IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A FRAMEWORK FOR ETL DESIGN EVALUATION BASED ON QUALITY FRAMEWORK

[1]Author: Madduru Samba Sivudu, research scholar at the department of Computer Science & Engineering at Sri Satya Sai University of Technology & Medical Sciences, Sehore-MP.

[2]Author: Dr. Pankaj Kawadkar, Professor at the department of Computer Science & Engineering at Sri Satya Sai University of Technology & Medical Sciences, Sehore-MP.

Abstract

Using Extraction, transformation, and loading ETL activities, an association might collect information from both inside and outside sources, change it, and afterward populate an information stockroom with the consequences of these cycles. As a model for business processes, the Business Process Modelling and Notation has been given. ETL procedures are expressed on a conceptual level. In this study, an alternative strategy is investigated. ETL is specified using relational algebra (RA), which has been enhanced with update procedures. A data warehousing architecture's Extraction, Transformation, and Loading (ETL) process is critical. ETL operations are often time-consuming and sophisticated. The design step is very crucial (though sometimes overlooked) in ETL development since it has an influence on the future phases, namely implementation. As well as execution When ETL quality is addressed throughout the design process, it is possible to take steps that will have a good and long-term impact.Process efficiency is improved at a cheap cost. Briand et alframework .'s (theoretical validation) was used. Internal metrics that we believe are linked to process efficiency are established in detail (framework for system artefacts). We also present empirical support for this relationship, as well as an interpretation of the data.the efficiency indicators that have a greater influence. ETL design quality has been addressed before, but this is the first time that measurements across ETL models have been used to anticipate the performance of these models.

Keywords: Data Integration Performance,ETL processes, Empirical Validation, Design Quality, Theoretical Validation.

1. INTRODUCTION

The ETL operations extract, transform, and load (ETL) activities extract, transform, and load (ETL) data from an organization's internal and external sources into a data warehouse (DW). It's vital to keep development and maintenance costs low since ETL procedures are sophisticated and costly. Modeling these processes at an abstract level might be helpful. In the absence of a shared conceptual paradigm for expressing such operations, existing ETL systems have to utilise their own specialised linguists. Work processes for information extraction and stacking (ETL) should be laid out. Accordingly, the exploration proposes two strategies for creating ETL processes. Utilizing the Business Process Modeling Notation (BPMN), which has arisen as the accepted norm for demonstrating business processes, the BPMN4ETL structure was made. defines such processes in a conceptual and implementation-agnostic manner. Relational algebra (RA) is a formal language for expressing ETL methods in relational databases, and the

second model is logical. Two causes are driving the investigation of these two options: Because BPMN is widely used to describe business processes, employing it for ETL seems like a smart idea. Those who are already acquainted with the language will have no problem using this tool with no learning curve. When it comes to RAG, however, it is both a formal and an informal language that has undergone extensive research. Data flow may be seen clearly thanks to the ETL's expressiveness. A data warehouse (DW) is a repository for data gathered from a variety of sources, allowing users to make informed choices based on the information it contains. Extraction, Transformation, and Loading are the three procedures that are used to consolidate data (ETL). The four steps of ETL development are depicted in Fig. 1 (Inmon, 2002). The ETL process has been commonly claimed to be complicated and time-consuming, to the point that it accounts for around 80% of a DW project (Inmon, 2002; A. Simitsis and Sellis,). For example, ETL processes are commonly implemented as SQL or Java apps that are tailored to the specific demands of the business. Since databases optimise SQL queries, ETL optimization can't be limited to that; it must also take the process structure into account. The task combination and order are the two most important factors of structural optimization. For instance, in (A. Simitsis and Sellis, 2005), an algorithm found the optimal process alternative by altering the order in which the various process tasks are performed. The combination aspect, on the other hand, has not been addressed in the literature. When faced with a similar integration difficulty, there are several process solutions, each with a different set of duties. and a combination of these two (for example, a process with a few jobs that individually perform heavy work might be similar to a process with many less loaded tasks). It's still impossible to determine the best process structure without a complete approach. There is a line of study aimed at refining the ETL process that suggests conceptual modelinglanguages to describe ETL workflows (Z. El Akkaoui and Zimanyi; Trujillo and Luj an-Mora, 2003;'P. Vassiliadis and Skiadopoulos; ' (U. Dayal and Wilkinson, 2009). In Section 2, we provide our thoughts on these ideas. Metrics for improving usability and maintenance are often proposed by them. ETL model design criteria haven't been thoroughly investigated and verified in terms of ETL process efficiency, to our knowledge. To address this issue, we provide tools in this paper that, depending on ETL design features, can aid in the prediction of process efficiency. Based on Briand et al. (L. Briand and Basili, 1996) theoretical framework, we apply a set of structural (also known as internal) metrics to verify the mathematical correctness of the approach. Since these parameters can be calculated throughout the design phase, they have a direct bearing on ETL efficiency, as we'll see.

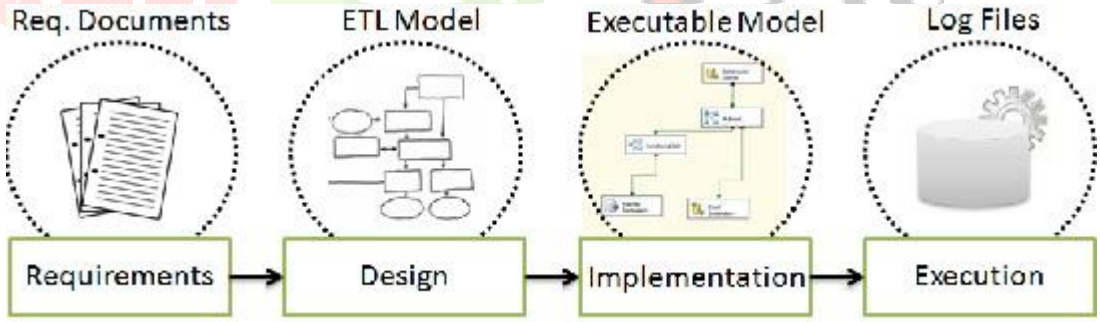


Fig 1: Steps in the development process and criteria for improvement

Specifically, with the throughput of the process (an external parameter). Since there are many different ETL models, it is possible to estimate which one is most likely to produce the highest performance without having to write a single line of code, thereby drastically cutting down on project expenses. This is an important breakthrough in ETL design.

2 MODELS FOR DATA AND QUALITY

2.1 Graph of the ETL Data Process

There will be an ETL model used in this article, therefore here are some details. Due to space limitations, we can only look at the ETL process from the data processing point of view, excluding the control process entirely.

T is a set of different types of nodes. T = "data input", "data output", "filter", "field lookup", "field derivation", "field generation", "join", "union", "aggregation", "sort", "pivot", "script" T = "data input", "data output", "filter", "field lookup", "field derivation", "field generation", "join", "union", "aggregation", "sort", "pivot", "script" There is also a set A that contains a list of possible actions that a node can take.A stands for "field manipulation," "field generation," "join," "lookup," "branching," "extraction," and "load," with "field manipulation" encompassing field computation, deletion, addition, sorting, pivoting, and splitting. A traversal stream is also present on a node:

((field1, field1.datatype),...,(fieldi, fieldi. datatype)) is the schema for a node's input stream.

Data Process Graph (DPG) is the first definition. In a data process graph (G(N,E)), the collection of data tasks and the set of edges between nodes are represented by the directed graph G(N,E). Node b receives input from node a through an edge with the value e = (a,b) E. The following is also true:

• n's type is mapped by a function having the signature N 7 T.

• A subset of A's activities are assigned to each node. There is a relationship between acts N and A that reflects this relationship.

• The input fields of a node are defined by a relation schema N S.

• A node represents a flow type. As for the rest of the nodes in your graph, they may be any of the following: A filter; A join; A union; A lookup; Or a single, unitary node.

• A node is identified by its script category. Non-script node "non-script" is either a script or a non-script node. These nodes may be referred to as script nodes if a data extraction script is contained in them. The absence of script nodes renders them useless.

• A stream category is allocated to a node based on the data traversal treatment it has received. Assuming it's an info yield hub, it's an information yield hub, a line by-column hub (otherwise called "field deduction" or "field age") ("information input," "information yield"). There are offbeat line by-column hubs (each line is p). Line set hubs are offbeat, while nonconcurrent hubs are offbeat (handling begins just when the total column set shows up) (handling starts just when the whole line set shows up).

• As free text, information comments can be appended to hubs and edges.

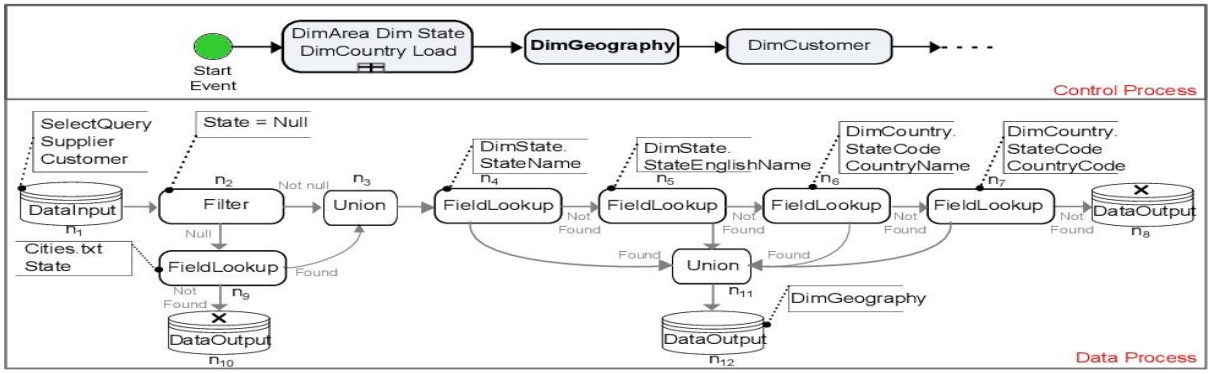


Fig 2: DimGeography is a data loader.

2nd Definition (Valid Data Process Graph). If the following conditions are met, a Data Process graph G is valid:

A "data input" and a "data output" node are included in G.

Every node in G follows the in- and outdegrees given for its type.

Every hub in G has something like one ancestor, and every hub plays out various undertakings. Rather than script hubs, non-script hubs just have one activity. ETL is shown graphically in Figure 2 as a tree with 12 nodes. There is just one node n1=DataInput, a row-by-row node with type(n1) = "data input" and actions(n1) = "extraction," according to the following criteria.

2.2 Families are being measured.

Complexity, connection, cohesiveness, and size are all quality characteristics that are frequently misunderstood. This led to the development of an approach to establishing quality metrics based on mathematical concepts by Briand and Basili (L. Briand and Basili, 1996). A multi-aspect and non-redundant set of measurements may be generated as a result. Mathematical properties described by the authors may be applied to a wide variety of artefacts, not simply software. We then demonstrate how the concept may be used in the context of an ETL process.

Each family of measures, known as quality dimensions, may be connected to create a set of linked measurements that "operationalize" the notion. Briand and colleagues' framework specifies the following families: A system's size, coupling, and cohesion all refer to how many components there are in the system as a whole. Cohesion refers to how well those components work together as subsystems or modules. It evaluates how tightly linked software features are arranged together in subsystems or modules. There are minimal interactions between the constituents of a highly cohesive system. (d) Complexity: this term is used to describe the behaviour of complex systems in general. It tells you how much time and effort it takes to run, change, and understand a system. A system's complexity is impacted by its surroundings.

According to the modular system decomposition (Briand and Basili, 1996), the DPG of Definition 1 is shown in Fig. 3a. A modular system consists of modules that incorporate components. The number of components in a system is given by the tuple N,E>. System elements (A) and edge (E) are two distinct sets of data. between the many features A module m is a subset of the total number of modules in the system's components, which includes all modules. In general, modules may be combined. The term "modular" refers to a system in which the nodes are partitioned into modules. For example, the DPG defines how nodes represent modules and actions represent components in the form of a modular system. To measure size and complexity is to measure size and complexity by definition. calculated throughout the whole modular system, whereasOther metrics are calculated over time.

Non-negativity: the measure (as determined by Size, Coupling, and Complexity) must not be inversely connected to the subject of study.

• Null value: When a system contains no elements, the measure must be null. (This applies to the four families of measurements.)

• Additivity: when numerous modules share no elements, the system's size is equal to the size of its modules. (This is true for all sizes.)

• Non-negativity and normalisation: the measure is unaffected by the system or module's size and corresponds to a specific range. (This is true for Cohesion.)

• Monotonicity asserts that increasing internal relationships between modules does not reduce the value of a measure (but adding an edge between modules does). (This is true for coupling and cohesion.) Non-negativity: the metric must not be inversely proportional to the metric.

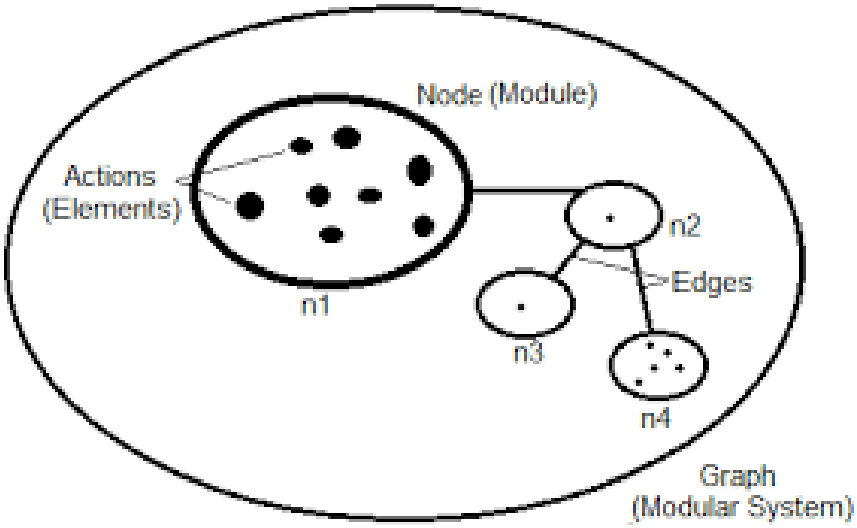


Fig 3: (a) The structure of a system in (L. Briand and Basili, 1996)

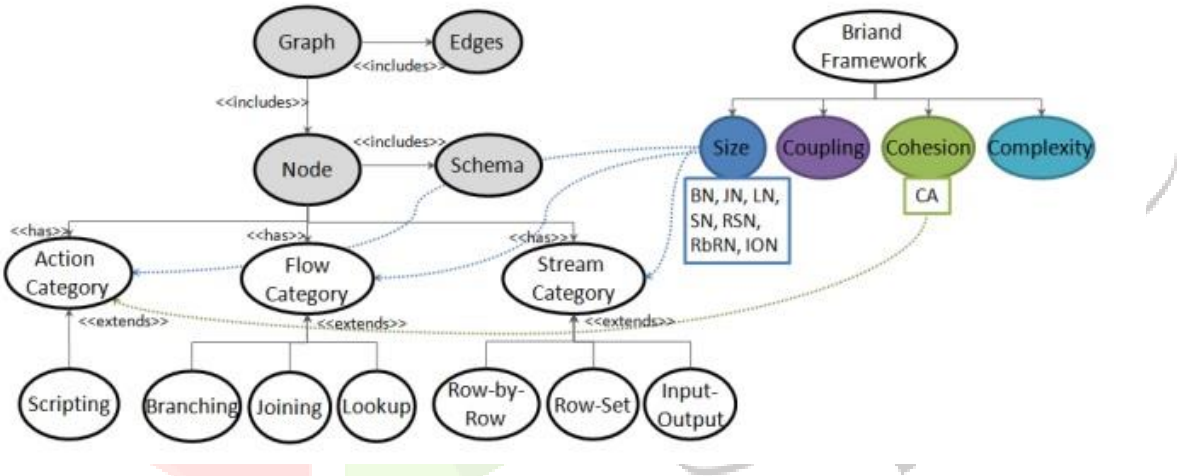


Fig 3 . (b) Its mapping to the DPG.

We may create the node classification hierarchy in Fig. 3b using Definition 1. We discover seven node categories at the bottom of this categorization.

Actually, the reason for these seven size measure categories is that we believe they have an influence on efficiency . Graph definitions and measure families are shown together in Figure 4b, as well (L. Briand and Basili, 1996). For instance, the Cohesion family is linked to the Action family. Various node types may be found in the Size family.

3 QUALITY MEASURES FOR ETL DESIGN

A model's ability to give adequate outcomes in proportion to the amount of resources spent under defined circumstances is evaluated using ISO 9126 quality dimensions of efficiency. We'll begin with external efficiency measures from ISO 9126 (Becker, 2008), which are often utilised in ETL process performance calculations. For the design level, we present a set of structural (internal) measures based on the measure families addressed in Section 4, with an emphasis on the definition of graph node combinations. Internal controls are likely to have an impact on the ETL process' efficiency.

3.1 External Controls

Functionality, dependability, usefulness, efficiency, maintainability, and portability are among the ISO 9126 quality dimensions. To quantify ETL execution efficiency, the ISO standard recommends using the following set of metrics: Execution Time (ET), for example, is the amount of time it takes a server to execute an ETL procedure. Throughput (Th) is a better statistic for measuring ETL efficiency since it takes into account how much data is processed by the ETL system at any one time. To measure throughput, rows are counted as they are processed per unit time. The usage of resources is another important performance measure. For example, during ETL execution, the Disk I/O counts the number of disc read/write operations. It's worth noting that evaluating external measurements necessitates a thorough examination.

Table 1: External measures.

Measure	*Name*	*Description*
ET	Execution Time	Time the server takes to complete the execution
NL	Network Latency	Time the network takes to transfer data from data sourcetarget to the data staging
Th	Throughput	Size of data served by the ETL model per unit of time
DIO	Disk IO	Number of disk readwrite
Me	Memory	Memory amount usage

Internal Controls

Then, we propose a bunch of inward measures, formalize them, and show that they match the highlights referenced in the article (L. Briand and Basili, 1996). We are based onthe formal formulation of the internal measurementsa DPG's definition (Definition 1). In order to portray theWe refer to G in (L. Briand and Basili, 1996) as a framework.

m = (N1,E1) is a module in G, and = (N1,E1) is a graph.

so that a set of nodes N1 is connected to a set of edges E1such that N1 N and E1 E are equal. A module may contain the following items.There is just one node.

How Big Is Your Family? ETL operations might become less efficient as the number of nodes in the network increases. Nodes in a deconstructed ETL network must wait for data to be sent between them before they can process it. Furthermore, because distinct transformations (rather than a single unified transformation) are executed to each row, a deconstructed graph has a latency time. Row-by-row, row-set, or inputoutput are all examples of stream categories, as specified in Definition 1. As a result, decomposed row-set nodes are predicted to have a greater latency. Because of this, we construct a size metric for each node type.

3rd Definition (Branching Nodes). Given a graph G, what is the best way to solve it?

The Branching Nodes metric, BN, measures the number of G branches (G). This is how you do it:

the nodes in the "branching" category's cardinality:

BN(G) = card(n N | type(m) BN(G) = card(n N | type(m) BN(G) = card(n N | type(m) BN

("filter"))

The 4th Term (Joining Nodes). JN(G) is the number of nodes that connect to each other in a given graph G. node in the "joining" category:

Card(N | Type(M) n) = JN

Two words spring to mind: "connect" and "union."

Assertion No. 5 (Lookup Nodes). According to G's Lookup Nodes metric, LN(G), there are a total number of nodes in G that only perform one operation (scripts are not included).

"Fieldlookup" is the type of "lookup" card (n N | type(m) "Fieldlookup"). "fieldlookup" card(actions(n)) = 1) = LN(G) = card(n N | type(m) "fieldlookup" is defined as: LN(G) = card(n N | type (m)

For the sixth time (Script Nodes). Nodes with more than two actions are counted as script nodes in the module m, and this number is the Script Nodes measure for G,SN (G).

SN(G) = card(n n | card(actions(n))

Exercising 1. In Fig. 3, we have: G= DimGeography

card(n2) = 1 BN(G) = BN(G) = BN(G) = BN(G) = BN(G) =

LN(G) = card(n4, n5,n6,n7,n9) = 5. JN(G) = card(n3,n11) = 2.

card(n1,n6,n7) = 3. SN(G) = card(n1,n6,n7) = 3.

Finally, we establish three stream category measurements. When dealing with large amounts of data, the row-set stream type consumes more resources since it utilises a blocking approach to postpone execution. The row-by-row technique is more efficient in terms of time spent. The info yield type is answerable for bringing in and trading information from and to data sets.

7th Definition (RS, RbR, and IO Nodes). The Row-Set (RS) Nodes estimate the number of nodes in a graph G.

Sorting, pivoting, and aggregation are all types of RSN(G). Sorting, pivoting, and aggregation are all types of RSN(G). Sort", "pivot", "RSN(G) = card(n N | type(n)

In G, the RbRN(G) metric measures the number of row-by-row nodes (RbR).

RbRN(G) = card(n N | type(n) " fieldderivation", " fieldgeneration") RbRN(G) = card(n N | type(n) " fieldderivation", " fieldgeneration") RbRN(G) = card(n N | type(n) " fieldderivation", " fieldgeneration

An Input-Output (IO) Nodes (ION) count is used to determine the number of G's data input and output nodes (G).

"datainput," "dataoutput," and "actions" are all cards in ION(G).

The second example There are no row-set nodes, no row-by-row nodes, and only four nodes in the G = DimGeography graph shown in Fig. 3 with RSN=0, RbRN=0, and ION=4 for this network.

The proposed ones, as demonstrated below, verify the mathematical requirements that must characterise any size measure (proofs omitted).

One is non-negativity. A graph G has all of the following properties: zero for all graphs G, zero for all graphs, zero for all graphs, and zero for all graphs.

G = BN(G) = 0, JN(G) = 0, LN(G) = 0, SN(G) = 0, RSN(G) = 0, RbRN(G) = 0, and ION(G) = 0; this is the null value. the fact that BN(G) is equal to the sum of the sum of the sum of modules is known as module additivity (G)

Family Cohesion The quantity of work processed by the graph nodes is related to cohesion. Some nodes may operate differently depending on the scripting category. actions that are more extensive than the rest Field lookup and data input nodes in particular can manage a large amount of data in scripts. The number of activities is unpredictably large. The Cohesion Action measure, which will be developed next, aims to represent theamount of work performed by such nodes A higher investment in time and resources was expected for nodes with a low cohesion (i.e., those that have a greater number of activities). contrasted with highly cohesive nodes One thing to remember about this measure is that it's specific to each individual system module or node.

The 8th Term (Cohesion Action). CA(m) is a measure of m's cohesion action and is defined as follows: We may get CA(m), the Cohesion Action measure of m, from the graph G by taking the following equations into consideration.

When LCAN is equal to CA(m), it means (m)

G, where CAC = nN card(actions (n)) and LCAN = nN card(actions (n))

The diagram G has s.t.actions(n) 2 CA of low-union hubs (m). Eventually, it decides how enormous of a job every hub plays in the gig. CAC is the absolute number of activities performed by the G hubs (m). Altogether, the chart does a work computation. There are several low-cohesion node activities that may be counted as LCAN(m).

4 VALIDATION BY EXPERIMENT

Here, we give a progression of examinations intended to test the connection between the suggested inward measurements and the outward proportion of throughput (Th). We picked Th as the outer measure since it very well may be assessed from the execution time kept in the framework log record and give important outcomes. Our speculations incorporate the accompanying:

• H1: The number of join, lookup, script, row-set, and input-output nodes is linked to Th;

• H2: Th is not affected by the number of row-by-row nodes.

• H3: Th is unaffected by the addition of branching nodes.

• H4: The data input node with a script is the optimal script node form. This is preferable to using a dedicated script node.

H5: Using a data input script node instead of a join node increases Thperformance. .

Then again, supplanting search hubs with information input script hubs doesn't upgrade the exhibition of the application.

The ETL designer will be guided by the confirmation of these hypotheses in addressing key issues like: Which ETL transformation should I do in the most efficient manner? What are the design factors that determine whether throughput is increased or decreased? Can we predict which of two graphs will have the highest throughput?

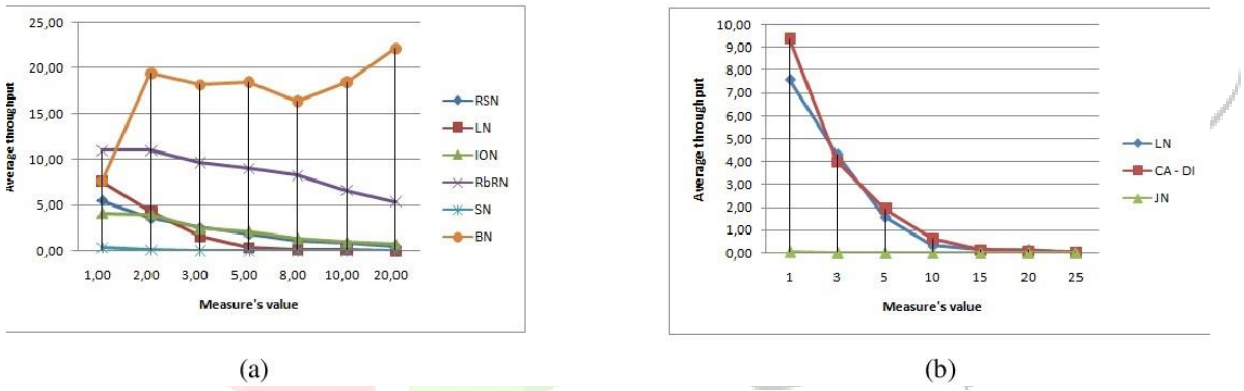
Results Discussion

Now we'll look at the findings of our studies in terms of the hypothesis H1 through H5. For starters, Using a variety of data sources, we examined the link between each metric and Th (1x, 100x,..). Table 2 shows the results, with measurements labelled as BN, JN, and so on. Except for BN, JN, and CA-S, all of the measures are substantially linked with throughput, as shown in Table 2. When it comes to branching nodes, this indicates that increasing the number of branching nodes has a negligible impact on throughput. SSIS's join nodes perform poorly, as seen by the low JN and CA-S throughput figures. Consequently, it is required to do further testing with a wide range of equipment.

The throughput is inversely associated with the measurements, indicating that increasing nodes reduces Th. As a first result, these high correlations validate our measurements and indicate their effect on the throughput external measure, which represents the efficiency target in our study. For the CA measure, we ran two experiments: (a) modifying the data input (DI) script to add node actions, and (b) modifying the particular script (S) to add node actions, both of which resulted in the CA measure (denoted CA-S) being calculated on a specific script node. There were excellent correlations for the (CA-DI) measure but no correlations for the (CA-S) measure could be calculated owing to inadequate throughput.

Table 2: Correlations.

Size	BN	JN	LI	SN	RSN	RbRN	ION	CA - DI	CA-S
65Kb	-0.82	-	-0.90	-0.93	-0.81	-0.94	-0.88	-0.97	-
6.5Mb	-0.69	-	-0.88	-0.74	-0.72	-0.93	-0.81	-0.97	-
32.5Mb	0.13	-	-0.82	-0.62	-0.81	-0.97	-0.93	-0.96	-
65Mb	0.64	-0.51	-0.80	-0.69	-0.78	-0.99	-0.93	-1.00	-



(a) (b)

Figure 4: (a) Throughput vs. # of nodes; (b) Lookup nodes vs. join nodes vs. script nodes

The second study, presented in Fig. 5a, covers the throughput variation for each metric (for the biggest dataset). On the X-axis, the total number of nodes of each kind is shown, while the throughput is shown on the Y-axis. Th is considerably reduced when all types of nodes (excluding branching nodes) are included, as seen in Fig. 5a.

The addition of branching nodes has little effect on throughput, as demonstrated in Table 2. Some external measures may benefit from the addition of these additional BNs (e.g., related with parallel execution). The findings were the same regardless of the size of the dataset. As a result of these observations, hypothesis H1 through H3 are supported.

H4 and H5 are the next possibilities we'll examine. An example of how script nodes (CA-DI) might be utilised instead of join nodes to enhance performance can be seen in Figure 5b (right) (especially for the biggest dataset).

Similarly, we may boost Th by combining numerous input nodes into a single script node. Lookup nodes are also superior than script nodes, as seen in Fig. 5b. Because of the ETL server's ability to manage nodes of this kind efficiently, this is an understandable result. ETL throughput may be boosted if more efficient

nodes are selected during the design phase, according to our results. However, this is not always possible (for example, grouping many row-set nodes in the same script is not a good idea).

CONCLUSION

To better understand how well a system would work, we've developed and scientifically tested a set of metrics that can be applied to a variety of ETL process graphs (representing various combinations of activities). They shall treat each other with respect while they carry out their duties. Even developing a single line of computer code is a challenge. As a result, we may make a set of conclusions based on our findings.design considerations for effective ETL operations While this article focused on the individual (partial) effect of each item, we plan to investigate the complete impact of the proposed policies in the future.

REFERENCES

[1] A. Simitsis, P. V. and Sellis, T. (2005).Optimizing ETL processes in data warehouse environments.In ICDE'21, 21st International Conference on Data Engineering. IEEE Computer Society Press.

[2] Ali, S. and Wrembel, R. (2017). From conceptual design to performance optimization of etl workflows: current state of research and open problems. The VLDB Journal, 26(6):777–801.

[3] B. Boehm, J. Brown, H. K. M. L. G. M. and Merritt, M. (1978).Characteristics of Software Quality (TRW series of software technology).Elsevier. Becker, S. (2008). Performance-related metrics in the ISO 9126 standard.

[4] In I. In Eusgeld, F. F. and Reussner, R., editors, Dependability Metrics, pages 204–206. Springer, Berlin, Heidelberg.

[5] El Akkaoui, Z. and Zimanyi, E. (2012).Defining ETL ´ worfklows using BPMN and BPEL.In DOLAP'09, 9th International Workshop on Data Warehousing and OLAP.ACM Press.

[6] G. Kougka, A. G. and Simitsis, A. (2018). The many faces of data-centric workflow optimization: a survey. International Journal of Data Science and Analytics, 6(2):81–107.

[7] G. Papastefanatos, P. Vassiliadis, A. S. and Vassiliou, Y. (2009).Policy-regulated management of ETL evolution. Journal Data Semantics, 5530:146–176. Inmon, W. (2002).Building the Data Warehouse.

[8] Wiley. Kimball, R. and Ross, M. (2002).The Data Warehouse Toolkit, 2nd.Ed. Wiley.

[9] L. Briand, S. M. and Basili, V. (1996).Property-based software engineering measurement. IEEE Transactions on Software Engineering, 22(2):68–86.

[10] L. Munoz, J. M. and Trujillo, J. (2010). A family of exper- ˜ iments to validate measures for UML activity. Information and Software Technology, 52(11):1188–1203.

[11] P. Vassiliadis, A. Simitsis, P. G. M. T. and Skiadopoulos, S.A generic and customizable framework for the design of ETL scenarios.Information Systems, 30(7).

[12] T. Majchrzak, T. J. and Kuchen, H. (2011).Efficiency evaluation of open source ETL tools. In SAC'11, 11th Proceedings of the ACM Symposium on Applied Computing. ACM Press.

[13] Trujillo, J. and Lujan-Mora, S. (2003). A UML-based ´ approach for modeling ETL processes in data warehouses. In ER'22, 22nd International Conference on Conceptual Modeling.Springer.

[14] U. Dayal, M. Castellanos, A. S. and Wilkinson, K. (2009). Data integration flows for business intelligence. In EDBT'09, 9th International Conference on Extending Database Technology.ACM Press.

[15] Z. El Akkaoui, J. Mazon, A. V. and Zim ´ anyi, E. (2012). ´ BPMN-based conceptual modeling of ETL processes. In DAWAK'12, 12th International Conference on Data Warehousing and Knowledge Discovery.Springer.

[16] Z. El Akkaoui, E. Zimanyi, J. M. A. V. and Trujillo, J. ´ (2013). A bpmn-based design and maintenance framework for ETL processes. International Journal of Data Warehousing and Mining IJDWM, 9(3):46–72.