CHAPTER 4

DESIGN AND DEVELOPMENT OF MANUFACTURING EFFECTIVENESS AND PROCESS IMPROVEMENT MODEL

The author has developed Manufacturing Effectiveness and Process Improvement Model (MEPIM), Figure 4.1 that overcome the drawbacks of PASIM (Edosomwan 1996). The MEPIM is a process improvement approach that requires continuous process assessment and an organized use of information to find easier and better ways of doing work, as well as streamlining the manufacturing activities to ensure that the products are manufactured at minimum cost. In MEPIM, IMS and Data Warehousing are used to provide the appropriate factors for decision-making. The MEPIM improvement strategy focuses on production, wastage, non-value added operation, automation of work, error rate and work repetition. It improves the productivity, employee morale and organizational competitiveness. The steps followed are:

Step 1: Study the current systems

The present organizational procedure and process are studied and the data are used as a foundation for developing plans for process improvement. Key performance factors are identified to provide feedback on performance
Monitor and Control

Multifunctional Team

Product Quality

Capacity Utilization

Environment

Employee Turnover

Organization Competitiveness

Better Co-ordination

Product Quality

Market Complaints

Employee Turnover

Multifunctional Team

Fast feedback procedure

Implementation of Reengineering:
- Combining the duplicate activities
- Eliminate multiple reviews and approvals
- Process in parallel
- Outsource inefficient activities
- Fast feedback procedure

Measure and Evaluate

Productivity

Employee Satisfaction

Organization Competitiveness

Productivity

Employee Satisfaction

Organization Competitiveness

Figure 4.1 MEPIM Conceptual Framework for implementation of BPR
progress. The operational parameters are studied to understand the current system.

Step 2: Develop the new procedure and process

This step focuses on a detailed review of tasks performed by individual, work groups, departments; the methods used and the flow of work from one task to another and among the members of the work group to develop new system and process to improve their effectiveness. It is also important to identify the volume of work, frequency and time involved in the performance of each task. The teamwork approach should be implemented to minimize resistance to change. Use managerial judgment along with facts and factors to assess the degree of tangible and intangible benefits.

Step 3: Analyze

Analyze current operation to determine the magnitude of problem identified. The feasibility and suitability of all parameters should be determined based on cost, value added, time and quality. Other factors are work safety, job satisfaction and employee morale. Analyze each task for its improvement. Five approaches are recommended as depicted in MEPIM conceptual framework for operation improvement by reengineering the process. They are:

- Combining the duplicate activities
- Eliminates multiple reviews and approvals
- Process in parallel
- Outsource inefficient activities
- Fast feedback procedure
The process improvement efforts follow a structured improvement methodology. This methodology requires data to establish measures for all process components and assess improvements. The improvement methodology is a cyclic and infinite process. The IMS and Data warehousing are used in this model to identify and prioritize new opportunities for process improvement.

Step 4: Measure and Evaluate

Measurement is an essential element of transformation and continuous improvement processes. It focuses on the effectiveness of improvement efforts. A basic need in all improvement efforts is the ability to measure the value of the improvement in units that are meaningful to the specific task.

Step 5: Monitor and Control

Follow up the issues and focus on a continuous improvement in all sectors of the manufacturing environment such as production, efficiency, inventory, machine down time, maintenance expenses, water consumption, energy, capacity utilization, product quality, market complaints, employee morale, better co-ordination and building-up multifunctional team.

The above process must be a permanent fixture in the organization. Approaches to positive transformation for continuous improvement that have limited lifetimes will become ineffective if left unattended. Review progress with respect to improvement efforts and modify existing approaches for the next progression of methods. This constant evolution reinforces the idea that continuous improvement through organizational transformation and
reengineering is not a program but rather is a new expectation for day-to-day behaviour and a way of life for each member of the organization.

4.1 DEVELOPMENT OF INTEGRATED MANAGEMENT SYSTEM

Every information has an economic value. Organization are willing to expend large sums to acquire it either through their own processing system or by purchase from an outside agency such as market surveyors conducted by an outside research firm. The value of decision making of information may be due to one or more of the following:

- **Availability**: Information was not previously available
- **Timeliness**: Obtained more quickly
- **Accuracy**: More accurate than previous
- **Completeness**: More Comprehensive & Complete
- **Presentation**: Presented in a manner more meaningful to the decision to be made

The last point is important to keep in mind. The problems are not in too little information, in its timeliness, in its completeness or in its accuracy. Rather it is the lack of relevant information presented such that the decision maker is unable to effectively use it. Information is viewed as a resource much like land, labour and capital. It is not a free good. It must be obtained, stored, retrieved, manipulated and analysed. An organization with a well designed information system, Figure 4.2, generally have a complete advantage over organization with poorer system.
Figure 4.2 IMS for Manufacturing Industries
4.1.1 Characteristics of IMS

Integrated Management System is the concept to deal with, and it can be viewed and analysed from many sides. The main characteristics of IMS are

- Management oriented
- Management directed
- Integrated
- Common data flows
- Heavy planning element
- Sub-system concept
- Data base

Management Oriented: This is the most significant characteristics of IMS. The system is designed top down approach. A marketing information system is an example. Basic sales order planning, shipment of goods to customers and the billing are fundamental operational control activities. However if the system is designed properly, the transaction information can be tracked by salesman, sales territory size and order, geography and product line.

Further more, if designed with integrated management needs in mind, external competition, market and economic data can be created to give a picture of how well the company's products are faring in the marketing environment and serve as a basis of new product or market place introduction.

Management Directed: Because of the management orientation of IMS, it is imperative that management actively direct the system development.
IMS process demands not a one time involvement from the management. A continued review and participation are necessary to ensure that implemented system meets the specifications that were designed. Therefore, management is responsible for setting system specifications and it must play a major role in the subsequent trade.

An important element of effective system planning is the process for determining the priority of application development. Management must control this process if an IMS is the objective. A company without a formal application approved cycle and a management steering committee to determine the priorities will never develop an IMS.

Integrated: Once the management is fully involved to use the information system that should be integrated to provide shared information. For example a Data entered in store section should reflect in production, inventory, despatch and accounts.

Common Data Flows: Because of the integration concept, there is an opportunity to avoid duplication and redundancy in data gathering, storage and dissemination. For example, customers orders are the basis for

- Billing the customer
- Setting up of the production
- Account Receivable
- Sales analysis
- Inventory and Godown operations.
Heavy Planning Element: IMS is not developed over night. Normally it takes from 3 to 5 years and might even take longer to stabilize within a company. Therefore a heavy planning element must be present in IMS Development. The IMS Designer must have the future objectives and needs of the company firmly in mind. The system designs must avoid system obsolesce before the system gets into operation. A phasing plan is an essential ingredient to IMS planning.

Sub System Concept: It is not possible to implement IMS as a single entity. It must be broken into digestible subsystems that can be implemented one at a time. The subsystem analysis is essential for applying boundaries to the problem thus enabling the design to focus on manageable entities that can be assigned and computerized by selected systems and programming teams.

Database: The database is the mortar that keeps the functional systems together. There are however many instances where different set of files that exist for different functional systems. To a possible extent this should be avoided. For example the supplier data should be available for both purchase and accounts.

4.1.2 Computer Based System

All industries are maintaining their data manually in ledgers and registers. If the volume of data increases, it is very difficult to get information in time. Hence the computer based system is in need to manage such a the situation. The computer based systems approach to managing has been brought about by the increasing complexity of business. This complexity is due to greater size (Number of employees), more complicated forms of organization
(Conglomerates), environmental restraints (legal, social) and accelerating change in technical and information factors, especially technological revolution and product changes.

In the last 30 years every industry has undergone vast changes in products, techniques output and productivity (e.g.) electronics industry, communications. Hence it is clearly that changes will continue at an accelerated pace and this change will demand in improved management. The manager of today must keep abreast of the factors influencing his products and future operations. This requirement demonstrates the need for on property designed IMS. Particularly with regard, an environment that includes competitors who are themselves using up-to-date methods.

As a decision makers, the modern manager knows that the ability to obtain, store, process, retrieve and display the rights information for the right decision is ultimate. This is the reason for an information system - better decisions.

The new techniques or developments that have become available in modern management are:

Information Feedback systems: These systems are concerned with the way information is used for the purpose of control. The basic trait of such a system is the output of the system leads to another decision.

Decision making: The central aspect of managing is the decision making. There are two types of decision making, (i) Programmed and (ii) Non programmed decision. Programmed decisions are of use because same
problems send themselves automated decisions using same techniques such as LP or specific programming.

**Non programmed Decisions:** Some decisions are not able to programme to the extent that they are unstructured, new or high consequence or complex or involve major commitments. Advertising budgets, new products decisions, acquisition or merger considerations are few examples of non programmed decisions.

**Management science techniques:** Management science techniques normally use the programmed decisions. LP, system simulation, game theory, probability theory and other quantitative techniques are available to the modern manager.

**Purpose of IMS:** The purpose of IMS is to raise managing from the level of piecemeal information, initiative guess work and isolated problem solving to the level of system insights, systems information, sophisticated data processing and problem solving. Managers always have sources of information but IMS provides a ‘system of information’ and aids in solving problems and making decisions.

The main information flow in a manufacturing unit is shown from Figure 4.3 to Figure 4.12 (Appendix). Once the information systems are identified such as the above then they can be integrated for data analysis.

IMS is implemented at M/s Balaji Distilleries Ltd., and data regarding all the operations are captured. The various sub-systems of the information
system are tabulated below in Table 4.1. This clearly indicates that the data capture is done at almost every point of operation in the manufacturing unit.

**Table 4.1 Various sub-system of IMS**

<table>
<thead>
<tr>
<th>System</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production</td>
<td>Line Production</td>
</tr>
<tr>
<td></td>
<td>Dispatch Details</td>
</tr>
<tr>
<td></td>
<td>Materials Planning</td>
</tr>
<tr>
<td></td>
<td>Inventory Control</td>
</tr>
<tr>
<td></td>
<td>Wastage Control</td>
</tr>
<tr>
<td></td>
<td>Maintenance System</td>
</tr>
<tr>
<td></td>
<td>Energy Consumption</td>
</tr>
<tr>
<td>Finance</td>
<td>Profit Planning</td>
</tr>
<tr>
<td></td>
<td>Capital Budgeting</td>
</tr>
<tr>
<td></td>
<td>Costing</td>
</tr>
<tr>
<td></td>
<td>Ledger Maintenance</td>
</tr>
<tr>
<td></td>
<td>Accounts Payable/Receivables</td>
</tr>
<tr>
<td></td>
<td>Billing</td>
</tr>
<tr>
<td>Personnel Management</td>
<td>Payroll</td>
</tr>
<tr>
<td></td>
<td>Personal records</td>
</tr>
<tr>
<td></td>
<td>Personal administration</td>
</tr>
<tr>
<td>Sales</td>
<td>Sales Record</td>
</tr>
<tr>
<td></td>
<td>Sales Forecast</td>
</tr>
</tbody>
</table>
4.1.3 Data Dictionary

The IMS at M/s Balaji Distilleries Ltd comprises of a data base and 120 tables in the VB-6 and MS SQL Environment that capture the information flow in the organization from the various functional areas discussed above. The distribution of operational databases based on functional area is given in Table 4.2. An extensive data dictionary is prepared to identify the location of the data required for the data warehouse so as to extract them into the data warehouse databases to be designed.

Table 4.2 Operational Databases in production

<table>
<thead>
<tr>
<th>Functional Area</th>
<th>Name of the Data base</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production</td>
<td>ELECCNSM.DBF</td>
</tr>
<tr>
<td></td>
<td>(Power Consumption)</td>
</tr>
<tr>
<td></td>
<td>ELECTRAN.DBF</td>
</tr>
<tr>
<td></td>
<td>(Fuel Oil Consumption)</td>
</tr>
<tr>
<td></td>
<td>LNPROD.DBF</td>
</tr>
<tr>
<td></td>
<td>(Bottling Operation)</td>
</tr>
<tr>
<td></td>
<td>LNBRTR.DBF</td>
</tr>
<tr>
<td></td>
<td>(Machine Maintenance)</td>
</tr>
<tr>
<td></td>
<td>PMMMAST.DBF</td>
</tr>
<tr>
<td></td>
<td>(Machinery Master)</td>
</tr>
<tr>
<td></td>
<td>ENGHIST.DBF</td>
</tr>
<tr>
<td></td>
<td>(Spare parts Consumption)</td>
</tr>
</tbody>
</table>
4.1.4 Reports Generated

The main characteristic of a IMS is reports generation on periodic transactions in the business. The Table 4.3 gives a brief list of various reports generated in the production area by the information system and the data sources used for these reports.

Table 4.3 Reports generated and Databases utilized

<table>
<thead>
<tr>
<th>Report</th>
<th>DataBases Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bottling performance</td>
<td>LNPROD.DBF</td>
</tr>
<tr>
<td>Day/Month/Year/Period/Line No</td>
<td>LNRTRTR.DBF</td>
</tr>
<tr>
<td>Power Consumption Area wise</td>
<td>ELECCNSM.DBF</td>
</tr>
<tr>
<td>Fuel/Power factor</td>
<td>ELECTRAN.DBF</td>
</tr>
<tr>
<td>Energy Consumption</td>
<td>ELECTRAN.DBF</td>
</tr>
<tr>
<td>Machinery wise Spare parts Consumption</td>
<td>ENGHIST.DBF</td>
</tr>
</tbody>
</table>

4.2 IMPLEMENTATION OF DATA WAREHOUSE

4.2.1 The need for Data Warehousing

Everyday, organizations’ systems create countless data bytes that describe aspects of their business and produce information about customers, products, processes and their personnel. However, this gold mine is usually not utilized by companies.
Acquiring information about business by means of quantitative measures is vital for any organization. Aligning the organization around strategic directions, establishing core values and visions, identifying improvement potentials, motivating individuals and teams, creating organizational learning and improving decisions through fact-based management are some of the general functions a measurement system should carry out. Organizations collect data in the normal course of business operations. The purpose of a data warehouse is to consolidate and organize this data so it can be analyzed and used to support business decisions. In many cases a data warehouse contains the living history of the organization.

Data warehouses usually contain historical data, often collected from a variety of disparate sources such as OLTP systems, legacy systems, text files, or spreadsheets.

The two main technical goals that should be kept as the base during the design and implementation of a data warehouse are given as follows:

**To provide easy access to corporate data**

**The end user access tools:** It should live on the PC in an environment familiar to the end user. They must have a short learning curve and be highly intuitive. If it has high level managers asking questions of the data warehouse, they must have a simple way to interact with it.

**The access should be graphic:** People have been using Microsoft Excel. Spotting trends in a columnar report listing 500 customers’ sales for the
quarter is a very tedious process. However, if you take the same list and chart it graphically, it is easy for the human eye to spot the winners and losers.

The access should be manageable by end users: MIS is no longer the end user of the tools that access the data. End users are analysts, mid-level managers, even in some cases, high level managers. They must be able to easily get answers to their questions, and ask new questions, all without getting MIS involved.

The process must be fast. Questions leapfrog, one should get answers fast. The very nature of data analysis is that not all requests are known beforehand, so there's a lot of unpredictable, ad hoc inquiry. In an analytical environment, the end user is directly interacting with the data, rather than looking at a printed report. The answers have to be delivered fast, before users lose their train of thought.

To provide data for analysis that is clean and reliable

For consistent analysis, the environment must be stable: Now that users can create their own reports, they must have consistent data. One department doing an analysis must get the same results as any other. Two managers can't show up at a meeting with different production figures for previous month.

Source conflicts must be resolved: Transactional (production) systems often have data stored in several different applications. Customer addresses, for example, may be in the invoicing system, as well as the sales order system. State Bank of India may be stored as “SBI” in one system and
“State Bank of India” in another. A straight load of data for this customer into the data warehouse would result in two customers being represented – and with invoices being listed with no supporting sales orders and vice versa. These conflicts in the source data need to be resolved and merged as they move into the analytical environment.

**Historical analysis must be possible:** There are two aspects to historical data that must be considered. The amount of historical data in the warehouses has to be sufficient to satisfy the need to spot trends over time. To see the growth in sales of a product over 3 years, perhaps to help predict sales of future products, you need three years of data. The second historical consideration is the ability to see the state of affairs at a historical point in time. Take, for example, a situation where we want to see what our inventory amounts have been over the last year to help determine if we can carry fewer inventories and still satisfy demand. We may need to be able to see what our inventory amount was for product X on the 3rd, 4th, 5th of July. As data is added to the data warehouse, it needs to be added and date stamped – not updated like it would be in the transactional inventory system.

**Performance of operational systems must not be impaired:** The Data warehouse will run on a separate and dedicated system to avoid conflicts with the performance needs of the operational systems. Extracting production data to populate the data warehouse will have some impact on the operational system. Consideration must be given to whether full loads of data are practical, or whether incremental loads are needed, using only changes since the last load.
4.2.2 Benefits of Data Warehousing

Data Warehouse gives substantial advantages in the quest for success. Among these, advantages are:

- Consolidation of business information that originates from inside or outside the boundaries of the organization. This makes it possible to conduct powerful analyses by combining new information with information that was dispersed earlier.
- Efficient management of data and information. Data is managed in one place, which greatly facilitates administration and distribution of business information.
- By combining Data Warehouse with user-friendly client tools, business information becomes readily available in a customized format for decision-makers on all levels within the organization.

4.2.3 Key characteristics to evaluate a Data warehouse

Subject oriented: Does the data warehouse apply to the subjects the business is most interested in? Does the data warehouse have the data elements needed to analyze the business and is the data organized in a manner that is useable to solve real-world business problems? Also, are the proper data elements available for each subject area such as production, finance distribution, inventory.

Integrated: Can the data be viewed within and across subject areas? Any functional data warehouse (or data mart or OLAP) must navigate in at least
three directions (sometimes called dimensions): geography (sales territories, market sectors, cost centers, production areas, and distribution channels), objects (parts, products, shipping units, sub assemblies, accounts, service types, etc.) and time (hours, days, weeks, years). The ability to navigate within a subject area is obviously critical, but navigation across subject areas is also important for understanding complex business relationships.

**Time invariant:** The ability to measure incremental improvement (or the lack of it) over time is the single most important function of a data warehouse. If we are making poor decisions over time, without visibility into the impact of those decisions, we will continue to repeat our mistakes, costing our organizations significantly. But if we have visibility into the data, we can see the results of successful best practices, which can be expanded to improve overall organizational performance, and the results of poor practices also can be evaluated and corrected.

**Non-volatile data:** The data has to be accurate and insulated from the turbulence of the transaction-based systems. Transaction-based, online transaction processing systems are volatile by their very nature. OLTP - based systems run the business – sales orders come in and are processed by the order entry system; the order entry system, in turn, generates information to have the product manufactured or pulled out from inventory.

The finished product is shipped, the sale is recorded in the general ledger and information is generated all along the way. It is this generated information that serves as the source data that incrementally (hourly, daily, weekly, etc.) feeds the data warehousing process. It is the incremental nature of the data warehouse process that creates non-volatile environments for analysis.
4.2.4 Data Warehousing and OLTP

A relational database is designed for a specific purpose. As the purpose of a data warehouse differs from that of an OLTP, the design characteristics of a relational database that supports a data warehouse differ from the design characteristics of an OLTP database.

A data warehouse supports an OLTP system by providing a place for the OLTP database to offload data as it accumulates, and by providing services that would complicate and degrade OLTP operations if they were performed in the OLTP database.

Without a data warehouse to hold historical information, data is archived to static media such as magnetic tape, or allowed to accumulate in the OLTP database. If data is simply archived for preservation, it is not available or organized for use by analysts and decisions makers. If data is allowed to accumulate in the OLTP so that it can be used for analysis, the OLTP database continues to grow in size and requires more indices to service analytical and report queries. These queries access and process large portions of the continually growing historical data and add a substantial load to the database. The comparison of OLTP and Warehouse database is given in the Table 4.4.
Table 4.4 Difference between Data Warehouse Database and OLTP Database

<table>
<thead>
<tr>
<th>Data Warehouse database</th>
<th>OLTP database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designed for analysis of business measures by categories and attributes</td>
<td>Designed for real-time business operations</td>
</tr>
<tr>
<td>Optimized for bulk loads and large, complex, unpredictable queries that access many rows per table</td>
<td>Optimized for a common set of transactions, usually adding or retrieving a single row at a time per table</td>
</tr>
<tr>
<td>Loaded with consistent, valid data; requires no real time validation</td>
<td>Optimized for validation of incoming data in transactions; uses validation data tables</td>
</tr>
<tr>
<td>Supports few concurrent users</td>
<td>Supports thousands of concurrent users</td>
</tr>
</tbody>
</table>

4.2.5 Technical Architecture for Data Warehousing

The Figure 4.13 shows various components of technical architecture of Data Warehouse.

- Design component, for designing warehouse databases.
- Data acquisition component, for capturing data from source files and databases, and for cleaning, transporting, and applying it to data warehouse databases
- Data manager component, for creating, managing, and accessing warehouse data
- Management component, for administering data warehouse operations
Information directory component, for providing administrators and business users with information about the contents and meaning of data stored in warehouse databases.

- Data access component, for providing business end users with the tools they need for accessing and analyzing warehouse data.
- Middleware component, for providing end-user tools with access to warehouse databases.
- Data delivery component, for distributing warehouse data to other warehouses and external systems.

Figure 4.13 Database Associates Technical Architecture for Data Warehousing System (Colin White 1997)
4.3 CREATING THE DATA WAREHOUSE DATA MODEL

4.3.1 Life Cycle for Designing a Data Warehouse

The life cycle for designing a data warehouse for decision support generally includes the following tasks:

- Determine business areas to be supported in the first phase of development.
- Determine what legacy and transaction systems are associated with the selected business areas.
- Document the associated legacy and transaction systems.
- Determine overlap and relationships between legacy and transaction systems.
- Specify queries for decision support.
- Specify data mapping among legacy, transaction and data warehouse databases.
- Design the data warehouse database.
- Specify required data transformations between legacy, transaction and data warehouse databases.
- Design management system for mapping data and making the required data transformations.

The life cycle of the typical data warehouse is shown in Figure 4.14. In a traditional environment, requirements are gathered, the application and database are designed, programs are coded, tested, and moved into production. Although the system may be modified after it is placed in production, requirements are (or should be) well understood up front. In sharp contrast, the
design of the data warehouse is based on a set of general requirements that may identify the subject area and provide few hints on the expected usage of the data. One or more subject databases will be designed using the enterprise data model as a foundation. If processors were infinitely fast and database management systems were perfectly efficient, the design could stop there. It would be possible to answer every query by accessing the databases containing data in detail. There would be no need for summary tables, multi-dimensional databases, or specialized subsets of data. Unfortunately, warehouses that have been implemented with this approach have usually failed due to poor performance.

The real world has not yet produced infinitely fast processors. Physical design must consider performance. It is important that the design of the detail data be correct before adding summary levels and other types of structures to the data warehouse. The warehouse may be accessed through specialized decision support applications, ad hoc queries, or both. The warehouse is usually implemented, tested by users (who provide feedback to developers on their level of satisfaction), and optimized for performance. There will be frequent modifications to meet constantly changing user requirements. Depending on feedback from users, there will be a need to recycle design, coding, and performance optimization in an iterative procedure. Requirements are constantly changing, and meeting them is like trying to hit a moving target.

4.3.2 Data Mapping and Transformation Requirements

The data included in the legacy and transaction systems have different physical characteristics associated with them. This may be named differently,
Figure 4.14: Data Warehouse Life Cycle (Denis Kosar, 1997)
have a different length, or may even be a different type altogether. When this is the case, the data must be transformed before it is brought into the data warehouse.

Data transformation is also required when two or more columns are merged from transaction systems into a single column in the data warehouse. For example, separate date and time fields in transaction systems may be joined in a single date and time field in the data warehouse. Even if the data doesn't require transformation, the relationships must be specified between tables and columns in the transaction and legacy systems and tables and columns in data warehouse and data marts, so that the data warehouse and data marts can be updated from the transaction systems.

The information required for this mapping is distributed throughout the previous phases of design process:

- Each column in the data mart is derived from a column in the data warehouse.
- Each column in the data warehouse is derived from one or more columns in one or more tables in one or more legacy or transaction bases. By tracing the origins of a warehouse data, we can determine the necessary mapping and transformations for data. This information is then entered into the Data Warehouse Management System.
4.3.3 Data extraction and populate methods

The main extract methods used are the following:

- **Full copy** unloads the entire file or table. A bulk export or copy command is often the most efficient means. Automated tools may use SQL.

- **Selective Extraction** generally uses SQL either directly against the source data or via a database gateway or transparency layer to create a subject of the source data. For some data sources particular proprietary data manipulation languages or export specification are more advantageous.

- **Net Changes (selective)** is a variant of selective extraction where the criteria are turned to extract records.

- **Net Changes (log)** determines net changes by reading the log file of the source DBMS or main file system instead of accessing the source data directly.

- **Net Changes (bucket)** accumulates net changes in a bucket or hold file under the control of the source or producer applications – no external extract process is used with this method.

- **Triggered Replication** is a record or transaction level process that determines new or updated data automatically when a new event occurs.
The main methods to populate Data Warehouse are:

- **Refresh** completely reloads the target warehouse table each time. This can be done by either truncating (emptying) the table or dropping and then creating the table. The loading process may either use SQL, a bulk load utility, or proprietary commands.

- **Logical Update** appends rows to the table and additionally supports logical update via effective dates. Use of a single as of date is the norm, but begin-date and end-date pair is occasionally used for reference data.

- **Physical Update** issues an actual update command to modify values in an existing row or to do a physical update and should never be used on base tables in the data warehouse. This destroys information by modifying history data.

- **Refresh operations** are by definition set-oriented. Append and logical update operations can be either record-oriented or set oriented.

- A **set-oriented approach** may use either a bulk load utility or proprietary commands. A record-oriented approach is generally not recommended except in conjunction with streaming transport.
4.3.4 Data Integration

Data from different DBMS (Data base management system), external information providers, and various standard applications can be obtained from the Data warehouse for analysis.

Tasks:

- Extraction (accessing different databases)
- Cleaning (resolving inconsistencies)
- Transformation (different formats, languages)
- Replication (importing a whole DB)
- Analyzing (detecting invalid values)
- Checking for data quality (correctness, completeness)
- Update metadata, if necessary

4.3.5 Data Uploading activities

To populate a data warehouse, regular loading or propagation of data from operational sources will be needed. A schedule for summarizing, loading, and marking the information available to the user community should be developed and presented to the user community. For example, weekly summary data may be available by 10 a.m. Monday morning (or the first business day following a holiday), and monthly data will be available on the second business day of the following month. In addition, it is important for users to know and when the data was actually loaded. If problems occur in the production environment, the load that should have occurred by 10 A.M. on Monday may not actually have occurred until 3 P.M. on Monday. Data
regarding actual loads may be made available through metadata in an end-user directory.

It is also necessary to develop procedures for managing the results of a bad load. What if the data loaded by 10 A.M. on Monday is corrupted due to an operational error and the erroneous data is not discovered until 3 P.M.? How will it be reloaded? What will be the impact of reloading during peak Hours? User notification regarding corrupted data should be part of the procedure.

Loading substantial amounts of data into the warehouse may require a lot of time. Loads may need to be scheduled for periods in which a sufficient batch window exists if daytime availability is a priority. The amount of data in a data warehouse of moderate size normally favors the use of utilities for loading rather than techniques that simply insert data. These utilities, whether supplied as part of the DBMS or purchased as additional products, provide the benefit of speed, which is often achieved by turning off database logging during the load process.

4.4 DATA STRUCTURES IN THE WAREHOUSE

Data Warehouse consists of any combination of the following types of data: detail data, summary data, external data, multidimensional data, data subsets, specialized data caches, replicated data, and archived data, Figure 4.15. The database designer must determine the types of data structures needed to meet design objectives. Physical design in the warehouse environment is an iterative process. The database designer will make a first cut at design based on expected data usage, which may not be well defined. The design may require
modification to achieve desired performance when actual usage patterns become more apparent. The data flow shown in Figure 4.16 contains the basic structures that may be used in the construction of a data mart or data warehouse.

Figure 4.15 Data Warehouse Architecture (Joyce Bishoff 1997)

4.5 DATA WAREHOUSE DEVELOPMENT METHODOLOGIES

There are three types of data warehouse development methodologies which are tabulated below of which data driven is simple in nature, Table 4.5

Data Warehouse design process steps are:

- Identify requirements and project scope.
- If not already available, develop subject area data model, including an entity relationship diagram and associated metadata. A subject area usually covers a particular aspect of
the business: for example, sales information, financial information, or customer information.

- Develop a data warehouse logical data model from the subject area data model. Although the logical data model should be as complete as possible, a decision may be made to implement only part of the model. This will be represented in the data warehouse data model.

- Develop Data warehouse architecture if a general warehouse architecture has not already been developed.

- Design the physical database.

- Populate an end-user-oriented repository/directory with metadata for the physical database.
### Table 4.5 Comparison of Data Warehouse development methodologies

<table>
<thead>
<tr>
<th>Methodology Criteria</th>
<th>Data Driven</th>
<th>User-Driven</th>
<th>Goal-Driven</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic approach</td>
<td>Bottom-up</td>
<td>Bottom-up</td>
<td>Top-Down</td>
</tr>
<tr>
<td>Supported Management Method</td>
<td>Taylorism Classical School of Management</td>
<td>None company is Culture Reflected</td>
<td>Management By Objectives</td>
</tr>
<tr>
<td>Project Support</td>
<td>None</td>
<td>Department</td>
<td>Top-Management</td>
</tr>
<tr>
<td>Application Area / Requirement Domain</td>
<td>Data Exploration and Data Mining</td>
<td>Raise the Acceptance of a system</td>
<td>Foundation for Decision Support</td>
</tr>
<tr>
<td>Targeting Organizational Level</td>
<td>Operational Partly Technical</td>
<td>Depends on the Group of Interview Partners</td>
<td>Strategic Tactical Operational</td>
</tr>
<tr>
<td>Focus</td>
<td>Short-term Focus</td>
<td>Short – term Focus</td>
<td>Long – term Focus</td>
</tr>
<tr>
<td>Extent to End User Involvement</td>
<td>None</td>
<td>High</td>
<td>Moderate</td>
</tr>
<tr>
<td>Project Duration</td>
<td>Low</td>
<td>Very High</td>
<td>High</td>
</tr>
<tr>
<td>Skills of project Members</td>
<td>Data Warehouse Designer</td>
<td>Moderator Data Warehouse Designer</td>
<td>Moderator Economist Data Warehouse Designer</td>
</tr>
<tr>
<td>Number of Measures</td>
<td>Many</td>
<td>Many</td>
<td>Few</td>
</tr>
<tr>
<td>Type of Measures</td>
<td>Non-Financial and Quantitative Time-Based and Frequently-Based</td>
<td>Non-Financial and Quantitative Time-Based and Frequently-Based</td>
<td>Balanced: Financial and Non-Financial as well as Qualitative and Quantitative</td>
</tr>
<tr>
<td>Level of Granularity</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Number of Dimensions</td>
<td>Few</td>
<td>Many</td>
<td>Few</td>
</tr>
<tr>
<td>Type of Dimensions</td>
<td>Represents the Basic Structure of the Application</td>
<td>Represents the Basic Structure of the Application and External Sources</td>
<td>Represents the Strategic Building of the Organization</td>
</tr>
<tr>
<td>Number of Source Systems</td>
<td>Low</td>
<td>Moderate</td>
<td>High</td>
</tr>
<tr>
<td>Longevity / Stability of Data Model</td>
<td>Long</td>
<td>Short</td>
<td>Long</td>
</tr>
<tr>
<td>Cost</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>
• Identify the source of data from operational systems or external sources for each target field in the data warehouse.

• Develop or purchase programs to extract, cleanse, transform, integrate, and transport data from the legacy systems to the warehouse. Populate the warehouse using these programs.

• Test for user satisfaction with the data warehouse, including data quality, availability, and performance.

• Rework the design as needed to achieve ongoing user satisfaction.

4.5.1 Requirements Gathering

Task Description: The first thing the developer should engage in is gathering requirements from end users. Because end users are typically not familiar with the data warehousing process or concept, the help of the business sponsor is essential. Requirement gathering can happen as one-to-one meetings or as Joint Application Development sessions, where persons with diverse backgrounds are talking about the project scope in the same meeting. The participants at a joint application development session at Balaji Distilleries Ltd were Manager-Engineering, Systems Manager, Production executives and the developer. This session was conducted for extract requirements capture.

The primary goal of this phase is to identify what constitutes as a success for this particular phase of the data warehouse project. In particular, end user reporting / analysis requirements will be identified, and the project team will spend the remaining period of time trying to satisfy these requirements.
Associated with the identification of user requirements is a more concrete definition of other details such as hardware sizing information, training requirements, data source identification, and most importantly, a concrete project plan indicating the finishing date of the data warehousing project.

**Deliverables**

- A list of reports / cubes to be delivered to the end users by the end of this current phase.
- A updated project plan that clearly identifies resources loads and milestone delivery dates.

### 4.5.2 Data Modeling

**Task Description:** Once the requirements gathering comes to an end the next task of data modeling starts in the project cycle. As it is difficult to talk about data modeling without going into some technical terms, let us have a look at several terms that are used commonly in the data modeling field:

- **Dimension:** A category of information. For example, the time dimensions.
- **Attribute:** A unique level within a dimension. For example, Month is an attribute in the Time Dimension.
- **Hierarchy:** The specification of levels that represents relationship between different attributes within a hierarchy. For example, one
possible hierarchy in the Time dimension is Year— > Quarter— > Month— > Day.

- **Fact Table**: A Fact table is a table that contains the measures of interest. For example, production quantity in number of cases would be such a measure. This measure is stored in the fact table with the appropriate granularity. For example, it can be production quantity in number of cases by Line number by month. In this case, the fact table would contain three columns: A month column, a line number column, and a production quantity in number of cases column.

- **Lookup Table**: The lookup table provides the detailed information about the attributes. For example, the lookup table for the Quarter attribute would include a list of all of the quarters available in the data warehouse. Each row (each quarter) may have several fields, one for the unique ID that identifies the quarter, and one or more additional fields that specifies how that particular quarter is represented on a report (for example, first quarter of 2001 may be represented as “Q1 2001” or “2001 Q1”).

In designing data models for data warehouses the most commonly used schema types are **Star Schema**, **Snowflake Schema**, and **Federated Star Schema**. The data modeling technique used in this project is star schema as it is simpler to construct and interpret. The star schema gives the analysis dimension for the data.

**Star Schema**: In the Star Schema design, a single object (the fact table) sits in the middle and is radially connected to other surrounding objects
A star schema can be simple or complex. A simple star consists of one fact table; a complex star can have more than one fact table.

**Deliverables**

- Identification of data sources.
- Logical data model.
- Physical data model.

### 4.5.3 Extraction, Transformation and Loading

**Task Description:** The ETL (Extraction, Transformation, and Loading) process typically takes the longest to develop, and this will easily take up to 50% of the data warehouse implementation cycle or longer. The reason for this is that it will take time to get the source data, understand the necessary columns, understand the business rules, and understand the logical and physical data models. The ETL in this project at Balaji distilleries Ltd will be carried out by means of sub-routines developed for that specific purpose. The ETL will bring the data needed for analysis from the operational databases to the data warehouse.

**Deliverables**

- Data Mapping Document
- ETL Script / ETL Package in the ETL tool
4.5.4 OLAP Cube/Reports Design

Task Description: It was the design of the output format in which the results of the analysis was required by the company. In this project the company needs the output in the form of graphs as the resultant of pivotal analysis.

Usually the design of the OLAP cube can be derived from the Requirement Gathering phase. More often than not, however, users have some idea on what they want, but it is difficult for them to specify the exact report / analysis they want to see. When this is the case, it is usually a good idea to include enough information so that they feel like they have gained something through the data warehouse, but not so much that it stretches the data warehouse scope by a mile. Remember that data warehousing is an iterative process – no one can ever meet all the requirements all at once.

Deliverables

- Documentation specifying the OLAP cubes dimensions and measures, Figure 4.17.
- Actual OLAP cube / report.

4.5.5 Front End Development

Task Description: Regardless of the strength of the OLAP engine and the integrity of the data, if the users cannot visualize the reports, the data warehouse brings zero value to them. Hence front end development is an important part of a data warehousing initiative.
When choosing vendor tools, make sure it can be easily customized to suit the enterprise, especially the possible changes to the reporting requirements of the enterprise. Possible changes include not just the difference in report layout and report content, but also include possible changes in the back-end structure. For example, if the enterprise decides to change from Solaris/Oracle to Microsoft 2000/SQL Server, will the front-end tool be flexible enough to adjust to the changes without much modification? Another area to be concerned with is the complexity of the reporting tool. For example, do the reports need to be published on a regular interval? Are there very specific formatting requirements? Is there a need for a GUI interface so that each user can customize his reports?
Deliverables

- Front End Deployment Documentation

4.5.6 Performance Tuning

Task Description: There are three major areas where a data warehousing system can use a little performance tuning.

- **ETL:** Given that the data load is usually a very time-consuming process (and hence they are typically delegated to a nightly load job) and that data warehousing-related batch jobs are typically of lower priority, that means that the window for data loading is not very long. A data warehousing system that has its ETL process finishing right on-time is going to have a lot of problems simply because often the jobs do not get started on-time due to factors that is beyond the control of the data warehousing team. As a result, it is always an excellent idea for the data warehousing group to tune the ETL process as much as possible.

- **Query Processing:** Sometimes in a system where the reports are run directly against the relationship database, query performance can be an issue. A study has shown that users typically lose interest after 30 seconds of waiting for a report to return.

- **Report Delivery:** It is also possible that end users are experiencing significant delays in receiving their reports due to
factors other than the query performance. For example, network traffic, server setup, and even the way that the front-end was built sometimes play significant roles. It is important for the data warehouse team to look into these areas for performance tuning.

4.5.7 Rollout to Production

Task Description: Once the performance tuning is done, it is time for the data warehouse system to go live. Developing on the number of end users, it sometimes takes up to a full week to bring everyone on line. The roll out to production will be done on a pilot projet basis in the initial stages at the company.

Deliverables

- Delivery of the data warehousing system to the end users.

4.5.8 Production Maintenance

Task Description: Once the data warehouse goes into production, it needs to be maintained. Tasks such as regular backup and crisis management becomes important and should be planned out. In addition, it is very important to consistently monitor end user usage. This serves two purposes

- To capture any runaway requests so that they can be before slowing the entire system down.
To understand how much users are utilizing the data warehouse for return-on-investment calculations and enhancement considerations.

**Deliverables**

- Consistent availability of the data warehousing system to the end users.

### 4.5.9 Incremental Enhancement

**Task Description:** Once the data warehousing system goes live, there are often needs for incremental enhancements. These could be simply small changes that follow the business itself. For example, the original number of production lines may be different, the company may originally have 10 production lines in operation, but now because sales are going so well, they need 15 production lines.

**Deliverables**

- Change management documentation
- Actual change to the data warehousing system

The tasks that were explained briefly above should be undertaken for the successful implementation of the project. The above discussed methodology will be implemented step by step for the analysis and design of the data
warehouse. The methodology discussed so far is data driven methodology for the development of the data warehouse.

4.6 DATA ANALYSIS

Today, most data warehouses consist of internal, enterprise-owned, structured data placed in relational data stores in support of individual business units. Data warehouses that support the solving of any repetitive, predictable types of problems can be called predetermined warehouses. Although most of activity is in building predetermined warehouses (also called data marts), the trend is toward expanding the scope toward more encompassing adhoc warehouses.

The tools (or applications) can be classified as data inquiry, data interpretation, multidimensional analysis, information discovery, and browsers.

- Data inquiry: Typical request for a set of data based on some search criteria
- Data interpretation: Manipulation and visualization of a set of data (for example, statistical analysis)

Data inquiry and interpretation tools are the traditional decision support tools, and most data warehouses use them. Trends for these tools are toward more advanced visualization approaches for interpreting the results and further hiding of data access mechanisms from the end user by providing a business semantics layer on top of relational tables for formulating the queries. As unstructured data becomes more prevalent in corporate databases, these tools
are expanded to support it. The data should be represented according to the granularity of the analysis to be performed. Thus the data are represented as detailed data, rolled up data, summary data. So these constraints should be considered while designing the database that will serve as the data warehouse. The data warehouse databases that are used as the source for the Pivotal analysis consists of following databases.

- **DW_UTENRG** ---- Energy Details
- **DW_UTLINE** ---- Line Production Details
- **DW_UTMNTC** ---- Maintenance Details
- **DW_DISPAT** ---- Dispatch Details

**Data Modeling:** Modeling a data warehouse is a much different process than the one that drives an OLTP data modeling effort. Most data modeling tools today support the entire relationship modeling technique that seeks to group data together in common relationship with the primary focus of eliminating redundancy in the database. Entire relationship modeling is very efficient in the OLTP environment where the goal is to get the data into the database in the most efficient way possible.

But in case of warehousing projects, dimensional data modeling is more advantages. A dimensional data model always is centered on a central fact table that contains multiple keys and numerous facts which are normally figures or units of measure that can be summed or averaged so as to determine trends or facts for a particular area. Thus the star schema for the data warehouse given in Figure 4.18 is designed to facilitate the dimensional analysis to be performed on the data to be captured in the data warehouse.
Data Warehouse Load Routines: MS SQL is used for the development data warehouse, and for extracting, transforming and loading the data. Since most of the administrative software used has been written in-house, integration of tables is done under a single unit code. It proved to be quick and easy to write and maintain, and is found powerful and efficient in its operation.
The steps involved

- Select data needed to support analysis needs from data source
- Extract source data and derive additional data where necessary
- Perform summarization and rollup to accepted level of granularity
- Perform data integration and denormalization

All the data in both the Operational Data Source (ODS) and the Data Warehouse are recoverable from production data and archive directories. The code that loads the project data needs to be completed so that it has the proper logic for dealing with projects. Then the ODS and Data Warehouse can be deleted, regenerated, and the data can be reloaded.

The subroutines are written in MS SQL and the data are extracted from the central data repository. The initial extraction is done successfully by means of data integration. The following tasks are undertaken during the process:

- Multiples source data mapping issues
- Key structure discrepancies
- Conversion of multiple description data to one standard form
- Created default data

**Output Cube Design:** The data captured by means of the data warehouse is analyzed multi-dimensionally using Pivotal chart of Excel 2000 and represented graphically for the decision makers to analyze the trend in a more presentable form.