubbles.databrewery.org/

[10] https://www.northconcepts.com/

[11] http://www.oracle.com/technetwork/middleware/data-integration/overview/index.html

[12] https://www.informatica.com/

[13] http://www-03.ibm.com/software/products/en/infosphere-information-server/

form. Some enterprise tools like Pentaho Data Integration [14], Talend Studio [15] etc. come with new features to upload data into big DW of Apache Hadoop. They use to work following scale-out architecture to handle a high volume of data. Still, they were mainly following a batch-oriented approach. The next generation tool demands a data pipeline to ensure high volume as well as the high velocity of data. Moreover, the previous tools were not suitable to extract real-time data feed. Presently, many vendors are coming to handle real-time data streams coming from varied sources.

Now a days, many leading ETL vendors are offering new capabilities to move towards cloud based solution. Some of promising cloud-enabled vendors in this field have been identified and studied. This section discusses a comparative review on some ETL cloud service providers based on their features.

Informatica Cloud Manages global and distributed data by producing seamlessly integrated and secured data in more scaled up and synchronized way [16]. With respect to architectural view, it has moved the meta-data and application layer elements into cloud. But the actual data integration tasks are done on-premise. It is very much suitable for existing cloud DW system like Azure SQL Data Warehouse, AWS Redshift, Snowflake etc.

Microsoft Azure Azure Data Factory provides a variety of options to move ETL work in cloud. Azure Data Factory [17] provides a hybrid data integration service to accelerate data processing in cloud where data can be placed either in cloud or locally. Overall ETL processing is maintained by data pipelines with additional support of running SQL Server Integration Service packages (SSIS) package in cloud. Basically it provides PaaS based services. But to do some advanced transformation task, any external engines for execution can be used through ADF activities.

Dell Boomi AtomSphere Provide a PaaS based cloud solution for their clients who want to integrate their various cloud base applications with other in-house applications [18]. Upgraded version allows integrating number of data sources along with reservation of cloud resources for real-time data flow during the integration.

Mulesoft It presents a data integration framework to perform the ETL process along with analytical support [19]. CloudHub provide an iPaaS service for connecting with Saas, various API and in house applications. Mule ESB is used to communicate between in-house enterprise application to cloud. Mule ESB presents a event driven any-point platform to create an API supported network consist of data, devices and applications. Till now its pre-build connectors are designed to integrate data from Google Cloud Storage, Oracle databases and Salesforce. Pre-built templates created by *DataWeave* language can be useful for complex type of integration.

Snaplogic One of the most promising cloud data integration vendor SnapLogic is gain-

[14] http://www.pentaho.com/product/data-integration

[15] http://www.talend.com/products/data-integration

[16] https://www.informatica.com/in/products/cloud-integration/cloud-data-integration.html

[17] https://docs.microsoft.com/en-in/azure/data-factory/

[18] https://solutionsreview.com/data-integration/cio-names-top-10-cloud-tools-3-data-integration-solutions/

[19] https://www.etltools.net/mulesoft.html

ing very much popularity . The architecture [20] is focused on different application integration and managing data paths. the concept of *Snaplex* can provide an elastic execution over the network which manages data stream within data sources and application. It is enabled to execute on cloud behind the firewall also. It has a simple web based interface to manage and schedule the integration pipelines and workload performance very efficiently. S3 and Amazon's EC2 is used for storage and computing purpose which will manage all transformations en-route.

Some leading cloud based ETL tools have been discussed in this section. Some other well known ETL cloud service providers are Alooma, Fivetran, Matillion, Stitch Data, Rivery etc.

In the academic world, a list of pioneering solution have been presented to promote near real time ETL in cloud. In the academic world, a solution has been proposed by [121] to implement incremental data loading in micro batch approach. A Lazy ETL approach has been designed by [133] where only the required data are extracted and loaded for targeting low cost data loading technique. Here ETL logic is integrated at query processing layer. For addressing the big data features, a novel solution has been proposed [155] by implementing ETL jobs in Map-Reduce framework. A programming framework ETLMR has been build to achieve parallel process and supporting snowflake schema, star schema and slowly changing dimensions (SCD) features. CloudETL [154] a cloud-enabled ETL framework uses Hadoop to parallelize ETL job and to process data in Hive. Here user can uses high level of constructs and various transformation features without bothering about MapReduce technical details. A new distributed architecture of ETL named Striim [203] has been developed to support real time data transformations over data streaming.

3.4 ETL Application Area

ETL has a promising research value for each competitive industry sector, where data is precious for profitability, successiveness, and faultless decision-making. Many research contribution has been proposed in the different application field. The main objective of this survey is to identify the various real-life application domain of ETL. This section will discuss the importance of various application domains as well as their research advancement with respect to ETL technology. We have investigated several research work over ETL processing and categories these research approach and development areas in a step by step manner. Figure 3.8 shows Classification by various research area on the ETL application. Here we have classified nine broad domain of ETL application. They are Agriculture, E-governance, Economic, Healthcare, Hospitality, Retail, Environment, Social network and Transport industry. Figure 3.9 presents all the identified application areas in a nice way. We will gradually discuss each application field and their research impact in the succeeding sub-sections.

[20] https://www.snaplogic.com

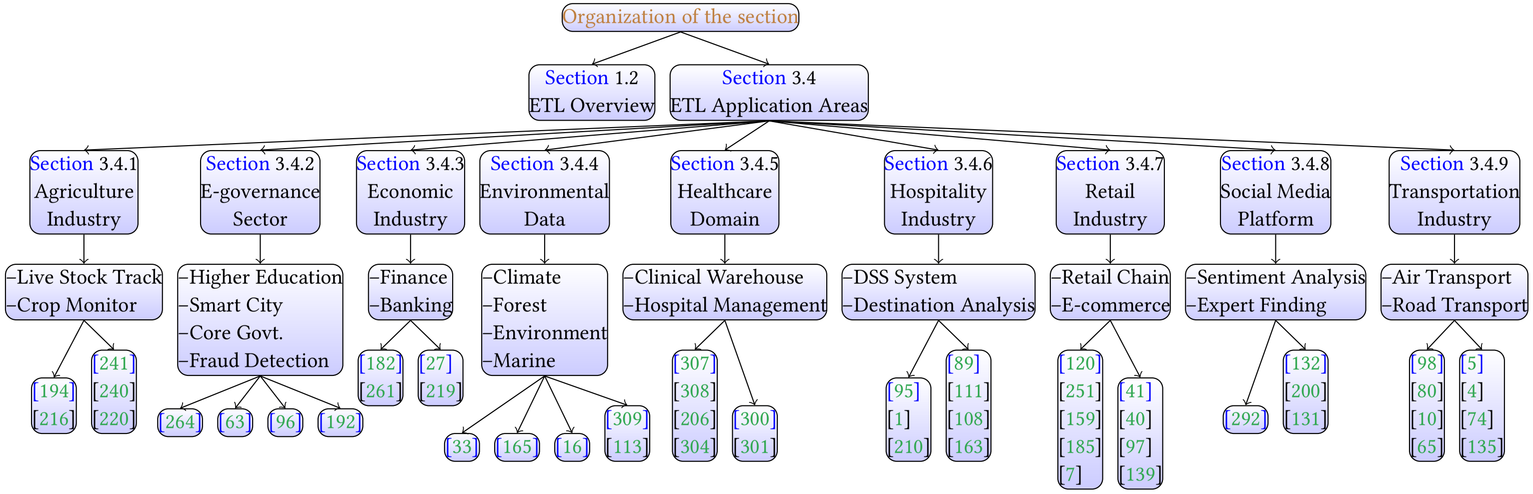


Figure 3.8: Organization of the article.

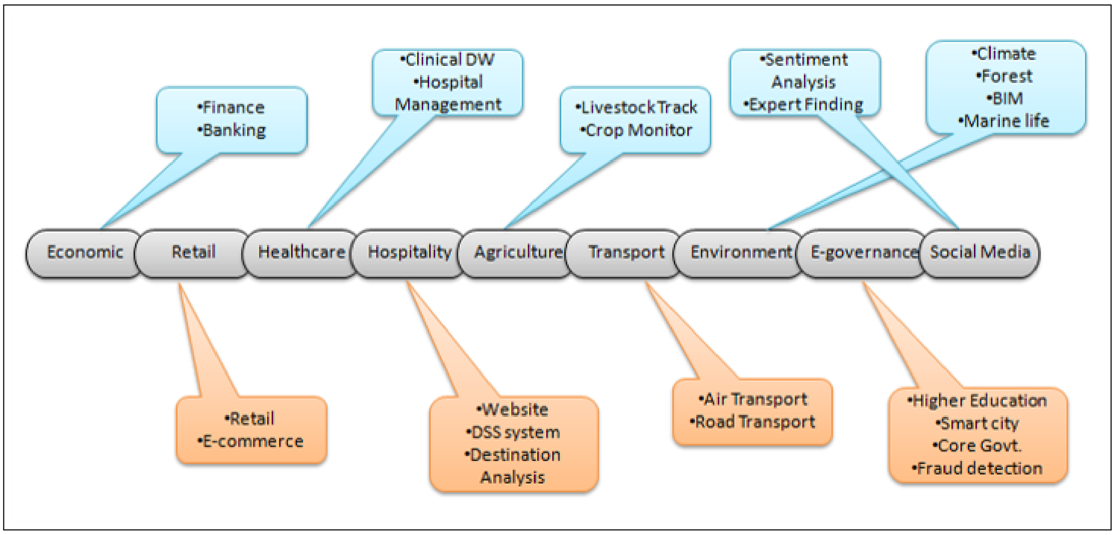


Figure 3.9: Important Research Area in ETL Application

In this section, some application area or industry has been identified where ETL processing is carried out. There is a short descriptive session about each industry and their research progress.

3.4.1 Agriculture

Nowadays, the combination of traditional agriculture and modern information and communications technologies (ICT) is an urgent need to meet market trends and reduce human effort. The application of modern technology can gain more profit and increase the productivity of farmers. The agricultural domain can be greatly enriched such as automated irrigation systems, precision planting, applications of pesticides, and nutrients supported by agriculture-based decision support systems (DSS). The DSS system can be further enhanced for crop management data analysis based on machine learning and data mining.

The agriculture industry mainly covers crop, livestock, forestry, and fisheries sector. There is a potential to integrate data from various industry and government sectors and enable a DW based solution with the help of ETL technology. The agricultural sector has an important perspective by providing a uniform data repository to address better data accessibility, transparency, and macro-level decision-making issue.

• **Livestock tracking** A project initiated by Govt. of India entitled “Integrated National Agricultural Resources Information System (INARIS)" focuses on developing a Data Warehouse is addressed in article [194]. Various issues regarding implementing an agricultural DW and ETL processing is explored in the literature. Here the dimensional model of data marts is mainly concerned about livestock resources.

An elaborated work is discussed in the article [216] by providing a guideline to design and development of hierarchical data mart and the dimensions in the agricultural sector particularly in the field of livestock management. ETL server uses to process and data mapping job upon data that are stored in the staging area. Transformed data can be in the form of OLAP cubes, web portal or dynamic reports types. Automatic updating of data mart and On-line Analytical Processing (OLAP) functionalities are addressed in this work.

- **Crop management** A conceptual ETL model named AGRETL [241] is developed and implemented by S. Sharma et al. This tool is specifically designed for agricultural aspects. The tool is evaluated by a real-life dataset and shows better performance compared to other generic ETL tools. Only some basic data transformation tasks are available in this design. There are many scopes to refine the model. Further, an extended version of work is suggested in the article [240]. The outliers detection algorithm is proposed and incorporated with the tool.

A personalized helping system for Indian farmers is designed and developed by P. K. Reddy et al. in the article [220]. Here the expert advice is available by accepting and analyzing images of crops send by the coordinators at regular time intervals. All information goes from the local sSagu center to the main center. After analyzing by the experts, the advice is sent back from the main center to the local enters. Finally, they are distributed by the coordinators to the specific farmers. The DW is populated by a huge volume of images regarding crops from diverse climate and locality through the ETL process. It is still a prototype that has a number of issues and future development scopes. A comparative overview of research work on the Agriculture Industry is presented in Table 3.17.

Objective	Tool	Dataset	Methodology	Strength	Weakness
Animal and Crop					
Improve agricultural production [194, 216]	Commercial	Data collection agency	Data mart/cube, bus architecture	Macro level support	Lacks network infrastructure
ETL tool create [241]	AGRETL	UP Govt.	Code based tool develop	Customizable	Cunt handle missing value
Outliers Detection [240]	AGRETL	UP Govt.	Algorithm develop	Quality Improve	Evaluated on small dataset
Agri DW design [220]	Commercial	Workers send	Architecture design	Give expert advice	Prototype

Table 3.17: Comparative overview of ETL research work on Agriculture Industry

3.4.2 E-governance

It is an e-media process through which all the government services can be made accessible to the citizens of any country [66, 222]. Through the e-governance procedure, more transparent and faster communication can be established between the government and citizens. It reflects government rules, laws, judicial, and accountability in a much-organized way. The overall system can be categories into four modules. Government-to-government (G2G) for Inter/Intra Govt. services like registration, revenue, land, agriculture, hospital, etc. Government-to-citizen (G2C) services include Inter-government organizations monitor, control, and communicate. Government-to-employees (G2E) services assure various policy enforcement, liability. Finally, Government-to-business (G2B) addresses the conduct and control of e-tenders. Many research initiatives from a different country have been taken to formalize and implement of e-governance framework in a standard way.

N. Agnihotri et al. [3] has discussed several issues and challenges over Big-Data management in this area. A Hadoop based framework is designed to efficiently manage a large amount of data as well as perform analytics for perfect decision-making purposes in the article [234]. The main objective of this work is to enable real-time reporting from the big volume of government data. All the initiatives will lead to improved better integration of information and stable decision-making system.

- **Higher Education:** Some work has been proposed for effectively managing and monitoring the higher education system. A star schema based multidimensional e-government architecture model is designed by S. Suresh et al [264]. SAS-Integration Studio is suggested for managing the ETL operations. Some other work related to the education system monitoring field is found in [171, 169].

- **Smart City Maintain:** The mission of implementing smart cities need to the efficient management of city operations through e-services. It requires to handle a huge amount of data in e-governance applications. P. Desai et al. [63] has worked on modeling of DW and application of OLAP for managing the registration system of a municipality's birth record. M. Mohammed et al. [170] has proposed a metadata-based solution for enforcing and monitoring of e-government system. For this purpose some ETL, OLAP, and BI tool is employed for better managing and analyzing of good quality data.

- **Core Government** A practical approach is done by [96] for implementing a data warehouse addressing social security in Tunisia. The project is mainly concerned with the insured person's data about different schemes, laws, and regulations. ETL tool Oracle Warehouse Builder (OWB) is used for data processing from oracle and access databases. The multidimensional strategy is taken for handling materialized views and OLAP cubes.

- **Fraud Detection** The use of modern technologies leads to a huge volume of data generation and speedy decision-making an urgent requirement in a competitive market. The trend to store a high inflow of electronic data having complex structure needs continuous monitoring and taking strategic decision-making within a firm time-bound. Various e-commerce, banking, Online payment introducing increasing rates of Online trading are the focus of criminal activities. There is evidence to occur in-store fraud also. Fraud schemes involving insurance, credit card, insurance, etc. creates great issues for business and government. The smart data analysis method can detect as well as prevent fraud activities.

Currently, it is a promising domain with many research publications. Some of them are discussed below. This article by A.R. Bologa et al. [46] mainly covers the advantages of Big data and different methods of the fraud detection technique. Here, the main concerning domain is health insurance. Processing big data generating from e-health cards will be employed on the Hadoop platform. Finally, some data mining, expert systems and machine learning-based fraud detection techniques are discussed.

An architecture named Sense & Response Service Architecture (SARESA) is developed by T.M. Nguyen et al. [192] for real-time data analysis and detecting fraud activities and proving an event-driven BI based proactive response to it. The case study is concerned with mobile phone call based fraud activities. The prototype is evaluated using OLAP and DW platforms. Traditional ETL processing does not provide the facility of real-time data monitoring, analysis, and reporting. The use of active DW facilitates real-time ETL to provide support to minimize the gap between transactional event and BI operation. A comparative overview of ETL research work on the E-governance Industry is found in Table 3.18.

Objective	**Tool**	**Dataset**	**Methodology**	**Strength**	**Weekness**
Higher Education, Smart city, Core govt., Fraud Detection					
Student info. repository [264]	SAS-Integration Studio	Proprietary data	DW architecture design	Real life experiment	Basic level
Birth reg. system [63]	Microsoft SQL Server	Proprietary data	ETL, DW, OLAP and DM	Effective support	Low security
DSS in Insurance [96]	OWB	Proprietary data	ETL, Oracle OLAP	Multidimentional model	Traditional reporting
Mobile call analysis [192]	SQL Server 2005	Sample	BI architecture designed	Real time solution	Prototype

Table 3.18: Comparative overview of research work on E-governance Industry

3.4.3 Economic Industry

From the last decades, most of the banking organizations are choosing for the Data Warehouse based solution for managing their daily internal and external data. An Online Transaction Processing (OLTP) system maintains daily transactional pursuit like account creation, deposit, withdrawal, loan, interest rate, commission, etc. The banking organizations have to deal with a large volume of customer data and transactional data per day. These data have an important role in risk analysis, liability, and asset analysis, market trends analysis, impose government rules and reporting, retain customer demographics. Moreover, the On-line banking system should be efficient to reply in real-time. Optimized management leads to success for the highly dynamic banking industry. So there is a need to integrate those data into a specific format. The more efficient ETL process ensures the more precise data in the Data Warehouse. All these data integration and data transformations are done by the ETL process. This process can be applied to other financial organizations like insurance and banking sectors also. Now some significant literature in the Financial and Banking domain are reviewed here.

• **Financial Service** G. Muhammad et al.[182] discusses BI based application for performing business analysis over any financial organization. Advantages and key features of BI oriented Knowledge Management (KM) solution are briefly discussed. They have designed the overall architecture where DW is maintained by deploying an efficient ETL process.

Significant work is found in paper [261] by P.J. Stromcuist et al. depicting about ETL module designing a strategy for any financial organization opting for a computerized solution. ETL package is used to regulate the data from various source databases to the target DW. The proposed ETL module defines a transformation class to hold the transformation object. The object conduct data mapping task within the source and target data.

- **Banking Service** Some technical papers have done optimized modeling of ETL architecture for banking organizations [157, 217]. Three detailed case study is observed in banking DW scenario at article [27]. It covers all the aspects from requirement gathering, DW architecture designing, ETL process modeling, and overall maintenance. Efficient data modeling of the ETL process is an important factor to construct a financial DW (FDWH) project. Any popular commercial ETL tool or customized development of its own ETL tool can be utilized in this exercise.

G.K. Rao et al. [219] has designed a framework to assimilate BI as well as KM (Knowledge Management) solutions in the context of a banking industry scenario. ETL is used as a platform in the framework design to combine and process data from various sources in the bank. From Table 3.19 we can get an overview of ETL Research work on the Finance industry.

Objective	**Tool**	**Dataset**	**Methodology**	**Strength**	**Weakness**
Financial Service					
BI application [182]	Not mentioned	Not specific	Theoretical analysis	Brief insight	Not validated
Computerized System [261]	Hand coded	Random	Object model for ETL	Improved methods	Not validated
Banking Service					
Specifics of banking DW [27]	Not required	Not required	Case study	Observation real FDWH	Theoretical
Framework Design [219]	Any tool	Any banking data	Integrate BI and KM	Customer-centric solutions	Theoretical

Table 3.19: Comparative overview of ETL research work on Finance Industry

3.4.4 Environmental Data

Revealing of inexperienced ecological patterns, data integration from the different ecological field can play a vital role. Integration of spatial dataset which spreads into multiple disciplines can explore new unanticipated insight, new research idea, and question as well as finding out missing links within the data. Some data integration task in the domain of climate, marine, forest, and building information modeling has been discussed in this section.

• **Climate** OLAP is not compatible to handle multi-temporal complex data. A new proposal is given by Bernier et al. [33] for the combination of OLAP and Geographic Information Systems (GIS) for storage and managing geographical data. For data integration and ETL operations, FME (Feature Manipulation Engine) has compatibility with the GIS tool. The spatial OLAP system is used to analyze the occurrence of some health-related threats that have a relation to climate change.

• **Forest** Spatial DW integrating the spatial ecological data has the potential to explore new patterns and hidden information. M. McGuire et al. [165] have combined OLAP with spatial DW in the field of ecology. This work also highlights about ETL work-flow in respect of spatial dimension integration. For better visualization, a web-based interface is also designed.

Building Information Modeling Combining BIM with a GIS system for property and geometry information has become an emerged research issue nowadays. T.W. Kang et al. [128] proposed an architecture by using the ETL technique, which will deploy BIM over the GIS platform. For visualizing GIS object information taken from Industry Foundation Classes (IFC) surface model information is transformed, and property-related data are processed using an open-source ETL tool Talent. This technique is suitable for construction design, energy, and facility management system to use cases.

• **Marine** J. Zubcoff et al. [309] has initiated an approach to evaluate time series analysis over fisheries data in marine protected areas. As an example, the ratio of fish captured per season or capture of any particular fish in any particular area can be evaluated. They have employed an extended version of UML for the multidimensional modeling of DW at a conceptual level. On top of that model, data mining analysis can be performed. In this framework, ETL is executed for data preparation purposes.

D. Huang et al. [113] have indicated a data heterogeneity problem and to overcome the issue an ontology-based ETL solution is chosen. At the first stage, data from heterogeneous sources is captured and mapped with local ontology. The second stage does the transformation task using mapping functions. The next process creates the global ontology which helps to accomplish the loading task in DW. A case study shows better results for handling various marine data like temperature, current, tide, etc. Some other significant works related to spatial data warehouse creation and maintenance can be found in [16, 205]. Comparative overview of ETL research work on Geospatial Data is presented in Table 3.20.

3.4.5 Healthcare Industry

Maintaining a data repository can help physicians to keep track of past records of any patient. Whenever a patient is treated, his personal and medical data will be stored securely.

Objective	Tool	Dataset	Methodology	Strength	Weakness
Climate, Forest, Environment, Marine					
Climate surveillance [33]	FME	Firsthand data	Combine OLAP and GIS	Strong spatio-temporal analysis	Require large-scale investment
Ecosystem Study [165]	Hand coded	Ecological data	Spatial DW	Multidimensional model	Navigating large data cubes
Hotspot monitoring [16]	Geokettle, QuantumGIS	FIRMS & GIB	Spatial ETL	Near Real-Time processing	Tool dependent
Resolve semantic heterogeneity [113]	Hand coded	Raw marine data	Ontology-based ETL	Automatic ETL	Not evaluated

Table 3.20: Comparative overview of research work on Environmmentl Data

The transactional server of any healthcare sector should have a strong perspective to maintain data quality and improve its performance. Patient admission/discharge, pathology, radiology, laboratory, report history are the various operating unit of data sources need to collaborate in a healthcare DW.

In earlier days patient data was manually inserted into a database from the patient's medical record. It was a laborious task as well as error-prone. Electronic processing of data using ETL can accelerate the data propagation system to the clinical DW in a more feasible, accurate, and efficient way [206, 87, 88]. Recently, noticeable growth is found in the field of genomics, functional genomics, proteomics, and biomedical research. Here data preparation, formatting, and storage is an important issue. Some important work in this sector is reviewed here.

• **Clinical Data Warehouse** The health sector can establish Data Warehouse with several objectives such as quality management, population follow-up, clinical investigations, intervention studies, etc. With the support of ETL raw clinical data can be stored and managed more efficiently and enhance the whole healthcare activity.

Author X. Zhou et al. [307, 308] have modeled a clinical DW for processing and integration of large scale data generated from various operational sources for clinical decision support and knowledge discovery purposes. Additionally, they have generated a large number of OLAP report and performed some data mining analysis over it. To fit with their requirement, a customized ETL tool named Medical Integrator (MI) is developed rather than choosing any commercial tool for data integration and normalization purpose.

A methodology is proposed by F. Pecoraro et al. [206] to design the ETL tool applicable to a clinical DW repository. The use of an electronic healthcare record (EHR) collecting system can represent heterogeneous data into a unified standard HL7 CDA. This designed ETL tool simplified the ETL transformation and loading task. The main contribution is to propose a conceptual framework based on dimensional modeling perspective suitable for an EHR system.

This work of Eric Zapletal et al. [304] is based on developing a clinical DW system with the help of the I2B2 framework at Pompidou University Hospital in Paris. The healthcare data components are consolidated by an Enterprise application integration (EAI) approach. Data sources are managed by using open source ETL tool Talend which integrates all data in the clinical DW.

• **Hospital or public health management** Apart from that, keeping employee details of any healthcare organization is required for attendance, payment, appraisal maintenance tasks. Author F. Yang [300] has designed an ETL process for performance appraisal monitoring of a hospital. A popular ETL tool Oracle Warehouse Builder (OWB) and SQL is utilized for physical implementation purposes. Another important work for the hospital monitoring system was done by S. Yoo et al. in the article [301]. It was successfully evaluated in a South Korea hospital. From Table 3.21, we can go through a comparative overview of ETL research work on the Healthcare Industry.

Objective	**Tool**	**Dataset**	**Methodology**	**Strength**	**Weakness**
Clinical DW					
Knowledge discovery and decision support [308]	Medical integrator (MI)	SEMR data	Physical data model	Structured attribute transformation	No privacy and security concern
Methodology to design ETL tool [206]	Designed	Real clinical data	Logical data map	Dimensional model design	Not Evaluated
Clinical Information Systems (CIS) [304]	Open Source Talend	Raw clinical data	Star schema based DW	Global methodology established	No continuous update from source
Clinical indicators system [301]	Not specified	EMR data	Medical DW	Paperless, Fully electronic	
Performance Appraisal System [300]	OWB	Patient data	KPI method	Implemented successfully	Validation not clear

Table 3.21: Comparative overview of ETL research work on Healthcare Industry

3.4.6 Hospitality Industry

Mainly this industry is concerned with foods - beverage, travel - tourism, and accommodation related services. These industries can be greatly benefited by utilizing DW based applications for their promotional and advertisement purpose. ETL works as the backbone of this kind of system. With the technological up-gradation, the hospitality industry needs an online-based information support system with the ability to give real-time feedback to the customer according to their search preferences. Application of some analytical tool over the preserved data can be very beneficial for analyzing customer recent booking trends, interest, requirements, sentiments, etc. Moreover, there is a necessity to keeping information about all the clients, employees, stakeholders, competitors, etc. It proves to be a better alternative for revenue management in an efficient manner. ETL has become a critical tool for airlines and hotel industries for booking and reservations from their websites for quality services. Some prominent research work in this domain is discussed below.

- **Destination Analysis** M. Fuchs et al. [89, 111] have designed a multidimensional data modeling framework for supporting a tourism-based system. Here the structured and unstructured data gathering and processing tasks are handled by ETL. Data management through ETL is followed by leading solution providers (Rapid Analytics BI server). Knowledge generation is performed over the consolidated data by applying some Data mining and BI operations. The prototype is successfully implemented in a Swedish tourism farm.

A. Hendawi et al. [108] has suggested a prototype DW model that can reflect different views from a single instance of data. A four-layer DW architecture is discussed where ETL has an important role. Adoption and validation of data from various types of sources, scrubbing and erasing all the erroneous data, and refreshing it into the DW dimensions and facts are performed using ETL. Standard ETL operations such as stored procedures or triggers are evaluated by SQL code / TSQL codes A case study is given for experimenting about how this proposed model can be implemented in an Egypt tourism organization.

D. Martins et al. [163] have discussed some challenges for handling Big data in the context of the hotel industry and applying effective BI analysis. Regarding the tourism industry, data is accumulated using web crawlers from various web sources and stored in any NoSQL database. This unstructured data is further processed via ETL and moved into secondary data storage. Any type of analytical operation is performed over this data for decision-making purposes.

- **Decision Support System (DSS)**

An initial noticeable work on the tourism domain was building Dimensional Fact Model. It was a conceptual model for the tourism Data warehouse was designed by [95] Matteo Golfarelli et al. A DW prototype is designed for the Egyptian tourism sector by Tamer A. et al in the article [1]. To populate the DW named Galaxy, the ETL process is executed. The next phase consist of data cubes and data mart creation which has an additional web-based data view management and OLAP based application for easy decision making for the managerial peoples.

Changing market trends, promote personalized and flexible service, long term development and maintenance strategy are the challenges of the tourism industry. X. Qiao et al. [210] has designed a decision support system to address all these issues for China's tourism industry. The system is based on a tourism data warehouse with an additional BI facility. Various tourism data are collected and preserved in the DW. When analysts view the data, they can correlate and perceive meaningful information from it. From Table 3.22 a Comparative overview of research work on the Hospitality Industry can be found.

Objective	Tool	Dataset	Methodology	Strength	Weakness
Destination Analysis					
BI based tourism framework [89, 111]	Rapid Insight	Small tourism farm dataset	customer preference based system	Dynamic DSS, Validated	Prototype
Egyptian tourism DW [108]	SQL code / TSQL query	Egyptian Ministry of Tourism (MoT) data	Tourism DW architecture with OLAP	Prototype implemented	Not suitable for real time data
Destination Analysis [163]	Hospitality big DW	Designed	Web data	Web Crawler, NoSQL database	Unstructured data consolidation
Decision Support System					
DW Prototype design [95]	Not mentioned	Tourisn data	Galaxy DW schema	Unified data model	Initial prototype
Tourism DW [210]	SSAS	UNWTO data and website	Data Transform Service (DTS)	Multi-dimensional data model	Heterogeneous data issues

Table 3.22: Comparative overview of research work on Hospitality Industry

3.4.7 Retail Industry

This domain is highly focused on research and development in DW and ETL applications. From a retailer point of view, DW can serve the purpose of keep track between producer and consumer information, item details, pricing, profit, etc. in an organized way. This information gathers by ETL processing can help to analyze the trend of consumer purchase behavior, market research, inventory management, market basket analysis, etc. Some research contribution in the retail domain are taken for consideration.

• **Retail** Thomas Jörg et al. [120] introduces a near real-time data propagation through the ETL approach in the DW using incremental data loading technique. With respect to the near-real-time concept, data need to be refreshed in a very short time window. Practically the DW cannot be in idle mode for a longer time. This article can suggest handling some refreshment anomalies. Finally, this approach is evaluated using a sales DW use case.

Population to a DW tracking all sales-related data is presented by A. Simitsis et al. [251]. Data processing can be done in an optimized way adopting QoX driven quality metrics. Physical level implementation is done in the sales DW for managing customer, sales, employee and vendor-related data.

Significant work has been done by S. Luján-Mora et al. [159] for physical level modeling of a sales DW. In this context, the significance of ETL is briefly discussed. UML language is employed for designing the model with the claim that it can reduce the implementation complexity and time. Another chain of work from this set of the author is about ETL process modeling using UML [187] and automatic generation of the physical model from the conceptual ETL model [185].

Z. El Akkaoui et al. [6, 9, 7] introduces a Business Process Model and Notation (BPMN) based conceptual ETL process modeling approach regarding a sales DW. A series of work is employed by executable (BPEL) model generation from the BPMN based model then updating and maintenance task handing in any physical model of any retail DW scenario.

- **E-commerce** Due to the availability of the Internet, lots of companies are choosing to buy and sell their product through e-commerce platforms. For this purpose, many E-commerce vendors are using DW based solutions for building and maintaining sales and marketing through their websites. Trend analysis, web marketing, market segmentation, etc. offer great advantages in the e-commerce industry. A new approach is introduced by N. Biswas et al. [41, 40] by using Model-based systems engineering (MBSE) supported Systems Modeling Language (SysML) for conceptually modeling any ETL process. The proposal is evaluated in an e-commerce based case study. The overall data propagation of a sample E-commerce system is described by using the SysML requirement diagram and activity diagram.

Web mining is a young domain which offers many benefits regarding e-commerce based applications. W. Grossmann et al. [97] designed a web mining based theoretical framework where ETL works as a building block for constructing a DW. Clickstream data is an important source here for the analysis of customer's behavior.

The E-commerce architecture of Blue Martini software company is discussed in the article [139]. It covers clickstream data collection, data preparation using the ETL process, data warehousing, and data mining challenge discussion. The designed data generation process named DSSGen performs better compared to any traditional ETL process. Table 3.23 represents a comparative overview of ETL research work on the retail domain discussed before.

3.4.8 Social Media

Social media websites (Facebook, Twitter, LinkedIn, etc.) are going to be an integral part of online activities nowadays. People can connect with their families and friends, share their thoughts, know about current trends, promote business, etc. via social media.

Huge numbers of users interact and share digital content frequently. To manage the high volume of data generated at an increasing rate is a great challenge to the technology providers and the research world for giving special care to this domain. Moreover, social media-generated data are mainly unstructured and dynamic nature which poses a critical task to store, manage and analyze using advanced technologies. The data warehouse constructed by ETL proves to be an effective platform for the storage of social media generated big data and takes a strategic decision on a business goal like targeting advertisements, marketing strategy, personality prediction. Behavior analysis and sentiment analysis are two types of categories for designing DW on social media [168, 142]. Some significant work related to behavior pattern analysis of any user in the social account is discussed here.

Objective	Tool	Dataset	Methodology	Strength	Weakness
Sales					
Incremental data load [120]	IBM InfoSphere DataStage	Random	Change Data Capture (CDC)	Near real-time DW	Theoretical Implementation
ETL Model Design [251]	Hand coded	Click-stream data	Quality QoX-Driven metric	Cost optimize	Manual Process
Physical level modeling [159]	CASE	Sales and CRM data	UML language	Unified DW framework	Validation required
Model, execute & update [6, 9, 7]	BPMN 2.0, Ecore Eclipse	Operational data	BPMN notation	Unique and not tool dependant	Validation required
E-commerce					
ETL Conceptual modeling [41]	Magic Draw	Random	SysML language	Unique representation	Overview level
Business model evaluation [97]	Any vendor tool	Clickstream data	Web mining	Framework design	Theoretical
B2C e-Commerce [139]	DSSGen	Clickstream data	Bottlenecking analysis	Weblogs data mining	Prototype

Table 3.23: Comparative overview of research work on Retail Industry

- **Sentiment analysis** It is a process of inspecting how a user reacts over an incident or product. Nowadays millions of users express their opinions on social media. These opinions have a good deal of importance with respect to mercantile applications. But it is a really challenging task for the underlying technologies to offer a quick view of the huge data stream.

ETL is an integral part of the DW system. But the details of ETL processing is not elaborately discussed in each research article. A. Walha et al. [292] have designed the ETL process using BPMN language. This proposal integrates user comments on Facebook. Comments are extracted, processed, perform opinion analysis, classified, and finally loaded into Data WeBhouse (DWB). The task of ETL is to identify positive, negative or neutral text polarity. The ETL design collects facebook data by its API graph explorer, reformat the data ,and refresh the processed data into the DWB. The disadvantage of this proposal is that it is not evaluated on a large data set.

- **Expert Finding** Lots of peoples are there in the social network platform nowadays with various information sharing activity. To validate that information and to justify their level of knowledge is the purpose of expert findings. Author A. Kardanet et al. [132] has a novel proposal on this purpose. Social network data is extracted by the ETL tool and stored in the star schema based DW from where the data will be scrutinized. A ranking algorithm (SNPageRank) is proposed for determining the level of expertise. The result is also validated by spearman's correlation function. Some other notable work targeting expert finding where ETL is evaluated are [200, 131]. Both of the work are followed by the mainly context-based algorithm.

Some other significant work in this domain can be found in the article [92, 221]. Table 3.24 represents a comparative overview of the research work on Social Media domain is discussed here.

Objective	**Tool**	**Dataset**	**Methodology**	**Strength**	**Weekness**
Social Network					
Opinion analysis [292]	BPMN 2.0	Facebook page - Sephora	Lexicon approach	Data WeBhouse	Small test collection
Expert Analysis [132]	Microsoft BI Development Studio	Friendfeed Data	SNPageRank Algo.	Determine the expertise level	Modified PageRank Algorithm

Table 3.24: Comparative overview of research work on Social Media Industry

3.4.9 Transportation Industry

This sector of application is responsible for the planning, blueprint design, establishment, management, and finally maintenance of the roads, water, and air-related transit systems facilities. While planning for long-term transport infrastructure targeting smart city facilities for its inhabitants, a scalable and streamlined framework is a challenge.

A smart Transportation system requires to manage a large set of real-time data to manage, monitor, process, and analyze. It leads to an efficient data integration method for managing incoming data related to transit, traffic, vehicles, etc. Some important research contribution regarding ETL processing in the transportation industry is discussed below.

- **Road Transport** G. Guerreiro et al. [98] has work in implementing the highway traffic managing system. They have designed an ETL architecture which is accountable to provide data for traffic prediction purpose. On the basis of this analytical result, a dynamic toll charging system in various highways will work. A suggested solution is implemented by *big data* compliance technology as Spark on MongoDB and Hadoop. An extended work if done by these set of authors in the article [80] by providing a web-based interface.

J. Almeida et al. [10] has designed a prototype of a fuel efficiency monitoring system for transport vehicles like a bus. The main objective was to identify and motivate fuel-efficient drivers for their Eco-friendly activities. To implement the ETL process, SSIS (Microsoft SQL Server Integration Services) platform is evaluated. The system enables to OLAP based analysis using naive based Data Mining (DM) approach.

The article [65] written by D. Dzemydiene et al. has given a proposal of Decision support system (DSS) architecture for monitoring the risk of road transportation on dangerous goods. Some mobile wireless devices are used to get location-based information on automobile transport. Here ETL does the task of processing raw data generated from sensors. Data mining task are incorporated with DSS to identify any situation and control any accident situation.

- **Air Transport** T. Ahmed et al. has worked on [5, 4] a multidimensional data warehouse based solution for handling airport baggage tracking information. The proposed indoor baggage tracking system is tested by Radio Frequency Identification (RFID) data. This work also includes the challenges of managing data flow by the ETL tool. The overall framework can significantly give a better solution such as a better reply to complex baggage related queries and valuable data management.

The architecture of a data warehouse to manage the air traffic management (ATM) system is proposed in the article [74] by M.M. Eshow et al. A web-based interface is designed which enables to process, parse raw data about live data stream of flight information and weather report and deliver as well as reply to queries. The ETL process is established using Pentaho Data Integration (PDI). Further, their extended work is found in literature [135] focusing on heterogeneous data regarding baggage, air traffic control, ticketing, fuel, catering, etc. All those data located in different locations are combined by evaluating the semantic integration process. The prototype will transform the source data into a uniform semantic format into an ontology directed triple store. Table 3.25 represents a comparative overview of the research work on the Transport domain which is discussed in this section.

Objective	Tool	Dataset	Methodology	Strength	Weakness
Road Transportation					
Traffic prediction [98]	Spark SQL, MongoDB	Sensor data	DATEX-II model, CRISP-DM	Big data management	Small data set used
Web Interface, Dynamic toll [80]	Apache Spark, MongoDB	Sensor data, GNSS data	DATEX II, CRISP-DM	Big Data pipeline	Data processing latency
Analysis fuel uses [10]	SQL SSIS	OLAP based feedback	Naive-Bayes DM	Operational & Meteorological	Prototype model
Risk assessment, DSS Architecture [65]	Proprietary	Sensor data	DSS Architecture	Mobile technology uses	Compatibility issues
Air Transportation					
Airport baggage tracking [5, 4]	C# code	RFID data	Multi-dimensional DW	Handle data imbalance prob.	Need Scale up
Air traffic control [74]	Hand coded	Sherlock Repository	Semantic integration	Proof-of-concept	Initial prototype
DW for ATM [135]	Pentaho Data Integration (PDI)	Raw data from ATM	Ontological approach	Able to large-scale analytics	Cross-source queries disable

Table 3.25: Comparative overview of ETL research work on Transport Industry

Part II

Research Proposal

Chapter 4

Conceptual Modeling

4.1 Conceptual ETL Process Modeling

Data modeling [310] gives an abstract view of how the data will be arranged in an organization and how they will be managed. The relationship between different data items can be visualized by applying data modeling techniques. The modeling concept has a significant benefit over organizational data to manage it in a structural way. At the starting phase, making efficient modeling and design of the total workflow is highly recommended. Due to the expensive nature of DW implementation, good modeling and documentation should be maintained. Based on the report [68], designing a well-established ETL workflow consumes almost one-third of the cost and effort in a DW implementation. A well-designed ETL process is one of the important aspects of accomplishing an effective DW. Each vendor-provided tool has its specific methodology for designing the ETL process [20, 136]. It requires an understanding of the functionality, language, standards, etc., of that particular tool. Moreover, the integrated design is not suitable for execution on other platforms.

During the ETL processing, conceptual modeling reflects the high-level view of entities and their relationship. It only provides an abstract workflow view instead of the implementation details. Different research work has been done for the conceptual modeling of ETL. UML, BPMN, and Semantic web are commonly used for conceptual modeling techniques. We proposed a new way of modeling an ETL process using a system modeling language (SysML). Although there are many contributions toward ETL abstract modeling is done, we think that SysML is a new direction for conceptualizing and validating ETL workflow. There is a lot of research scope using SysML to practically implement the ETL model, validation, simulation, and executable code production in a specific way for the sake of both technical and non-technical users. The Figure given below 4.1 shows the relation between SysML and UML language.

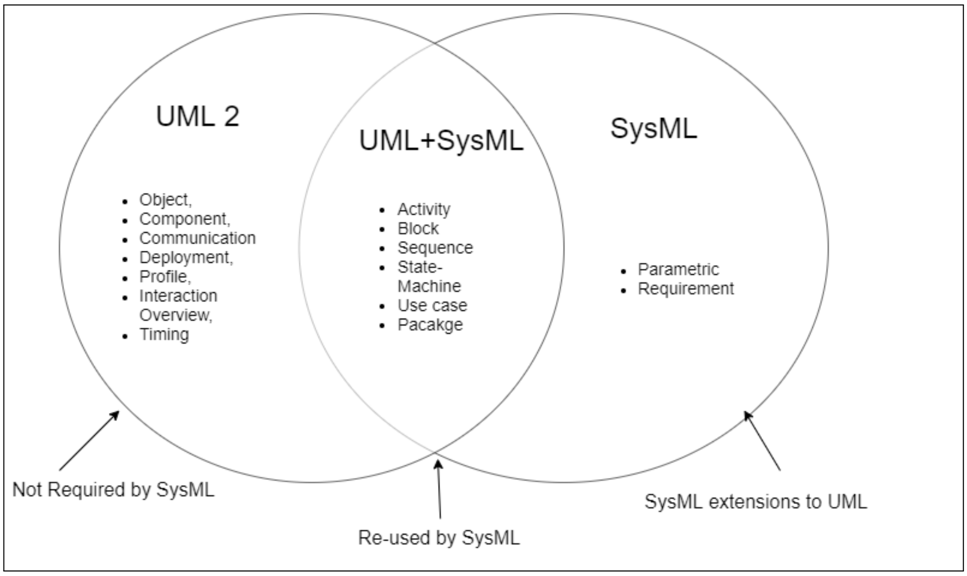


Figure 4.1: Relation between SysML and UML

This research aims to propose a new technique for designing a conceptual model of

ETL using the SysML standard supporting the Model-based System Engineering (MBSE) approach. SysML is a general purpose system modeling language that facilitates the system by identification, analysis, design, test, and validation [85]. It supports system modeling for broad categories of an organization like aerospace, automotive, health care, etc. SysML is a new modeling language standardize by Object Management Group (OMG) [199] and International Council on Systems Engineering (INCOSE) [105, 106]. It can be used to model a high-level view of the ETL process and justify the system validation by applying the simulation process.

The drawback of UML is of having a software-centric point of view and the shortfall of clear semantics. Moreover, the relationship between software and hardware is not representable by UML. SysML offers more facilities over UML by adapting some core features and extending many new directions. Whereas the BPMN language is suitable for business users to graphically model complex business processes of an organization. An initial model of the overall process is created by the business users. After that, technical developers implement that model. But implementing any SysML model is much more flawless for the technical developers as it is developed from a systems engineering viewpoint. SysML is derived from the UML model but compared to UML, SysML is very much flexible and expressive, capable of better requirement analysis and defining performance and quantitative parameters of a broad range of systems from the perspective of a system engineer and not from software centric views like UML. SysML can efficiently capture the continuous nature of the system with requirements and the parametric relation of a system model.

4.1.1 Model Based Systems Engineering (MBSE)

MBSE is an OMG supported new standard for system engineering domain featuring requirement-driven and functional analysis, design, integration, validation, and simulation of system design throughout the life-cycle of System development defined by INCOSE [75, 105]. MBSE promotes model-based approaches instead of prevalent document-oriented design methods. Functions of MBSE are shown in Figure 4.2. UML or SysML are visual modeling languages that can be used to describe the system model.

MBSE is gaining popularity in the industry for creating complex systems in the multi-disciplinary environment scenario. SysML is a visual modeling language that can be used to describe the system model. SysML is one of the key components of MBSE, having properties for capturing requirements, architecture, constraints, and hierarchical or multi-layered views of the system model. It allows linking different types of models that come from different engineering disciplines. MBSE [164] improves system modeling techniques through advanced communication, better system complexity management, standard data management, better quality product, upgraded information capture, and risk minimization.

4.1.2 System Modeling Language (SysML)

SysML is a general-purpose graphical modeling language that can be termed an extended version of UML. For modeling a system, SysML supports the system's requirements, the

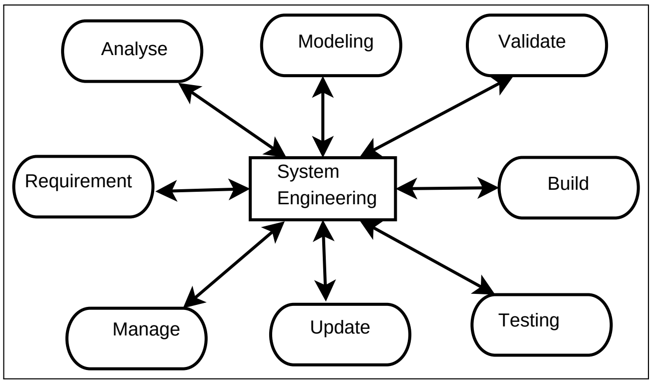


Figure 4.2: MBSE Features

system's functional and behavioral structure, and their interrelationship [86]. As it is originated from UML, it reuses many UML notations with some additional extensions [199]. SysML support various type of diagrams, which represent the structural and behavioral nature of a system shown in Figure 4.3. The Activity diagram, Block diagram, and Internal block diagram indicated by the bubble box are modified versions of the basic UML diagram. Parametric and requirement diagrams indicated by the dashed box are totally new types of diagrams incorporated in SysML. Other basic diagrams of UML can also be drawn in SysML.

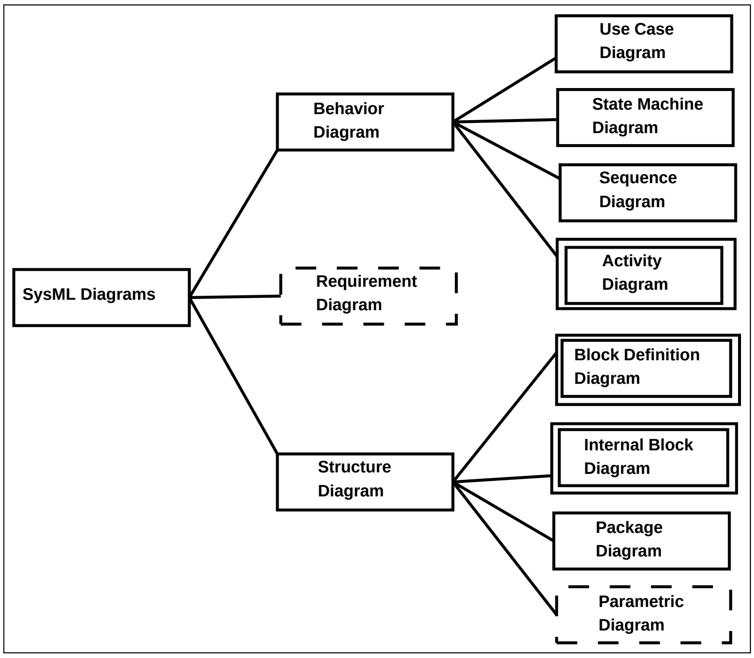


Figure 4.3: SysML supported diagrams

4.1.3 SysML Notation

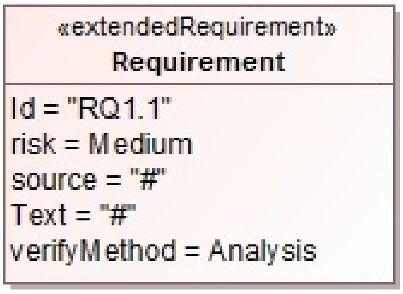
In this work, we are using the SysML Requirement diagram and Activity diagram for expressing ETL processes. The requirement diagram represents test-based requirements using a graphical construct. Whereas the activity diagram explores system behavior by showing flow of control and data within activities [293].

In SysML each modeling elements can be characterized by their *Stereotype*. There are a set of different standard stereotypes for SysML diagrams. Stereotype notation provides a new way to define system elements according to user requirements. Stereotypes are expressed by enclosing their type within double chevrons such as $\ll discrete \gg$, $\ll continuous \gg$, $\ll allocated \gg$, etc. For our proposed ETL conceptual model, we will need to understand the SysML requirement diagram notation and Activity diagram notation.

Requirement Diagram Notation

Requirement diagram in completely new concept compared to UML diagram. It supports text-based *requirements*, their *relationship* and *test cases* to verify the requirements.

A basic SysML requirement block is displayed in Figure 4.4. A SysML requirement rectangular block contain its stereotype mentioned as $\ll requirement \gg$, its unique identifier number *Id=RQ1.1* and *Text=“#"* for describing textual requirement details. There are some extended requirement properties such as verification method, source priority, risk, etc. can be selected by the designer. Requirements can be customized into more additional subcategories like *business*, *functional*, *interface*, *usability*, *performance*, *physical* etc. *Derive*, *Satisfy*, *Nesting*, *Trace*, *Verify* and *Refine* are different relationships types that can be used in requirement diagram for describing the relationship. Satisfy relationship represent that the model element satisfies a particular requirement. Trace relationship represents that the model element can be traced according to a requirement. Finally, refine relationship shows how the model elements and requirements are used to refine other model elements to an extended level. Verify relationship is used to represent how any test case can verify a requirement



«extendedRequirement»
Requirement

Id = "RQ1.1"
risk = Medium
source = "#"
Text = "#"
verifyMethod = Analysis

Figure 4.4: SysML Requirement block

Activity Diagram Notation

For SysML activity diagram, some of the notations are same as UML activity diagram as shown in Figure 4.5. *Initial State* It represents the initial action state. It is the starting point

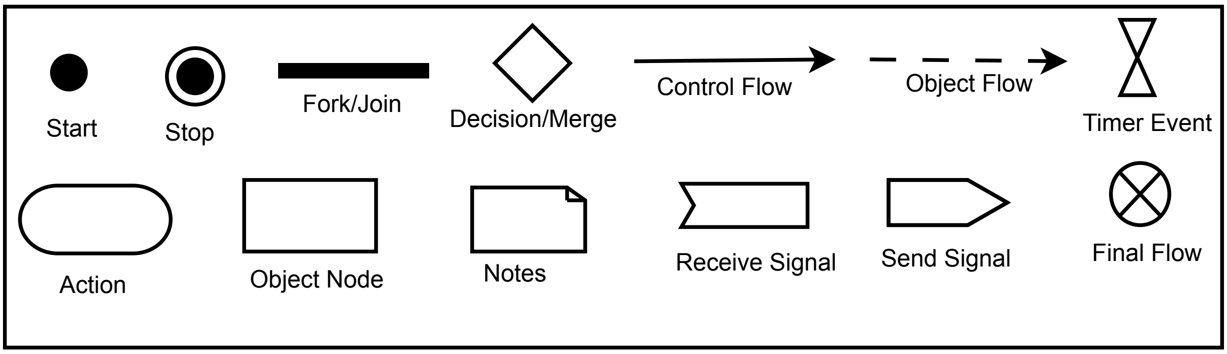


Figure 4.5: Activity diagram notations of UML and SysML

of any activity diagram.
Action State It represents the uninterruptible action of any object.
Control Flow It is an arrow symbol. It shows the transition path from one action to another action.
Object flow It is a dotted arrow symbol. It shows the transition path from an action to an object. It indicated specific data passing.
Decision Node This symbol is used as a point of conditional progression. The processing task continues based on whether the condition is true or false.
Fork node It is used to splitting a single control flow into double or any required number of concurrent flows.
Join node It is used to join any number of concurrent flows into a single flow.
Send Signal It shows how any signal can be sent to an Activity diagram.
Timer Event After any fixed time interval, the timer event action will be activated.
Final Flow This symbol indicates the termination of a control flow.
Stop When the control reaches the Stop symbol, all the activity reaches its termination point. Other new incorporated features are described below. SysML diagram is an extension of the UML diagram with some additional features. Continuous Flow is the feature that gives control over the rate at which the entities move along the Activity edges. This feature ensures the availability of the most updated data to Action. Probability is another feature of activities having the probabilistic verification over the edge coming from decision nodes and object nodes.

Activity edge can be characterized by mentioning its stereotypes like $\ll discrete \gg$ or $\ll continuous \gg$. Actual rate of the object flow can also be mentioned by using *constraint* notation like $\{rate = expression\}$. Assigning *probability* to any activity edge (mostly control flow) is another new feature like $\{Probability = value\%\}$ in SysML diagram. It expresses the probability of traversal for any particular edge. The behavior of any object node can more precisely be expressed by using stereotype $\ll nobuffer \gg$ or $\ll overwrite \gg$. For the first case, the object node will be discarded if the next action is not prepared to receive it. For the second case, the object node will be overwritten if the next action is not prepared to receive it. For example, applying *interruptible regions* a group of elements in the Activity diagram can be separately identified by a dashed box.

4.2 Conceptual Modeling of ETL Processes

During the design and planning phase of any Data warehouse, the ETL processing model at conceptual the level should also be developed. This model represents the whole process, and besides that, it includes the mapping between source and target data, shows and verifies required data transformations, requirements verification, and the overall structure. Based on the conceptual model, the ETL processes are developed. If any redesign of the process and maintenance, database schema alternation, etc., is needed due to new business requirements, it gives an extra advantage to the ETL developers.

The main purpose of conceptual ETL modeling is to establish a relationship between the source data schema and the target warehouse data schema. It provides a high-level view of the system, which does not include any logical or physical implementation details.

We have designed a high-level model of the ETL process. At first, we designed a SysML requirement diagram for the ETL scenario. After that, we modeled the conceptual ETL process using the SysML activity diagram. SysML is a general purpose language standardized by OMG and INCOSE. Each element of the SysML model is specified by its simulation-specific characteristics. Uses of SysML language is an entirely new attempt in the field of ETL conceptual modeling.

4.2.1 Example Scenario

For representing the ETL scenario, we are taking an example of an e-commerce (electronic commerce) system where a database is maintained for daily transactions. Here, buying or selling of products, payment process, and data transfer are done over an electronic network.

Operational data are stored in relational format. This data needs to be converted and deposited according to the Data warehouse format. For the e-commerce system, total sales for each day are calculated and stored in the Data warehouse. Moreover, all information related to the customer, supplier, website, and products is stored in the warehouse.

The database schema is the layout of the database. There are three types of basic Data warehouse schema: Star schema, Snowflake schema, and Fact Constellation schema. Here we are following the Snowflake type of schema. The structure of the target Data warehouse logical schema is shown in Figure 4.6. The fact table contains key attributes of dimension tables, basic facts, and derived facts. Here the *Fact_Sales* table has six dimensions of *Customer*, *Supplier*, *Product*, *Date_Time*, *Website* and *Promotion*. Each dimension table contains a set of attributes about their respective fields.

All the Dimension tables can have an aggregation level hierarchy. Dimension_Website → Dimension_Navigation is an example of hierarchy maintenance. Dimension_Address → Dimension_State → Dimension_City, these three level of the hierarchy is shared by both Dimension_Customer and Dimension_Supplier.

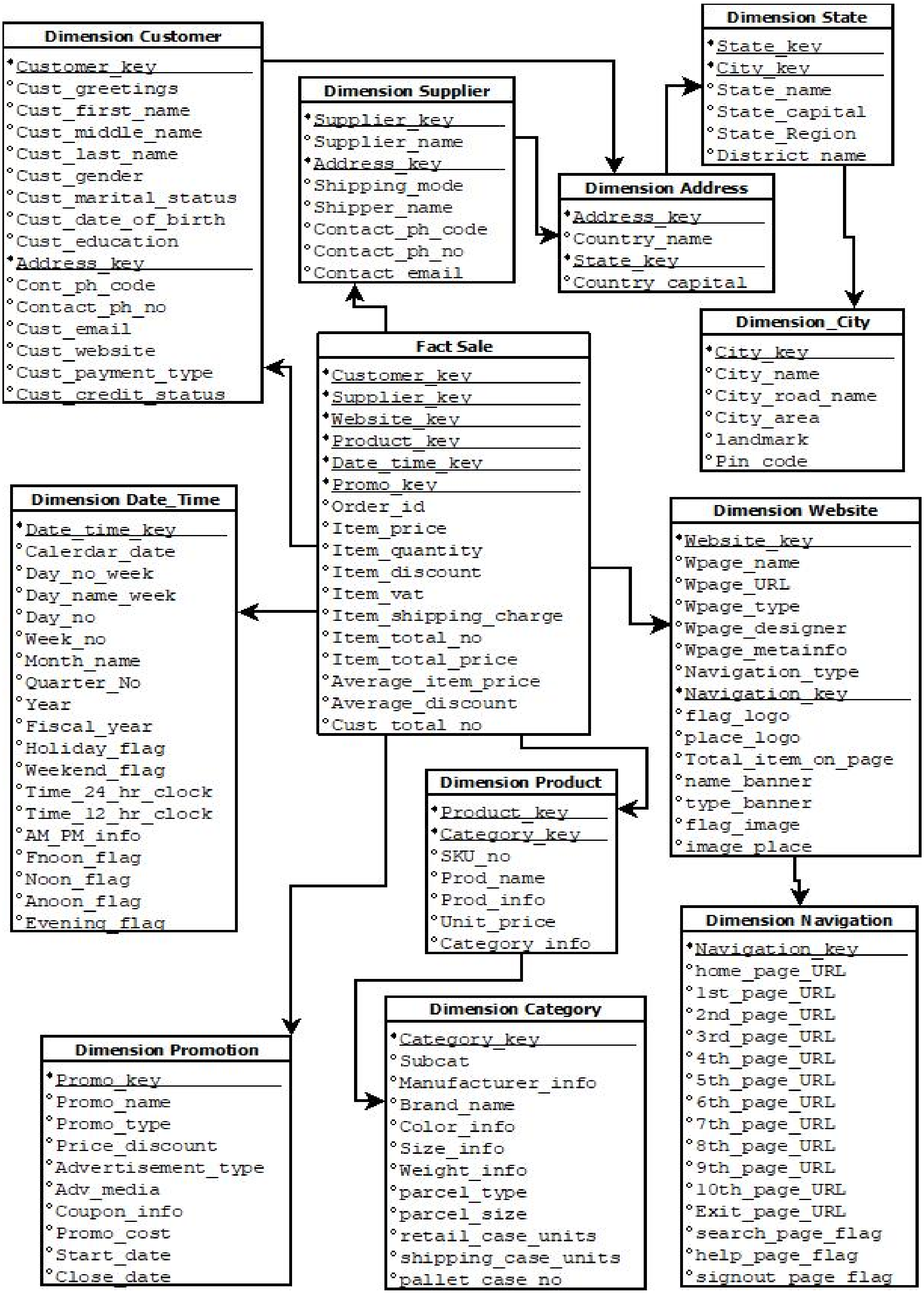


Figure 4.6: Data Warehouse schema for E-commerce System

4.3 Requirement Diagram

A requirement is a condition that a system must satisfy. There are two types of requirements, Functional requirements and Non-Functional requirements. A Functional Requirement defines the functions that a system must perform. A Non-Functional Requirement defines the qualities that can be utilized for testing the the effectiveness of system functions.

A requirement diagram is a structural diagram that represents the relationship between the requirement construct, system elements that satisfy dependency among them, and test cases to verify the dependency. The objective of the requirement diagram is to identify the Functional and Non-Functional Requirements within the system model.

Before starting the conceptual modeling, we need to identify the requirement for the ETL process. For this purpose, a SysML requirement diagram will help to visualize the requirements and their interrelations. Figure 4.7 represents an example of the requirement of ETL process for a e-commerce system using MagicDraw.

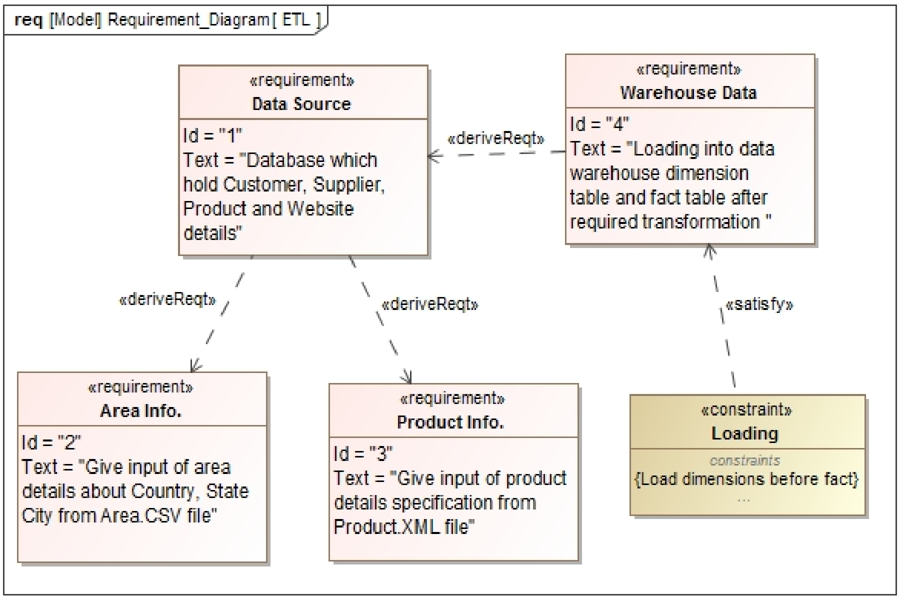

req [Model] Requirement_Diagram [ETL]
«requirement»
Data Source
Id = "1"
Text = "Database which hold Customer, Supplier, Product and Website details"
«deriveReqt»
«requirement»
Warehouse Data
Id = "4"
Text = "Loading into data warehouse dimension table and fact table after required transformation "
«deriveReqt»
«deriveReqt»
«satisfy»
«requirement»
Area Info.
Id = "2"
Text = "Give input of area details about Country, State City from Area.CSV file"
«requirement»
Product Info.
Id = "3"
Text = "Give input of product details specification from Product.XML file"
«constraint»
Loading
constraints
{Load dimensions before fact}
...

Figure 4.7: SysML Requirement Diagram for ETL in e-commerce system

In the Figure, we can observe that the operational databases (data source) provide data for loading to the Data warehouse. Two other data sources are shown here, from where data about customer address and product details are derived. The warehouse data are derived from these source databases. The restriction before loading to the warehouse is described in the constraint block. The system must satisfy the loading condition. Here the condition is, loading the Dimension tables before the Fact table. This requirement diagram represents the relationship between the various data sources and data sink options.

4.4 Activity Diagram

A SysML activity diagram is used to portray the dynamic behavior of any system that satisfies functional requirements by the uses of control and object (data) flow. It is a powerful tool that is capable of presenting the sequence of Actions for describing the nature of a Block. Control flows maintain the sequence. The Actions has an Input and Output pin. It works as an intermediary of items that flow from one action to another. The items can be energy, data, physical material, power, or anything else that is invoked or returned based on the system description. An activity presents the flow of functional behaviors, including its object flow. Here the object and the control flow can be parallel or sequential type based on the condition. The Activity diagram can be decomposed repeatedly by exchanging Call Behavior Action usages and the activity definitions.

To represent the ETL process, a SysML activity diagram is shown in Figure 4.8 using MagicDraw. The flow of data and control within different activities for loading in an E-commerce sales Data warehouse is shown.

From starting to ending node, each object flow and control flow stereotypes are indicated for describing their nature of flow. By using constraint, the rate of data flow is shown. Opaque action and call behavior action are used for describing the unit activity and sub-activity as per the requirement. The value type for each input and output pin of the action node is specified. The join node joins parallel edges, and single paths are split into parallel outgoing edges by the fork node.

At first, source databases are accessed for performing data extraction tasks. Here we can see another two data sources from an external source. They are Area.XML and Product.CSV. Address of Customer and Supplier comes from Area.XML file, and list of product catalog are fetched from Product.CSV file. After verifying the key attributes, a list of data about the dimensions are updated by the loader into their respective dimension tables. Here we can see how the six dimensions of *Customer*, *Supplier*, *Product*, *Date_Time*, *Website* and *Promotion* are loaded. During dimension loading, aggregation level hierarchy is maintained. For example, Dimension Navigation will be loaded prior to the Dimension Website. Another example is that data about the product is firstly loaded to Dimension Category and then to Dimension Product. Sub activity for loading the Area is given in Figure 4.9. In the diagram, it is shown that the dimension of the city is loaded prior to the dimension of the state. The data of this sub-activity is coming from the Area.XML file. After loading the Area, the Dimension Address is loaded. After loading the Dimension Address, it is shared and loaded by Dimension Supplier and Dimension Customer. From the figure, we can observe how the dimensions are loaded.

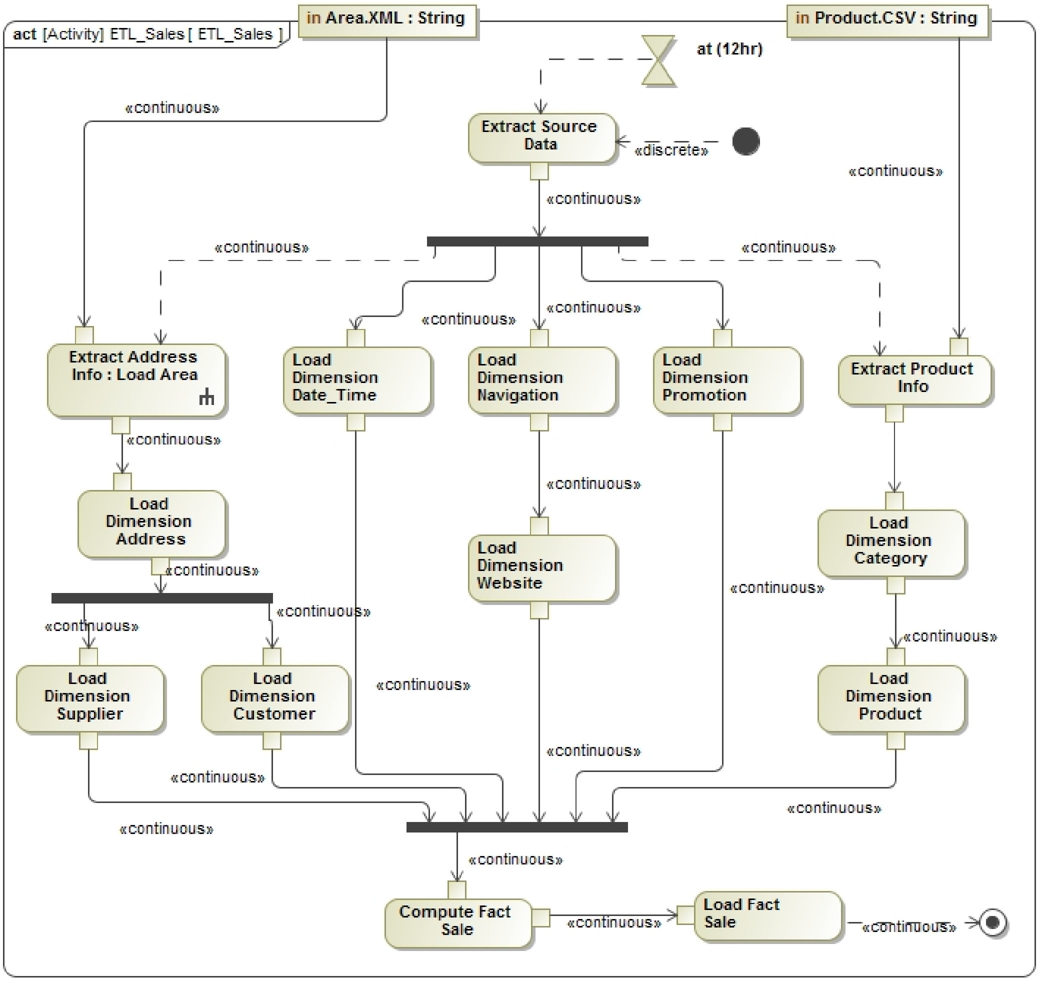


Figure 4.8: Example of ETL Conceptual Model using SysML

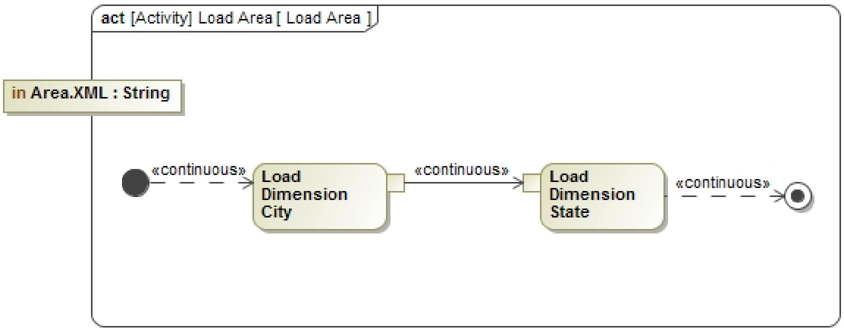


Figure 4.9: Sub-activity for loading Address

After loading six Dimensions, basic facts, derived facts, and non-additive facts are stored in the fact table. Here the basic Facts are price, quantity, and discount. Derived Facts are vat, shipping charge, total price, item total no. Moreover, Non-additive Facts are average item price, average discount, etc. Some computational tasks are done here. The overall ETL process is executed every 12 hours at intervals, as mentioned in the figure. In this example, extraction and loading processes are shown. Some other common ETL transformation tasks like Aggregation, Filter, Correction, Conversion, Joining, Splitting, Merging, and Log generation can also be represented in the conceptual model.

Post designing the system model, the SysML model is transformed into its corresponding executable code. XMI format is the standard platform-independent code of a SysML model. Therefore, this conceptual model can be transformed into its corresponding XMI format. Part of this XMI code is given in here.

Listing 4.1: executable code

```
language=xml,
tabsize=3,
%frame=lines ,
caption=XMI code of SysML diagram ,
label=code:sample ,
frame=single ,
rulesepcolor=\color{gray},
xleftmargin=20pt ,
framexleftmargin=15pt ,
keywordstyle=\color{blue}\bf ,
commentstyle=\color{OliveGreen},
stringstyle=\color{red},
numbers=left ,
numberstyle=\tiny ,
numbersep=5pt ,
breaklines=true ,
showstringspaces=false ,
basicstyle=\footnotesize ,
emph={food ,name, price}, emphstyle={\color{magenta}}
```

4.5 Summary

The ETL process is responsible for the selection and extraction of data from several sources, their cleansing, transformation according to the desired format, and finally updating into a DW. ETL process modeling is a way to design data orientation and establish relationships throughout the ETL processing activity. In this proposal, the main focus is to model the ETL process at the conceptual level. Previously various research work has been done for ETL process modeling by UML, BPMN, or Semantic web approaches.

In this chapter, we are proposing an MBSE-oriented system model for the ETL process in a Data warehouse environment. For this objective, a new modeling language, SysML is used, which is gaining popularity in organizations nowadays. It is derived from the UML language by giving some additional facilities to the system engineers. By using SysML, the system model can be designed in a more expressive as well as flexible way. An example of an e-commerce system for ETL process modeling is discussed in this work. Propagation of data from sources to DW is explained. This developed model is a platform Independent by nature and simple to understand by both technical and non-technical users. After designing the ETL model using SysML language, its corresponding executable XMI code is generated.

Although MBSE provides several benefits, there are number of drawbacks also. Some of the key drawbacks of MBSE include switching costs, a lack of standardization, scaling problems, the snowball effect, false assumptions, and various unidentified hazards. The lack of rules for combining or converting SysML models to UML models for engineering teams that comprise both software engineers and system engineers is one of the drawbacks of SysML.

Our developed model is a platform Independent by nature and simple to understand by both technical and non-technical users. After that transformation of the SysML model to its corresponding executable code is generated. In the future, we intend to simulate the proposed model to analyze system behavior and requirements more precisely and to extend the model view at the logical and physical levels.

Chapter 5

ETL Model Simulation

5.1 Simulation of Conceptual ETL Model

To handle the increasing complexity of any system model, it is preferable to go through the verification and validation process in the early stage of system development. MBSE is one of the current system engineering methodologies which covers all of the key aspects of system modeling. It combines various aspects of the system model from requirements analysis, design, and simulation throughout the system development life cycle.

Firstly a SysML conceptual model of ETL is designed in our previous work. In this paper, we are extending our previous work and presenting an MBSE based tooled approach to automate the SysML models validation by using the No Magic simulator. Here The main objective is to overcome the gap between modeling and simulation and examine the performance of the SysML model.

Gradually for the last few years, the complexity of any system is noticeably growing up. Integration of heterogeneous system components like electrical, software, mechanical, etc. is the reason behind it. Besides this, the system developers are always bound to maintain their goal of building the correct product within a low cost and fixed delivery date. Besides, a clear perception of the overall project scope is required to verify compliance with requirements.

Till now, the design of a correct system has been a big obstacle for system engineers. On the other hand, sometimes erroneous system design, which remains unrecognized at an early phase of system design, can be uneconomical. Therefore, validating any complex system design as early as possible is very practical.

Model-Based System Engineering (MBSE) has come to tackle all these problems. MBSE restores the previous document-oriented approach with a model-based approach. MBSE is a formalized way of modeling each phase of the system development life cycle. MBSE methodology-guided modeling phases are requirement analysis, designing, analyzing, verification and verification. The model can express all functional and non-functional requirements and structural and behavioral components of a system. Various high-level system models for a complex embedded system designed by MBSE supported SysML [1] language is gaining popularity these days.

As per INCOSE, system modeling and simulation are common today for system requirements and functionality evaluation. Some portion of the system or whole system can be validated as per the requisite. Some analytic routines can be executed for formal analysis purposes or any simulation-oriented analysis can also be used.

This proposal is focused on creating an automated executable SysML modeling proposal based on an activity diagram. For this purpose, a SysML execution engine is adapted. The proposed method is demonstrated by taking a running example of an E-commerce website case study.

[1] http://www.omg.org/spec/SysML/1.4/

5.2 Model Simulation Approaches

There is some existing approach for producing simulation code from a SysML model. SysML supports different diagrams which can be utilized to simulate a system model of various viewpoints. Generally any SysML model including simulation-specific profiles is designed using a modeling tool. This model is exported to an XML Metadata Interchange (XMI) format. After that, it is transformed into any simulator-specific models with the help of model transformation language (ATL, OCL, QVT, etc.) and MOF meta-model. Finally, the simulation model can be executed in the particular simulator.

SysML4Modelica [198] profile launched by the OMG aiming to transformation of SysML model to Modelica-specific executable simulation code. A ModelicaML profile was introduced for incorporating simulation ability to SysML. Query/View/Transformation (QVT) [2] language is used for transforming SysML model into executable Modelica model with the help of MOF meta-model. A guideline is given for bi-directional transformation within two languages by successfully transferring all modeling details within SysML and Modelica model. The entire process is developed under a model-driven engineering framework.

A DEVSys framework is proposed in [129], for simulating SysML model in DEVS simulator. Primarily a SysML model enriched with DEVS profile is to be defined. Block diagrams can be used to represent internal system structure, and state machine, activity, and parametric diagrams are used for system behavior. A DEVS meta-model is used for model transformation with the help of relational QVT transformation language. In this way, an executable DEVS simulation code can be produced.

In Arena [21], manufacturing-based system models can be done in SysML language. SysML-to-ARENA model transformation is done by ATL [3] language is employed with an additional meta-model. The transformed model can be executed in the Arena simulator. The limitation of this work is that only SysML block definition diagrams and activity diagrams are supported by Arena model profile specification. There is some other process also for generating simulation code.

5.3 Simulation

The system model is built for the design, analysis, and understanding of any complex system. According to model-based system engineering (MBSE), all modeling activities like a requirement, analysis, design, validation, and verification can be performed on a single platform. Recent focus is given on the issue of model execution through computer simulation experiments. Nowadays, Modeling and Simulation-based Systems Engineering (M&SBSE) is also coming along with the MBSE methodology.

Simulation is a common practice for analyzing as well as verifying any particular system model. Simulation is generally performed during a system development phase. SysML

[2] http://www.omg.org/spec/QVT/1.1

[3] https://eclipse.org/atl/

is an enabling MBSE language support simulation process for system validation. There are various research efforts for simulating the SysML model. Different tools, as well as methods, are proposed for this purpose. MagicDraw [4], Enterprise Architect[5], Visual Paradigm [6], Papyrus [7] are some of the most popular system modeling tools.

In this proposed work, SysML models for the ETL processing system are executed with the help of MagicDraw's CST execution engine.

5.4 Model Execution

The above-mentioned SysML tools are still treated as graphical modeling tools. In this work, we want to go beyond the graphical modeling by making them executable. So the modeler will get a new experience by making the models "alive". Execution of the model also provides debugging facility , which enables the modeler to judge if the behavioral model is functioning as expected or not. The main challenge in this method is to understand and define the execution semantics correctly.

Sometimes it is impossible to examine actual system behavior due to resource cost, time, and other risk constraints. Simulation provides another way to validate system functionality and identify unwanted errors in the early stage of system development. It does not need to manipulate the real system.

For SysML modeling purposes, we have used the MagicDraw tool. Models are executed by using a simulation engine. In this paper, for simulation purposes, we have used Cameo Simulation Toolkit (CST) [8]. It is supported by a Plugin that enables MagicDraw for model execution. CST provides an Execution environment standardized by OMG fUML and W3C SCXML (State Chart XML). It offers MagicDraw the features of execution, animation, debugging of the designed state machine, and activity models to validate the system behavior in a realistic environment, including the user interface. It defines specific model semantics and a UML basic virtual machine that helps designed models convert into different executable forms. It helps integrate and instantiate behavioral and structural models (especially UML Activity and Parametric model diagram).

5.4.1 Execution Engine

The execution semantics are defined in the modeling language. For model execution, semantics need to be correctly defined. Moreover, it helps to validate the particular model correctly. As it is previously stated that SysML is inherited from UML. Executable UML (xUML) was introduced to expand UML features by making it executable by providing behavioral specifications.

[4] https://www.nomagic.com/products/magicdraw

[5] http://www.sparxsystems.com/products/ea/

[6] https://www.visual-paradigm.com/

[7] https://eclipse.org/papyrus/

[8] https://www.nomagic.com/product-addons/magicdraw-addons/cameo-simulation-toolkit

fUML Foundational UML (fUML) [9] is a subset of xUML, standardized by the OMG. It specified fUML execution semantics. It defines the structural and behavioral semantics of a system. As an extended profile of UML, SysML succeeds in these semantics. fUML defines a virtual machine for the UML. The abstraction of the enabled component models transformed into several executable forms for integration, verification, validation, and deployment purposes. It supports activity and action support from UML language, which includes object and control flows, operation calls, synchronous and asynchronous behavior of the system, input and output signal, timer, pins, structures, activity nodes, parameters, and many other features.

SCXML State Chart XML (SCXML) defines the State Machine notations for control abstraction. It provides a general state machine–based execution environment. It can describe complex state-machines, sub-states, concurrency, time events, history, and more. SCXML engines enable business process flows, interaction management, view navigation bits, and many other features. By SCXML, we can simulate executable models for demonstrating tools as well as review the system behavior. CST supports exporting of SCXML file format from UML state machine models for transformation processes or further analysis.

fUML and SCXML integration the fUML execution model is treated as a backbone for any type of CST execution. But it does not have support for the state machine models. To address the gap, SCXML engine is integrated into the fUML Execution interface. As a result, State machine behavior is integrated into CallBehaviorActions.

5.4.2 Example Model Simulation

In our previous work, we designed a conceptual model of an ETL process using SysML language. An activity diagram is used to represent the ETL model. The next objective of this work is to simulate this model.

CST produces an activity simulation engine that permits to execute of an Activity Diagram. There is a list of activity elements that are supported by fUML. At first, a new simulation project needs to be created, and after that, we need to create a class. A class is the context of any activity which is created in the containment browser. For simulating any UML activity, which may be called a classifier, we need to specify the classifier behavior in the activity.

For giving input of Excel or CSV files, an Excel Import plugin [10] is applied. It is able to import any Excel or CSV file as schema classes. Row-wise data is imported from a file as instance specifications of the selected schema classes. When a table heading is imported, The Excel import plugin allow creating a schema class. It apply a stereotype *fileSchema* including the tagged value of the *fileName* tag. Figure 5.1 shows a mapping diagram of supplier schema class.

Excel import plugin also generates a class mapping. The mapping diagram represents the relation between schema class elements to target elements. Figure 5.2 represents a

[9] https://www.omg.org/spec/FUML/

[10] https://www.nomagic.com/product-addons/no-cost-add-ons/other-no-cost-add-ons

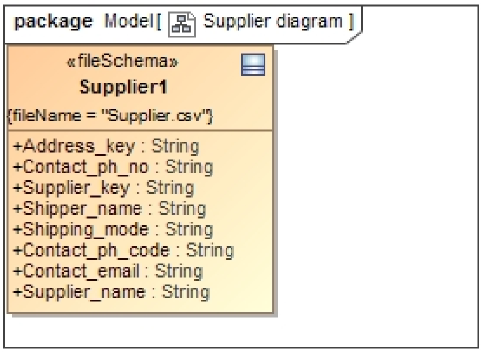


Figure 5.1: Schema class Diagram

mapping diagram of the supplier. Create Mapping wizard helps to create a class mapping. During the mapping process, you need to specify a source, a schema class, and finally, a target element. Here the schema class is the source element. The target element can be a SysML profile or UML profile, or a user model. A Composite Structure diagram is created if the schema class and the target element are successfully mapped. It is the mapping diagram. The plug-in will automatically connect the elements of the source and the target elements is the property names matches.

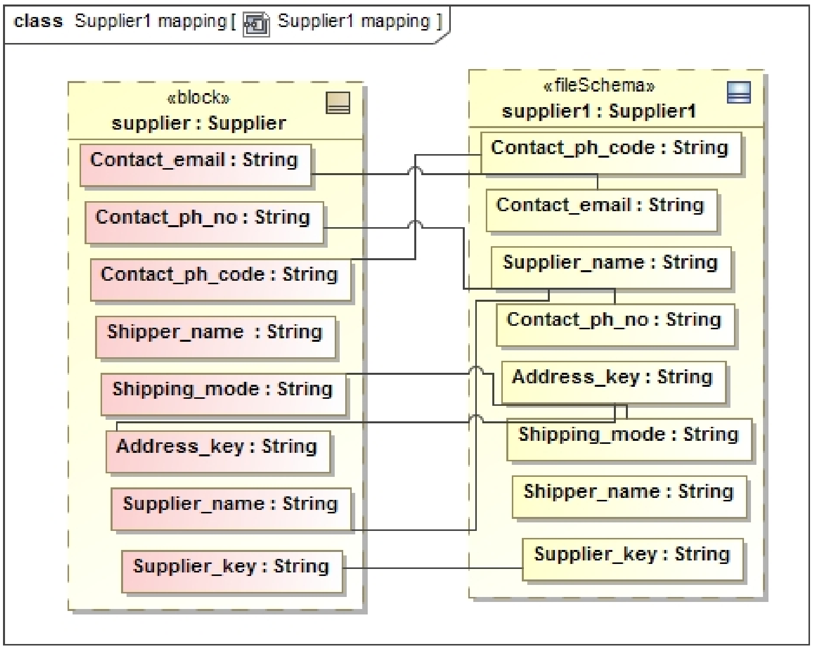


Figure 5.2: Mapping Diagram

Figure 5.3 represents the simulation model of the ETL process. The sales activity simulation diagram shows the overall processing of the system. Processing of Supplier, Customer, Promotion, Product, Website, and Date/Time behavior actions are done simultaneously. Figure 5.4 shows the simulation of the sub-activity diagram of the supplier with a more detailed view. At the starting point, the Excel or CSV files are extracted. All data are loaded into the dimension table, and the key values are loaded into the Fact table. This procedure is applied to each Customer, Promotion, Product, Website, and Date/Time action. For each action, data are loaded into the fact and dimension, respectively.

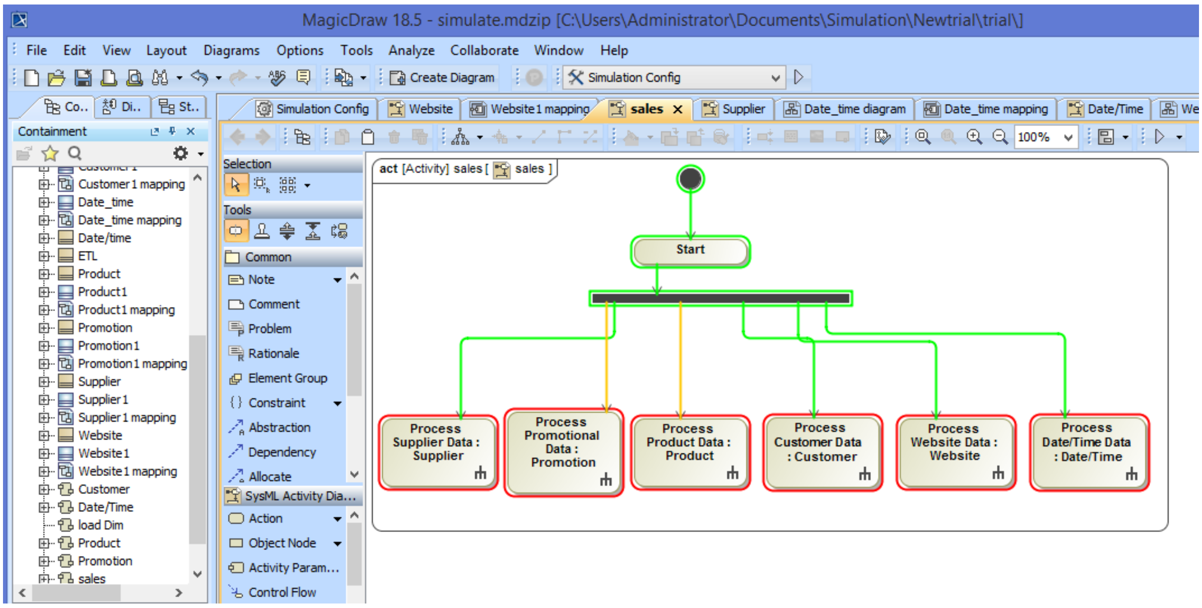

MagicDraw 18.5 - simulate.mdzip [C:\Users\Administrator\Documents\Simulation\Newtrial\trial\]
File Edit View Layout Diagrams Options Tools Analyze Collaborate Window Help
Simulation Config
act [Activity] sales [sales]
Start
Process Supplier Data : Supplier
Process Promotional Data : Promotion
Process Product Data : Product
Process Customer Data : Customer
Process Website Data : Website
Process Date/Time Data : Date/Time

Figure 5.3: Activity Model Simulation

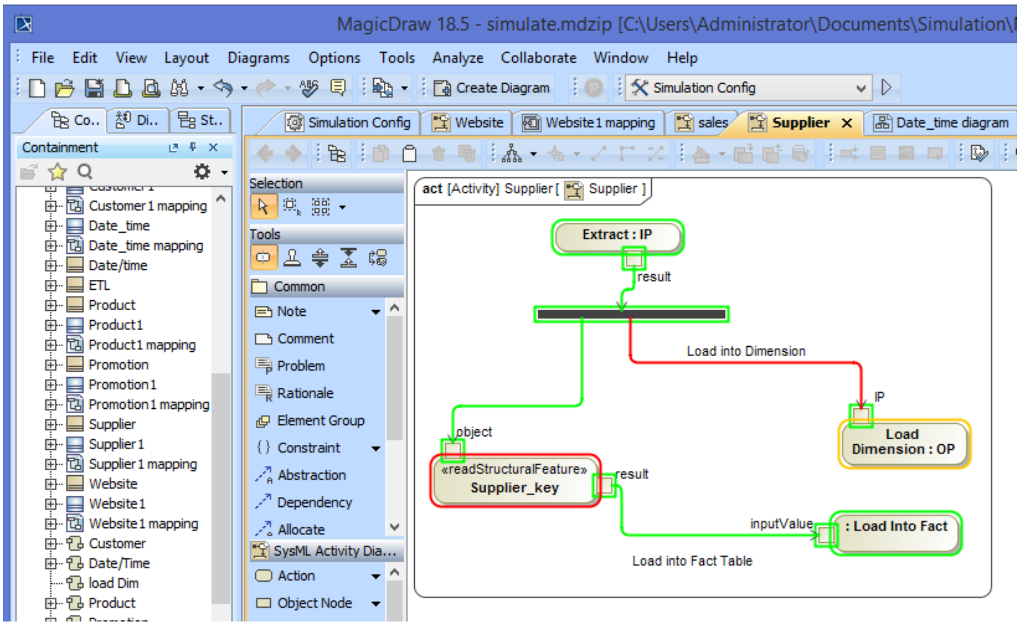
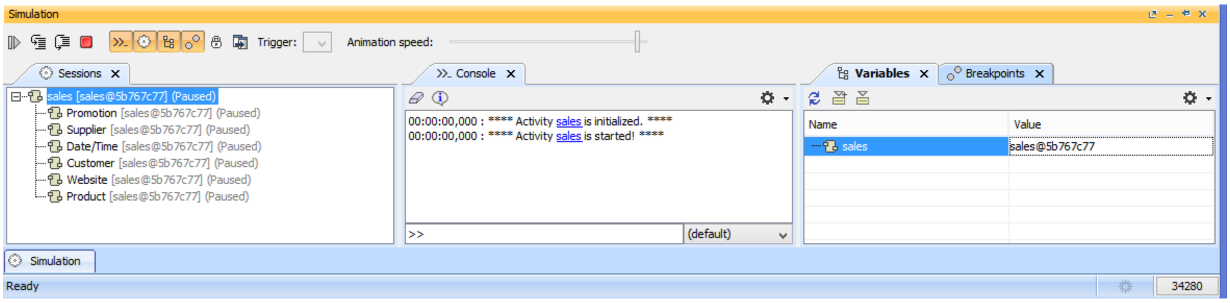

MagicDraw 18.5 - simulate.mdzip [C:\Users\Administrator\Documents\Simulation\
File Edit View Layout Diagrams Options Tools Analyze Collaborate Window Help
Simulation Config
act [Activity] Supplier [Supplier]
Extract : IP
result
Load into Dimension
IP
Load Dimension : OP
object
«readStructuralFeature» Supplier_key
result
inputValue
: Load Into Fact
Load into Fact Table

Figure 5.4: Supplier Sub-activity Simulation

The model can be executed either from the diagram pane or from the Context Menu in the containment browser. A simulation window will open as shown in Figure 5.5. There is three sections in the simulation window: Sessions, Console, and Variables. The simulation session holds the running elements of the model, and the Console pane shows the outputs and The variables pane shows the run time object of the main Activity.

After running the simulation project, the sample output generated on the console pane



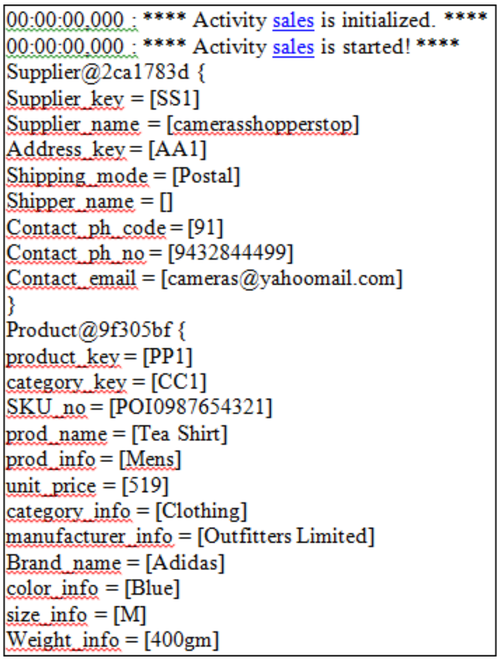
```
sales [sales@5b767c77] (Paused)
    Promotion [sales@5b767c77] (Paused)
    Supplier [sales@5b767c77] (Paused)
    Date/Time [sales@5b767c77] (Paused)
    Customer [sales@5b767c77] (Paused)
    Website [sales@5b767c77] (Paused)
    Product [sales@5b767c77] (Paused)
```

```
00:00:00,000 : **** Activity sales is initialized. ****
00:00:00,000 : **** Activity sales is started! ****
```

Name	Value
sales	sales@5b767c77

Figure 5.5: Model Simulation Window

is given in Figure 5.6 is generated at the console pane. The overall activity simulation project is successfully extracting the source data and loading the data into the Fact and Dimension tables.



```
00:00:00,000 : **** Activity sales is initialized. ****
00:00:00,000 : **** Activity sales is started! ****
Supplier@2ca1783d {
Supplier_key = [SS1]
Supplier_name = [camerasshopperstop]
Address_key = [AA1]
Shipping_mode = [Postal]
Shipper_name = []
Contact_ph_code = [91]
Contact_ph_no = [9432844499]
Contact_email = [cameras@yahoomail.com]
}
Product@9f305bf {
product_key = [PP1]
category_key = [CC1]
SKU_no = [POI0987654321]
prod_name = [Tea Shirt]
prod_info = [Mens]
unit_price = [519]
category_info = [Clothing]
manufacturer_info = [Outfitters Limited]
Brand_name = [Adidas]
color_info = [Blue]
size_info = [M]
Weight_info = [400gm]
```

Figure 5.6: Simulation Output

5.5 Summary

The ETL process is responsible for the selection and extraction of data from several sources, then cleaning and transformation according to the desired format is done and finally updated into a DW. ETL process modeling is a way to design the orientation of data and establish their relationship throughout the ETL processing activity.

In this chapter, the main focus is to model an ETL process at the conceptual level. A significant number of works have been done for ETL process modeling by UML, BPMN, or Semantic web-based methods. In this work, we proposed an MBSE-oriented system model for the ETL process for the data warehouse environment. For the job, a new modeling language called SysML is used, which is gaining popularity for modeling nowadays. It is derived from UML by giving some additional facilities to the system engineers. By using SysML, the system model can be designed in a more expressive as well as flexible way. An example of an e-commerce system for ETL process modeling is discussed in this work. Particularly, propagation of data from sources to DW is explained as a use case of the model. Our developed model is a Platform independent by nature and simple to understand by both technical and non-technical users. After designing the ETL model using SysML language, its corresponding executable XMI code is generated.

SysML makes it possible to use a Model-Based Systems Engineering (MBSE) methodology to increase output and quality while lowering risk when developing systems. The simulation model's primary benefits include: Without creating the system, examine its behaviour. Since, the principles are based on study and actual experiences, it is exceedingly challenging to develop an absolutely realistic model or simulation. The biggest drawback of simulations is that they aren't accurate representations of reality, and occasionally it might be challenging to comprehend the outcomes.

In this extended work, the ETL model is validated using a visual modeling tool provided simulation environment. We have chosen a tooled approach to simulate the proposed model. From the simulation model, it is possible to analyze how different data files are extracted from the source and loaded into the fact and dimensions.

In the future, we intend to extend the model view at the logical and physical level and create an integrated MBSE-based framework for implementing the ETL process.

Chapter 6

Empirical Analysis

6.1 Empirical Analysis of Programmable ETL Tools

ETL tools [281] come as a solution provider by offering a user-friendly graphical user interface (GUI) to map data items between the source and target system in a fast hassle-free manner. In spite of developing and maintaining custom hand-coded ETL systems, it is easier and faster to select and use any ETL tool. The User needs to configure the tool as per their requirement. Many open-source (e.g.Pentaho Kettle [1], Talend [2]) and commercial (e.g. Informatica[3], SAS, ODI [4], IBM[5]) ETL tools comes with nice GUI which is easy to use for non-programmers. Using this type of tool, developers design the visual flow of data throughout the ETL process. One disadvantage of this kind of visual approach is, sometimes it is difficult to design a specific ETL scenario with the limited item available in the graphical tool.

Writing a few lines of code can be a better way for this type of problem. Because it is tricky to drag icons, draw flow lines, setting properties for a complex case design with respect to writing own customizable ETL codes. Here, one of the main considerations should be the productivity of any system. Using any GUI-based tool cannot assure more productivity compared to a code-based approach. Generally, ETL development is done by skilled technical people. So it is justified to go for a code-based ETL option rather than GUI based option. We agree that a graphical program is effective for self-documentation and standardized features. But still, there is some aspect where a code-based approach can give an effective solution. Coding your own data pipelining for extraction is a fascinating job. But it is a difficult task. Now, many companies are opting to write their own code/ scripts for data integration in the cloud environment. One of the main benefits of the code-based approach is that any type of customizations can be done which sometimes is not offered by the existing GUI-based ETL solution.This code-based approach can be beneficial in terms of flexibility, performance optimization, and self-services. There are some issues also for any code modification and maintaining purpose. Only skilled technical people can handle it. But for GUI based approach any non-technical people can also handle workflow scheduling, mapping, tasks, and jobs after a little bit of training session.

Some review article [237, 161, 202] over ETL tools is done so far. However, they are typically done over commercial ETL tools available in the market. And most of the tools are open source. Only high-level view is included by those works without covering any technical details. But no such work is noticed so far regarding the code based ETL tool developed by academic peoples. The focus of this work is to give an integrated analysis report in the research domain of programmable ETL system. For this purpose four prominent work on ETL framework is selected namely Pygrametl, Petl, Scriptella and R_etl. Each of the evaluated ETL tools is discussed with their unique features in next Section.

[1] http://www.pentaho.com/product/data-integration

[2] http://www.talend.com/products/data-integration

[3] https://www.informatica.com/in/products/data-integration.html

[4] http://www.oracle.com/technetwork/middleware/data-integration/overview/index.html

[5] http://www-03.ibm.com/software/products/en/infosphere-information-server/

6.2 ETL Tools Overview

There are numerous ETL tools available in the market. Each of the tools offers its own features and limitations. But most of the ETL tools are GUI based. The availability of less degree of customization facility for modeling and integrating extension environment in GUI-based ETL tools has leads many organization to go for programmable solutions for ETL process. In this work, we have selected some code based ETL tools. All these tools are open source and no graphical user interface is offered. Basic introduction about these selected tools are discussed below.

6.2.1 Pygrametl

Most remarkably, Pygrametl [274, 273] is an open source python based ETL framework first released in 2009. This software is licensed under BSD. Till now continuous up-gradation is done on this tool.

Without drawing any ETL process using GUI based tool, Pygrametl [6] suggest performing ETL tasks by writing python codes. It offers some commonly used ETL functionality to populate data in DW. The data flow can be achieved into three stages, namely extraction, cleaning and insert into DW. Data is represented using python dictionary having key and value pair. PostgreSQL, MySQL, Oracle are the supported databases. Seamless integration of any new kind of data source can be done using merge-join, hash-join, union-source functions. Both the batch or bulk load can be performed as per the requirement.

It is easy to populate fact and dimension tables from the source data through one iteration. It offers to insert data into star dimension or snowflake dimension which span into several tables. Besides it provides advancement on dimension support applying SCD type 1 and 2.

6.2.2 Petl

Most notably, Petl [7] is a general purpose Python package which is able to perform conventional tasks of ETL. This package is supported under MIT License. Petl provides support both object-oriented and functional programming style. A well explained documentation is available to implement general ETL tasks. Petl can handle wide range of data sources with structured file like CSV, Text and semi-structured file like XML, JSON etc. PyMySQL, PostgreSQL, SQLite are three compatible databases with this package.

Petl support maximum transformation patterns required in any ETL process. Besides timing, materialized view, lookup etc. utility function provide extra benefit to the developer. Addition of any third party package can be easily done within it. Efficient use of memory

[6] http://www.pygrametl.org/

[7] http://petl.readthedocs.io/en/latest/

is implemented by the use of lazy evaluation and iterator. ETL data flow are synchronized using ETL pipelines. However it does not have SCD or parallelism handling mechanism.

6.2.3 Scriptella

Scriptella [8] is another script based ETL tool written in Java. It is licensed under Apache Version 2.0 [9]. Plain SQL queries are executed using JDBC bridge in this scripting language. In case of non-JDBC provider can be added using mixed SQL script. For describing various ETL task, XML script is used. SQL or other scripting language can be used for transformation purpose.

The main application is focused on executing script those are written in SQL, JEXL, Javascript and velocity for the purpose of ETL operations to/from various databases as well as file format like text, CSV, XML, LDAP etc. A thin wrapper created by XML script can give extra facility to make dynamic SQL script.

Multiple data sources can be added to an ETL program with additional support to some JDBC features like batching, escaping etc. No installation is required for deploying the tool or it can be worked as *Ant* task. Only JDK or JRE with version above 5.0 is required. Execution of this tool is also very simple. It is compatible with many popular databases having JDBC/ODBC compliant driver. For non-JDBC data sources a Service Provider Interface (SPI) is developed. It's integration provision cover Java EE, JMX, Spring framework, java mail, JNDI for easy scripting with enterprise standards.

Basic ETL task can be executed but with limited transformation support. Both batch load and bulk load can be implemented through this tool. It does not provides any support for parallelism as well as warehouse specific facility like SCD. Scriptella does not provide any GUI facility.

6.2.4 R_etl

Now a days, R is a promising language which is gaining popularity in the field of Data Science. A newly developed package for R [23] named *etl* is selected for this piece of work [10]. It is licensed under CC0 with version 0.3.7 and available in CRAN [11]. It provides a pipeable framework to execute core ETL operations. It is suitable for working with medium size data.

This *etl* package can work as a basis for extending its dependent packages for managing any particular data sets. Seven open source and cross-platform dependent packages are available to easily access and analyze publicly accessible medium data sets (PAMDAS). This *etl* package can be extended to perform ETL operation for any data which is stored in an R package.

[8] http://scriptella.org/

[9] https://github.com/scriptella/scriptella-etl/wiki

[10] https://cran.r-project.org/web/packages/etl/README.html

[11] http://github.com/beanumber/etl

RPostgreSQL, RPostgreSQL, RSQLite are the DBI drivers for R is compatible with this package. It is suitable to handle data which can reside either in the local or remote database. Database creation or management can be done without having any expertise in SQL. Some utility functions like dbRunScript, smart_download, smart_upload, src_mysql_cnf etc. can provide some additional benefits to the developers. Very few lines of code is required to implement this tool. But only some basic ETL functionalities are enabled here. It does not meet the requirements of current ETL technologies.

6.3 Characteristic based benchmarking

After deploying each tool on-premise different features have been identified. On the basis of these characteristics, a comparison table is done. The comparison matrix is given in Table 6.1 represents a brief overview of these tools. These are the general characteristics which can be taken as a criteria when evaluating any ETL tool. Benchmarking of these tools can be done upon these selected characteristics.

On the basis of the characteristic specification analysis we can have a deep insight into the selected code based ETL tools. The overall observation is that, Easy usability plays an important role in ETL tool benchmarking. The tool should be easy to use, and easily understandable. Data centric approach should be followed there. Functionality used in these tools should be reusable. Tools should support basic and complex transformation tasks. Scalability is one of the desired features. It includes bulk data handling with partitioning, clustering and parallelism support. Data mapping should be transparent and easy. Unstructured data support gives an extra benefit to the modern users. Cloud enabled ETL tools having big data handling mechanism are the most demanding features now a days. Apart from that there are many other specifications, that are considered for the comparative analysis.

6.4 Experimental Analysis

The availability of different functionality about these tools makes it difficult to create a ranking. Because all of them has some special type of features. So, respective aspect is the main point to choose any tool for use. General specification of these tools are discussed in the previous section. This section will discuss about performance evaluation of each tools based on their characteristics.

6.4.1 Performance Analysis

All the ETL tools have been deployed in the local machine. The hardware specifications of the machine and the software description is given below.

Hardware Specifications: The hardware configuration is as follows:

- Processor: Intel(R) Core(TM) i3-4170 CPU @3.70 Ghz 3.70Ghz

Table 6.1: Feature comparison matrix on ETL solution provider

Specifications	Pygrametl	Petl	Scriptella	R_etl
Easy usability	N	Y	Y	Y
Data centric approach	Y	Y	Y	Y
SOA-enabled	N	N	N	N
Reusable functionality	Y	Y	N	Y
Single installation	Y	N	N	N
Big Data handle	N	Y	Y	N
Data segregation	Y	Y	Y	N
Real-time triggers	N	N	N	N
Unstructured data support	N	Y	Y	N
Multiple source join	Y	Y	Y	N
Complex transformation	N	Y	N	N
Data validations	N	Y	N	Y
SCD Support	Y	N	N	N
Parallelism Support	Y	N	N	N
Bulk Load	Y	N	Y	N
Data pipeline	N	Y	N	Y
Easy data mapping	Y	Y	Y	Y
Lookup support	N	N	Y	N
Code Re-usability	Y	Y	N	Y
Exception Handling	Y	Y	Y	N
Documentation available	Y	Y	N	Y
Third-party dependency	N	Y	Y	Y
Version control	Y	Y	Y	Y
Deploy in cloud	N	N	N	N
Licensed	Y	Y	Y	Y
Community based forum	Y	Y	Y	Y

- Installed Memory(RAM): 4.00 GB
- System Type: 64-bit Operating System, X64 based processor
- Operating System: Windows 8.1

Tool Specifications: Four ETL tool/package have been installed in the machine. Tool specifications are give here.

- *Pygrametl* Version 2.6 has been installed with database PostgreSQL and MySQLdb for deploying purpose. Python 3.6 version is employed in IDE Spyder.
- *Petl* Version 1.1.1 is used. It does not have any installation dependencies. SQLAlchemy and PostgreSQL databased has been used. Here also Python 3.6 is used in Spyder.
- *Scriptella* Version 1.1 has been installed along with HSQLDB database.
- *etl* package version 0.3.7 and PostgreSQL is installed along with DBI RPostgreSQL. For IDE RStudio is utilized along with R version 3.4.4.

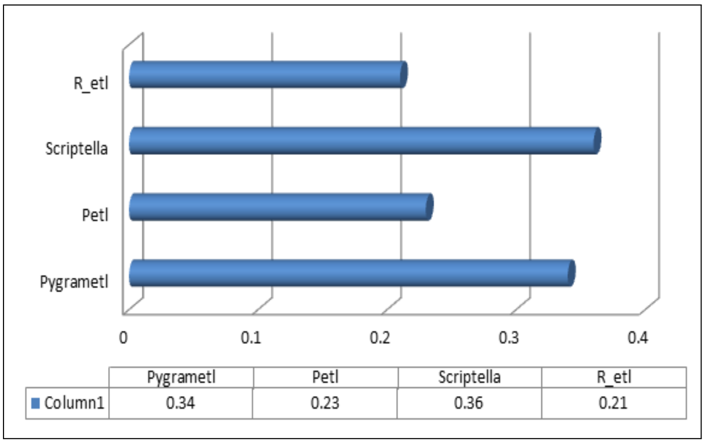
We have evaluated the performance of four code based ETL tools on the basis of four criteria. The criteria are time of execution time, transformation support, throughput, and code length. Each of the cases has been evaluated on these tools. After evaluating, the results have been graphically represented.

Execution Timing

Quantifying the amount of time taken to execute any ETL process is the most important case to analyses any tool. For this purpose, each of the tools is executed on premise. The execution time taken for each tool are collected and plotted in a graph in Figure 6.1. Among the two sources input file, one file contains 120 row elements and another file contains 8 row elements which need to move in DW. The execution time is calculated in number of seconds. It is observed that R_etl is much efficient than other three options with respect to execution timing taken.

Transformation Support

Performing required transformation is the most important task in ETL process. The key process is to selecting proper source data and apply required transformation rules. Generation of correct data depends upon the successful completion of the transformation process. So it is a key aspect for any data integration tool is to provide a good range of data transformation support. In this study, a list of transformation supported by each tool has been identified. Based on the range of transformation support provided by each tool a comparative graph has been drawn in Figure 6.2. It is observed that Petl is the highest number of



	Pygrametl	Petl	Scriptella	R_etl
■ Column1	0.34	0.23	0.36	0.21

Figure 6.1: Execution Timing

transformation variety provider. Transformation type supported by Petl are Basic transformations, Header manipulation, Regular expressions, Unpacking compound values, Converting values, Selecting rows, Transforming rows, Sorting, Deduplicating rows, Reducing rows (aggregation), Joining, Set operations, Reshaping tables, Missing values filling, Intervals and Validation.

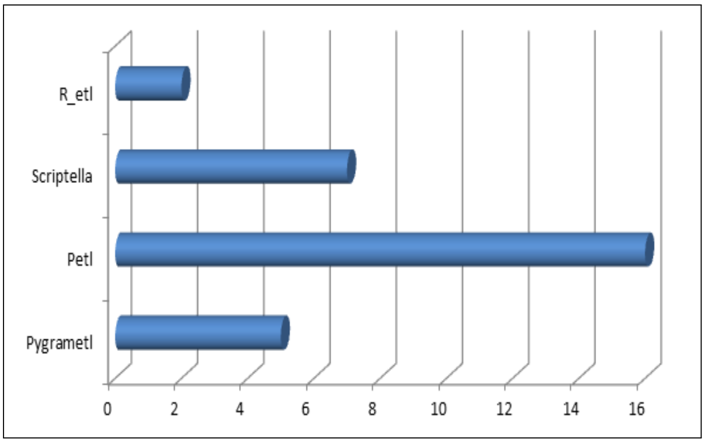


Figure 6.2: Transformation Support

Throughput

To evaluate any system efficiency, throughput is one of the important metric. Throughput calculation is one of the most common task for evaluating system performance. Here, throughput is the amount of data processed per time. For any data integration project, always high throughput is desirable. Here throughput of each ETL tool is measured after deploying each four system on premise. The graphical Figure 6.3 presents the throughput performance of them. Here the throughput is measured with respect to the number of row processed per second. It is observed that performance of R_etl is better compared to other ones.

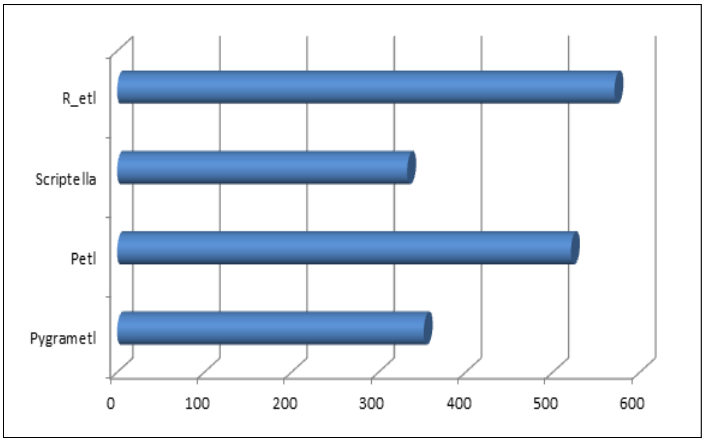


Figure 6.3: Throughput Rate

Line of Code

It is a measure of how many lines of code (LOC) is required for accomplishing the total ETL process. Any comment or blank lines are not counted. LOC is one of the metric used in software engineering for cost estimation. LOC used in assessing a project's efficiency. Besides it helps to predict the effort as well as time required to construct any software project. Here we have taken LOC as a measurement parameter within the selected tools. A comparative graph is presented in Figure 6.4 will show the line of code required for each tool to establish the task. A approximate value is considered for this case. Less line of code means how easily and in less time a code for implementing ETL can be written. From the graph, it is visible that *etl* using R takes less number of code than the others.

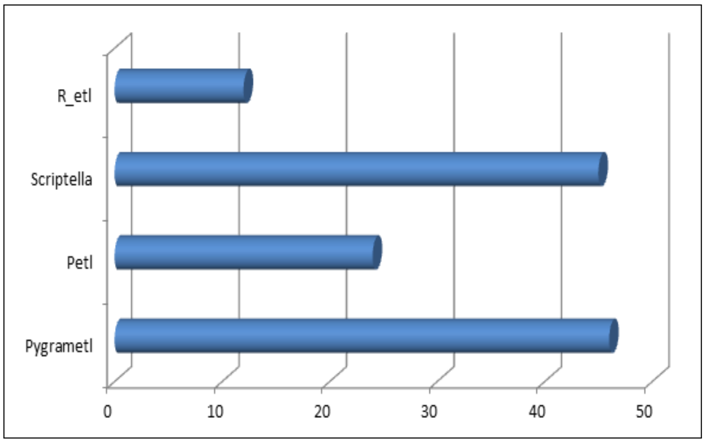


Figure 6.4: Line of Code

6.5 Summary

At present, the requirement of the continuous and increasing amount of data handling in more complex environment is a great challenge in the research domain. It demands standardized ETL process which has a great business impact on the BI industry. Most of the organization opt for taking any vendor made GUI based product for their ETL solution. But still in some of the cases custom coded ETL can be the best option in respect of performance.

This work has chosen the second option for ETL solution. Four promising code based ETL tool Pygrametl, Petl, Scriptella and R_etl has been chosen here. Their overall characteristics has been studied as well as deployed. An experimental and feature analysis is presented on these type of tool.

Regarding the drawback of the work presented in this chapter, some more variants of code base ETL tools can be selected. For performance evaluation, here mainly four criteria (Execution Timing, Transformation Support, Throughput Rate, Line of Code) are considered. Some more criteria (like cloud data integration, complex transformation, big data handling, data streaming etc.) can be scrutinized.

This proposal summarizes as well as evaluated recent work in the domain of programmable ETL development approach in a novel way. The main objective does not point out on which tool is good or bad. It totally depends on particular requirement and competence of any organization like scalability, costing, infrastructure support and more. Hope that this piece of work will help to grow a deep perspective in the field of ETL process.

Chapter 7

Real-Time ETL

7.1 Real-Time Data Integration

The traditional ETL process is the backbone of modern data integration approaches. From small to big multinational companies depend upon the ETL process for maintaining and updating their everyday transactional data, having deep insight over it for making better competitive managerial decisions and serving their valuable customers. Traditional ETL approaches were mainly batch-oriented. This Means ETL jobs were performed in batch mode at fixed time intervals. For example, a first food restaurant chain needs to perform batch ETL to store and calculate its daily revenue. It will schedule the ETL job execution daily at midnight after the restaurants are closed. Traditional ETL exactly fits these requirements. It doesn't need a real-time ETL process.

But nowadays, many organizations are choosing real-time ETL solutions [283, 146]. Here real-time means the data should be propagated into the Data warehouse as soon as they are available on the source side. For example, for fraud detection purposes in any banking system, real-time ETL is a smarter choice compared to the traditional approach. Whenever any credit card holder makes an online transaction, the company needs to investigate if it is a fraud or not, and an alert should be sent to the owner immediately (within minutes) about the suspicious transaction. If any transaction is done in any unusual shop, new city or country, or any unusual time, it will give an alert.

With the continuation of code based ETL tool approach, a new solution model is proposed to meet the near real-time ETL demand. The target is achieved by implementing incremental loading [119] model with the assistance of the CDC (Change data capture) approach [269]. The contribution of this work is to highlight a new area by programmable ETL development technique. The work is continued through designing a new ETL-based data integration technique. The proposed solution makes the data integration much more efficient by incrementally populating only the changed data in the DW at the right time. Afterward, the proposed ETL model is discussed with algorithmic details.

7.2 Modeling ETL Jobs

The main objective of the work is to design a model for a code-based ETL tool by which we can reduce the data flow and latency for achieving near real-time ETL processing in the DW environment with respect to incremental loading by utilizing the CDC technique. This approach can noticeably cut down the ETL processing latency by reducing the amount of data propagation throughout the extraction, transformation, and loading stages of the ETL workflow.

7.2.1 Incremental Loading

The loading task in the DW is executed as a background process at a certain time interval. Initial load is performed for the foremost DW population purpose. Afterward, if any modification or new data arrives at the source side, DW refreshing is done by a full reloading process to make the repository up to date. However, nowadays, with the increasing data size, high data rate, and complex data structure, full reloading becomes inefficient and inadequate. The more practical approach is continually updating DW on the changes made in the source data since the previous reload. This approach is termed as incremental loading [119, 121]. This type of loading is much more efficient compared to full load. Incremental loading can be implemented using two methods: Source Change Identification and Destination Change Comparison.

The main idea of Source Change Identification is to capture only the changes in source data and propagate them to the DW. The process of extracting only the new data and propagating it into the Data warehouse is implemented using the CDC technique. If the source system of the ETL process does not support source change identification, then source-to-destination comparison can be the other way to identify the changed data and select the data that should be inserted.

7.2.2 Change Data Capture (CDC)

CDC is one of the most appropriate techniques for implementing real-time features for any ETL application. It is a new real-time data integration approach that identifies and captures changes that occur to data sources and delivers only the changed data to the operating system [70]. This approach does not need DW downtime or batch windows of ETL. Some CDC technologies operate in batch mode with a pulling technique. Means that the ETL tool periodically receives a batch for all new changes made up to the last received and execute them. The real-time CDC solutions apply a continuous streaming “push” approach to delivering data. The data changed at the source side are captured and delivered immediately to the target.

The advantage of CDC is the latency can be cut down to minutes or even seconds, which makes the data instantly available, eliminating the use of batch windows. Besides, it minimizes the amount of data flow therefore resource requirement is minimized, and data flow speed and efficiency are maximized. CDC addresses some business needs like building Operational Data Stores (ODS), Business Activity Monitoring (BAM), Application Integration, Real-time Dashboards, data quality improvement, etc. There are several techniques by which CDC techniques [13, 45] can be implemented for detecting the change.

Transactional log: Most of the data sources maintain a change log which keeps a record of all changes performed on it. This log keeps track of the date and time for the last modification in the table. This transactional log can be used to capture the change. This log is automatically created by an active database trigger. Analyzing those log files does not affect the operational database.

Database log scraping & log sniffing : This technique takes a snapshot of the transaction logs maintained by the database system for backup and recovery at a scheduled time. The later techniques involve “pooling" of the active log file and identifying changes in real-time. The first approach has a higher latency value.

Snapshot Differential: During the extraction phase a snapshot of the complete source table is taken. Changed data can be identified by comparing consecutive snapshots [153, 145]. This technique is introduced in, and improvement is going on literature [211]. It is a time and resource-consuming process but an easy approach.

Timestamped index The operational system often maintains a timestamps column for keeping track of the last update. This column refereed as *audit column* generate a new timestamp for any modification in the tuple. These audit columns can be used to identify new changes since the last loading cycle.

Database Trigger It is a special type of activity in a database system that is fired basis on some predefined function [262]. The trigger can be set on every (add, delete and update) event for finding new data. The output of the trigger program, which is stored in another file, can be used for extracting data. This type of application is suitable for source systems having database applications. However, the trigger-based system can have a performance impact on the source system.

7.3 Proposed Approach

For implementing incremental loading, the proposed work is divided into three parts. In the first part of the work, snapshot-based CDC is implemented, which captures only the changes of data from the input dataset and loads it into the data warehouse. Here the main motive is to indicate an efficient comparison between the previously loaded record, and it's corresponding new record and capture the changes and load only those changes efficiently in the DW. An algorithm for dimension processing is described in the second part. Finally, in the third part, an algorithm for fact processing is represented formally.

For the algorithm, four database tables and one input dataset are taken. The first table is the main table which contains previously loaded data. The second and third tables are dimension tables that will be used at the time of dimension processing. The fourth one is a fact table Tab_{fact} and will be used at the time of fact processing. Let's assume that the name of the main table is Tab_{main} and the name of first dimension table is $Tab1_{dim}$ and $Tab2_{dim}$. Now let's focus on the input dataset. It is considered as a new record New_{data} which needs to be extracted from the source and needs to be loaded into the main table.

For Algorithm 1, At first New_{data} input dataset is taken as a pandas *DataFrame* DF_{new}. Next, previously loaded record Tab_{main} table is fetched in a *DataFrame* $DFTab_{main}$ from the database. It is considered a previously loaded record. Now we will briefly discuss Algorithm 1. At first, we've taken New_{data} dataset as input and it is taken in a pandas *DataFrame* DF_{new}. As discussed earlier, we can consider it a new record. Next, we've fetched Tab_{main} table in a Pandas *DataFrame* $DFTab_{main}$ from the database. We can consider it as a previously loaded record or the old record. After that, check whether this $DFTab_{main}$ *DataFrame* is empty or not. If the main table $DFTab_{main}$ is empty, load data from the *DataFrame* DF_{new} in Tab_{main} table using bulk load.

If $DFTab_{main}$ is not empty, then we can determine the changes that occurs in New_{data} dataset by comparing the newly inserted record and the old record. For this purpose first we concatenated the *DataFrame* of newly inserted record, DF_{new} with *DataFrame* of the old

Algorithm 1: Change Data Capture Algorithm

```
Result: Dataframe of new, updated and deleted records
$DF_{new} \leftarrow$ Take $New_{data}$ dataset as Input from Machine;
$DFTab_{main} \leftarrow$ Fetch $Tab_{main}$ table from Database;
if $DFTab_{main}$ is empty then
    Load $DF_{new}$ in the Database ;
else
    $IN_{df} \leftarrow$ Concat $(DF_{new}, DFTab_{main}, DFTab_{main})$. drop_duplicate();
    $UP_{temp} \leftarrow$ Concat $(DF_{new}, DFTab_{main}, DFTab_{main})$.drop_duplicate
      (subset=key);
    $UP_{df} \leftarrow$ Concat $(IN_{df}, UP_{temp})$.drop_duplicate();
    $DEL_{df} \leftarrow$ Concat $(DF_{new}, DF_{new}, DFTab_{main})$.drop_duplicate (subset=key);
end
```

record, $DFTab_{main}$ and again concatenated this resulting *DataFrame* with the *DataFrame* of the old record, $DFTab_{main}$. Then we removed the duplicate rows from the resulting *DataFrame* and got the *DataFrame* IN_{df} of new elements which were present in the new record.

Afterwards, for getting the updated values from input *DataFrame*, again we've concatenated the *DataFrame* of the inserted dataset, DF_{new} with *DataFrame* of the old record, $DFTab_{main}$ and again concatenated this resulting *DataFrame* with the *DataFrame* of the old record, $DFTab_{main}$. Then we've removed the duplicate rows according to values of the key attribute from the resulting *DataFrame* and got a temporary *DataFrame* UP_{temp}. In the next step again we performed concatenate operation on UP_{temp} and *DataFrame* of new record DF_{new} and after deleting the duplicates using *drop_duplicate()* method we finally got the *DataFrame* of updated elements UP_{df}.

For doing this, first we've taken the *DataFrame* of inserted dataset DF_{new} twice and concatenate it with the *DataFrame* of old records $DFTab_{main}$ and then deleted the duplicate records according to values of the key attribute from the resulting *DataFrame* by using *drop_duplicate()* method and got the *DataFrame* of deleted elements DEL_{df}.

7.3.1 Dimension Processing Algorithm

For Algorithm 2, here we will discuss the dimension processing for real-time data load using CDC. In the first step of the dimension processing algorithm, we've fetched dimension attributes from recently updated main table Tab_{main} in a pandas *DataFrame* $DFDim_{new}$ from the database. We can consider it as a recently updated record or a new record. In the next step, we've fetched a previously loaded dimension table or dimension table which loaded with old data $Tab1_{dim}$ in a Pandas *DataFrame* $DFDim_{old}$ from the database.

Afterwords, for getting the changes in dimension table concatenate *DataFrame* $DFDim_{new}$ with $DFDim_{old}$. After dropping the duplicate, changed data needs to be loaded into the

Algorithm 2: Dimension Processing Algorithm

```
Result: Dimension Table
$DFDim_{new}$ ← Fetch dimension attributes from recently updated main table
    $Tab_{main}$ from database;
$DFDim_{old}$ ← Fetch old dimension table $Tab1_{dim}$ from database;
$DIM_{df}$ ← Concat ($DFDim_{new}, DFDim_{old}, DFDim_{old}$). drop_duplicate();
Load $DIM_{df}$ DataFrame in old dimension table $Tab1_{dim}$ of the database;
```

database dimension table $Tab1_{dim}$.

7.3.2 Fact Processing Algorithm

Algorithm 3: Fact Processing Algorithm

```
Result: Fact Table
$DFTab_{new}$ ← Fetch recently updated main table $Tab_{main}$ from database;
$DFDim1_{new}$ ← Fetch recently updated dimension table $Tab1_{dim}$ from database;
$DFDim2_{new}$ ← Fetch another recently updated dimension table $Tab2_{dim}$ from
    database;
$DFFact_{old}$ ← Fetch Fact table $Tab_{fact}$ with old data from database;
$Fact1_{df}$ ← Merge($DFTab_{new}, DFDim1_{new}$) using common attribute Com_attr1;
$Fact2_{df}$ ← Merge($Fact1_{df}, DFDim2_{new}$) using common attribute Com_attr2;
$Fact3_{df}$ ← Only select the Key attributes from $Fact2_{df}$;
$Fact_{df}$ ← Concat ($Fact3_{df}, DFFact_{old}, DFFact_{old}$).drop_duplicate();
Load $Fact_{df}$ dataframe in old dimension table of the database;
```

For Algorithm 3, the fact processing part of the algorithm. In the first steps of the fact processing algorithm, we have fetched the recently updated main table Tab_{main} in a pandas *DataFrame* $DFDim_{new}$ from the database. We can consider it as a recently updated record or a new record. In the next two step we've fetched two recently updated dimension table $Tab1_{dim}$ and $Tab2_{dim}$ in Pandas *DataFrame* $DFDim1_{new}$ and $DFDim2_{new}$ from database. Then in the next step, we have fetched a fact table Tab_{fact}, which is a table with old records.

Now for determining the changes and updating it in the fact table Tab_{fact}, first we've merged $DFTab_{new}$ and $DFDim1_{new}$ with respect to a common attribute Com_attr1 in the next step. The resulting *DataFrame* is $Fact1_{df}$. Then in the next step, we merged $Fact1_{df}$ *DataFrame* and $DFDim2_{new}$ *DataFrame* with respect to a common attribute Com_attr2. This time the resulting dataframe is $Fact2_{df}$. For merging two *DataFrame* we have used merge() method from pandas dataframe. From $Fact2_{df}$ *DataFrame* we just need to select the key attributes which can reference the dimension tables and make another *DataFrame* $Fact3_{df}$. Then in the next step, we concatenated the *DataFrame* $Fact3_{df}$ and two $DFFact_{old}$ *DataFrame* and dropped the duplicates from the resulting *DataFrame*. In the last step, we have loaded the change data in the fact table Tab_{fact}.

7.3.3 Example Scenario

For our example scenario, A small DW is designed for a book store management system. The book stores are located in various cities. All the transaction records in each book store are saved in this Data warehouse on a daily basis. The Data warehouse contains a single fact table and three dimension tables. The dimension tables are used for storing information about book details, time of sale, and location of book stores. Figure 7.1 shows the star schema constructed with corresponding fact tables and dimension tables.

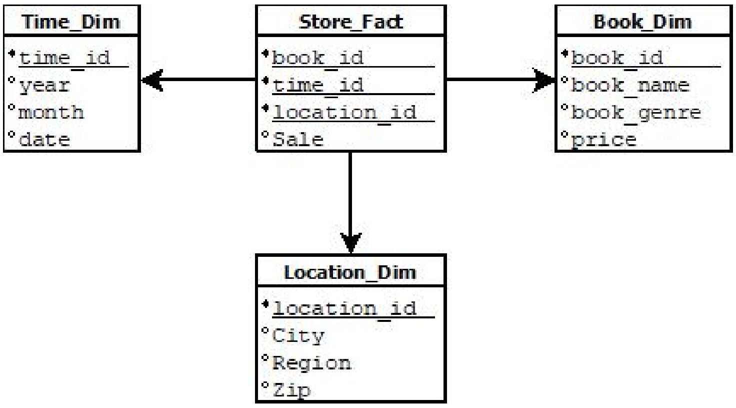


Figure 7.1: DW schema design

Dimensional modeling is an established and very popular methodology for DW design. It reflects the logical schema design of a particular DW. Fact tables usually contain measures of business processes that are referred to as facts. Dimension tables stores textual descriptions of the business entities. The dimension attributes are used to select, group, and aggregate facts of interest in the DW queries. Dimensions can often represent multiple hierarchical relationships in a single table. Dimension tables are usually highly denormalized.

7.3.4 Experimental Analysis

Here, experimental results are provided to display the benefit of incremental loading over full reloading. To implement and maintain the DW, input data is taken from a database table and an external CSV file. A dataset of 60000 tuples was first inserted into a database table. Then six datasets are created having 1, 5, 10, 20, 25, 30, 40, 45 percent of changed data and inserted in different steps. At each step, same numbers of tuples need to be inserted, deleted as well as updated. Full reloading and incremental loading differ in the sense that the former performs *lookups* operation to decide whether tuples need to be inserted, updated, or it should keep unchanged. But the latter on it computes two separate datasets. Deletions are not propagated, because historical data is stored in the Data Warehouse.

The **hardware and software requirement** to run the tests was a PC having Intel(R) Core(TM) i5-4200U (2.3 GHz) processor, a DDR2 4 gigabyte main memory on a 1 terabyte hard disc, SATA standard, transfer rate of 3.0 Gbit/s and 5,400 rpm (rotations per minute). The equipment had Microsoft Windows 8.1 system software, Anaconda Python 3.6 compiler, and PostgreSQL 10.4 DBMS.

7.3.5 Result Discussion

The tests were done in two stages. In the first stage, the time to compute change data for full reloading is measured. A fictitious database containing a single table was used. This table contains 6000 tuples. As previously mentioned, six datasets are given as an input for the time measurement to compute the change data for DW refreshment. In the second stage, the time to compute change data for incremental loading is measured. We have followed the same steps we did in the first stage for measuring the time to compute change data for DW refreshment. A comparative graph is provided in Figure 7.2 for both full load and incremental load using CDC technique. As expected, the time for full reload is considerably slower than the incremental loading, where only the changes are captured and loaded into the data warehouse. However, incremental loading clearly outperforms full reloading. Furthermore, there is no effect on the performance of the proposed method when the source relation is changing dramatically.

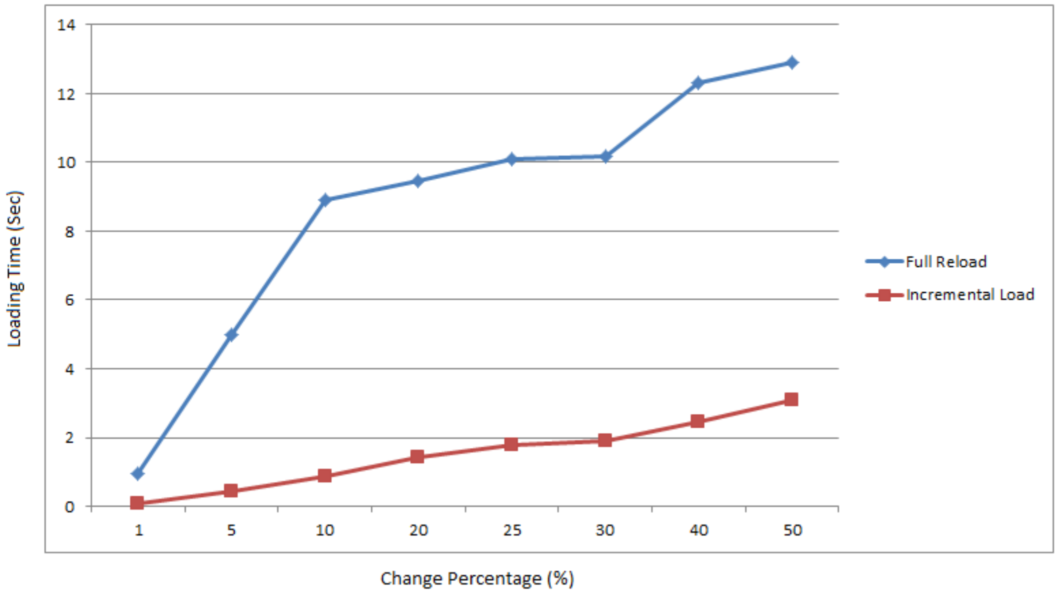


Figure 7.2: Time Comparison between Full load and Incremental load

This proposed model aims to reduce processing overhead throughout the ETL stages and minimize the time complexity compared to the full loading technique. Performance of incremental loading proves quite good compared to full loading. It is observed that the CPU time grows linearly with the size of the dataset growing for both cases. From Figure 7.2 it is clearly visible how the incremental method outperforms full reloading of data into the Data warehouse. The result is expected because when we are performing full reloading, then we need to load the whole table containing the changed data as well as old data in the Data warehouse. However, for the incremental loading, only the changed data is being identified and loaded into the Data warehouse. That's why the size of the data which needs to be loaded becomes much small. For this reason, incremental loading is very efficient for refreshing DW in near real-time.

7.4 Summary

Currently, the requirement of continuous and increasing data handling within a more complex environment is a great challenge in the research domain. It demands standardized ETL process has a great business impact on the BI industry. Most organizations opt for taking any vendor-made GUI-based product for their ETL solution. However, in some cases, custom-coded ETL can still be the best option in respect of performance. This proposal has chosen the second option for the ETL solution.

The main objective of the work presented in this chapter is to design a model for a code-based ETL tool by which we can reduce the data flow and latency for achieving near real-time ETL processing in the DW environment with respect to incremental loading by utilizing the CDC technique. This approach can noticeably cut down the ETL processing latency by reducing the amount of data propagation throughout the extraction, transformation, and loading stages of the ETL workflow. A rich transformation library housed on the Cloud platform will be designed as part of the ongoing development, which seeks to create a unified code-based ETL framework supporting relational and NoSQL databases. Additionally, one can concentrate on sophisticated transformation operators including pivoting, outer joins, and aggregation of data. At the same time, one can have a strategy for making better use of the staging space. It can allow ETL tasks to store data that will later be used as supplementary input in the staging area. Utilizing the staging area can help to some extent eliminate CDC restrictions. Using the staging area, I can use data that has been modified partially. This can anticipate performance enhancements from enduring intermediary results.

Part III

Research Proposal: Recent Trends

Chapter 8

ETL Automation

8.1 Machine Learning in ETL Automation

Wide data storage requires a data warehouse (DW) [115] where the main purpose is for analytical reporting in the future. To construct a DW, data is generally collected from heterogeneous data sources, clean and restructured as per the required standard, and finally loaded into the DW. This is widely known as ETL (Extract Transform Load) [281] which is one of the important components in DW. It is observed that the ETL process consumes a significant amount of time, cost, and complexity overhead of any DW.

Generally, sources of data for DW are operational systems and external systems. An organization's operational systems can be an Enterprise Resource Planning (ERP) system, Customer Relationship Management systems (CRM), and On-Line Transaction Processing (OLTP) system. Any organization does not manage external sources. It can be open data services or other services. Moreover, some data sources are unstructured or semi-structured, like web pages, emails, documents, spreadsheets, texts, or images.

Nowadays, the way of accessing an organization's data is rapidly changing. They want to access real-time transactional data to make an immediate decision. Many industries such as stock exchange, e-commerce, telecommunication, air traffic control, etc., require a correct report based on fresh data in DW to make speedy operational decisions. This kind of decision-making cannot be performed on yesterday's status report. Besides, the volume of data for analysis is becoming very high, and the response time is shortening. So, the demand for a superior and more advanced ETL tool is increasing. Therefore the time window can be shortened for loading in DW. So, the main focus for business intelligence (BI) lies in the DW, and the ETL process for supporting continuous data flow [272, 207] and decreasing downtime.

The main technical challenge of DW regarding the source system is to identify any changes and promptly propagate them into DW. For the near real-time ETL process, some well-known extraction techniques can be considered [13, 45]. They are Enterprise Application Integration (EAI) Middleware, log Sniffing, triggers, timestamping, snapshot differential [283, 211] etc. All the options mentioned above have their own advantages and disadvantages. In this chapter, we will explore if there is any other option that can track changes and automatically initiate the process of loading data in DW.

Fixing data quality issues [19] is a continuous procedure. Data is precious when it is cleaned and processed. Here we will explore how to use the Machine-learning (ML) based pre-processing [141] of data before loading it into the DW. In this chapter, we propose and show how an automated ETL process can be made to manage the growing quantity and variety of data with better quality maintenance. In this work, we will suggest how various Machine learning approaches can be leveraged in this automation.

8.2 Case Study

This section presents three case studies: Marketing, Retail, and Financial Services. The main objective is to showcase the necessity of modernizing the Data warehouse and the

ETL process by automating its processes. A brief discussion is given here.

8.2.1 Retail Domain

In this competitive age, all industries must deliver value faster to improve the user experience and be ahead of competitors. Amazon deploys code every 11.7 sec in a day for production [1]. Netflix deploys code at least a thousand times per day. These farms are adopted in a data-driven culture. Hence deployment and release consist of two parts - application code release and database change release [2]. The process of application code deployment has evolved significantly by adopting DevOps culture [3] and using advanced tools and technologies. So, they can shorten the application release cycle and release more applications in less time. However, the process of database change remains static. The current database release process delays overall application releases by creating a bottleneck in the process. In the current process shown in Figure 8.1, they are facing challenges in keeping track of database changes and synchronizing the databases. Database schema tends to mismatch in different environments, which can cause missing of some critical data. Manual intervention is required for schema changes. Schema changes are not reflected automatically in DW. Overall it takes a longer time to make a change in production that fulfills a crucial need which generates opportunity for competitors.

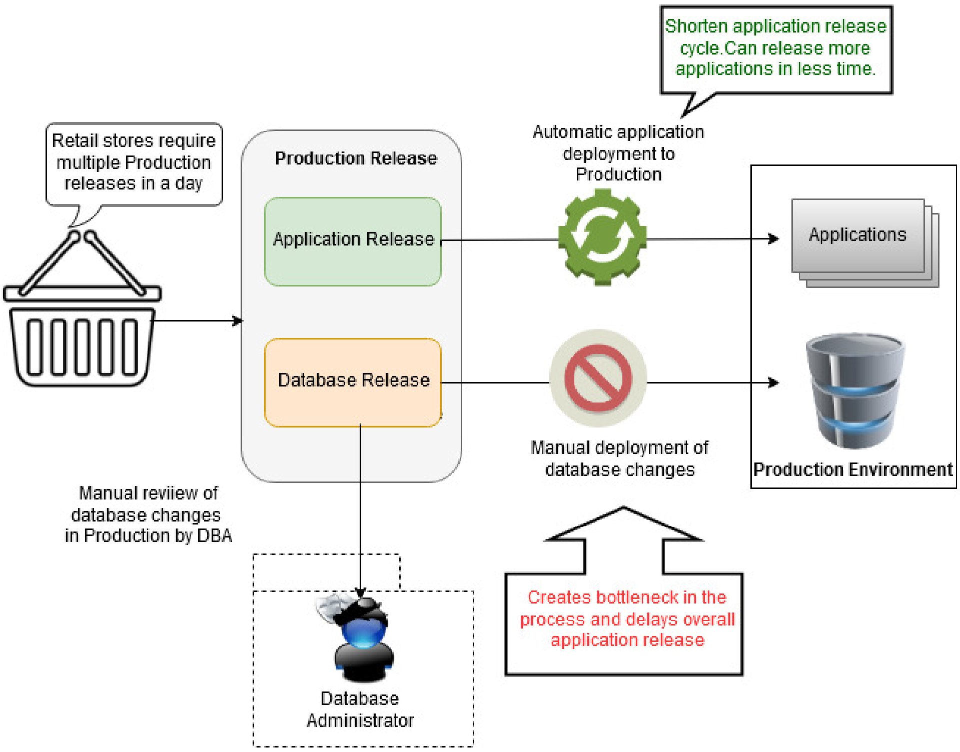

Shorten application release cycle.Can release more applications in less time.
Retail stores require multiple Production releases in a day
Production Release
Application Release
Database Release
Automatic application deployment to Production
Applications
Production Environment
Manual deployment of database changes
Manual reviiew of database changes in Production by DBA
Database Administrator
Creates bottleneck in the process and delays overall application release

Figure 8.1: Challenges In Retail Domain

[1] https://blog.newrelic.com/technology/data-culture-survey-results-faster-deployment/

[2] https://www.datical.com/database-automation/what-is-database-release-automation/

[3] https://docs.microsoft.com/en-us/azure/devops/learn/what-is-devops

8.2.2 Marketing Domain

Consider a Philippine-based retail affiliated marketing company with a large customer database. The company wants to do finer target marketing by emails, an identity which products should up-sell, bring down its churn rate, and gain overall customer satisfaction through DW and BI implementation [4]. This company deals with an extremely large quantity of data. From its 500-plus store locations, 200 million transaction records need to be processed yearly. Running a DW for this company requires multiple operations (around 15-20) performed in the correct sequence at accurate times and conditions. Also, it requires coordination among several teams, including application, database, and operation teams. In the current system shown in Figure 8.2, all these operations are not automated, and manual intervention is required, which magnifies the risk of human-induced errors. Now the challenge is how to automate these processes to ensure that all processes are executed successfully. Data are coming from different operative systems and in multiple formats on this farm. ETL processes do the basic pre-processing and transformation before loading into DW. However, the quality of data is not up to the mark. Missing data or non-accurate data are also causing serious implications in many cases [5]. There are some scenarios where bad quality data disturbs making important decisions. Hence data quality is an area of concern that needs to be addressed.

8.2.3 Financial Services Domain

The success of a financial firm largely depends on gaining new customers and providing accurate investment advice. Most importantly, market analytics and sales strategies should be reliable and responsive daily. Hence the IT department plays an important role in a financial institution to support business needs efficiently. One of the major investment advisory firms [102] has established a Data Warehouse system. In the DW ecosystem, they have used Informatica as an ETL tool and Congo's BI as a reporting tool. This farm is struggling to manage increased information demands and integrate them with the BI processes, as shown in Figure 8.3. Real-time analytics plays an important role when providing investment advice or making any strategic decision. If the company can react quickly and efficiently to new market trends, it will have the advantage of being competitive. Hence this farm wants to address the pain points between the data warehouse and the Cognos BI reporting to increase query efficiency and enable more timely analytics.

8.3 Proposed Solution

For the case study explained in Section 8.2.1, Database Version Control [81] tool is used to track database changes. Also, another tool (Liquibase) [6] can able to identify database schema changes. This tool is integrated with the Database Release Automation component,

[4] http://hosteddocs.ittoolbox.com/aa_data_warehouse_wp_us.pdf

[5] https://www.theseus.fi/bitstream/handle/10024/146311/Aunola_Jere.pdf?sequence=2&isAllowed=y

[6] https://www.liquibase.org/

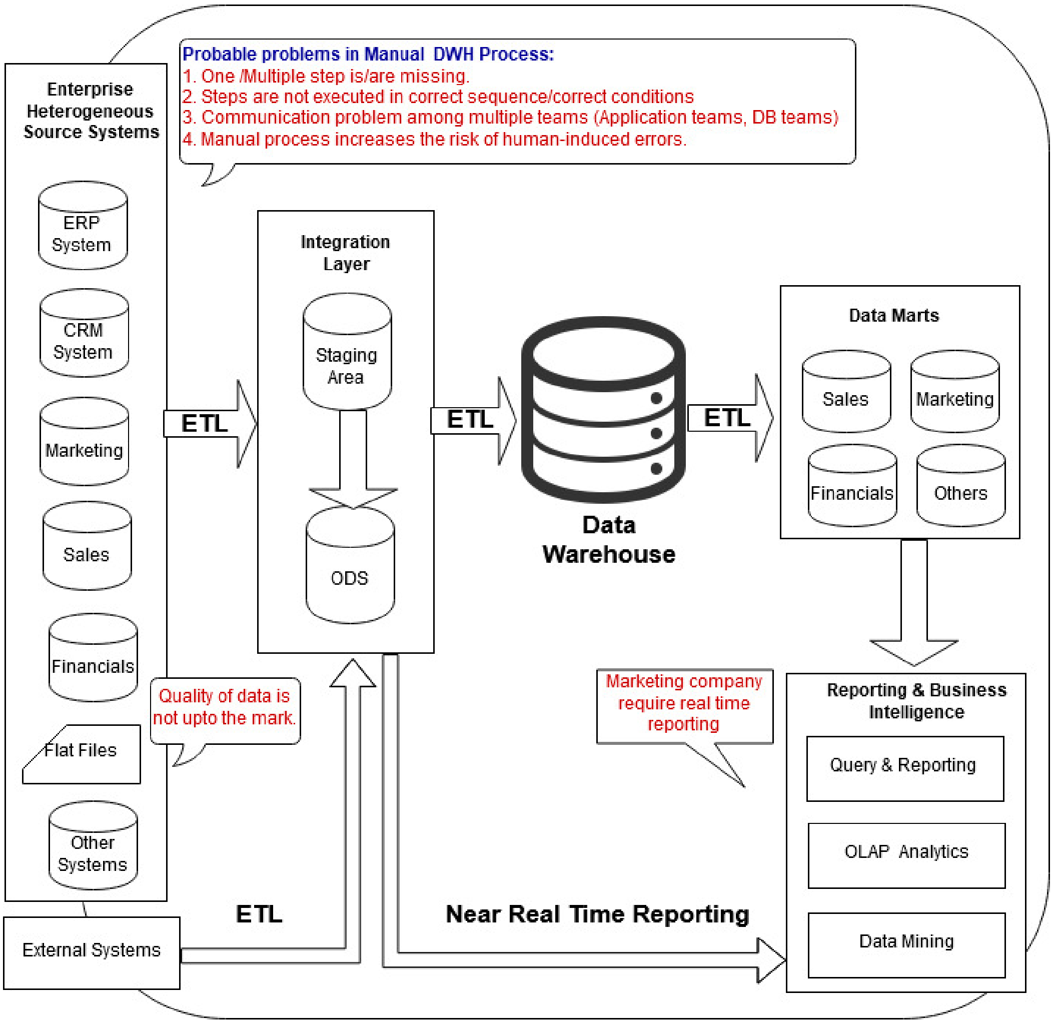

Probable problems in Manual DWH Process:
1. One /Multiple step is/are missing.
2. Steps are not executed in correct sequence/correct conditions
3. Communication problem among multiple teams (Application teams, DB teams)
4. Manual process increases the risk of human-induced errors.
Enterprise Heterogeneous Source Systems
ERP System
CRM System
Marketing
Sales
Financials
Flat Files
Other Systems
External Systems
ETL
Integration Layer
Staging Area
ODS
ETL
Data Warehouse
ETL
Data Marts
Sales
Marketing
Financials
Others
Quality of data is not upto the mark.
Marketing company require real time reporting
Reporting & Business Intelligence
Query & Reporting
OLAP Analytics
Data Mining
ETL
Near Real Time Reporting

Figure 8.2: Challenges In Marketing Domain

which automates the release process of database changes. The company can easily support multiple weekly deployments by using the Database Release Automation component.

For the case study explained in Section 8.2.2, the automated ETL process ensures that all processes in the ETL process are executed in the correct sequence and the correct manner. Machine learning-based data pre-processor is used to pre-process the data more rigorously. It reduces the processing time significantly and produces a good quality of data from the data warehouse.

Similarly, for the case study explained in Section 8.2.3, the automated ETL process helps to reduce the delay occurring between the Data warehouse and BI reporting tool. As soon as database changes are committed to database version control, the automated data integration process is initiated, and changes in data are reflected in the data warehouse. BI tool can generate a report based on real-time data, which is stored in a data warehouse.

Figure 8.4 shows a proposed solution to address the above mentioned problems in the case study section.

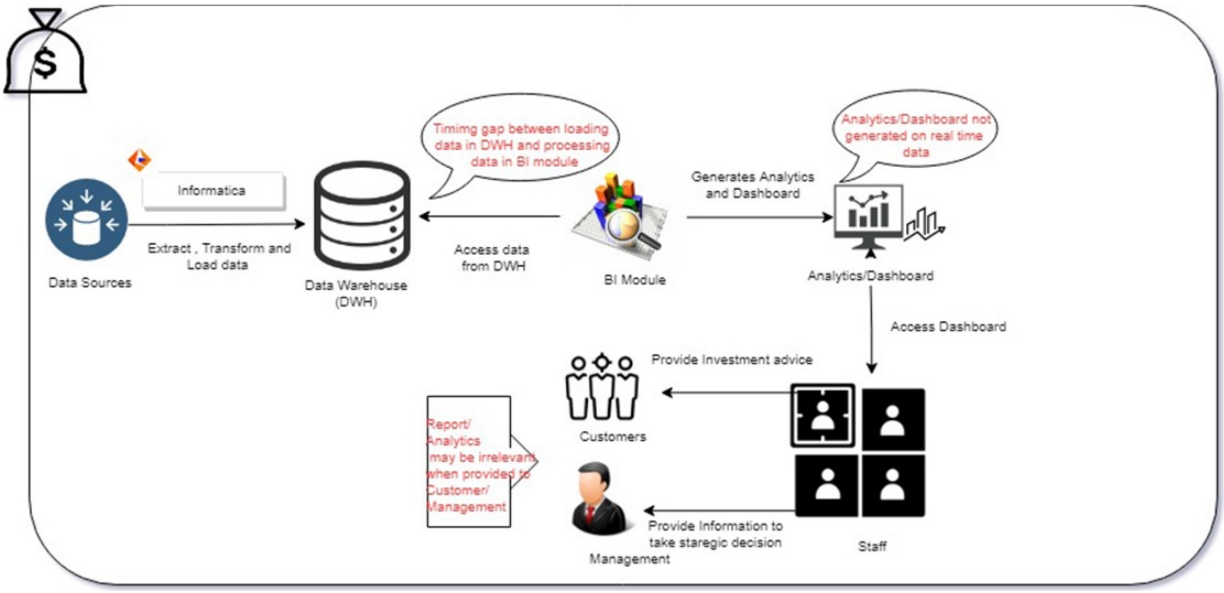

Informatica
Extract , Transform and Load data
Data Sources
Data Warehouse (DWH)
Timimg gap between loading data in DWH and processing data in BI module
Access data from DWH
BI Module
Generates Analytics and Dashboard
Analytics/Dashboard not generated on real time data
Analytics/Dashboard
Access Dashboard
Provide Investment advice
Customers
Report/ Analytics may be irrelevant when provided to Customer/ Management
Provide Information to take staregic decision
Management
Staff

Figure 8.3: Challenges In Financial Service Domain

An architecture is designed to address the challenge faced in the practical application of near real-time ETL processing. Figure 8.5 shows the overall architectural design of the automated ETL system. Proposed data integration steps are discussed in the following subsection.

8.3.1 Automated Data Integration

This work's main objective is overall automation of the data integration process. For this purpose, a Continuous Integration (CI) [7] platform is used to automate the ETL process. Data Integration tool Informatica integrates with Jenkins [8] to build, test and release database changes faster and more frequently. Jenkins is an open-source CI tool that orchestrates the ETL processes with automation. Jenkins pipeline is set up to execute automated scripts to perform the following steps of the ETL process sequentially.

- *Trigger Build Jobs*: Database changes are captured and tracked by the Database Version control. In the proposed solution, Liquibase by Dactical is considered as source control for databases. Liquibase tracks the database changes, including schema changes. A new changelog file is created in XML/YML/JSON/SQL format where changeset is added. Changelog file is committed to source control after running Liquibase update. Jenkins hook detects this change and triggers the build process.

- *Data Load into Stagging Table*: A custom rule engine would be used to classify the data as structured or unstructured using ML classification algorithms. Based on the type, data would be loaded in the staging database.

[7] https://kb.informatica.com/whitepapers/4/Documents

[8] https://jenkins.io/

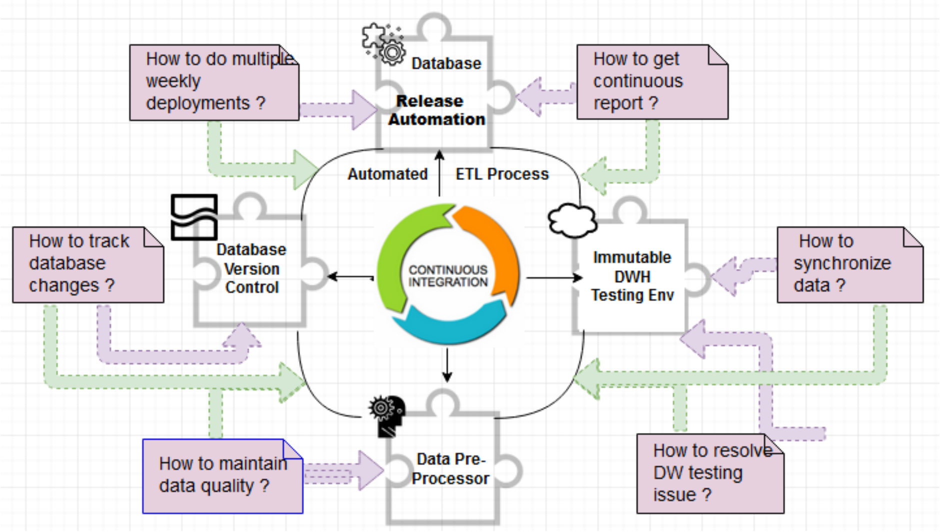

How to do multiple weekly deployments ?
Database
Release Automation
How to get continuous report ?
Automated
ETL Process
How to track database changes ?
Database Version Control
CONTINUOUS INTEGRATION
Immutable DWH Testing Env
How to synchronize data ?
How to maintain data quality ?
Data Pre-Processor
How to resolve DW testing issue ?

Figure 8.4: Proposed solution

- *Data Pre-Processing*: ML-based custom pre-processor processes the data stored in the staging database to improve data quality.

- *Code Build*: Instead of doing a complete database build, an incremental database is built using Liquibase.

- *Automated Code Review and Analysis*: Newly built code is scanned programmatically to identify coding standard violations and other issues like code duplication etc. The proactive Monitoring addon of Informatica reviews the code in an automated manner. The complex event processing engine responds to events and does static code analysis.

- *Automated Test case genereration and Unit Testing*: Data validation test cases are generated using Powercenter Data Validation Option (DVO) addon of Informatica [9]. Unit testing is executed in an automated manner by running pre-build test cases.

- *Save Package*: After successful unit testing, the package is stored in a binary repository. JFrog Artifactory can be used as a binary repository. The binary package can also be stored in the in-build repository of Informatica.

- *Automated Deployment to Pre-production*: The Testing environment is immutable infrastructure hosted in the cloud, which is going to be created on demand for test execution. After completion of testing, the pre-production environment is going to be deleted. Data in the pre-production environment is also generated on demand. The executable package is retrieved from Artifactory by the python scripts and deployed to testing environment.

- *Automated Integration Testing*: There are different tools available in the market to do ETL testing. Informatica Data validation (DVO) tool is used for integration testing.

[9] https://kb.informatica.com/proddocs/Product%20Documentation/4/DVO_100_UserGuide_en.pdf

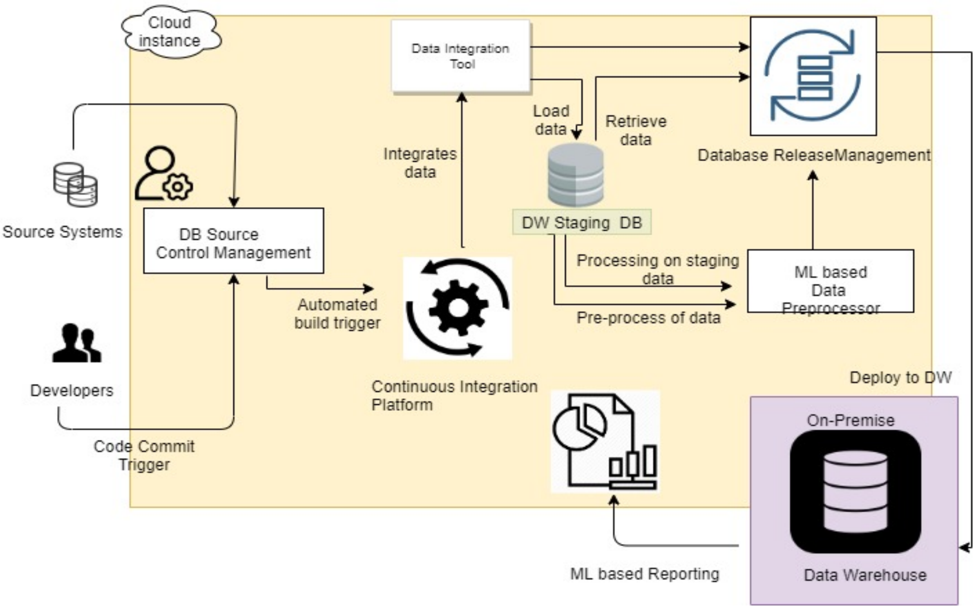


Figure 8.5: Proposed Architecture of Automated ETL Process

Testcases are generated automatically by DVO. Other tools like QuerySurge can also be used for DWH testing.

- *Automated deployment to Production*: After successful integration testing, data packages are deployed in production. The production deployment happens in the same way deployment to the pre-production environment happens.

- *Real time reporting*: Real-time reports are generated from DW using a custom Machine Learning based reporting module.

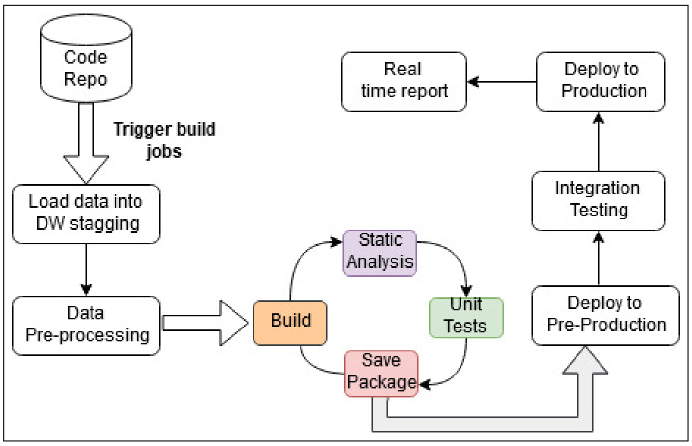


Figure 8.6: Pipeline of Process

8.3.2 Details of Major Components

Database Version Control

In today's world, data is an integral part of any application since the data volume of structured and unstructured data is increasing daily. It is going to be impossible for the traditional database systems to handle it [81]. However, there are some tools like Liquibase, and Flyway available in the market for managing databases. Liquibase is one of the top players in this field. Hence in this paper, we will consider the open source Database management tool Liquibase. As a future scope, we can explore other related tools like flyway, or we can think of building a custom framework for managing database changes. Nathan Voxland developed Liquibase in 2006 for making the database change process easy. Its database-independent library efficiently tracks and manage any database schema changes. Liquibase scripts support updating the schema of any RDBMS.

Custom Rule Engine

Data is coming from different sources in different format. A custom rule engine built on machine learning can be used to specify the class to which data belongs (structured or unstructured). Based on the class [140, 2] of data, appropriate rules can be used to load or transform data. Below classification algorithms have been studied, which can be used to build the machine learning model for the custom rule engines.

Data Preprocessor

Pre-processing of data is a crucial step as far as data quality is concerned. The success of the machine learning model [238] largely depends on the data quality. This stage select target data, prepare it by simplifying it through various filtering and transformation processes and makes it available for ML application.

Database Release Automation

Quicker database code deployment is a practical demand for any enterprise now a day. Database release automation [10] enhances data security. It assures better database code by eliminating errors that can cause application performance issues or downtime. Common mistakes that make databases more vulnerable to breaches and data theft are eliminated.

[10] https://www.datical.com/database-automation/what-is-database-release-automation/

8.4 Summary

The advantages of ETL automation include increased productivity, precision, and dependability, as well as increased customer satisfaction and scalable processes. The significance of automation is made more apparent as businesses develop and consumers want quicker processing of goods and services. As a result, process automation can help your organization grow while also ensuring its continuation. However, there are significant flaws in this proposal. Near-real time or on-demand data integration is normally not that excellent with this ETL processing approach. It performs admirably when working in batch mode.

In this Chapter, we have concentrated on automating the ETL process so that with minimum or no manual intervention, data can be loaded in DW, and analytics jobs can be done based on real-time data. The proposed approach supports automating data processing, including database release automation. Proof-of-concept has been performed on the Liquibase tool, which is used as database source control for managing database changes. A study has also been done on the machine learning-based automation process. Some area, including Data Pre-Processor, Custom rule engine, and reporting, has been identified where ML can be utilized. As a next step, work will be carried out to implement the other parts of the proposed approach. The study will be continued with machine learning approaches and the identification of other areas of the automated process where machine learning algorithms can be incorporated to automate the ETL process fully.

Chapter 9

ETL in Cloud

9.1 Integration of ETL in Cloud

Cloud computing is the most popular buzzword with the belief that it will give a new direction to the IT industry. Cloud comes with scalability and flexibility, which can be highly beneficial for data integration applications. Faster data movement with reduced cost and expanding capacity is the main objective of integrating the ETL process with the cloud environment.

As per IDC worldwide tracker [1] the year-wise investment of IT industry towards cloud deployment is represented in Figure 9.1. It is visible that in present days, most enterprises are going partially or fully toward cloud-based IT solutions. Furthermore, the trend is increasing rapidly. It can be beneficial to respect cost and time compared to deploying traditional batch mode ETL on-premises. The potential for speedy processing of increasing volume of data has semi/Unstructured type is the main plus point of a cloud-based ETL solution [228].

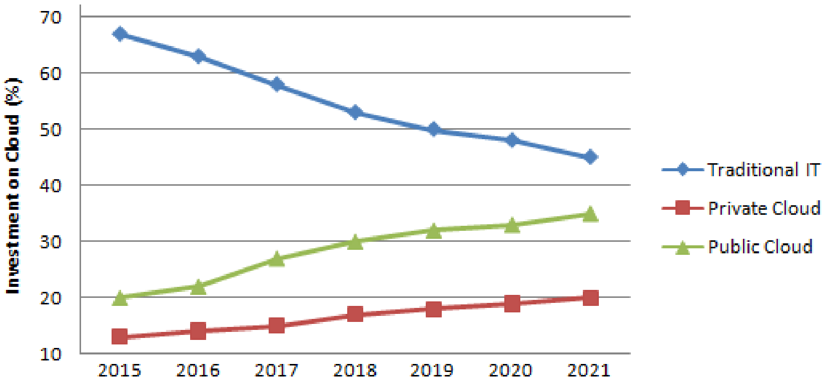


Figure 9.1: IDC Market forecast in cloud IT infrastructure

The Traditional Data warehouse (DW) use to store static data. In DW, strategic analysis is performed on Business data which is integrated from the heterogeneous data sources. The data is captured, aggregated, cleaned, and analyzed to derive better decisions. The overall set of processes for preparing data for future analysis is called ETL [281, 115]. Organizing and storing data in one place seems to be very simple, but in reality, it is a very complex process to establish an efficient ETL process [68]. In traditional batch processing ETL, DW refreshment is performed in an offline mode on a daily, weekly, or monthly basis.

In today's computing atmosphere, Big Data [227] placed at different locations having a diverse format and file sizes need to be processed in near real-time [117, 38]. It gives a great challenge to the data integration task. In the past, data could be managed by simply joining some manual scripts and adapters to manage different applications' data sets within any organization. But now, the source of data can be inside or even from the outside place, having the demand of manage, process, synchronize, and store in real-time [283]. Traditional ETL systems fail to meet all these requirements. Cloud service can address most of the previous issues, Not only for storing or getting data but also for managing complex integration tasks.

[1] https://www.idc.com/home.jsp

9.2 Current Requirements of ETL

What are the most demanded features from the ETL vendors nowadays? Here is a list of features identified which the organizations at present mostly require. These are some general characteristics that should be present in any ETL tool.

- The potential to communicate and collect data from all types of sources, databases (RDBMS, NoSQL), flat files, Big Data enabled technologies (Hadoop, Spark), sensors generated data, API, and more.
- Source to target data mapping through a GUI enable system providing drag and drop facility.
- GUI enabled data mapping and synchronization functionalities to retain consistent data on both sides with enabled workflow orchestration function.
- Capabilities to support team-based system build-up for collaborating work with release management and version control characteristics.
- Some basic transformation, data cleansing and data quality monitoring and integrated data profiling capabilities.
- Built in meta-data managing functionalities and documentation for different transformation and business rules.
- Efficient job scheduling and control over error handling, alerting and logging.
- Integrated data profiling capability which enables to inspect source data before starting of the ETL process.
- The most wanted feature is to integrate data in both in-site as well as on cloud. In the next section, we will briefly discuss about the characteristics of cloud enabled ETL tools.

9.2.1 Benefits of ETL in cloud

To establish an organization's cloud infrastructure, a variety of options are available now. A public cloud, private cloud, or a hybrid cloud can also be a choice as per their requirement. For cloud-based data interrelation deployment services, the commonly used option goes for Software as a Services (SaaS) or Infrastructure as a Service (IaaS). But when database, tool development, and middle-ware is concerned, the choice goes for Platform as a Service (PaaS) in maximum cases [2].

Figure 9.2 represents the basic architectural view of arranging IT service resources [263] in cloud. It gives the view of managing IT-related services among the consumers

[2] http://www.oracle.com/us/products/middleware/data-integration/ioug-di-for-cloud-survey-2596248.pdf

and providers. This architecture assures the quality of service (QoS) in terms of cloud services. QoS includes performance, throughput, accessibility, and proper utilization of service without concern about particular technical details of any tool. The architecture consists of three layers. The bottom layer collects information from cloud service resources (CSR). The middle layer manages QoS along with bottom layer information. The upper layer consists of IT service monitor tasks for CSR.

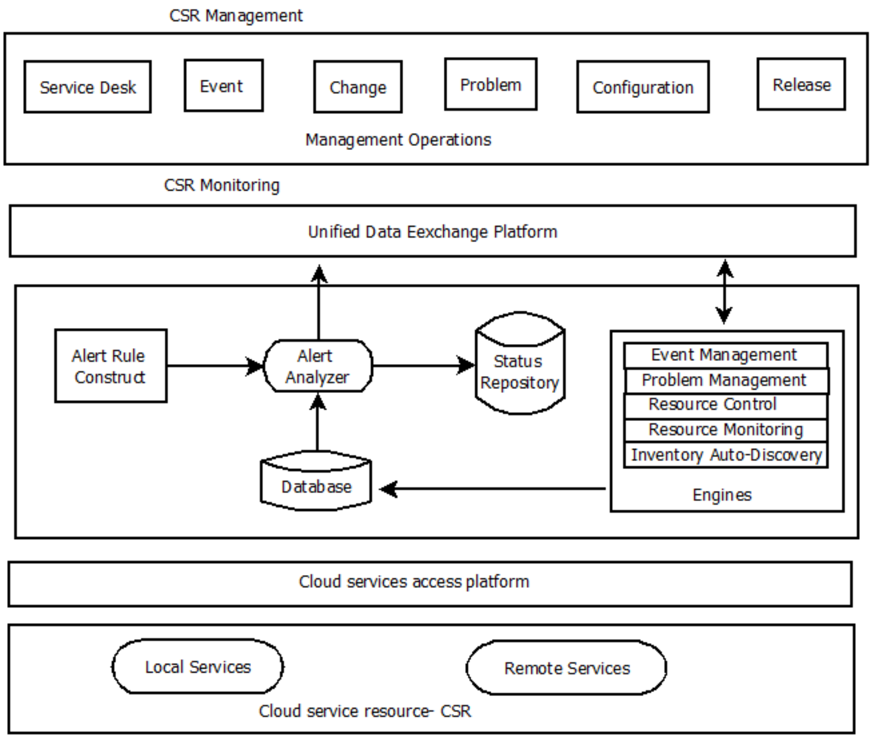


Figure 9.2: Architecture for managing cloud service resources

Some benefits of cloud ETL adoption are listed below.

Low Cost On-premise Data warehouse and ETL establishment and maintenance are quite expensive and time-consuming. On the other hand, in the cloud, you only need to pay for what you use. The total cost of resource overhead for an ETL system is minimized in the cloud. It also minimizes the maintenance and third-party service cost.

Scalability Deploying an ETL system in Cloud offers more scalability than operating it on a local site. Broad network access, and on-demand resource pooling are the key factors to meet increased performance.

MapReduce This technology has emerged as a key enabling technology to parallel processing massive data.

Elasticity Proving on-demand resource capacity is one of the most popular among the cloud's plus points. This capability is useful for businesses that have frequent demands of resource changes. When the demand is high, it will provide an adequate amount of resources. Again when the demand is low, the resource supply can shrink. It is a much more cost-effective provision.

Processing rate Choosing a cloud ETL tool can offer unlimited data processing capability. Cloud environments can support numerous data streams coming with various velocities. Cloud can take advantage of multiple servers to balance the data load at an increased processing speed.

Storage Extensive storage capacity is provided for cloud-based solution. The "pay-as-you-go" model is also a cost-saving option. The cloud data storage system promotes flexibility and agility. The data engineers can change the cluster size, RAM, and CPU as per the project requirement. Any new project can be deployed quickly without annoying the company's architecture or budget.

Flexibility As cloud-based ETL system does not depend on local resources. It offers more flexibility to its consumers providing access best equipment anytime and from anywhere.

Data recovery As data are kept in distributed locations; the cloud environment strongly provides data recoverability. It makes cloud ETL much more reliable compared to traditional systems.

9.2.2 Challenges of ETL in Cloud

Integration of cloud technology comes with many benefits as well as some additional challenges. To implement the mechanism for unidirectional movement of data between data stores and organizational applications without considering their location is a hard task [64]. Following is a list of challenges in cloud-based ETL solutions. Moving ETL to the cloud offers the host system to deploy the overall process at a decreasing cost but with increased performance. Integrating heterogeneous data into a homogeneous format for analysis at a given time is a challenging task. Another concern is that any BI platform should be interacting with these data can again reside on-site or in the cloud.

Security But when the cloud is coming to the scenario, one of the main challenges is security. It is an important question about how the relationship between the ETL tool and DW should be established. Should the data be kept in the cloud or on-premise? There can be a major risk of data confidentiality.

Data Recovery As the data is stored in the cloud, there can be a possibility of data violation. Dependency on the third party can be the reason for taking a long time and formalities in recovering data.

Data Transfer Moving huge data into the cloud in a particular time window can be a significant issue. Because it depends on the network capacity. If the storage and server resource are kept in a different place for security purposes. There can be latency in accessing data which will affect the data processing performance.

Interoperability Each cloud service providers have its own way of managing customers and services in its system. This leads to difficulty in selecting any vendors concurrently to optimize tasks at different levels for any organization. Many times it becomes very complex to migrate/connect existing systems with cloud API.

Selecting Vendor Sometimes, it becomes confusing to select a particular vendor as all cloud ETL vendors are offering some special features. All the facility offered by different vendors needs to be listed. All the features need to be verified and chosen which is suitable for any organization.

9.3 Migrating ETL workload in Cloud

Migration of ETL activity can be of two types - i) lift and shift and ii) re-architect. The first type is about relocating the overall work into the cloud without changing anything. This migration style will avail one of the cloud feature elasticity. The second type re-architecture, means designing the entire work scenario again to avail various cloud facilities like service registry, CloudIVS, etc. The factors we need to consider while migrating ETL workloads into the cloud are:

1. **Vendor selection** For ensuring multi-cloud foundation, vendor agnostic approach should be taken while selecting any vendor.
2. **Data migration approach** Data migration tool should follow an incremental data uploading approach enables on-premise system data to cloud storage data mapping efficiently.
3. **Application migration approach** For re-architecting the existing ETL process for cloud migration, some issues need to give extra attention. They are rapid prototyping, test ability, processing capacity, and upstream and downstream capacity.
4. **Workload management** While moving the ETL workload into the cloud, check out the security issues, overall monitoring process, and service-level agreement (SLA) details, and the master life cycle management.
5. **Structured design approach** It needs to follow a structured approach to import and manage the framework into the cloud. Points need to ensure our infrastructure portability, appropriate framework design, and implement suitable business logic.

9.4 Proposed Solution: Streaming ETL Framework

Studying the present technological requirement, we are proposing a new framework. The solution is unable to process, query, and analyze Big Data. Nowadays, a large volume of data is generated by new sources and existing systems that need to be processed and analyzed with the aim of many uncover insights. Among the characteristics of Big Data (Volume, Variety, Veracity, Value, Velocity), we are considering the huge volume of data that cannot be managed by any traditional data integration system. This solution will integrate a wide new variety of data also, like text data, graph data, etc., along with batch or real-time streaming types of source data. Moreover, it is suitable to process the overall workload in the cloud environment. Following Figure 9.3 presents the basic format of our proposed ETL framework.

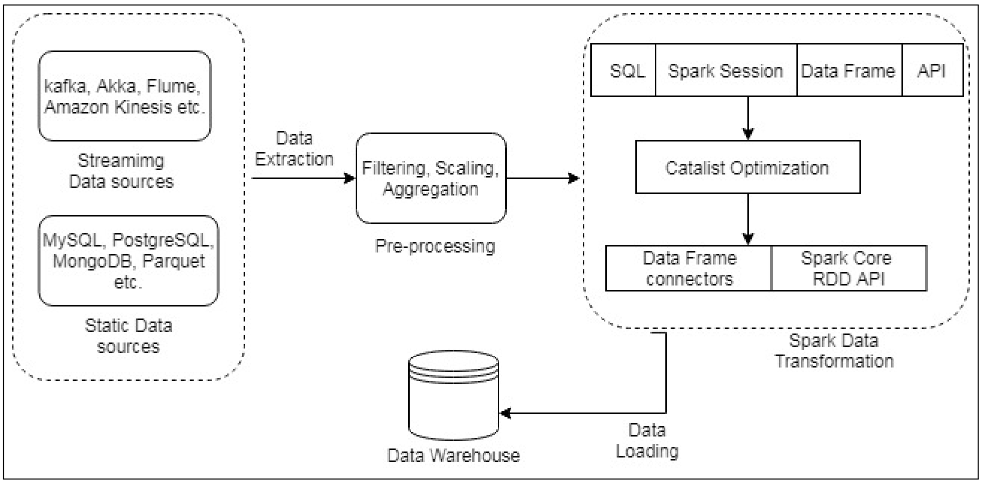


Figure 9.3: Spark Based Streaming ETL Framework

In our proposed solution, we are going to implement a robust ETL pipeline using Apache Spark. Apache Spark has API support which can convert various data formats into unified Data frames, and also SQL, which can be used for future analysis purposes. Spark is an open-source cluster-computing framework with having simple programming interface for handling data parallelism as well as the processing and querying of Big Data. It can process real-time data 100 times faster than its strong competitor MapReduce or other traditional data processing approaches. Writing Spark program solutions are very easy and can be done either in Python, R, and Scala.

Streaming Data Extraction: Many Big Data repositories can be integrated using Spark. Data sources having different formats will be extracted and converted into data frames. Spark is capable of handling streaming data as well as static data. Real-time data pipelines will extract data from sources as a chain of streaming events. Structured Streaming features provide a simple way to convert these traditional batch jobs into a real-time data pipeline. In this stage, some basic pre-processing tasks can be done in these stages.

Real-time Transformation: The next phase basically runs the data transformation task. Some custom reusable transformation functions can be defined, which will take the data frames as arguments and return it to transform the extracted file.

Continuous Data Loading In the loading phase, Spark Data Frame writers can be used to define the functions for writing the data frames in the target Data storage systems. This is the basic framework to promote ETL processes in the cloud environment. A real-time data pipeline will continuously move transformed data into the destination storage system.

A comparative analysis of some existing cloud-based ETL frameworks is presented in the Table 9.1. These are the existing solution for cloud-based ETL process. A list of pioneering solutions has been presented in the academic world to promote near real-time ETL in the cloud. In the academic world, a solution has been proposed by [121] to implement incremental data loading in the micro-batch approach. A Lazy ETL approach has been designed by [133] where only the required data are extracted and loaded for targeting low-cost data loading technique. Here ETL logic is integrated into the query processing layer.

For addressing the Big Data features, a novel solution has been proposed [155] by implementing ETL jobs in the Map-Reduce framework. A programming framework ETLMR has been built to achieve parallel process on large-scale data and supporting snowflake schema, star schema, and slowly changing dimensions (SCD) features. CloudETL [154] a cloud-enabled ETL framework that uses Hadoop to parallelize ETL jobs and process data in Hive. Here users can use a high level of constructs and various transformation features without bothering about MapReduce technical details. This framework support different dimensional concepts like star schemas, snowflake schemas, and SCD maintenance. A new distributed architecture of ETL named Striim [203] has been developed to support real-time data transformations over data streaming. Apache Kafka-based declarative transformation engine claim to handle streaming real-time data rapidly in the cloud.

Frameworks	**Principal Logic**	**Architecture**	**Special Feature**	**Supporting Property**
ELTMR [155]	MapReduce	Hadoop	Parallelize ETL processes	Different dimensional concepts
CloudETL [154]	MapReduce	Hadoop, Hive	Scalable data processing	Different dimensional concepts
Striim [203]	Real-time CDC	Hadoop, NoSQL and Kafka	In memory transformations	Real-time data integration

Table 9.1: Comparative analysis of different cloud based ETL Framework

9.5 Summary

This study covers the limitation of traditional data integration systems regarding current demands towards the ETL vendors. 90% data generated in today's digital evolution is semi-structured or unstructured type. ETL process should handle this type of data. To cope with the requirements of the present world, how the cloud can act as a key resolution technique is analyzed. Cloud-based ETL solution architecture will represent the deployment environment. This study reveals that still, for many organizations moving confidential data to the cloud is the main concern due to privacy issues. It is observed that currently, some side-line functionality like application programs or IT management systems is moved frequently in the cloud. Core activities are kept within the organizational access. As a result, IaaS is more accepted compared to SaaS for most organizations.

This chapter covers the demanding features of ETL tools nowadays. A brief discussion has been done about the limitations of traditional ETL tools. In the current technological scenario, Cloud has come to overcome the above challenges. Some basic benefits and challenges to going for the cloud-based ETL solution are highlighted in this write-up. A comparative review can give a deep insight into their promising characteristics. Finally, a new Spark-based ETL framework has been developed, which will efficiently handle real-time data stream in the cloud platform. Hope that this work will present a visible scenario of a cloud-based ETL solution. In the future, we are planning to integrate real-time machine learning with the proposed ETL framework for predictive analytics of streaming data.

Big data processing can be done using the suggested approach. It has several advantages that make it an effective choice for ETL processing. It is appropriate for ETL processing operations involving large datasets because it is designed to quickly process massive amounts of data. It is straightforward to use for ETL workloads because of its intuitive process. It is suitable for data integration jobs that may require the processing of very large datasets since it can be quickly scaled to handle large data volumes. It is a flexible option for ETL processing because it is simple to combine with other big data tools and platforms, such as Hadoop and Amazon Web Services (AWS). Streaming ETL is an effective framework for processing and analyzing massive amounts of data in near real-time, but it might not be the ideal choice for all streaming use cases.

Practically, real-time data integration is not literally instantaneous. Because it takes at least a fraction of a second at each stage of data collection, transfer, transformation, migrating, and uploading task. So the idea for real-time ETL is to process and transfer the data as quickly as it is collected at the source side. Adopting Spark-based Streaming analytics frameworks can be very useful for ETL processing in near real-time.

Chapter 10

Conclusion And Future Work

In this dissertation, we have worked towards the modeling, simulation, and empirical analysis of traditional and real-time ETL processes and advanced proposal of ETL workflows management by use of machine learning and shifting the ETL workload in the cloud environment. In this chapter, I am presenting the conclusion of the report. A summarized report on each proposal is presented with their corresponding future scope.

10.1 Conceptual model

The ETL process is in charge of choosing and extracting data from various sources, cleaning it, transforming it into the desired format, and then updating the DW. Data orientation and their interaction during the ETL processing activity can be designed using ETL process modeling.

At first, we proposed a novel model of ETL at a conceptual level [41]. The conceptual model is the foundation of any ETL process. Previously all of the ETL conceptual models were mainly developed by UML, BPMN, and Semantic web-based approaches. In our proposal, we are using MBSE oriented system model for designing the ETL process. We have selected a new modeling language, SysML, suitable for system engineering applications. By using SysML, the system model can be designed in a more expressive as well as flexible way. UML language has a limitation of having a software-centric point of view. SysML is an extension of the UML language, offering some noteworthy enhancements over UML.

An example of an E-commerce system has been evaluated for designing the ETL model. Data Warehouse schema for E-commerce system gives the basic idea about the E-commerce system. At first, the requirement diagram notation and the activity diagram notation of the SysML language are briefly discussed. After that, these two SysML diagrams are designed as well as explained at a conceptual level. Propagation of data from the source system to the destination DW is visible from the diagrams.

The use of SysML in conceptual ETL modeling is the very first approach in this domain. Hope that this proposal will enhance the modeling of a broad range of ETL systems in addition to its software, hardware, processes, personnel, information, and facilities. Our designed paradigm is naturally platform-independent and easy for both technical and non-technical users to comprehend. After that, the SysML model is converted into the relevant executable code. To more accurately analyse system behaviour and requirements in the future and to broaden the model's perspective at the logical and physical levels, we plan to simulate the suggested model.

10.2 Model Simulation

SysML language is used for modeling complex systems in a standard way. Therefore, an enormous curiosity is there for creating the simulation models form the designed SysML model. Simulation is a well-accepted method for the validation of any system. There are many approaches from both research and industrial communities for system model simulation.

In this extended work, we intend to simulate our proposed ETL model that was designed by SysML language [40]. Successful execution of simulation should have a number of potentials. Two main features are a designed conceptual model to be simulated and translation of the model into a simulation program or computational process. In this work, we have described how OMG supported new graphical systems modeling language, SysML can be evaluated to form a platform-independent model (PIM) at a conceptual level, and after that, how this model can be translated to a simulation program. For demonstrating the process, we have used MagicDraw, which is a visual modeling tool for simulation purposes. The ETL activity model is validated using this modeling tool provided simulation environment.

In this work, we have proposed an MBSE-oriented system model for the ETL process for the Data warehouse environment. MBSE is the standard practice of system modeling techniques for supporting system requirements, design, verification, analysis, validation, and finally, documentation activities. This proposal is focused on creating an automated executable SysML modeling proposal based on an activity diagram. For this purpose, a SysML execution engine is adapted. The proposed method is demonstrated by taking a running example of an E-commerce website case study.

In future work, there is a scope to design the SysML Parametric diagrams of the ETL processing system. It will explore the dynamic nature of execution of mathematical models designed in the SysML parametric model. There is another scope for a code-based simulation approach for any ETL system.

10.3 Code based ETL tools

In the current technology era, most of data-centric applications demand real-time data processing capability with an increasing amount of data handing methodology in a more complex environment. It is coming with great challenges in the research domain also. There are numerous ETL tools available in the market. Most of the vendor-made ETL tools are commercial, licensed tools or a few open-source, free tools. Generally, most organizations want to go for taking any vendor-made GUI-based product for their ETL solution. But still, in some cases, custom-coded ETL can be the best option in respect of performance optimization and flexibility.

This work has chosen the option for a code-based ETL solution [39]. There is no other work found for proving a survey or comparative evaluation over code-based ETL approaches. This is the uniqueness of the work. For this proposed work, four reputed code-based ETL tools Pygrametl, Petl, Scriptella, and R_etl have been chosen. Their overall characteristics have been studied as well as a benchmarking is done. After that, an in detail experimental and feature analysis is presented on these selected tools.

This proposal presents a fresh summary and evaluation of current research developments in the area of programmable ETL development techniques. Numerous organizations still desire to develop their own data pipeline infrastructure. A wide range of technical abilities are needed for this strategy. This work's primary goal is not to recommend a certain tool as being good or poor. It depends on the specific needs and expertise of any organization, including its capacity to scale, manage costs, support infrastructure, maintain confidentiality, and more. I hope that this piece of work will contribute to developing a rich perspective on the ETL process. We can compare further code-based ETL technologies with our current efforts in future development.

10.4 Real-time ETL

Currently, a major difficulty in the research sector is the demand of the constant and increasing volume of data handling within a more complicated context. It requires a standardized ETL procedure, which has a significant economic impact on the BI sector. We have here presented an ETL framework with a real-time data integration facility, continuing the code-based ETL approach [38]. Our primary goal is to reduce data latency. We are choosing the incremental loading approach out of the two available loading strategies for data warehouse upgrades. The CDC approach is linked to incremental loading. Only newly altered data from the source side will be extracted and propagated into the data warehouse. The demands for real-time ETL are the primary goal of this work.

We argued that incremental loading is more efficient than full reloading unless the operational data sources happen to change dramatically. That's why incremental loading is generally preferable. This chapter's main contribution is modeling a near real-time ETL process with the persuasion of incremental loading. However, the development of ETL jobs for incremental loading is not at all well-supported by existing ETL tools. In fact, separate ETL jobs for initial loading and incremental loading have been created by ETL programmers so far. So incremental load jobs are considerably more complex and error-prone. However, this approach proves much better compared to the former one.

The future work aims to design a unified code-based ETL framework supporting relational and NoSQL database with a rich transformation library hosted on the Cloud platform. Also, we will focus on advanced transformation operators such as aggregation, outer joins, and data restructuring such as pivoting. Further, we have a plan to utilize the usage of the staging area in a better way. We can allow ETL jobs to persist data in the staging area, serving as additional input for subsequent runs. CDC limitations can be reduced to some extent by utilizing the staging area. We can use partially changed data by utilizing the staging area. Moreover, we expect performance improvements from persisting intermediary results. In this chapter, we have concentrated on automating the ETL process so that with minimum or no manual intervention, data can be loaded in DW, and analytics jobs can be done based on real-time data . The proposed approach supports automating data processing, including database release automation. Proof-of-concept has been performed on the Liquibase tool, which is used as database source control for managing database changes. A study has also been done on the machine learning-based automation process. Some area, including Data Pre-Processor, Custom rule engine, reporting, has been identified where ML can be utilized. As a next step, work will be carried out to implement the other parts of the proposed approach. The study will be continued with machine learning approaches and identification of other areas of the automated process where machine learning algorithms can be incorporated to automate the ETL process fully.

10.5 ETL Automation

ETL process frequently call for vast amounts of data to be taken from diverse sources, formatted appropriately, and loaded into a destination system. Manually carrying out these processes can be time-consuming, prone to error, and ineffective. Automation enables the automation of complicated and repetitive activities, leading to a faster and more precise data integration.

In this work, we have focused on automating the ETL process so that data can be put into DWs with little to no manual assistance and analytical tasks can be carried out using real-time data [172]. The suggested strategy facilitates automating data processing, which includes automated database release. The Liquibase tool, which serves as database source control for managing database updates, has undergone proof-of-concept testing. The automation technique based on machine learning has also been studied. There are other areas where ML can be used, such as data pre-processing, custom rule engines, and reporting. Work will then be done to put the other components of the suggested method into practise.

10.6 Integration of ETL in Cloud

This work is mainly concerned with finding out the limitation of traditional ETL processing systems regarding the current demands of the ETL vendors. Major amount of data generated in today's digital evolution is semi-structured or unstructured type. It is a challenging task to manage the growing speed of complex type of data volume. Cloud can act as a critical resolution technique to cope with the requirements of the present world. Many organizations have started to move into Cloud for their ETL processing task to get the advantage of robust and automated ETL pipelines, which can be deployed within a few minutes.

In the current technological scenario, Cloud has the capability to overcome the above challenges. Some basic benefits and challenges to going for the cloud-based ETL solution are highlighted in this write-up. A comparative analysis is presented about the existing research progress on taking ETL in the Cloud. Finally, a new Spark-based ETL framework has been developed, which will efficiently handle real-time data stream in the cloud platform [37]. We are taking the advantage of Apache Spark, which is suitable for large-scale data processing the task in a distributed framework. It has a huge potential to process big data coming from real-time applications. Hope that this work will present a notable solution for the cloud-based ETL processing approach. In the future, we are planning to integrate real-time Machine Learning with the proposed ETL framework for predictive analytics of streaming data.

Bibliography

[1] T Abdul-Aziz, I Moawad, and W. M. Abu-Alam. “Decision Support System Utilizing Data Warehouse Technique for the Tourism Sector in Egypt”. In: *The 7th International Conference on Information Technology.* 2015.

[2] C. C. Aggarwal. *Data classification: algorithms and applications.* CRC press, 2014.

[3] N. Agnihotri and A.K. Sharma. “Big data analysis and its need for effective E-governance”. In: *International Journal of Innovations & Advancement in Computer Science* 4 (2015), pp. 219–224.

[4] T. Ahmed, T. Calders, and T.B. Pedersen. “Mining risk factors in RFID baggage tracking data”. In: *Mobile Data Management (MDM), 2015 16th IEEE International Conference on.* Vol. 1. IEEE. 2015, pp. 235–242.

[5] T. Ahmed, T.B. Pedersen, and H. Lu. “A data warehouse solution for analyzing RFID-based baggage tracking data”. In: *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on.* Vol. 1. IEEE. 2013, pp. 283–292.

[6] E. El Akkaoui and E. Zimányi. “Defining ETL worfklows using BPMN and BPEL”. In: *Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP.* ACM. 2009, pp. 41–48.

[7] Z. El Akkaoui et al. “A BPMN-based design and maintenance framework for ETL processes”. In: (2013).

[8] Z. El Akkaoui et al. “A model-driven framework for ETL process development”. In: *Proceedings of the 14th international workshop on Data Warehousing and OLAP.* ACM, 2011, pp. 45–52.

[9] Z. El Akkaoui et al. *BPMN-based conceptual modeling of ETL processes.* Springer, 2012, pp. 1–14.

[10] J. Almeida and J. Ferreira. “BUS public transportation system fuel efficiency patterns”. In: *2nd International Conference on Machine Learning and Computer Science (IMLCS’2013).* 2013.

[11] S. Amer-Yahia and S. Cluet. “A declarative approach to optimize bulk loading into databases”. In: *ACM Transactions on Database Systems (TODS)* 29.2 (2004), pp. 233–281.

[12] N. Anand and M. Kumar. “An Overview on Data Quality Issues at Data Staging ETL”. In: (2013).

[13] I. Ankorion. “Change Data Capture Efficient ETL for Real-Time BI”. In: *Information Management* 15.1 (2005), p. 36.

[14] T. Anton. “XPath-Wrapper Induction by generalizing tree traversal patterns”. In: *Lernen, Wissensentdeckung und Adaptivitt (LWA) 2005, GI Workshops, Saarbrcken.* 2005, pp. 126–133.

[15] L. Arge et al. “Efficient bulk operations on dynamic R-trees”. In: *Workshop on Algorithm Engineering and Experimentation*. Springer. 1999, pp. 322–341.

[16] W. Astriani and R. Trisminingsih. “Extraction, transformation, and loading (ETL) module for hotspot spatial data warehouse using Geokettle”. In: *Procedia Environmental Sciences*. Vol. 33. Elsevier, 2016, pp. 626–634.

[17] F. Atigui et al. “A Unified Model Driven Methodology for Data Warehouses and ETL Design.” In: *ICEIS (1)*. 2011, pp. 247–252.

[18] S. Ayhan et al. “Predictive analytics with aviation big data”. In: *Conference on Integrated Communications, Navigation and Surveillance (ICNS’13)*. 2013, pp. 1–13.

[19] D P. Ballou and G K. Tayi. “Enhancing data quality in data warehouse environments”. In: *Communications of the ACM* 42.1 (1999), pp. 73–78.

[20] J. Barateiro and H. Galhardas. “A Survey of Data Quality Tools.” In: *Datenbank-Spektrum* 14.15-21 (2005), p. 48.

[21] O. Batarseh and L. McGinnis. “System modeling in sysml and system analysis in arena”. In: *Proceedings of the Winter Simulation Conference*. Winter Simulation Conference. 2012, p. 258.

[22] C. Batini, M. Lenzerini, and S. B. Navathe. “A comparative analysis of methodologies for database schema integration”. In: *ACM computing surveys* 18.4 (1986), pp. 323–364.

[23] B. Baumer. “A Grammar for Reproducible and Painless Extract-Transform-Load Operations on Medium Data”. In: *arXiv preprint arXiv:1708.07073* (2017).

[24] R. Baumgartner, S. Flesca, and G. Gottlob. “Visual web information extraction with lixto”. In: *VLDB*. Vol. 1. 2001, pp. 119–128.

[25] R. Baumgartner, W. Gatterbauer, and G. Gottlob. “Web data extraction system”. In: *Encyclopedia of Database Systems*. Springer, 2009, pp. 3465–3471.

[26] R. Baumgartner et al. *Web data extraction for business intelligence: the lixto approach*. Citeseer, 2005.

[27] W. Behrmann and M. Räkers. “Specifics of Financial Data Warehousing and Implications for Management of Complex ISD Projects.” In: *16th European Conference on Information Systems, ECIS 2008*. 2008, pp. 1740–1751.

[28] O. Belo et al. “Automatic Generation of ETL Physical Systems from BPMN Conceptual Models”. In: *Model and Data Engineering*. Springer, 2015, pp. 239–247.

[29] J. Bercken and B. Seeger. “An evaluation of generic bulk loading techniques”. In: *Proc. of VLDB*. 2001, pp. 461–470.

[30] S. Bergamaschi, F. Guerra, and M. Vincini. “A data integration framework for e-commerce product classification”. In: *International Semantic Web Conference*. Springer. 2002, pp. 379–393.

[31] A. L Berger, V. J D. Pietra, and S. A D. Pietra. "A maximum entropy approach to natural language processing". In: *Computational linguistics* 22.1 (1996), pp. 39–71.

[32] N. Berkani, L. Bellatreche, and C. Ordonez. "ETL-aware materialized view selection in semantic data stream warehouses". In: *2018 12th International Conference on Research Challenges in Information Science (RCIS)*. IEEE. 2018, pp. 1–11.

[33] E. Bernier et al. "Easier surveillance of climate-related health vulnerabilities through a Web-based spatial OLAP application". In: *International Journal of Health Geographics*. Vol. 8. 1. BioMed Central, 2009, p. 18.

[34] *Best Practices for Real-time Data Warehousing*. White Paper. Oracle, 2012.

[35] M. Bilenko and R. J. Mooney. "Adaptive Duplicate Detection Using Learnable String Similarity Measures". In: *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*. ACM, 2003, pp. 39–48. ISBN: 1-58113-737-0.

[36] A. Bilke and F. Naumann. "Schema matching using duplicates". In: *21st International Conference on Data Engineering (ICDE'05)*. IEEE. 2005, pp. 69–80.

[37] N. Biswas and K. C. Mondal. "Integration of ETL in Cloud Using Spark for Streaming Data". In: *International Conference on Emerging Applications of Information Technology*. Springer. 2021, pp. 172–182.

[38] N. Biswas, A. Sarkar, and K. C. Mondal. "Efficient incremental loading in ETL processing for real-time data integration". In: *Innovations in Systems and Software Engineering* 16.1 (2020), pp. 53–61.

[39] N. Biswas, A. Sarkar, and K. C. Mondal. "Empirical Analysis of Programmable ETL Tools". In: *International Conference on Computational Intelligence, Communications, and Business Analytics*. Springer. 2018, pp. 267–277.

[40] N. Biswas et al. "A New Approach for Conceptual Extraction-Transformation-Loading Process Modeling". In: *International Journal of Ambient Computing and Intelligence (IJACI)* 10.1 (2019), pp. 30–45.

[41] N. Biswas et al. "SysML Based Conceptual ETL Process Modeling". In: *International Conference on Computational Intelligence, Communications, and Business Analytics*. Springer. 2017, pp. 242–255.

[42] J. A. Blakeley, P. Larson, and F. W. Tompa. "Efficiently updating materialized views". In: *ACM SIGMOD Record*. Vol. 15. 2. ACM. 1986, pp. 61–71.

[43] R. Bliujute et al. *Systematic change management in dimensional data warehousing*. Tech. rep. Time Center Technical Report TR-23, 1998.

[44] C. Böhm and H. Kriegel. "Efficient bulk loading of large high-dimensional indexes". In: *International Conference on Data Warehousing and Knowledge Discovery*. Springer. 1999, pp. 251–260.

[45] M. B. Bokade, S. S. Dhande, and H. R. Vyavahare. “Framework Of Change Data Capture And Real Time Data Warehouse”. In: *International Journal of Engineering Research and Technology*. Vol. 2. 4. ESRSA Publications. 2013.

[46] A.R. Bologa, R. Bologa, A. Florea, et al. “Big data and specific analysis methods for insurance fraud detection”. In: *Database Systems Journal*. Vol. 4. 4. Academy of Economic Studies-Bucharest, Romania, 2013, pp. 30–39.

[47] D. Calvanese et al. “Information integration: Conceptual modeling and reasoning support”. In: *In Proceedings 3rd International Conference on Cooperative Information Systems (IFCIS’98)*. IEEE. 1998, pp. 280–289.

[48] P. Carreira et al. “Data mapper: an operator for expressing one-to-many data transformations”. In: *International Conference on Data Warehousing and Knowledge Discovery*. Springer. 2005, pp. 136–145.

[49] P. Carreira et al. “On the performance of one-to-many data transformations.” In: *QDB*. 2007, pp. 39–48.

[50] P. Carreira et al. “ONE-TO-MANY DATA TRANSFORMATION OPERATIONS”. In: (2007), pp. 21–27.

[51] P. Carreira et al. “One-to-many data transformations through data mappers”. In: *Data & Knowledge Engineering* 62.3 (2007), pp. 483–503.

[52] M. Castellanos et al. “Automating the loading of business process data warehouses”. In: *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. ACM. 2009, pp. 612–623.

[53] S. Chaudhuri, V. Ganti, and R. Motwani. “Robust identification of fuzzy duplicates”. In: *21st International Conference on Data Engineering (ICDE’05)*. IEEE. 2005, pp. 865–876.

[54] L. Chen, W. Rahayu, and D. Taniar. “Towards Near Real-Time Data Warehousing”. In: *24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*. 2010, pp. 1150–1157.

[55] N. Choudhary. “A Study over Problems and Approaches of Data Cleansing/Cleaning”. In: *International Journal of Advanced Research in Computer Science and Software Engineering ISSN* 2277 (2014).

[56] S. Cluet and G. Moerkotte. “Classification And Optimization of Nested Queries in Object Bases.” In: *In Bases de Donnes*. 1994, 331–349.

[57] S. S. Conn. “OLTP and OLAP data integration: a review of feasible implementation methods and architectures for real time data analysis”. In: *In Proceedingsof IEEE SoutheastCon*. IEEE. 2005, pp. 515–520.

[58] V. Crescenzi, G. Mecca, P. Merialdo, et al. “Roadrunner: Towards automatic data extraction from large web sites”. In: *VLDB*. Vol. 1. 2001, pp. 109–118.

[59] A. Cuzzocrea, N. Ferreira, and P. Furtado. “Real-Time Data Warehousing: A Rewrite/Merge Approach”. In: *Data Warehousing and Knowledge Discovery*. Springer, 2014, pp. 78–88.

[60] Datawarehouse4u. *Slowly Changing Dimensions*. http://datawarehouse4u.info/SCD-Slowly-Changing-Dimensions.html.

[61] U. Dayal et al. “Data integration flows for business intelligence”. In: *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. Acm. 2009, pp. 1–11. ISBN: 978-1-60558-422-5.

[62] U. Dayal et al. “Optimization of analytic data flows for next generation business intelligence applications”. In: *Topics in Performance Evaluation, Measurement and Characterization*. Springer, 2011, pp. 46–66.

[63] P. Desai and A. Desai. “The Study on Data Warehouse and Data Mining for Birth Registration System of the Surat City”. In: *IJCA Proceedings on International Conference on Technology Systems and Management (ICTSM)*. 4. 2011, pp. 1–5.

[64] Tharam Dillon, Chen Wu, and Elizabeth Chang. “Cloud computing: issues and challenges”. In: *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*. IEEE. 2010, pp. 27–33.

[65] D. Dzemydiene and R. Dzindzalieta. “Development of architecture of embedded decision support systems for risk evaluation of transportation of dangerous goods”. In: *Technological and Economic Development of Economy* 16.4 (2010), pp. 654–671.

[66] Z. Ebrahim and Z. Irani. “E-government adoption: architecture and barriers”. In: *Business process management journal* 11.5 (2005), pp. 589–611.

[67] M. J. Eccles, D. J. Evans, and A. J. Beaumont. “True Real-Time Change Data Capture with Web Service Database Encapsulation”. In: *Services (SERVICES-1), 2010 6th World Congress on*. IEEE. 2010, pp. 128–131.

[68] W. Eckerson and C. White. “Evaluating ETL and data integration platforms”. In: *Report of The Data Warehousing Institute* 184 (2003).

[69] W. W. Eckerson. “Data quality and the bottom line: Achieving business success through a commitment to high quality data”. In: *The Data Warehousing Institute* (2002), pp. 1–36.

[70] *Efficient and Real Time Data Integration with Change Data Capture*. White Paper. http://attunity.com. Attunity Ltd., 2009.

[71] S. H. A. El-Sappagh, A. M. A. Hendawi, and A. H. E. Bastawissy. “A proposed model for data warehouse ETL processes”. In: *Journal of King Saud University – Computer and Information Sciences* 23 (2011), 91–104.

[72] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. “Duplicate record detection: A survey”. In: *IEEE Transactions on Knowledge and Data Engineering* 19.1 (2007), pp. 1–16.

[73] D. W Embley et al. "Conceptual-model-based data extraction from multiple-record Web pages". In: *Data & Knowledge Engineering* 31.3 (1999), pp. 227–251.

[74] M.M. Eshow, M. Lui, and S. Ranjan. "Architecture and capabilities of a data warehouse for ATM research". In: *Digital Avionics Systems Conference (DASC), 2014 IEEE/AIAA 33rd.* IEEE. 2014, 1E3–1.

[75] J. A. Estefan. "Survey of model-based systems engineering (MBSE) methodologies". In: *Incose MBSE Focus Group* 25.8 (2007).

[76] S. Faisal and M. Sarwar. "Handling slowly changing dimensions in data warehouses". In: *Journal of Systems and Software* 94 (2014), pp. 151–160.

[77] I. Fellegi and A. Sunter. "A theory for record linkage". In: *Journal of the American Statistical Association* 64.328 (1969), pp. 1183–1210.

[78] R. Fenk et al. "Bulk loading a Data Warehouse built upon a UB-Tree". In: *International Symposium on Database Engineering and Applications.* IEEE. 2000, pp. 179–187.

[79] E. Ferrara et al. "Web data extraction, applications and techniques: A survey". In: *Knowledge-Based Systems* 70 (2014), pp. 301–323.

[80] P. Figueiras et al. "User interface support for a big ETL data processing pipeline an application scenario on highway toll charging models". In: *Engineering, Technology and Innovation (ICE/ITMC), 2017 International Conference on.* IEEE. 2017, pp. 1437–1444.

[81] M. Fischer, M. Pinzger, and H. Gall. "Populating a release history database from version control and bug tracking systems". In: *International Conference on Software Maintenance, 2003. ICSM 2003. Proceedings.* IEEE. 2003, pp. 23–32.

[82] G. Fiumara. "Automated information extraction from web sources: a survey". In: *Proc. of Between Ontologies and Folksonomies Workshop.* 2007, pp. 1–9.

[83] E. Franconi and A. Kamblet. "A data warehouse conceptual data model". In: *Proceedings of 16th International Conference on Scientific and Statistical Database Management.* 2004, pp. 435–436.

[84] D. Freitag. "Machine learning for information extraction in informal domains". In: *Machine learning* 39.2 (2000), pp. 169–202.

[85] S. Friedenthal, A. Moore, and R. Steiner. *A practical guide to SysML: the systems modeling language.* Morgan Kaufmann, 2014.

[86] S. Friedenthal, A. Moore, and R. Steiner. "OMG Systems Modeling Language OMG SysML™ Tutorial". In: *INCOSE International Symposium.* Vol. 18. 1. Wiley Online Library. 2008, pp. 1731–1862.

[87] R. R Friedlander, R. A Hennessy, and J. R Kraemer. *System and method for a multiple disciplinary normalization of source for metadata integration with ETL processing layer of complex data across multiple claim engine sources in support of the creation of universal/enterprise healthcare claims record.* US Patent 7,788,213. 2010.

[88] R. R Friedlander, R. A Hennessy, and J. R Kraemer. *System and method for semantic normalization of source for metadata integration with etl processing layer of complex data across multiple data sources particularly for clinical research and applicable to other domains*. US Patent App. 11/760,636. 2008.

[89] M. Fuchs, W. Höpken, and M. Lexhagen. “Big data analytics for knowledge generation in tourism destinations–A case from Sweden”. In: *Journal of Destination Marketing & Management* 3.4 (2014), pp. 198–209.

[90] H. Galhardas et al. “AJAX: an extensible data cleaning tool”. In: *ACM Sigmod Record*. Vol. 29. 2. 2000, p. 590.

[91] H. Galhardas et al. “Declaratively cleaning your data using AJAX”. In: *In Journees Bases de Donnees*. 2000.

[92] E. Gallinucci, M. Golfarelli, and S. Rizzi. “Meta-stars: multidimensional modeling for social business intelligence”. In: *Proceedings of the sixteenth international workshop on Data warehousing and OLAP*. ACM. 2013, pp. 11–18.

[93] S. Geisler. “Data stream management systems”. In: *Dagstuhl Follow-Ups* 5 (2013), pp. 275–304.

[94] R. Gill and J. Singh. “A Review of Contemporary Data Quality Issues in Data Warehouse ETL Environment”. In: (2014).

[95] M. Golfarelli, D. Maio, and S. Rizzi. “The dimensional fact model: A conceptual model for data warehouses”. In: *International Journal of Cooperative Information Systems* 7.02n03 (1998), pp. 215–247.

[96] M. S. Gouider and A. Farhat. “Building a Data Warehouse for National Social Security Fund of the Republic of Tunisia”. In: *Computing Research Repository (CoRR)*. Vol. abs/1006.0876. 2010. arXiv: 1006.0876. URL: http://arxiv.org/abs/1006.0876.

[97] W. Grossmann, M. Hudec, and R. Kurzawa. “Web usage mining in e-commerce”. In: *International journal of electronic business* 2.5 (2004), pp. 480–492.

[98] G. Guerreiro et al. “An architecture for big data processing on intelligent transportation systems. An application scenario on highway traffic flows”. In: *Intelligent Systems (IS), 2016 IEEE 8th International Conference on*. IEEE. 2016, pp. 65–72.

[99] N. Guerreiro and O. Belo. “Predicting the Performance of a GRID Environment: An Initial Effort to Increase Scheduling Efficiency”. In: *Future Generation Information Technology*. Springer, 2009, pp. 112–119.

[100] A. Gupta, H. V. Jagadish, and I. S. Mumick. “Data integration using self-maintainable views”. In: *International Conference on Extending Database Technology*. Springer. 1996, pp. 140–144.

[101] A. Gupta, I. S. Mumick, et al. “Maintenance of materialized views: Problems, techniques, and applications”. In: *IEEE Data Engineering Bulletine* 18.2 (1995), pp. 3–18.

[102] Keen Hahn. “Industry Case Study: Modernizing the Data Warehouse for Finance IT”. In: (2019).

[103] A. Halevy, A. Rajaraman, and J. Ordille. “Data Integration: The Teenage Years”. In: *Proceedings of the 32rd International Conference on Very Large Data Bases (VLDB’06).* VLDB Endowment, 2006, pp. 9–16.

[104] H. Han. *Conceptual modeling and ontology extraction for web information.* The University of Texas at Arlington, 2002.

[105] Laura E. Hart. *Introduction To Model-Based System Engineering (MBSE) and SysML.* http://www.incose.org/docs/default-source/delaware-valley/mbse-overview-incose-30-july-2015.pdf. July 30, 2015.

[106] M. Hause. “The sysml modelling language”. In: *15th European Systems Engineering Conference.* Vol. 9. 2006.

[107] M. Helfert, G. Zellner, and C. Sousa. “Data quality problems and proactive data quality management in data-warehouse-systems”. In: *Proceedings of BITWorld* (2002).

[108] A Hendawi and H El-Shishny. “Data Warehouse Prototype for the Tourism Industry: A Case Study from Egypt”. In: *International Conference on Informatics and Systems.* 2008.

[109] M. A. Hernández and S. J. Stolfo. “Real-world data is dirty: Data cleansing and the merge/purge problem”. In: *Data mining and knowledge discovery* 2.1 (1998), pp. 9–37.

[110] A.D.T. Hoang and B. T. Nguyen. “An integrated use of CWM and ontological modeling approaches towards ETL processes”. In: *2008 IEEE International Conference on e-Business Engineering.* IEEE. 2008, pp. 715–720.

[111] W. Höpken et al. “Multi-dimensional data modelling for a tourism destination data warehouse”. In: *Information and communication technologies in tourism 2013.* Springer, 2013, pp. 157–169.

[112] C. Hsu and M. Dung. “Generating finite-state transducers for semi-structured data extraction from the web”. In: *Information systems* 23.8 (1998), pp. 521–538.

[113] D. Huang, M. Du Y.and Zhang, and C. Zhang. “Application of ontology-based automatic ETL in marine data integration”. In: *Electrical & Electronics Engineering (EEESYM), 2012 IEEE Symposium on.* IEEE. 2012, pp. 11–13.

[114] B. Hüsemann, J. Lechtenbörger, and G. Vossen. *Conceptual data warehouse design.* Universität Münster. Angewandte Mathematik und Informatik, 2000.

[115] W. Inmon. *Building the data warehouse.* John wiley & sons, 2005.

[116] R. J. and J. Bernardino. “Real-time data warehouse loading methodology”. In: *Proceedings of the 2008 international symposium on Database engineering & applications.* ACM. 2008, pp. 49–58.

[117] T. Jain, R. S, and S. Saluja. "Refreshing datawarehouse in near real-time". In: *International Journal of Computer Applications* 46.18 (2012).

[118] C. Jermaine, A. Datta, and E. Omiecinski. "A novel index supporting high volume data warehouse insertion". In: *VLDB*. Vol. 99. 1999, pp. 235–246.

[119] T. Jörg and S. Dessloch. "Formalizing ETL Jobs for Incremental Loading of Data Warehouses." In: *BTW*. 2009, pp. 327–346.

[120] T. Jörg and S. Dessloch. "Near real-time data warehousing using state-of-the-art ETL tools". In: *Enabling Real-Time Business Intelligence*. Springer, 2009, pp. 100–117.

[121] T. Jörg and S. Dessloch. "Towards Generating ETL Processes for Incremental Loading". In: *Proceedings of the 2008 International Symposium on Database Engineering Applications (IDEAS'08)*. ACM, 2008, pp. 101–110. ISBN: 978-1-60558-188-0.

[122] T. Jörg and S. Dessloch. "View maintenance using partial deltas". In: *Datenbanksysteme für Business, Technologie und Web (BTW)* (2011).

[123] A. Kabiri and D. Chiadmi. "A method for modelling and organazing ETL processes". In: *Second International Conference on Innovative Computing Technology (INTECH'12)*. IEEE. 2012, pp. 138–143.

[124] A. Kabiri and D. Chiadmi. "Survey on ETL processes". In: *Journal of Theoretical and Applied Information Technology* 54.2 (2013).

[125] A. Kabiri, F. Wadjinny, and D. Chiadmi. "Towards a framework for conceptual modeling of ETL processes". In: *Innovative Computing Technology*. Springer, 2011, pp. 146–160.

[126] K. Kakish and T. A. Kraft. "ETL evolution for real-time data warehousing". In: *In: Proceedings of the Conference on Information Systems Applied Research* (2012). ISSN: 2167-1508.

[127] S. Kambhampati et al. "Havasu: A multi-objective, adaptive query processing framework for web data integration". In: *ASU CSE*. Citeseer. 2002.

[128] T.W. Kang and C.H. Hong. "The architecture development for the interoperability between BIM and GIS". In: *Proceedings of the 13th International Conference on Construction Applications of Virtual Reality, London, UK*. 2013, pp. 30–31.

[129] G. Kapos et al. "An integrated framework for automated simulation of SysML models using DEVS". In: *Simulation* 90.6 (2014), pp. 717–744.

[130] A. Karakasidis, P. Vassiliadis, and E. Pitoura. "ETL Queues for Active Data Warehousing". In: *Proceedings of the 2Nd International Workshop on Information Quality in Information Systems (IQIS'05)*. ACM, 2005, pp. 28–39.

[131] A. Kardan, A. Omidvar, and M. Behzadi. "Context based expert finding in online communities using social network analysis". In: *International Journal of Computer Science Research and Application* 2.1 (2012), pp. 79–88.

[132] A. Kardan, A. Omidvar, and F. Farahmandnia. “Expert finding on social network with link analysis approach”. In: *2011 19th Iranian Conference on Electrical Engineering*. IEEE. 2011, pp. 1–6.

[133] H. Kargin Y.and Pirk et al. “Instant-on scientific data warehouses lazy ETL for data-intensive research”. In: (2013).

[134] V. Kashyap and A. Sheth. “Semantic and schematic similarities between database objects: a context-based approach”. In: *The VLDB Journal—The International Journal on Very Large Data Bases* 5.4 (1996), pp. 276–304.

[135] R.M. Keller et al. “Semantic representation and scale-up of integrated air traffic management data”. In: *Proceedings of the International Workshop on Semantic Big Data*. ACM. 2016, pp. 1–6.

[136] V. A. Kherdekar and P. S. Metkewar. “A Technical Comprehensive Survey of ETL Tools”. In: *International Journal of Applied Engineering Research* 11.4 (2016), pp. 2557–2559.

[137] R. Kimball. “A dimensional modeling manifesto”. In: *DBMS* 10.9 (1997), pp. 58–70.

[138] R. Kimball and J. Caserta. *The data warehouse ETL toolkit: Practical techniques for extracting, cleaning, conforming, and delivering data.* John Wiley and Sons, 2004.

[139] R. Kohavi et al. “Lessons and challenges from mining retail e-commerce data”. In: *Journal of Machine Learning* 57.1-2 (2004), pp. 83–113.

[140] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas. “Supervised machine learning: A review of classification techniques”. In: *Emerging artificial intelligence applications in computer engineering* 160 (2007), pp. 3–24.

[141] SB Kotsiantis, D. Kanellopoulos, and PE Pintelas. “Data preprocessing for supervised leaning”. In: *International Journal of Computer Science* 1.2 (2006), pp. 111–117.

[142] M.B. Kraiem et al. “Modeling and OLAPing social media: the case of Twitter”. In: *Social Network Analysis and Mining* 5.1 (2015), p. 47.

[143] N. Kushmerick. “Wrapper induction: Efficiency and expressiveness”. In: *Artificial Intelligence* 118.1-2 (2000), pp. 15–68.

[144] James J. L. “A Data Model for Data Integration”. In: *Electronic Notes in Theoretical Computer Science* 150 (2006), 3––19.

[145] W. Labio and H. Garcia-Molina. “Efficient Snapshot Differential Algorithms for Data Warehousing”. In: *Proceedings of the 22th International Conference on Very Large Data Bases (VLDB’96).* Morgan Kaufmann Publishers Inc., 1996, pp. 63–74. isbn: 1-55860-382-4.

[146] J. Langseth. *Real-time data warehousing: Challenges and solutions.*

[147] M. L. Lee et al. “Cleansing data for mining and warehousing”. In: *International Conference on Database and Expert Systems Applications.* Springer. 1999, pp. 751–760.

[148] M. Lenzerini. “Data Integration: A Theoretical Perspective”. In: *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS’02).* ACM, 2002, pp. 233–246. ISBN: 1-58113-507-6.

[149] N. Leone et al. “The INFOMIX system for advanced integration of incomplete and inconsistent data”. In: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data.* ACM. 2005, pp. 915–917.

[150] C. Li and B. Liu. “An improvement on window snapshot differential algorithm”. In: *Computer Applications and Software* 4.47 (2010), pp. 140–142.

[151] W. Li and C. Clifton. “Semantic integration in heterogeneous databases using neural networks”. In: *vldb.* Vol. 94. 1994, pp. 12–15.

[152] W. Li and C. Clifton. “SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks”. In: *Data & Knowledge Engineering* 33.1 (2000), pp. 49–84.

[153] B. Lindsay et al. *A snapshot differential refresh algorithm.* Vol. 15. 2. ACM, 1986.

[154] X. Liu, C. Thomsen, and T. B. Pedersen. “CloudETL: Scalable Dimensional ETL for Hadoop and Hive”. In: *History* (2012).

[155] X. Liu, C. Thomsen, and T. B. Pedersen. “ETLMR: a highly scalable dimensional ETL framework based on mapreduce”. In: *Transactions on Large-Scale Data-and Knowledge-Centered Systems VIII.* Springer, 2013, pp. 1–31.

[156] X. Liu, C. Thomsen, and T.B. Pedersen. “Mapreduce-based dimensional ETL made easy”. In: *Proceedings of the VLDB Endowment* 5.12 (2012), pp. 1882–1885.

[157] A. Lohiya et al. “Optimize ETL For Banking DDS: Data Refinement Using ETL Process For Banking Detail Data Store (DDS)”. In: *Imperial Journal of Interdisciplinary Research* 3.3 (2017).

[158] A. Lönnqvist and V. Pirttimäki. “The measurement of business intelligence”. In: *Information Systems Management* 23.1 (2006), p. 32.

[159] S. Luján-Mora and J. Trujillo. “Physical modeling of data warehouses using UML”. In: *Proceedings of the 7th ACM international workshop on Data warehousing and OLAP.* ACM. 2004, pp. 48–57.

[160] K. Ma and B. Yang. “Log-based change data capture from schema-free document stores using MapReduce”. In: *Cloud Technologies and Applications (CloudTech), 2015 International Conference on.* IEEE. 2015, pp. 1–6.

[161] Tim A Majchrzak, Tobias Jansen, and Herbert Kuchen. “Efficiency evaluation of open source ETL tools”. In: *Proceedings of the 2011 ACM Symposium on Applied Computing.* ACM. 2011, pp. 287–294.

[162] C. D Manning, H. Schütze, et al. *Foundations of statistical natural language processing.* Vol. 999. MIT Press, 1999.

[163] D. Martins et al. “Challenges in building a big data warehouse applied to the hotel business intelligence”. In: *Proc. 6th Int. Conf. on Applied Informatics and Computing Theory (AICT’15), in Recent Research in Applied Informatics*. 2015, pp. 110–117.

[164] *MBSE Wiki*. http://www.omgwiki.org/MBSE/doku.php.

[165] M. McGuire et al. “A user-centered design for a spatial data warehouse for data exploration in environmental research”. In: *Ecological Informatics* 3.4-5 (2008), pp. 273–285.

[166] P. Mitra, G. Wiederhold, and J. Jannink. “Semi-automatic integration of knowledge sources”. In: *Proceedings of Fusion’99, July 1999* (1999).

[167] P. Mitra, G. Wiederhold, and M. Kersten. “A graph-oriented model for articulation of ontology interdependencies”. In: *International Conference on Extending Database Technology*. Springer. 2000, pp. 86–100.

[168] I. Moalla et al. “Data warehouse design approaches from social media: review and comparison”. In: *Social Network Analysis and Mining*. Vol. 7. 1. Springer, 2017, p. 5.

[169] M. Mohammed and A. Hasson. “Metadata technique with E-government for Malaysian Universities”. In: *International Journal of Computer Science Issues (IJCSI)* 9.4 (2012), p. 234.

[170] M. Mohammed et al. “Meta-data and Data Mart solutions for better understanding for data and information in E-government Monitoring”. In: *International Journal of Computer Science Issues (IJCSI)* 9.6 (2012), p. 78.

[171] M. A. Mohammed et al. “E-government architecture uses data warehouse techniques to increase information sharing in Iraqi universities”. In: *E-Learning, E-Management and E-Services (IS3e), 2012 IEEE Symposium on*. IEEE. 2012, pp. 1–5.

[172] K. C. Mondal, N. Biswas, and S. Saha. “Role of Machine Learning in ETL Automation”. In: *Proceedings of the 21st International Conference on Distributed Computing and Networking*. 2020, pp. 1–6.

[173] A. E. Monge. “Matching algorithms within a duplicate detection system”. In: *IEEE Data Engineering Bulletine* 23.4 (2000), pp. 14–20.

[174] A. E. Monge, C. Elkan, et al. “The Field Matching Problem: Algorithms and Applications.” In: *KDD*. 1996, pp. 267–270.

[175] R Mooney. “Relational learning of pattern-match rules for information extraction”. In: *Proceedings of the Sixteenth National Conference on Artificial Intelligence*. Vol. 328. 1999, p. 334.

[176] S. L. Mora, P. Vassiliadis, and J. Trujillo. “Data Mapping Diagrams for Data Warehouse Design with UML”. In: *In Proc. 23rd International Conference on Conceptual Modeling* (November 2004), pp. 191–204.

[177] P. Mork, A. Halevy, and P. Tarczy-Hornoch. “A model for data integration systems of biomedical data applied to online genetic databases.” In: *Proceedings of the AMIA Symposium*. American Medical Informatics Association. 2001, p. 473.

[178] M. Mrunalini, T. S. Kumar, and K. R. Kanth. “Simulating secure data extraction in extraction transformation loading (ETL) processes”. In: *Third UKSim European Symposium on Computer Modeling and Simulation (EMS’09).* IEEE. 2009, pp. 142–147.

[179] M Mrunalini, TV S. Kumar, and K R. Kanth. “Secure ETL Process Model: An Assessment of Security in Different Phases of ETL”. In: *International Journal of Software Engineering* 6.s1 (2013).

[180] M Mrunalini et al. “Modeling of secure data extraction in ETL processes using UML 2.0”. In: *Proceedings of the IASTED Asian Conference on Modelling and Simulation.* ACTA Press. 2007, pp. 230–235.

[181] M. Mrunalini et al. “Modelling of Data Extraction in ETL Processes Using UML 2.0”. In: *DESIDOC Bulletin of Information Technology* 26.5 (2006), pp. 3–9.

[182] G. Muhammad et al. “Business intelligence as a knowledge management tool in providing financial consultancy services”. In: *American Journal of Information Systems* 2.2 (2014), pp. 26–32.

[183] H. Müller and J. C. Freytag. *Problems, methods, and challenges in comprehensive data cleansing.* Professoren des Inst. Für Informatik, 2005.

[184] L. Muñoz, J. N. Mazón, and J. Trujillo. “A family of experiments to validate measures for UML activity diagrams of ETL processes in data warehouses”. In: *Information and Software Technology* 52.11 (2010), pp. 1188–1203.

[185] L. Muñoz, J. N. Mazón, and J. Trujillo. “Automatic generation of ETL processes from conceptual models”. In: *Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP.* ACM. 2009, pp. 33–40.

[186] L. Muñoz, J. N. Mazón, and J. Trujillo. “Measures for ETL processes models in data warehouses”. In: *Proceedings of the first international workshop on Model driven service engineering and data quality and security.* ACM. 2009, pp. 33–36.

[187] L. Muñoz et al. “Modelling ETL processes of data warehouses with UML activity diagrams”. In: *Workshops On the Move to Meaningful Internet Systems: OTM.* Springer. 2008, pp. 44–53.

[188] R.P.D. Nath, K. Hose, and T.B. Pedersen. “Towards a programmable semantic extract-transform-load framework for semantic data warehouses”. In: *Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP.* ACM. 2015, pp. 15–24.

[189] R.P.D. Nath et al. “SETL: A programmable semantic extract-transform-load framework for semantic data warehouses”. In: *Information Systems* 68 (2017), pp. 17–43.

[190] T. B. Nguyen, A. M. Tjoa, and R. R. Wagner. “An object oriented multidimensional data model for OLAP”. In: *Web-Age Information Management.* Springer, 2000, pp. 69–82.

[191] T. M. Nguyen et al. “An approach towards an event-fed solution for slowly changing dimensions in data warehouses with a detailed case study”. In: *Data & Knowledge Engineering* 63.1 (2007), pp. 26–43.

[192] T.M. Nguyen, J. Schiefer, and A.M. Tjoa. “Sense & response service architecture (SARESA): an approach towards a real-time business intelligence solution and its use for a fraud detection application”. In: *Proceedings of the 8th ACM international workshop on Data warehousing and OLAP.* ACM. 2005, pp. 77–86.

[193] M. Niinimäki and T. Niemi. “An ETL process for OLAP using RDF/OWL ontologies”. In: *Journal on Data Semantics XIII.* Springer, 2009, pp. 97–119.

[194] S. Nilakanta, K. Scheibe, and A. Rai. “Dimensional issues in agricultural data warehouse designs”. In: *Computers and electronics in agriculture* 60.2 (2008), pp. 263–278.

[195] B. Oliveira and O. Belo. “BPMN patterns for ETL conceptual modelling and validation”. In: *Foundations of Intelligent Systems.* Springer, 2012, pp. 445–454.

[196] B. Oliveira and O. Belo. “ETL Standard Processes Modelling - A Novel BPMN Approach”. In: *Proceedings of the 15th International Conference on Enterprise Information Systems.* 2013, pp. 120–127. isbn: 978-989-8565-59-4.

[197] B. Oliveira, V. Santos, and O. Belo. “Pattern-based ETL conceptual modelling”. In: *Model and Data Engineering.* Springer, 2013, pp. 237–248.

[198] OMG. *SysML-Modelica Transformation (SyM).* http://www.omg.org/spec/SyM/1.0/.

[199] *OMG Systems Modeling Language.* http://www.omgsysml.org/.

[200] A. Omidvar, M. Garakani, and H. Safarpour. “Context based user ranking in forums for expert finding using WordNet dictionary and social network analysis”. In: *Information Technology and Management* 15.1 (2014), pp. 51–63.

[201] *Oracle9i Data Warehousing Guide.* https://docs.oracle.com.

[202] A. S. Pall and J. S. Khaira. “A comparative Review of Extraction, Transformation and Loading Tools”. In: *Database Systems Journal* 4.2 (2013), pp. 42–51.

[203] A. Pareek et al. “Real-time ETL in Striim”. In: *Proceedings of the International Workshop on Real-Time Business Intelligence and Analytics.* 2018, pp. 1–10.

[204] S. Parent C.and Spaccapietra. “Issues and Approaches of Database Integration”. In: *ACM Communication* 41.5es (1998), pp. 166–178. issn: 0001-0782.

[205] K. Patroumpas et al. “TripleGeo: an ETL Tool for Transforming Geospatial Data into RDF Triples.” In: *EDBT/ICDT Workshops.* 2014, pp. 275–278.

[206] F. Pecoraro, D. Luzi, and F. L Ricci. “Designing ETL Tools to Feed a Data Warehouse Based on Electronic Healthcare Record Infrastructure.” In: *MIE.* 2015, pp. 929–933.

[207] N. Polyzotis et al. “Supporting streaming updates in an active data warehouse”. In: *IEEE 23rd International Conference on Data Engineering (ICDE’07).* IEEE. 2007, pp. 476–485.

[208] P. Ponniah. *Data warehousing fundamentals: a comprehensive guide for IT professionals.* John Wiley & Sons, 2004.

[209] N Prasath and J Sreemathy. "A New Approach for Cloud Data Migration Technique Using Talend ETL Tool". In: *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS).* Vol. 1. IEEE. 2021, pp. 1674–1678.

[210] X. Qiao et al. "Constructing a data warehouse based decision support platform for China tourism industry". In: *Information and Communication Technologies in Tourism 2014.* Springer, 2013, pp. 883–893.

[211] W. Qu et al. "Real-Time Snapshot Maintenance with Incremental ETL Pipelines in Data Warehouses". In: *Big Data Analytics and Knowledge Discovery.* Springer, 2015, pp. 217–228.

[212] V. Radhakrishna and K. SravanKiran V.and Ravikiran. "Automating ETL process with scripting technology". In: *Nirma University International Conference on Engineering (NUiCONE).* IEEE. 2012, pp. 1–4.

[213] M. Radonić and I. Mekterović. "ETLator-a scripting ETL framework". In: *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017 40th International Convention on.* IEEE. 2017, pp. 1349–1354.

[214] E. Rahm and P. A. Bernstein. "A survey of approaches to automatic schema matching". In: *The VLDB Journal* 10.4 (2001), pp. 334–350.

[215] E. Rahm and H. H. Do. "Data cleaning: Problems and current approaches". In: *IEEE Data Engineering Bulletine* 23.4 (2000), pp. 3–13.

[216] A. Rai et al. "Design and development of data mart for animal resources". In: *Computers and electronics in agriculture* 64.2 (2008), pp. 111–119.

[217] M. Räkers. "A communication efficiency model for ETL projects in financial data warehousing." In: *ECIS.* 2009, pp. 2444–2455.

[218] V. Raman and J. M. Hellerstein. "Potter's wheel: An interactive data cleaning system". In: *VLDB.* Vol. 1. 2001, pp. 381–390.

[219] G. K. Rao and R. Kumar. "Framework to integrate business intelligence and knowledge management in banking industry". In: *Review of Business and Technology Research* 4.1 (2011).

[220] P.K. Reddy, G.V. Ramaraju, and G.S. Reddy. "eSagu™: a data warehouse enabled personalized agricultural advisory system". In: *Proceedings of the 2007 ACM SIGMOD international conference on Management of data.* ACM. 2007, pp. 910–914.

[221] N. Rehman et al. "Building a data warehouse for twitter stream exploration". In: *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012).* IEEE Computer Society. 2012, pp. 1341–1348.

[222] A.M. Riad, H.M. El-Bakry, and H. Gamal. "A Novel E-Service for E-Government". In: *International Journal of Computer Science and Information Security* 9.1 (2011), pp. 193–200.

[223] J. Rodic and M. Baranovic. "Generating data quality rules and integration into ETL process". In: *Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP.* ACM. 2009, pp. 65–72.

[224] O. Romero, A. Simitsis, and A. Abelló. "GEM: requirement-driven generation of ETL and multidimensional conceptual designs". In: *Data Warehousing and Knowledge Discovery.* Springer, 2011, pp. 80–95.

[225] N. Roussopoulos, Y. Kotidis, and M. Roussopoulos. "Cubetree: organization of and bulk incremental updates on the data cube". In: *ACM SIGMOD Record.* Vol. 26. 2. ACM. 1997, pp. 89–99.

[226] A. Rudra and E. Yeo. "Key issues in achieving data quality and consistency in data warehousing among large organisations in Australia". In: *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on.* IEEE. 1999, 8–pp.

[227] P. Russom. "BIG DATA ANALYTICS". In: *TDWI Best Practices Report, Fourth Quarter 2011.* TDWI Research. 2011, pp. 1–38.

[228] A. I. Saada, G. A. El Khayat, and S. K. Guirguis. "Cloud computing based ETL technique using Warehouse Intermediate Agents". In: *International Conference on Computer Engineering & Systems (ICCES'11).* IEEE. 2011, pp. 301–306.

[229] A. Sahuguet and F. Azavant. "Building light-weight wrappers for legacy web datasources using W4F". In: *VLDB.* Vol. 99. 1999, pp. 738–741.

[230] O. Santos V.and Belo. "Slowly changing dimensions specification a relational algebra approach". In: *International Journal on Information Technology* 1.3 (2011), pp. 63–68.

[231] R. J. Santos and J. Bernardino. "Optimizing data warehouse loading procedures for enabling useful-time data warehousing". In: *Proceedings of the International Database Engineering & Applications Symposium.* ACM. 2009, pp. 292–299.

[232] V. Santos, R. Silva, and O. Belo. "Towards a low cost ETL system". In: *International Journal of Database Management Systems (IJDMS'14)* 6.2 (2014).

[233] V. Santos et al. "Configuring and executing etl tasks on grid environments-requirements and specificities". In: *Procedia Technology* 1 (2012), pp. 112–117.

[234] M. Sanyal, S. Das, and S. Bhadra. "BIG Data Analysis for Indian e-Governance Projects—A Proposed Framework to Improve Real Time Reporting". In: *Information Systems Design and Intelligent Applications.* Springer, 2015, pp. 659–667.

[235] C. Sapia et al. "Extending the E/R model for the multidimensional paradigm". In: *Advances in Database Technologies.* Springer, 1998, pp. 105–116.

[236] K. U. Sattler and E. Schallehn. "A data preparation framework based on a multidatabase language". In: *International Symposium on Database Engineering and Applications.* IEEE. 2001, pp. 219–228.

[237] N. Schmidt et al. "ETL Tool Evaluation–A Criteria Framework". In: ().

[238] F. Sebastiani. “Machine learning in automated text categorization”. In: *ACM computing surveys (CSUR)* 34.1 (2002), pp. 1–47.

[239] S. Sen, D. Datta, and N. Chaki. “An architecture to maintain materialized view in cloud computing environment for OLAP processing”. In: *2012 International Conference on Computing Sciences*. IEEE. 2012, pp. 360–365.

[240] S. Sharma and R. Jain. “Outlier Detection in Agriculture Domain: Application and Techniques”. In: *Big Data Analytics*. Springer, 2018, pp. 283–296.

[241] S. Sharma, R. Jain, and P. Mittal. “AGRETL: Tool for ETL activities for agriculture domain”. In: *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on*. IEEE. 2016, pp. 1877–1883.

[242] J. Shi et al. “Study on log-based change data capture and handling mechanism in real-time data warehouse”. In: *Computer Science and Software Engineering, 2008 International Conference on*. Vol. 4. IEEE. 2008, pp. 478–481.

[243] A. Simitsis. “Mapping conceptual to logical models for ETL processes”. In: *Proceedings of DOLAP* (2005), 67––76.

[244] A. Simitsis. “Modeling and managing ETL processes”. In: *CEUR Workshop/VLDB PhD Workshop Proceedings* 76 (2003).

[245] A. Simitsis, D. Skoutas, and M. Castellanos. “Representation of conceptual ETL designs in natural language using Semantic Web technology”. In: *Data & Knowledge Engineering* 69.1 (2010), pp. 96–115.

[246] A. Simitsis and P. Vassiliadis. “A Method for the Mapping of Conceptual Designs to Logical Blueprints for ETL Processes”. In: *Decision Support Systems* 45.1 (2008), pp. 22–40.

[247] A. Simitsis and P. Vassiliadis. “A methodology for the conceptual modelling of ETL processes.” In: *Proceedings of DSE* (2003).

[248] A. Simitsis et al. “Graph-based modeling of ETL activities with multi-level transformations and updates”. In: *Data Warehousing and Knowledge Discovery*. Springer, 2005, pp. 43–52.

[249] A. Simitsis et al. “Optimizing Analytic Data Flows for Multiple Execution Engines”. In: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 2012, pp. 829–840. isbn: 978-1-4503-1247-9.

[250] A. Simitsis et al. “Optimizing ETL workflows for fault-tolerance”. In: *IEEE 26th International Conference on Data Engineering (ICDE’10)*. IEEE. 2010, pp. 385–396.

[251] A. Simitsis et al. “QoXdriven ETL design: reducing the cost of ETL consulting engagements”. In: *ACM,SIGMOD Conference* (2009), pp. 953–960.

[252] R. Singh and K. Singh. “A descriptive classification of causes of data quality problems in data warehousing”. In: *International Journal of Computer Science Issues* 7.3 (2010), pp. 41–50.

[253] D. Skoutas and A. Simitsis. “Designing ETL processes using semantic web technologies”. In: *In Proceedings ACM 9th International Workshop on Data Warehousing and OLAP (DOLAP 2006)* (2006). Arlington, Virginia, USA, pp. 67–74.

[254] D. Skoutas and A. Simitsis. “Ontology-Based Conceptual Design of ETL Processes for Both Structured and Semi-Structured Data”. In: *International Journal of Semantic Web Information Systems (IJSWIS)* 3.4 (2007), pp. 1–24.

[255] D. Skoutas, A. Simitsis, and T. K. Sellis. “Ontology-Driven Conceptual Design of ETL Processes Using Graph Transformations”. In: *Springer Journal on Data Semantics* 5530 (2009). Special issue on Semantic Data Warehouses (JoDS XIII), pp. 119–145.

[256] S. Snezana and M. Violeta. “Business intelligence tools for statistical data analysis”. In: *Proceedings of the 32nd International Conference on Information Technology Interfaces (ITI’10)*. 2010, pp. 199–204.

[257] S Soderland. “Learning information extraction rules for semi-structured and free text”. In: *Machine learning* 34.1 (1999), pp. 233–272.

[258] J. Song, Y. Bao, and J. Shi. “A triggering and scheduling approach for ETL in a real-time data warehouse”. In: *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*. IEEE. 2010, pp. 91–98.

[259] K Srikanth, N. Murthy, and J Anitha. “Data Warehousing Concept Using ETL Process For SCD Type-1”. In: *International Journal of Computer Science & Applications (TIJCSA)* 1.10 (2012).

[260] K Srikanth, N. Murthy, and J Anitha. “Data Warehousing Concept Using ETL Process For SCD Type-3”. In: *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)* 2.5 (2013).

[261] P. J Stromquist, B. D Schoening, and D. A Austin. *Extraction, transformation and loading designer module of a computerized financial system*. US Patent 7,805,341. 2010.

[262] I. M. Sukarsa, N. W. Wisswani, and IK Gd Darma. “Change Data Capture on OLTP Staging Area for Nearly Real Time Data Warehouse Base on Database Trigger”. In: *International Journal of Computer Applications* 52.11 (2012).

[263] Y. Sun et al. “An Architecture Model of Management and Monitoring on Cloud Services Resources”. In: *International Conference on Advanced Computer Theory and Engineering (ICACTE)*. IEEE. 2010, pp. 27–33.

[264] S. Suresh and M. Mahale. “Student performance analytics using data warehouse in e-governance system”. In: *International Journal of Computer Applications* 20.6 (2011).

[265] S. Suresh et al. *Method and architecture for automated optimization of ETL throughput in data warehousing applications*. US Patent 6,208,990. 2001.

[266] A. Taa, M. S. Abdullah, and N. M. Norwawi. “RAMEPs: a goal-ontology approach to analyse the requirements for data warehouse systems”. In: *Transactions on Information Science and Applications* 7.2 (2010), pp. 295–309.

[267] M. Tanaka and T. Ishida. “Ontology extraction from tables on the web”. In: *Applications and the Internet, 2006. SAINT 2006. International Symposium on.* IEEE. 2006, 7–pp.

[268] D. M. Tank. “Reducing ETL load times by a new data integration approach for real-time business intelligence”. In: *IJEIR* 1.2 (2012), pp. 56–60.

[269] D. M. Tank et al. “Speeding ETL processing in data warehouses using high-performance joins for Changed Data Capture (CDC)”. In: *Advances in Recent Technologies in Communication and Computing (ARTCom), 2010 International Conference on.* IEEE. 2010, pp. 365–368.

[270] A. Ta’a and M. S. Abdullah. “Goal-ontology approach for modeling and designing ETL processes”. In: *Procedia Computer Science* 3 (2011), pp. 942–948.

[271] A. D. Hoang Thi and B. T. Nguyen. “A Semantic Approach towards CWM-based ETL Processes”. In: *Proceedings of I-SEMANTICS’08* (2008), pp. 58–66.

[272] M. N. Tho and A. M. Tjoa. “Zero-latency data warehousing for heterogeneous data sources and continuous data streams”. In: *5th International Conference on Information Integrationand Web-based Applications Services.* 2003, pp. 55–64.

[273] C. Thomsen and T. Pedersen. “Easy and effective parallel programmable ETL”. In: *Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP.* ACM. 2011, pp. 37–44.

[274] C. Thomsen and T. Pedersen. “pygrametl: A powerful programming framework for extract-transform-load programmers”. In: *Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP.* ACM. 2009, pp. 49–56.

[275] C. Thomsen and T. B. Pedersen. “Proceedings of 7th International Conference Data Warehousing and Knowledge Discovery”. In: Springer Berlin Heidelberg, 2005. Chap. A Survey of Open Source Tools for Business Intelligence, pp. 74–84. ISBN: 978-3-540-31732-6.

[276] J. Trujillo and S. L. Mora. “A UML based approach for Modeling ETL Processes in Data Warehouses”. In: *LNCS, Springer Verlag* 2813/2003 (2003), 307–320.

[277] A. Tsois, N. Karayannidis, and T. K. Sellis. “MAC: Conceptual data modeling for OLAP.” In: *DMDW.* Vol. 39. 2001, p. 5.

[278] V. Tziovara, P. Vassiliadis, and A. Simitsis. “Deciding the physical implementation of ETL workflows”. In: *Proceedings of the ACM tenth international workshop on Data warehousing and OLAP.* ACM. 2007, pp. 49–56.

[279] H. T Uitermark et al. “Ontology-based geographic data set integration”. In: *Spatio-temporal database management.* Springer. 1999, pp. 60–78.

[280] C. R. Valêncio et al. “Real time delta extraction based on triggers to support data warehousing”. In: *2013 International Conference on Parallel and Distributed Computing, Applications and Technologies.* IEEE. 2013, pp. 293–297.

[281] P. Vassiliadis. “A Survey of Extract – Transform – Load Technology”. In: *International Journal of Data Warehousing and Mining* 5.3 (2009), pp. 1–27.

[282] P. Vassiliadis and A. Simitsis. “Extraction, transformation, and loading”. In: *Encyclopedia of Database Systems*. Springer, 2009, pp. 1095–1101.

[283] P. Vassiliadis and A. Simitsis. “Near Real Time ETL”. In: *Springer Annals of Information Systems* 3.978-0-387-87430-2 (2008). Special issue on New Trends in Data Warehousing and Data Analysis.

[284] P. Vassiliadis, A. Simitsis, and E. Baikousi. “A taxonomy of ETL activities”. In: *Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP*. ACM. 2009, pp. 25–32.

[285] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos. “Conceptual modeling for ETL processes”. In: *Proc. DOLAP* (2002), 14–21.

[286] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos. “Modeling ETL activities as graphs.” In: *DMDW 2002*. Vol. 58. 2002, pp. 52–61.

[287] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos. “On the Logical Modeling of ETL Processes”. In: *Proc. International Conference on Advanced Information Systems Engineering* (2002), 782–786.

[288] P. Vassiliadis et al. “A Framework for the Design of ETL Scenarios”. In: *Advanced Information Systems Engineering*. Springer. 2003, pp. 520–535.

[289] P. Vassiliadis et al. “ARKTOS: towards the modeling, design, control and execution of ETL processes”. In: *Information Systems* 26.8 (2001), pp. 537–561.

[290] Z. Vassiliadis P.and Vagena et al. “ARKTOS: A tool for data cleaning and transformation in data warehouse environments”. In: *IEEE Data Eng. Bull.* 23.4 (2000), pp. 42–47.

[291] *Vertica*. http://www.vertica.com/the-analytics-platform/real-time-loading-querying/.

[292] A. Walha, F. Ghozzi, and F. Gargouri. “ETL design toward social network opinion analysis”. In: *Computer and Information Science 2015*. Springer, 2016, pp. 235–249.

[293] T. Weilkiens. *Systems engineering with SysML/UML: modeling, analysis, design*. Morgan Kaufmann, 2011.

[294] A. Wibowo. “Problems and available solutions on the stage of Extract, Transform, and Loading in near real-time data warehousing (a literature study)”. In: *International Seminar on Intelligent Technology and Its Applications (ISITIA)*. 2015, pp. 345–350.

[295] J Wiener and J Naughton. *Incremental loading of object databases*. Tech. rep. Stanford InfoLab, 1996.

[296] J. L. Wiener and J. F. Naughton. “OODB bulk loading revisited: The partitioned-list approach”. In: *VLDB*. 1995, pp. 30–41.

[297] K. Wilkinson et al. “Leveraging business process models for ETL design”. In: *Conceptual Modeling in ER*. Springer, 2010, pp. 15–30.

[298] W. Winkler. *Improved decision rules in the fellegi-sunter model of record linkage*. Tech. rep. Decision Rules in the Felligi-Sunter Model of Record Linkage,” Technical Report Statistical Research Report Series RR93/12, 1993.

[299] W. Winkler. *Methods for record linkage and bayesian networks*. Tech. rep. Technical report, Statistical Research Division- US Census Bureau, Washington DC, 2002.

[300] F. Yang. “Analysis and Design of ETL in Hospital Performance Appraisal System”. In: *Computer and Information Science* 2.4 (2009), p. 116.

[301] S. Yoo et al. “Electronically implemented clinical indicators based on a data warehouse in a tertiary hospital: its clinical benefit and effectiveness”. In: *International journal of medical informatics* 83.7 (2014), pp. 507–516.

[302] T. Yu. “A Materialized View-based Approach to Integrating ETL Process and Data Warehouse Applications.” In: *IKE*. Citeseer. 2006, pp. 257–263.

[303] G. Yuan, B. Li, and T. Xiao. “Improvement of snapshot differential algorithm based on hadoop platform”. In: *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC’11)*. Vol. 2. IEEE. 2011, pp. 1212–1214.

[304] E. Zapletal et al. “Methodology of integration of a clinical data warehouse with a clinical information system: the HEGP case.” In: *MedInfo*. Citeseer. 2010, pp. 193–197.

[305] X. Zhang et al. “Generating incremental ETL processes automatically”. In: *First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS’06)*. Vol. 2. IEEE. 2006, pp. 516–521.

[306] Z. Zhang and S. Wang. “A Framework Model Study for Ontology-Driven ETL Processes”. In: *4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM’08)*. IEEE. 2008, pp. 1–4.

[307] X. Zhou et al. “Building clinical data warehouse for traditional Chinese medicine knowledge discovery”. In: *2008 International Conference on BioMedical Engineering and Informatics*. IEEE. 2008, pp. 615–620.

[308] X. Zhou et al. “Development of traditional Chinese medicine clinical data warehouse for medical knowledge discovery and decision support”. In: *Artificial Intelligence in medicine* 48.2-3 (2010), pp. 139–152.

[309] J. Zubcoff, J. Pardillo, and J. Trujillo. “A UML profile for the conceptual modelling of data-mining with time-series in data warehouses”. In: *Information and Software Technology* 51.6 (2009), pp. 977–992.

[310] N. E. Çağıltay et al. “Abstract conceptual database model approach”. In: *Conference on Science and Information*. 2013, pp. 275–281.

Neepa Biswas