The next chapter is based on two published research papers


CHAPTER 2
GOAL STRUCTURED REQUIREMENT ENGINEERING FOR DATA WAREHOUSE

Data warehouse is a decision support system that is specifically designed for the business managers and executives for reporting and business analysis. It is a database that stores enterprise-wide data that can be used to deduce useful information. Business organizations can achieve competitive advantage by analyzing its historical data to learn from its past. However data warehousing is still maturing as a technology. In order to effectively design and implement a data warehouse for an organization, its goal must be understood and requirement must be analyzed in the perspective of the identified goal. This chapter presents a goal structured model for requirements engineering that also enables its users to manage traceability between the goals, decisions, business strategy and the corresponding business model.

To evaluate and improve the quality of a conceptual data model (if required), we must assess it in an objective way. This helps the data warehouse designer to make better decisions during design activities. Even when the designer has several conceptual schemas available with them, they can use this assessment to decide the best schema amongst them. We have defined a set of metrics that can be used to measure the complexity, understandability, usability,
accessibility, believability, data currency and modifiability which are defined as quality characteristics for data warehouse systems.

2.1 INTRODUCTION

Requirements have always been identified as a crucial force driving the technical and the design aspects of the data warehouse systems. It is the user’s requirements alone that decide:

- What will be the end user applications?
- What data will be collected and from where it will be collected?
- What will be the facts and dimensions?
- What level of quality is demanded?
- What will be the physical, technical and conceptual design of the warehouse?
- When, where and how the data warehouse will be deployed?
- What will be done for data warehouse maintenance and growth?

Many academia and professionals working in companies have given approaches for information requirements analysis. Most of the approaches are applicable for traditional database systems and a few are suitable for data warehousing projects. For example, these approaches lay emphasis on interviews, questionnaires and job analyses to take user’s input for traditional database
systems but these approaches of data acquisition cannot be applied for data warehouse systems in its entirety.

When a data warehouse is developed, the key issue to be considered is to predict information requirements of data warehouse users in future with available information supply and only few approaches seem to address this issue specifically. The two most widely used approaches are demand drive approach and supply driven approach [10]. The two approaches differ on whether information demand or information supply is more prominent in the matching process.

In demand driven approach, information requirements are collected from the data warehouse users. However, end users alone are unable to specify their requirements precisely and accurately. This is because the users do not have sufficient knowledge about all available information sources in the organization. They use only a specific business application and it is beyond their domain to predict or forecast what type of information the data warehouse can provide them.

However, in supply driven approach (also known as the data driven approach), the data warehouse development team starts with an analysis of transactional source systems to reengineer their logical data schemas. The end users can then be asked to specify their information requirements from such a consolidated data schema. This serves as an input for the data warehouse data schema.
Although the data driven approach [29] though simplifies the process of extraction, transformation and loading of data from the source systems but gives less importance to the user’s requirements. In contrast to this, the requirements driven approach gives prime importance to the user’s requirements. This in turn may lead to complicated ETL process.

Table 4: Comparison between Data Driven and Requirements Driven Approach

<table>
<thead>
<tr>
<th>Demand Driven</th>
<th>Supply Driven</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Information requirements are collected from the data warehouse users.</td>
<td>1. Analysis of source systems is done to reengineer their logical data schemas. The end users are then be asked to specify their requirements from such a consolidated data schema</td>
</tr>
<tr>
<td>2. But end users are unable to specify their requirements precisely and accurately as they do not have sufficient knowledge of all available information sources available in the organization.</td>
<td>2. Simplifies ETL process</td>
</tr>
<tr>
<td>3. Complicated ETL process</td>
<td>3. Gives less importance to user’s requirements</td>
</tr>
<tr>
<td>4. Gives importance to user’s requirements</td>
<td>4. Based on what is available is data repository</td>
</tr>
</tbody>
</table>

The supply driven approach is usually preferred when there is a lack of clearly defined requirements. So in this case, the data warehouse development team put
all it focus on the structure and technical details of the operational sources. The team integrates all the available data to present a comprehensive view of the business. But such an approach may not be in line with the business requirements. Table 4 gives a comparison between the two approaches.

A third approach which is a variant of demand driven approach derives information requirements by analyzing business processes and transforming the relevant data structures of business processes into data structures of the data warehouse [22].

For the requirements driven approach to develop a data warehouse, Jarke et al [73] had proposed to add a Conceptual Design Phase before the Logical Design Phase that could identify the conceptual objects like facts, dimensions, hierarchies, etc. But how these objects can be identified was not discussed. So another phase of Requirements Engineering is proposed to be added before the Conceptual Design Phase [10]. However, the Data Warehouse Requirements Engineering process must be closely related to the strategy formulation process because this is the main reason for using and deploying a Data Warehouse in an organization. Data warehouse systems are primarily used to support business users to take strategic decisions and analyze the performance of their business.

Moreover, business performance is directly related to the goals of the users. However, many domain-specific modeling approaches, treat the data warehouse and user’s goals separately. They assume that these two entities are not related to each other, which is not true. Therefore, there is a need for a model
that gives an insight into user’s goals, strategies and its performance. This enables users to make better quality decisions as they can easily learn from the decisions they took in the past.

2.2 GOAL STRUCTURED REQUIREMENT ENGINEERING AND TRACEABILITY MODEL

We propose a model as outlined in Figure 7 that can be used for requirements engineering and maintaining traceability in data warehouse. The model starts with identifying user’s goals.

![Diagram](image)

*Figure 7: Goal Structured Requirements Engineering and Traceability Model*
Key Terms used in the Model

**Goal:** The objectives of using the data warehouse. A goal can be simple or complex. A simple goal cannot be decomposed further but the complex goal can be decomposed in smaller goals that may itself be simple or complex. This makes a goal hierarchy.

**Stakeholder:** One or more person having interest in the project. They are actively involved in the project. Stakeholders may own more goals. A simple goal may not be shared between different stakeholders. Some examples of stakeholders include customer, the user group, etc.

**Strategy:** A plan to accomplish the goals.

**Decision:** A choice that is made between $n$ number of possible actions

**Quality Information:** Information that is timely, accurate, specific, organized for a purpose giving it a meaning and relevance to improve understanding and decrease uncertainty. The information is valuable only if is effective enough to affect a decision.

**Justification:** Strategy is associated with a justification to elaborate how the framed strategy can help to achieve business goal. Justification may be given using formal proofs.

**Business Model:** A framework that describes core aspects of a business, including purpose, strategies, infrastructure, products, services, organizational
structures, etc. Therefore, it gives a complete picture of an organization from a high-level perspective.

According to the proposed model, the first step in the Requirements Engineering process is to identify user’s goals. To accomplish these goals, formulate a sound strategy. However, to finalize a strategy a number of decisions may have to be taken. These decisions can be made only with the help of quality information. Moreover, every strategy is associated with a justification that helps to explain with proofs how the strategy can accomplish the identified goals. Last but not the least a business model is designed that includes goals, strategies, infrastructure and every technical aspect of the business. This business model is brought in action to be finally executed and achieve strategic advantage. Thus, we see that backbone of this model is the goals of the user and quality information stored in the data warehouse. With this model in place, the data warehouse development team can easily get an insight into what information is actually required to take the decisions.

Let us take an example to realize the proposed model.

**GOAL:** Increase market share by 15%

**SUB-GOAL 1:** Increase sales

**SUB-GOAL 2:** Retain Customer Loyalty

**STAKEHOLDER:** CEO of the business
DECISIONS:

a. Launch a new product in the market

b. Improve the quality of the existing products

c. Open new Retail Outlets

INFORMATION:

a. Information about existing products of the business

b. Information about competitor’s products

c. Information about user’s preferences and usage patterns

d. Information about sales trends in the market

e. Information about area where there is no or less number of stores

STRATEGY:

To launch a new product with better quality in the next two years and open a new retail outlet in the north-west region within a year

JUSTIFICATION:

To launch a new product requires R&D in the area which takes more time, so it can be accomplished within two years. However, opening a new area is a region where there are no/few stores can be done in less time. This strategy will enable the business to enhance its market share by 15%. Justification also includes
factual data to give a formal proof. This factual data can be obtained only from the information stored in the data warehouse.

In this model, when a user has a new goal, then the existing business model and the new goal are inputted in the proposed model to formulate a new strategy and thus a revised business model. This is shown in Figure 8.

![Diagram of the proposed model]

Figure 8: Revising an existing business model

2.3 IMPLEMENTING TRACEABILITY THROUGH THE PROPOSED MODEL

Traceability refers to having complete information about every step in the project. It enables users to chronologically inter-relate uniquely identifiable item, verify its history, location, or application of through proper documentation.

Most of the traceability models that we have till date focus on tracking relationships between requirements specifications and design, but this is not enough for a system like that of a data warehouse. Here, we need to give due importance to the goals of the users. The proposed model can be used to identify a number of traceability relationships between goals and the stakeholders.
In this model, we have a strategy for each goal and every simple goal is owned by a stakeholder. The strategy leads to a business model. So we can easily trace the relationship between a goal, its stakeholder and the corresponding business model that was designed. This helps to easily gather information about the decisions that were taken, the assumptions that were made and the justifications that were given to approve or reject a particular decision.

Moreover, in future when the requirements of the user’s change or when additional requirements pop up or when certain decisions turn out to be impractical, the effect of undoing a decision or modifying the strategy can be easily understood.

Since we can have a hierarchy of goals, the model helps to trace the relationships between goals and strategies through-out the hierarchy. This leads to traceability from abstract statements to documented requirements and further to design.

2.4 ANALYZING QUALITY

Data warehouses form the core of most of the current decision support systems, providing organization with several years of historical information for making strategic decisions. Poor quality of data warehouse system can result in technical and organizational losses like loss of clients, important financial losses or discontent amongst employees [46]. Therefore, it becomes very important for
an organization to guarantee the quality of information in its data warehouse right from the early stages of data warehouse development project.

The information quality in data warehouse systems by and large depends on presentation quality and data warehouse quality (Figure 9). Data warehouse quality in turn depends on the DBMS quality, data quality and data model quality (which can be considered at conceptual, logical and physical levels) [136].

![Data Warehouse Information Quality Diagram](image)

*Figure 9: Data Warehouse Information Quality*

In this chapter, we will focus on the quality of conceptual model because sooner the data warehouse development team concentrates on its quality higher will be the likelihood of implementing a high quality data warehouse [134]. Moreover, conceptual data models forms the foundation of all later design work. So the
conceptual model quality has a significant impact on the quality of the data warehouse.

To evaluate and if required to improve the quality of a conceptual data model first we must analyze it in an objective way. This would help the data warehouse designers to make better decisions during design activities. Even when the designers have several conceptual schemas with them, they can use this assessment to decide the best schema amongst them.

Till date, researchers all over the world have proposed many quality frameworks for conceptual data models but not much work has been done to specify valid metrics to calculate the quality of conceptual data models in an objective way. The scarcity of such metrics made us to identify a set of measures that could analyze our conceptual model of Goal Dimension Table.

2.4.1 Metric Definition Process

The process of metric definition is based on the organization’s measurement goals that are related to quality attributes of the conceptual schema. Figure 10 shows the steps involved in obtaining valid and useful metrics. The process has seven steps. First, the goals of the
metrics that can analyze and control the quality of conceptual model of data warehouse are identified. Then corresponding hypotheses are formed. For example, if the goal is to assess the structural complexity of the conceptual model then the hypotheses could be- “lesser the structural complexity greater is the understandability (which is a quality attribute).”

The next step is to define metrics by considering the hypotheses, specific characteristics of the model that has to be assessed and the experience of the designers of data warehouse systems. A goal-oriented approach as GQM (Goal-Question-Metric [12]) can also be very used for this step.

After defining the metrics, we need to validate them validation has to be done both theoretically as well as empirically. Theoretical validation specifies when and how to apply the metrics. It can be done either using formal frameworks or by measurement theory [136]. While formal framework merely specifies a set of formal properties defined for given software attributes for classifying the proposed metrics; the measurement theory, on the other hand, defines the scale to which a metric pertains. This helps us to know the statistics and transformations that can be applied to the metric.
Followed by theoretical validation, the metrics must be empirically validated to confirm and understand the implications of the measurement of the conceptual model. This can be done through experiments, case studies and surveys.

Once the metric has been validated, it can be simply accepted, redefined or discarded. This means that the final outcome of this step is a valid metric. The valid metric thus obtained is applied to be used in real world. Once it is applied, the metric must be monitored and if required adapted to application changing environment.

2.5 PARAMETERS FOR ANALYZING THE PROPOSED MODEL

In this section, we will define the parameters and the metrics that can be used to analyze the quality of our proposed conceptual schema of goal structured requirements engineering and traceability model. The quality parameters that we have identified are- understandability, modifiability, believability,

![Figure 11: The Goal Template for Assessing the Conceptual Model](image)

requirements engineering and traceability model. The quality parameters that we have identified are- understandability, modifiability, believability,
accessibility, usability, timeliness and data currency. So, our goal would be to evaluate the proposed conceptual schema. The goal and its template has been shown in Figure 11.

### 2.5.1 Understandability

Figure 12 depicts the relationship between structural complexity and quality of data warehouse schema. It has been found that the structural complexity of the conceptual schema affects the cognitive complexity which in turn affects its understandability which is an important external quality of any data warehouse system [136].

![Diagram showing the relationship between structural complexity, cognitive complexity, understandability, and external quality](image)

*Figure 12: Effect of Structural Complexity and Quality of Data Warehouse Schema*

So we see that in order to make the proposed conceptual model understandable, we need to analyze its structural complexity and cognitive complexity.

#### 2.5.1.1 Structural Complexity

The structural complexity of any conceptual schema including the proposed goal structured requirements engineering and traceability model can be evaluated based on the metrics given below:
1. Number of Tables

2. For each Dimension Table, we can calculate
   a. Number of attributes ($NA$)
   b. Relationship with other Dimension Tables
   c. Number of Dimension Hierarchies
   d. Depth of Chosen Hierarchy

3. Number of attributes repeating in different Dimension Tables

4. For Fact Table, we can calculate,
   a. Number of Attributes in Fact Table
   b. Number of Additive Measures
   c. Number of Semi-Additive Measures
   d. Number of Non-Additive Measures

5. Number of Degenerated Dimension Tables

Let the Number of Tables be $NT$

Let the Complexity of Dimension Table $i$ be calculated as $DTC_i$

Let the Complexity of Fact Table be calculated as $FTC_i$

Let the Number of Repeating Attributes be $NRA$
Let the Number of Degenerated Dimension Tables be $NDD$

*Structural Complexity* $= NT \times (\sum DTC_i) \times (\sum FTC_i) \times (1 + NRA) \times (1 + NDD)$

Here, we have added 1 to $NRA$ and $NDD$ because if we have a conceptual schema that has no attribute that repeats and no degenerated dimension, then its structural complexity must not evaluate to be zero.

Alternatively, we can also calculate the complexity for a simple star schema by taking into account the following metrics.

$nD$: Number of dimension tables in the start schema

$nT$: Total number of tables in the start schema. It is equal to $nD + 1$

$nAD$: Total number of attributes in all the dimension tables. It is the sum of number of attributes in each dimension table. Therefore, we can write $nAD = \sum_{i=1}^{nD} nAD_i$, where $nD$ is the number of dimension tables in the star schema

$nAF$: Total number of attributes in the fact table. It is defined as the sum of facts plus the number of foreign keys (attributes shared from the dimension tables). Therefore, $nAF = nF + nFKF$, where $nF$ is number of facts and $nFKF$ is the number of foreign keys.

$nAS$: Total number of attributes in the Star Schema is given as the sum of number of attributes in the fact as well as the dimension tables. Therefore, $nAS = nAD + nAF$

$RA$: Ratio of number of attributes in dimension tables to the number of attributes in the fact table. $RAS = nAD / nAF$
**RFK:** It defines the ratio of number of foreign keys in the fact table to the total number of attributes in the fact table. Therefore, $RFK = \frac{nFKF}{nAF}$

The above justification can be extended to calculate the complexity of a snowflake or a constellation schema. In such schemas, we need to consider some additional attributes that are given below.

**nFK:** Number of foreign keys in the schema. This is defined as the sum of foreign keys in the fact table as well as in each of the dimension tables. Therefore, we can write, $nFK = nFKF + \sum_{i=1,...,nD} nFKD$

**nSD:** Number of shared dimension tables in the schema

Moreover, in such a schema the number of attributes in the fact table will be calculated as $nAF = \sum_{i=1,...,nF} nFi + \sum_{i=1,...,nF} nFKFi$

Similarly, number of attributes in dimension table will be calculated as $nAD = \sum_{i=1,...,nD} nADi + \sum_{i=1,...,nD} nFKDi$, where $nFKDi$ is the number of foreign keys in each dimension table

**nA:** Total number of attributes in the schema can now be written as $nAF + nAD$

**nFKF:** Number of foreign keys in all the fact tables of the schema can be given as $\sum_{i=1,...,nF} nFKFi$

**nFK:** Total number of foreign keys in the star schema can be given as number of foreign keys in all the fact tables plus the number of foreign keys in all the dimension tables. Therefore, $nFK = \sum_{i=1,...,nD} nFKDi + \sum_{i=1,...,nF} nFKFi$

**nASD:** Number of attributes in shared dimension tables
Using the parameters we can calculate the ratios that can give us a clear insight into the complexity of the schema.

**RSD**: The ratio of shared dimension tables in a schema can be given as number of shared dimensions to the total number of dimension tables in the schema. Therefore, $RSD = \frac{n_{SD}}{n_{D}}$.

**RDF**: Ratio of number of dimension tables to the number of fact tables in the schema can be given as $RDF = \frac{n_{D}}{n_{F}}$.

**RA**: Ratio of number of attributes in dimension tables to the number of attributes in the fact tables. It can be written as, $RA = \frac{n_{AD}}{n_{AF}}$.

**RFK**: Ratio of foreign key is number of foreign keys in the schema to the total number of attributes. Therefore, $RFK = \frac{n_{FK}}{n_{A}}$.

**RSDA**: Ratio of attributes in shared dimension tables to the number of attributes in the schema. This can be written as, $RSDA = \frac{n_{ASD}}{n_{A}}$.

Let us take an example of a data warehouse schema as shown in figure 13 and apply our metrics on them.
Figure 13: Sample Star Schema

The value for the metrics for the above schema can be given as shown in Table 5.

Table 5: Metrics and their Values

<table>
<thead>
<tr>
<th>METRIC</th>
<th>VALUE</th>
<th>SCALE</th>
</tr>
</thead>
<tbody>
<tr>
<td>nA</td>
<td>25</td>
<td>Above Ordinal</td>
</tr>
<tr>
<td>nFK</td>
<td>6</td>
<td>Above Ordinal</td>
</tr>
<tr>
<td>nD</td>
<td>4</td>
<td>Above Ordinal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>nT</td>
<td>6</td>
<td>Ratio</td>
</tr>
<tr>
<td>nAD</td>
<td>15</td>
<td>Above Ordinal</td>
</tr>
<tr>
<td>nAF</td>
<td>10</td>
<td>Above Ordinal</td>
</tr>
<tr>
<td>RFK</td>
<td>6/25</td>
<td>Absolute</td>
</tr>
<tr>
<td>nF</td>
<td>2</td>
<td>Ratio</td>
</tr>
<tr>
<td>nSD</td>
<td>2</td>
<td>Above Ordinal</td>
</tr>
<tr>
<td>nASD</td>
<td>8</td>
<td>Ratio</td>
</tr>
<tr>
<td>RSD</td>
<td>2/4</td>
<td>Absolute</td>
</tr>
<tr>
<td>RDF</td>
<td>4/2</td>
<td>Absolute</td>
</tr>
<tr>
<td>RA</td>
<td>15/10</td>
<td>Absolute</td>
</tr>
<tr>
<td>RSDA</td>
<td>8/25</td>
<td>Absolute</td>
</tr>
</tbody>
</table>

**Metrics Formal Validation**

We will be validating the metrics using the formal framework proposed by Zuse [168]. The framework has been used to determine the scale to which a metric pertains.

The framework works with three main mathematical structures- the extensive structure, the independence conditions and the modified relation of belief.
Based on which structure a metric pertains to, it is categorized in a scale as shown in Table 6. The result of the formal validation of the metrics in the Zuse formal framework are summarized in Table 5.

**Table 6: Categorization Scale**

<table>
<thead>
<tr>
<th>Scale</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio</td>
<td>Extensive structure + Independence Conditions</td>
</tr>
<tr>
<td>Ordinal</td>
<td>Independence Conditions but not extensive structure</td>
</tr>
<tr>
<td>Above</td>
<td>Modified Relation of Belief but neither Independence Conditions</td>
</tr>
<tr>
<td>Ordinal</td>
<td>Conditions nor extensive structure</td>
</tr>
</tbody>
</table>

The scale of our metrics are also shown in the table 7. We see that the scale of all our metrics are either above ordinal or superior. According to the Zuse framework the specified metrics are therefore valid metrics.

**Table 7: Scale of the Metrics**
Cognitive complexity is a characteristic that indicates how complex or simple is the concept, and structure and depends on the amount of knowledge required to perform a task using a specific application. It is actually a summation of several factors that make things hard to see, use, understand, and contribute directly to our neural load [13].

In context of data warehouse conceptual schema, the main features that affect understandability are structural complexity and cognitive complexity. Structural complexity depends on parameters discussed in the above section, cognitive complexity, on the other hand, depends on people’s (data warehouse users and designers) interests, capability and perceptions.

Funke [54] has identified factors that affect the cognitive complexity. These factors include:

<table>
<thead>
<tr>
<th>MODIFIED EXTENSION STRUCTURE</th>
<th>nA</th>
<th>nFK</th>
<th>nD</th>
<th>nT</th>
<th>nAD</th>
<th>nAF</th>
<th>nF</th>
<th>nSD</th>
<th>nASD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>❌</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>☒</td>
<td>✔️</td>
<td>✔️</td>
<td>☒</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

| INDEPENDENCE CONDITIONS     | nA | ✔️ | ✗ | ✔️ | ✗ | ✔️ | ✔️ | ✔️ | ✔️   |
|                            | ✗ | ✗ | ✔️ | ✔️ | ☒ | ✔️ | ✔️ | ✔️ | ✔️   |
|                            | ✔️ | ✗ | ✔️ | ✔️ | ☒ | ☒ | ☒ | ☒ | ☒   |
|                            | ✔️ | ✔️ | ✗ | ✔️ | ✔️ | ✔️ | ✔️ | ✔️ | ✔️   |

| MODIFIED RELATION OF BELIEF | nA | ✔️ | ✔️ | ✔️ | ✗ | ✗ | ✔️ | ✔️ | ✔️   |
|                            | ✔️ | ✔️ | ✔️ | ✗ | ✗ | ✔️ | ✔️ | ✔️ | ✔️   |
|                            | ✔️ | ✔️ | ✔️ | ✗ | ✗ | ✔️ | ✔️ | ✔️ | ✔️   |
|                            | ✔️ | ✔️ | ✔️ | ✔️ | ✗ | ✔️ | ✔️ | ✔️ | ✔️   |

2.5.1.2 Cognitive Complexity
a. Multiple goals. People often have different goals and some of them may be contradictory so trade-offs are often required.

b. Situation complexity. Data modeling suffers when the data warehouse requirements analyst or designer fail to understand the requirements because of complexity of domain.

c. Connectivity. Connectivity under cognitive complexity does not imply cardinality as it used to in structural complexity. Here, connectivity means large number of interrelationships among a limited number of elements in complex problems.

d. Novelty. Novelty refers to situations that are new and unfamiliar to the designer. A new situation may not be complex if an analogue can be found. However, even then the design has to be made either from scratch or by a deft handling of mapping and manipulating from the analogue

2.5.2 Modifiability

Data warehouse collects its massive amount of data from production systems, external systems, internal systems and archived data. The source system’s schema may change with time based on user’s requirements. Such changes must also be incorporated in data warehouse systems. Such structural changes or schema changes in data warehouse systems may result in dimension updates, structural updates, instances updates, facts updates, attributes updates, hierarchy updates, quality updates and constraints updates.
We can assess our conceptual schema of goal structured requirement engineering and traceability model by making a checklist of following features and then analyzing whether the schema supports following features or not.

a. Add a table

b. Delete a table

c. Add a view

d. Delete a view

e. Alter view definition

f. Add an attribute

g. Delete an attribute

h. Change attribute domain

i. Add integrity constraint

j. Delete integrity constraint
k. Insert classification relationship

l. Delete classification relationship

m. Add a new level to the dimension

n. Delete an existing level from the dimension

o. Add a new attribute to given dimension or a given level

p. Delete an attribute from given dimension or a given level

q. Rename table, view, fact, dimension, level, or property

2.5.3 Accessibility

Data warehouse data elements must be well documented and easily accessible to the users so that if interested they can see the origin of data element and also validate their reports. The proposed model on goal structured requirements engineering and traceability model makes data sources clearly documented and traceable. For this, in the star schema, data elements have meaningful names so that their origin is clear to the users reading it. Moreover, the details of all these elements will be stored in the metadata according to a template specified by us in [87].

Here, the Goal is accessibility (Figure 13) and metric is percentage of data elements with their data sources properly documented.

*Figure 13: Data warehouse conceptual model quality template- accessibility*
2.5.4 Data Currency

While collecting requirements for the upcoming data warehouse project, the team must also ask the users how frequently they want the data to be refreshed. As a normal practice data warehouse data must be refreshed at least once in a week for example, during the weekends when data warehouse may not be available for strategic decision making. To measure this, the data warehouse development team must measure the frequency of refreshing the physical tables that are based on our proposed conceptual schema. The team must also measure the time it takes to refresh these tables.

Here, the Goal is data currency (Figure 14) and metric is frequency of data refreshes and the time taken to complete the refresh process.
While designing the conceptual schema of the data warehouse, the designers must give due consideration to the hierarchies that exist in the data and represent them in the star schema designs. Data warehouse users must be able to view summary data as well as detail data, which helps them to deduce how summaries have been calculated and also increases their trust in accuracy and completeness of data. In the proposed goal structured requirements engineering and traceability model, users can get a complete information at the lowest level of detail.

Here, the Goal is believability (Figure 15) and metric is the level of detail at which the data is stored.
Figure 16: Data warehouse conceptual model quality template - believability

2.5.6 Timely

Quality of a data warehouse is also affected by processing time or the time required to run reports. To evaluate the response time, each available report must be processed and the time taken to process that report must be measured and optimized.

Here, the Goal is data timeliness (Figure 16) and metric is time required (in seconds/minutes) to generate requested reports.
2.5.7 Usability

Data warehouse users always appreciate the availability of pre-defined reports that are frequently used by them this eases their work and also reduces the time they have to put in to generate that report.

Here, the Goal is usability (Figure 17) and metric is percentage of percentage of reports that are already available for the data warehouse users.
2.5.8 User Friendliness

User friendliness of a data warehouse in other words means the ease with which it can be accessed by its users. User friendliness is usually measured by number of steps the users perform to produce a particular report. Generally, users don’t want to perform more than five steps to make a report. Here, the Goal is user friendliness (Figure 18) and metric is number of steps the users have to perform to make a report.
2.6 MEASURE PROPOSED FOR QUALITY EVALUATION

In order to assess the proposed conceptual schema of goal structured requirements engineering and traceability model, we can use the Table 5 and get it filled by the data warehouse designers. While finalizing the schema, the designers may have more than one schema, so they can use this table to identify the best schema so that it can be used further with development process.

In this chapter, we have given metrics for eight goals. To use this table, the designers must follow the steps given below.

Step 1: Prioritize the quality goals

Step 2: Assess the schemas based on the proposed metrics. For this, give rating from 0-100 for every individual metric such that 0 is the least and 100 is the maximum rating

Step 3: Normalize the assessment result of top four quality goals so that its value lies in range of 0-15.

Step 4: Normalize the assessment result of next four quality goals so that its value lies in range of 0-10.

Step 5: Add the assessment of all the quality goals to get a final value in the range of 0-100.

Table 8: Assessing the proposed conceptual schema of goal structured requirements engineering and traceability model

<table>
<thead>
<tr>
<th>Quality Goal</th>
<th>Metric</th>
<th>Value</th>
<th>Normalized Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To utilize our quality metrics, let us prove mathematically how we can find out defects in our proposed model. Lesser the number of defects, higher will be the quality.

Suppose we have a data set of N records and M quality goals. An entry in the quality matrix $Q$ denoted by $Q_{i,j}$ will have a value 1 if there is a defect and the value 0 if there is no defect. Thus, the row $q_{i,j}$ depicts how much the $i^{th}$ record is defective with respect to $j^{th}$ attribute.

Accordingly, let the quality of a record be denoted as $Q^R$ and the quality of a particular attribute be denoted as $Q^A$, then

<table>
<thead>
<tr>
<th>Understandability</th>
<th>Structural complexity</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Modifiability</td>
<td>Provision for enhancement</td>
<td></td>
</tr>
<tr>
<td>Accessibility</td>
<td>Percentage of data elements with documented data sources</td>
<td></td>
</tr>
<tr>
<td>Data Currency</td>
<td>Frequency of data refreshes</td>
<td></td>
</tr>
<tr>
<td>Believability</td>
<td>Level of detail</td>
<td></td>
</tr>
<tr>
<td>Timely</td>
<td>Time taken to make a report</td>
<td></td>
</tr>
<tr>
<td>Usability</td>
<td>Percentage of reports already available</td>
<td></td>
</tr>
<tr>
<td>User friendliness</td>
<td>Number of steps performed to make a report</td>
<td></td>
</tr>
</tbody>
</table>

**EVALUATING OUR GOALS BASED ON PROPOSED METRICS**
\[ Q^R = \frac{1}{M} \sum_{j=1..M} q_{i,j} \quad (i) \]

\[ Q^D = \frac{1}{N} \sum_{i=1..N} q_{i,j} \quad (ii) \]

\[ Q = \frac{1}{MN} \sum_{i=1..N} \sum_{j=1..M} q_{i,j} \quad (iii) \]

Substituting (i) in (iii), we get

\[ Q = \frac{1}{N} Q^R \]

Substituting (ii) in (iii), we get

\[ Q = \frac{1}{M} Q^D \]

Using the Quality matrix \( Q \) we can easily calculate Quality\_Defects.

\textit{Quality\_Defects} = \frac{\text{Number of Defects}}{\text{Total Cases}}

We can easily find out both – Number of defects and Total Cases from our Quality Matrix \( Q \).

\textit{Number of Defects} = \sum_{i=1..N} \sum_{j=1..M} q_{i,j} \text{ where } q_{i,j} = 1.

\textit{Total Cases} = N \times M

This ratio is consistent with common structural definitions of quality measures specified by Redman [124]

For example, let us consider some of our Quality Goals to see how we would fill their corresponding entries in the matrix.

\textbf{Data Currency}. To make things clear we have replaced \( Q \) with \( DC \). A 1 is entered in \( DC_{ij} \) if the current data is present and a 0 otherwise.

\textbf{Timely}: A 1 is entered if the required data could be accessed within in the user specified time limits and a 0 otherwise.

\textbf{2.8 SUMMARY}

Data warehouse is an information delivery system that is used by the people at the upper management level to make strategic decisions. Generally data warehouse development teams fail
to focus on the delivery of business value and therefore, lose end user trust and fail to get final product accepted. This is because they take a data-centric approach. In the data centric approach we talk only about data. The data warehouse team usually has a prejudice that if they get the right data in the data warehouse, then the system can meet all possible future requirements of users.

For data warehouse users, data warehouse is just a tool or a product to support them to take relevant business decisions. They know that they have a certain goal that can be accomplished with the help of data warehouse. Therefore, we need a traceability model that extends beyond tracing the relationships between requirement specifications and design. The model must be structured on user’s goals. The proposed model focuses on these issues. The main highlights of this model are:

i. it helps to identify the requirements that would be affected when the designer of the system wants to undo a specific design path

ii. it helps to analyze the impact on existing requirements when a user’s changes the requirements of an ongoing project

iii. it helps to re-use the existing business model because the decisions, justifications and assumptions, if any can be easily understood.

The proposed model is also more useful for development team because IT is a field that is badly hit by influx of personnel as people keep leaving and joining organizations. The model enables a newly joined professional to have a better understanding of the project.

Data modeling plays a vital role in data warehouse development. The quality of these systems can suffer because of poor data modeling practices. So, in order to assure the quality of data warehouse systems, we need to guarantee the quality of the models used in their design. As of now, many
research studies are focused on comparisons of data modeling formalisms, but little has been said on developing techniques to manage complexity and enhance their quality.

We have proposed a set of quality goals and their corresponding metrics to assure quality of the proposed conceptual schema (goal structured requirements engineering and traceability model) used in the early stages of data warehouse design. These metrics have been useful in measuring the understandability of users and designers, modifiability, timeliness, data currency, believability, usability, user friendliness and accessibility.