MINING ASSOCIATION RULES FOR QUALITY RELATED DATA IN AN ELECTRONICS COMPANY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

YASEMİN KILINÇ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

FEBRUARY 2009

Approval of the thesis;

MINING ASSOCIATION RULES FOR QUALITY RELATED DATA IN AN ELECTRONICS COMPANY

submitted by **YASEMİN KILINÇ** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Nur Evin Özdemirel _____
Head of Department, **Industrial Engineering**

Prof. Dr. Nur Evin Özdemirel
Supervisor, **Industrial Engineering Dept., METU** _____

Prof. Dr. Sinan Kayalıgil
Co-Supervisor, **Industrial Engineering Dept., METU** _____

Examining Committee Members:

Prof. Dr. Gülser Köksal _____
Industrial Engineering Dept., METU

Prof. Dr. Nur Evin Özdemirel _____
Industrial Engineering Dept., METU

Prof. Dr. Sinan Kayalıgil _____
Industrial Engineering Dept., METU

Assoc. Prof. Dr. Tuğba Taşkaya Temizel _____
Informatics Institute, METU

Assoc. Prof. Dr. Murat Caner Testik _____
Industrial Engineering Dept., Hacettepe University

Date: 13.02.2009

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Yasemin Kılınç

Signature :

ABSTRACT

MINING ASSOCIATION RULES FOR QUALITY RELATED DATA IN AN ELECTRONICS COMPANY

Kılınç, Yasemin
M.S., Department of Industrial Engineering
Supervisor: Prof. Dr. Nur Evin Özdemirel
Co-Supervisor: Prof. Dr. Sinan Kayalıgil

February 2009, 102 pages

Quality has become a central concern as it has been observed that reducing defects will lower the cost of production. Hence, companies generate and store vast amounts of quality related data. Analysis of this data is critical in order to understand the quality problems and their causes, and to take preventive actions. In this thesis, we propose a methodology for this analysis based on one of the data mining techniques, association rules. The methodology is applied for quality related data of an electronics company. Apriori algorithm used in this application generates an excessively large number of rules most of which are redundant. Therefore we implement a three phase elimination process on the generated rules to come up with a reasonably small set of interesting rules. The approach is applied for two different data sets of the company, one for production defects and one for raw material non-conformities. We then validate the resultant rules using a test data set for each problem type and analyze the final set of rules.

Keywords: Data mining, association rules, Apriori algorithm, metarules, rule set reduction.

Öz

BİR ELEKTRONİK ŞİRKETİ İÇİN KALİTE VERİLERİNDEN BİRLİKTELİK KURALLARININ BELİRLENMESİ

Kılınç, Yasemin
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Nur Evin Özdemirel
Ortak Tez Yöneticisi: Prof. Dr. Sinan Kayalıgil

Şubat 2009, 102 pages

Günümüzde ürünlerdeki hata oranlarının azaltılmasının üretim maliyetini azaltma yönündeki etkileri gözlenmekte ve bu yüzden kalite konusu oldukça önem kazanmaktadır. Bu yüzden firmalar büyük miktarlarda kalite verisi üretmekte ve saklamaktadırlar. Bu verilerin analizi, problemlerin ve sebeplerinin anlaşılarak önleyici tedbirlerin alınması açısından önemlidir. Bu tezde bir veri madenciliği yöntemi olan birliktelik kuralları için bir metodoloji sunulmuştur. Kullanılan Apriori algoritması çok fazla kural üretmektedir. Bu kuralların çoğu gereksizdir. Bu yüzden gereksiz kuralların elenmesine yönelik üç aşamalı bir eleme yöntemi geliştirilmiştir. Bu yaklaşım bir elektronik firmasında üretim ve mal giriş kalite verileri üzerinde uygulanmıştır. Ortaya çıkarılan kurallar test verileri ile doğrulanmış ve sonuçlar analiz edilmiştir.

Anahtar Kelimeler: Veri madenciliği, birliktelik kuralları, Apriori algoritması, meta kurallar, kural kümesi küçültme.

To My Parents

ACKNOWLEDGEMENTS

I would like to express sincere thanks to my advisor Prof Dr. Nur Evin Özdemirel and co-advisor Prof. Dr. Sinan Kayalıgil for understanding the meaning of this work for me. I am very grateful to their patience, encouragement and support anytime I needed. I would not be able to do it without their help. I want to thank them for showing the light they give me in my most hopeless times.

I also would like to express my appreciation to the committee members of my defense; Prof. Dr. Gülser Köksal, Assoc. Prof. Dr. Tuğba Temizel Taşkaya and Assoc. Prof. Dr. Murat Caner Testik for their comments and suggestions.

I would like to thank to my manager Mehmet Ali Berk and my director Ali Fatih Bilgi for allowing me do this work and supporting the hardware I required for the thesis. I also would like to thank my managers in Quality and Production Management Departments letting me use their data and supporting me with all of the information I required. I also would like to thank to Pelin Ay for her endless support and encouragement both as a colleague and as a friend.

I would like to thank to my parents and my husband for their love and support. Especially I would like to thank my two sons for their presence and for the joy they add to my life.

I would like to thank my friends for their support and help.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Today, all industries have a basic goal of producing high quality and reliable products to satisfy their customers at minimum cost. Quality has become a central concern since it was observed that reducing defects will lower the cost of production. There are many definitions of quality. A very simple definition of quality is "conformance to requirements" by Phil Crosby given in given in Hamson et al. (2002). As the importance of quality improvement has become a major issue, quality improvement approaches have been introduced by some organizations via some standards such as ISO 9000, ANSI, AQAP that state the principles of quality improvement.

Companies need to analyze their processes to achieve the goal of producing a quality product. From the supply of raw materials to the delivery and maintenance activities, all phases of production have to be analyzed to find the problems, their causes, and measure the conformance of the product to the requirements of the customer. This often generates a high volume of data including nonconformity records, laboratory measurements and, field test results and all other information. Hence, the volume of data collected grows up day by day. As the data volume increases, analyzing the data and extracting knowledge from the data gets more difficult, with more time and effort spent on the analysis. For reducing the cost of analysis, some analysis tools have been developed in the form of business warehousing (BW) with OLAP (On-line Analytical Processing) functions. In a business warehousing system, the transactional data maintained in the operational system is transferred to the BW system. BW stores the data in cubes as dimensions and facts. Dimensions are the attributes and facts are the measurements or indicators. A cube is a combination of a fact table with its related dimension tables. These cubes allow

multidimensional analysis by drilling down each attribute or drilling up aggregating the facts. The OLAP functionalities, such as summarizing, aggregating or consolidating, support decision making for improvement of the processes. However, additional data analysis tools have been required for further investigation of the data characteristics. As a consequence of these requirements, data mining has emerged. Han and Kimber (2001) define data mining as "extracting knowledge from large amounts of data". This knowledge is hidden in the high volume data. Different data mining approaches search for this hidden knowledge called a hidden pattern. A pattern is interesting if it is new or surprising for the decision maker. One important task of data mining is exploring interesting patterns in the form of a rule.

"Association rules" is one of the most popular approaches in data mining applications for mining rules. An association rule is represented in the form $A \rightarrow B$. It shows the relationship of item sets A and B. Each attribute and its value in the data is called an item. One or more of these items together form each side of this association. The rule is read as "If A, then B". In this rule, A stands for the antecedent and B stands for the consequent. The antecedent part of the association rule consists of a number of items called an "item set". The consequent typically has only one item. The rules generated for quality data include the relationships of the attributes of production and the effects of these attributes on the yield of the product. Association rule mining for quality related data is used for two aims. The first aim is to find the effect of the input parameters (production attributes) on the output (yield). The rules give an idea on what the input parameters should be for the desired product. The second aim is to analyze the relationships among the attributes for non-conformities. This analysis can be made by OLAP functions to a certain extend. OLAP functions require some measurable attributes called "facts" in business warehousing literature. When there is a quantitative attribute, the data can be stored in cubes with many dimensions. However, when there is not a fact (the attributes are qualitative), it is not possible to analyze the relationships with OLAP.

The rules are formed according to the frequency of the antecedent items observed in the data set and the confidence of the consequent in the data set where the antecedent is supported. The percentage of the tuples in the data set

that include a given set of antecedent items is called the "support". Given a rule in the form A $\rightarrow$ B, "Confidence" is the percentage of the tuples containing B in the tuples that include the antecedent items. A rule is generally applicable and reliable only if it exceeds the support and confidence thresholds set by the decision maker. However, when these thresholds are set to low values or when the data set is large, the association rule algorithms produce many rules which are hard to interpret. Some other measures have to be used that measure the "interestingness" of a rule.

In this thesis, we aim to propose an application to discover the interesting association rules for quality related data of ASELSAN Inc. company. These rules will hopefully provide help to improve the quality of supply and production processes.

1.1 Problem Definition

ASELSAN Inc. is an electronics focused high technology company founded in 1975 to meet the requirements of Turkish Armed Forces. The sales revenue has increased as the product range has widened and it has become the 64th company in the "Top 500 Industrial Enterprises of Turkey" list announced by Istanbul Chamber of Industry in 2007. Today, ASELSAN Inc. has four main plants each specializing in a different product range. Although the mass production goes on for products such as different kinds of military radios, the production is mainly based on projects. These projects include research and development activities and the production of specialized military requirements in the area of communications, defense system technologies, radar, electronic warfare and intelligence systems, microelectronics, guidance and electro-optics.

Wide product range requires many suppliers. The number of suppliers is around 5000 including local and foreign subcontractors or manufacturers. All suppliers are evaluated based on the non-conformities of received components, the time of delivery and the cost of raw material supplied. A quality notification is opened whenever a non-conformance is observed. For each quality notification opened in the supply phase, the supplier is assigned a penalty in his note.

As an inevitable result of growth, increasing requirements of information management has lead ASELSAN Inc. managers to make a decision to migrate to an ERP system. To this end, a year's work has been performed on analyzing the processes and defining the requirements of an ERP system. ASELSAN Inc. has been using SAP as its ERP system since 2005. SAP offers a great deal of opportunities to store the data a company can ask for. Each plant can handle its own data in this system using the same set of modules.

With the use of SAP, Quality Departments of the plants started using notifications for the defects observed during quality inspections of component receipt or production phases. A quality notification is a document in SAP that stores all of the related data about a defect or a non-conformity observed during some inspection. Basically, when a raw material or component arrives at ASELSAN Inc., it undergoes acceptance inspections by the Quality Departments. For the supply phase, a sampling plan for each material type exists and a quality procedure is attached to it for the definition of inspection activities. This procedure is applied when that material arrives. These inspections are generally in the form of visual, mechanical and electronic inspections. If no defect is observed in inspections, then the material is accepted and moved to the depot. If a defect is observed in a lot, then a notification is created in the ERP system.

The procedure is slightly different in the production phase. 100% inspection is made in production. The production phase includes two types of inspection. One is realized by the Production Management including some tests such as vibration, environmental conditions, cable and others. The second type of inspection is realized by the Quality Departments according to the quality procedure of the material. These inspections are conducted in one work center during production. For each material produced, a routing is defined. In this routing, in addition to the production stages, inspection processes are included. In the production stages, there are approximately two or three stages for the inspections of the Quality Department and one or two stages for the inspections of the Production Management. The routings are matched with the work orders and in each work order a defined lot of material is produced. During the production stages, whenever a defect is observed one notification is created in the ERP system for every lot at each inspection stage. A new item record is added to the same

notification if more than one non-conformity or defect is observed for the same lot in the same inspection unit.

For the supply phase inspections are performed by around 20 technicians in each plant. For the production phase, the technicians of Quality Departments are around 10 and the technicians of Production Managements are around 5 in each plant. The quality inspection procedures are similar in all plants.

The notifications hold the data on the problem, the work center where the defect is observed, the material type and other related information. Approximately 1000 notifications per month in SAP for all plants, which may include more than one item are created. The quality coordinators distribute these notifications to the technicians creating a task on the notification. There are 5-10 technicians for supply and 30-40 technicians for production in each plant. The cause of the problem is searched by the assigned technician. The technician adds the cause data. They select one code and one activity suggestion on the defective material to the notification from the previously defined entries. This information is validated by the quality inspectors. The resultant activity may be accepting, repairing, or scrapping the defective material, as well as reworking it, returning it to the supplier, or using it regardless of its defect or non-conformity.

Different inspection types can be defined in the system. Two types of those notifications are the subject of this thesis. These are the supply and the production types of notifications. This is analyzed for one plant's data only as a representative case.

OLAP functions of SAP in BW allow the quality notifications to be analyzed for ASELSAN Inc. For each non-conformity, an entry is filled in. These entries are then aggregated in business warehousing cubes having the fact fixed one for each entry. This kind of OLAP processing gives the opportunity for multidimensional analysis. However, the analyst should know which dimensions (attributes) to use in the analysis.

Data mining opens up a new window for the analyst to see the relationships without having to select the proper data items (i.e., dimensions). This is because

rules emerge only if a group of attributes are associated. Hence, at present there is a lack of sufficient analysis of quality notifications data. Using OLAP and data mining may be useful in this respect.

1.2 Objectives

Continuous improvement and quantitative management of information are required by the well known quality standards ASELSAN Inc. has adopted such as ISO 9001:2000, AQAP-2110 and AQAP-160. Following the implementation of SAP, ASELSAN Inc. had the opportunity for multidimensional analysis by using the business warehousing capabilities offered by SAP. One can tour among the attributes and can discover the relationships between attributes. Hence, some quality related reports can be generated through the BW reports after SAP was put to use. However, the Quality Departments want to enlarge the analysis parameters and want to capitalize on the chance of improving the (supply and production) processes. Quality Departments search and implement statistical approaches to achieve this. The hidden patterns in the data can enhance these endeavors for investigative, corrective or preventive actions. This thesis work is expected to contribute to the Quality Departments in defining their approaches to analysis in near future. The results are also expected to show the adequacy of the details stored as the attributes of problems found or causes of the defects made available as part of the notification. If the resulting rules supply new knowledge on the problems and their causes and point to the right action for correction, then the attributes defined and the detail of the attribute codes entered in the system are sufficient. Otherwise, more work has to be done on these attributes and predefined attribute codes. The rules are expected to determine the analysis parameters with regard to which of the existing attributes to be included in the analysis, what other missing parameters shall we add to the analysis.

We start our analysis with data retrieval and data pre-processing. Data retrieval and data pre-processing phases are unavoidable in a data mining process. This phase has constituted almost 30% of the effort in this study.

One of the most important points of knowledge discovery with data mining is to end up with interpretable results. Association rule mining algorithms not only produce excessively large number of rules but also most of these rules are redundant. Generally, further analysis of the rules is required after the implementation of an algorithm. Our contribution towards this goal includes the proposal of an easy to implement methodology for ASELSAN Inc. It uses a combination of three approaches to eliminate the rules with no true contribution in the existence of the others.

The effect of decision parameters such as minimum support and minimum confidence level are examined through runs with different parameters. In each run, a series of redundant rule elimination approaches are applied. The first approach relates to missing values. Instead of deleting the tuples with missing values, we include all tuples and use the information in other attributes of those tuples with the promise to get the maximum information from the data. The second approach is related to seeking superior rules in terms of representation power using the SC optimality (Bayardo and Agrawal, 1999). We add some restrictions to this approach so that information loss is avoided. Finally, we use the metarules approach (Berrado and Runger, 2007) to find the equivalent rules. Although metarules approach in literature is restricted to the rules that have the same consequent, we implement this approach for the final set of rules which point to different consequents. The case study of ASELSAN Inc. conducted in the thesis work shows the applicability of these approaches in real life.

The following chapters of the thesis are organized as follows:

- Chapter 2 discusses the related work in literature. In the first section, the literature review of data mining work on quality related data is given. The algorithms used for such data are mentioned very briefly. The following sections, review the work in association rules algorithms, their use in quality related data and interestingness measures.
- Chapter 3 describes the methodology we use to end up with interesting rules. The algorithm used in discovering the association rules (Apriori) and the subsequent procedures to reduce the number of rules are discussed in this chapter. For this aim, SC optimality (Bayardo and Agrawal, 1999) with some restrictions added to avoid information loss

and metarules approach (Berrado and Runger, 2007) are explained. The procedure applied for validation of the rules and interestingness measures used are also described in this part.

- The application and the results of the proposed methodology to ASELSAN Inc. quality notifications data is given in Chapter 4.
- We finish the thesis with the conclusions and possible future work in Chapter 5.

CHAPTER 2

LITERATURE SURVEY

Data mining applications are widely used in many areas such as banking, marketing, and health care. However, the applications in manufacturing quality are relatively new due to some restrictions of the quality data. These restrictions are imbalanced distribution, curse of dimensionality and mixed type of data as stated by Rokach and Maimon (2006). Although there are many areas of use in manufacturing, related to this thesis, the following section includes only the data mining implementations on quality related data in manufacturing. In Section 2.2 the progress of the algorithms for association rules is mentioned. The implementations of association rule mining in quality related data for manufacturing are mentioned in Section 2.3. Finally, the interestingness measures in the literature are discussed in Section 2.4.

2.1 Data Mining for Quality Related Data

There are different analysis approaches used in manufacturing quality improvement. These include the techniques for production accuracy using the precise measurement or inspection devices. Statistical methods are used to analyze the defects and to search for the causes. Work in literature for the quality related data searches for the answers to the research question of if data mining can be used to analyze the defects and nonconformities so as to find the relations of nonconformities to other attributes in the whole process. The data mining implementations surveyed for this thesis are mainly based on rule induction and classification algorithms. The rule induction algorithms of rough set theory and association rules are the most popular algorithms used in this

area. The classification algorithms are used for analyzing the behaviors of the factors that affect the percentage of defectives or so called yield. Decision tree is one of the classification approaches used in literature. There are some reviews in literature (see, for example, Harding et al. 2006 and Wang et al. 2007) where the applications and implementations of these algorithms are mentioned in detail. The papers mentioned in these reviews constitute the basis for the literature survey in this thesis.

2.1.1 Rough Set Theory

A rough set is defined by Pawlak (1991) as "a formal approximation of a conventional set in terms of a pair of sets which give the lower and upper approximation of the original set". The main idea is that, deciding on a decision attribute, rules are created and a new object is classified according to these rules. Rough set theory applications are widely used in recognition algorithms, dimensionality reduction, decision support systems in medicine and intelligent control systems (Wu et al., 2004). Wu et al. (2004) mention in their review that this theory has first been used in rule discovery by Bell and Guan (1998). Bell and Guan applied rough set theory on the car test results.

Kusiak and Kurasek (2001) analyzed the production quality notifications of the printed circuit board (PCB). The cause of solder defects in PCBs were identified with data mining using the rough set approach. Data collected for this work included over a three month period. 2052 PCBs that include 89 defectives were analyzed. Fourteen attributes were included in the application. Three rules were generated at the end of the work, one for showing the conditions of no defect, one for the conditions of defect and one for alternative outcomes. They validated the results using 10-fold cross validation scheme. It was concluded that the rules provided a robust indication of where to focus.

Zhai et al. (2002) used rough set theory for feature extraction with the integration of genetic algorithms. An application developed by the authors was proposed in this study. The results showed that this application remarkably reduces the cost and time consumed on product quality evaluation without compromising the overall specifications of the acceptance tests.

Hou et al. (2003) studied on an intelligent remote monitoring and diagnosis of manufacturing processes. This study included the implementation of an application with an integrated approach that uses back propagation neural networks that monitors the production process and identifies faulty categories. For the diagnosis of the process they used rough set to extract the relationships between manufacturing processes and the product quality measure. They applied this approach to the manufacturing of industrial conveyor belts. Eight input attributes and the output (fault) for 27 records are used to train back propagation neural network and rules are generated by using the rough set approach. 15 records are used to test the results. Some abnormalities that cause faulty production were discovered with the help of this application.

The "Rough Set Based Decision Support System" software was developed by Tseng et al. (2005). They verified their approach with some historical data of a CNC machine process. Data collected for 1000 parts were used where 60 of them was faulty. Rules were generated using rough set theory and validated by bootstrapping (2/3 of the data for training, 1/3 of the data for testing). The results provided the relationships between the input parameters of the CNC and the acceptance of surface roughness. As a conclusion this approach showed the practical viability of the rough set theory approach for quality control.

Sadoyan et al. (2006) presented a new algorithm based on the rough set theory for manufacturing process control. As an implementation of the presented algorithm, the data was extracted by a thermal imaging system measuring the output parameters for the spray system. 1200 records included nine attributes. Three of these attributes had a value in some range. For these attributes, discretization was used to group the attribute values in clusters. Other attributes had at most four different codes that could be entered. These data was used to form the if/then rules choosing two of the attributes as process outputs. Using the generated rules, the ideal input parameters of the spray forming process were suggested. This article gives an idea about how knowledge obtained from data mining can be used in process control. The rules give the information on the input parameters for the desired output. The article also suggests an automated process control using the decision rules obtained from data.

2.1.2 Decision Trees

Kwak and Yih (2004) proposed "Competitive Decision Selector" software they have developed. Long-run and short run performances of the rules on various states of the system are observed by this application. Short term performance is observed by the classification rules of decision trees. This approach is applied on a simulation test bed where data is generated for a surface mount technology process.

Huang and Wu (2004) studied a case study for an ultra-precision manufacturing industry for the analysis of product quality improvement. They used decision trees to find the important factors that impact the product quality. 11320 ultra-precision optical products for three month data were analyzed. The gains chart produced by the decision tree was used to develop quality improvement strategies. The results showed that type of processing chain, precision requirement, product classes and raw material had an impact on the percentage of defectives according to the decision tree.

Chien et al. (2007) conducted a real life case which includes the data of wafer fabrication in a semiconductor foundry company in Taiwan. To reduce the costs that are caused by excursion, they found the root causes. 71 lots passing through 168 manufacturing stages are revised where 12 of them were in the bad group. K-means algorithm was used to cluster the wafer lots into two groups of good and bad lots. Since there were many processes, they applied the K-W test to examine the significant differences among the outputs of the machines used in the same process. Only those processes that have significant differences were chosen for the analysis. Finally the decision tree approach is used to form the if/then rules. These rules help domain engineers to find out the root cause when a defect occurs and help decision makers to understand how to overcome the problem by the analysis.

Rokach and Maimon (2006) presented a feature set decomposition methodology with the BOW (Breadth-Oblivious-Wrapper) algorithm. The main idea of this algorithm is decomposing the original set of features (input attributes) into several subsets, building a decision tree for each subset and then combining the

trees. They aimed to find the relationship between the quality measure and the manufacturing quality related process data. They tested the new algorithm in the fabrication of integrated circuits where the process begins with the production of a semiconductor wafer with two different data sets. The first dataset included only 70 tuples and 257 input attributes. The target attribute was chosen as the yield having a value of "high" or "low". The second data set included 395 tuples and 220 input attributes. The target attribute was chosen as a binary having the values "pass" and "not pass". With this work they examined the behavior of different parameters that affect the line throughput (the number of good products) and they obtained the decision trees. They compare their results with other decision tree methodologies. They concluded that setting the process parameters according to the classifier obtained by their algorithm (BOW) improved the yields. This approach can specifically be used in circumstances where there are many attributes.

There are also other applications. Shiue and Guh (2006) used decision trees for optimization of attribute selection for production control systems. Decision trees with cased based reasoning was studied to analyze the causes of the defects by Selvamani and Khemani (2005) in manufacturing steel strips.

2.2 Association Rules

The problem of discovering association rules was first introduced by Agrawal, Imielinski and Swami (1993). The problem is described as finding the relationship of items in a set of transactions for supermarket data. This problem is called the market basket analysis. In this work the problem is decomposed into two sub-problems: discovering the frequent item sets and exploring the association rules from the frequent item sets (frequent item sets are explained in Section 3.2). Frequent item sets are chosen from the candidate item sets where the fraction of tuples for combination of items in the candidate item sets is over a threshold (minimum support) given by the user. Association rules are explored from the frequent item sets where a minimum confidence level is achieved. The algorithm for solving these problems is introduced as AIS (the initials of the authors) algorithm. Given n items, 2^n candidate item sets are checked if they are frequent item sets with a bottom-up, breadth-first search that enumerates every

single frequent item set. Kotsiantis and Kanellopoulos (2006) state in their review that "the main drawback of the AIS Algorithm is too many candidate item sets that finally turned out to be small are generated, which requires more space and wastes much effort that turned out to be useless". Hence, some techniques are developed for efficiency of the algorithm by reducing the

- Number of candidates (2^n)
- Number of transactions (N)
- Number of comparisons ($Nx2^n$).

SETM algorithm was introduced from a set-oriented perspective using sorting and merging scan joins (Houtsma and Swami, 1995). This algorithm is based on the SQL queries. Hence, it takes the advantage of query optimization with an improved efficiency.

In both AIS and SETM algorithms, candidate item sets during the pass are generated as data is being read. The transaction is checked for the large item sets found in the previous pass. If there exists, then new candidate item sets are generated by extending these large item sets with other items in the transaction.

Apriori algorithm was proposed for reducing the number of candidate item sets by Agrawal and Srikant (1994). The algorithm is described in Section 3.2 as it is used in our methodology. This algorithm eliminates some item sets that are proved not to be large. The basic intuition of the algorithm is that all subsets of a frequent item set must be frequent. If they are not frequent, then the item set cannot be used further to generate a candidate item set. Hence, the item sets that contain any subset that is not frequent are eliminated according to this algorithm. This algorithm produces a much smaller number of candidate item sets.

An implementation of the Apriori algorithm was performed by Christian Borgelt (2002) based on the prefix tree concept. The program written by him can be accessed at (http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html) and is used in SPSS Clementine 7.1.2 (SPSS Inc. Clementine 11 Algorithms Guide, 2007).

Many other algorithms developed for association rules are derivatives or extensions of finding frequent patterns. Improving the efficiency, FP-Growth algorithm was presented with a compact data structure called frequent pattern tree or FP Tree (Han et al., 2000).

Bayardo (1998) introduced the Max-Miner algorithm for forming only the maximal frequent item sets. An item set is maximal frequent if it has no frequent superset. Any frequent item set is a subset of a maximal frequent item set. However, this algorithm is not suitable for generating association rules since it does not consider the confidence of the frequent item sets in the subset. That is, the frequent item sets that have a support over the threshold set by the user are only considered if the item set is maximal. If frequent item set is not maximal, then the confidence of items for that frequent item set is not calculated and the set is not considered anymore. Hence, there is information loss.

Zaki (2000) introduced the concept of non-redundant rules. According to his definition, a rule R_i is more general than another rule R_j if R_j can be generated by adding items to either consequent or antecedent of R_i. A rule R_j is redundant if there exist another rule R_i which is more general but having the same confidence level. For generating non-redundant rules with no loss of information, Zaki (2000) also introduced the closed frequent item set concept. Closed item sets uniquely determine the set of all frequent item sets and their exact frequency. So there is no information loss. CHARM Algoritm for mining all frequent closed item sets was presented by Zaki (2002, 2004).

A major problem with association rules approach is that it often generates a very large number of rules where many of these rules are redundant. This brings about the problem of eliminating redundant rules. This topic is also studied by Bayardo and Agrawal (1999) with SC optimality and Berrado and Runger (2007) with metarules approach. Methods proposed by these are explained in detail in Sections 3.3.2 and 3.3.3 as they are used in our methodology.

2.3 Association Rules in Quality

Chen et al. (2004) addressed the manufacturing defect detection problem which is defined as analyzing the relations between combinations of machines and the result of defect. They proposed an integrated processing procedure RMI (root cause machine identifier) to discover the root cause based on a novel interestingness measure. Nine real data sets from Taiwan Semiconductor Manufacturing Company were used in the study having 53 to 484 products passing through around 1200 stages and 2726 machines. The rules are then ranked according to an interestingness measure proposed as the minimum defect coverage. The rule with the highest value of this measure is treated as the root cause.

A study was presented by Tong et al. (2007) on the manufacturing quality data for fan blades including the thickness, width and height attributes. For 13 different sections of the fan blades these measurements are collected. For an illustration, they have analyzed 15 transactions. They applied Apriori algorithm and formed the association rules. They showed that these rules could then be used by the designer of the material to avoid the design mistakes.

Wuescher (2006), discussed the data mining implementation for shop floor information for aircraft, ship building or special machine industry. They retrieved the data by the OLAP operations. Although the analysis by OLAP operations is very important, it was shown that the major drawback of OLAP analysis is that analysis is possible for only numerical data where the user knows which dimensions or attributes he will analyze. The answer to the question of "which components are frequently involved together in disturbances" was searched with the Apriori algorithm. In the data pre-processing step, Deviation Based Outlier Detection, Attribute Oriented Induction and Analytical Characterization approaches were used. They concluded that the results could be used in corrective actions. However, methodological support and motivation of the people involved was founded as important as the related processes.

Shahbaz et al. (2006) studied the implementation of association rules for the maintenance and repair data of a company specialized in distribution panel

assembly in the packaging and shipping channels of traditional industries. They examined the faulted materials of which the design should be changed. They also implemented a particle swam optimization (PSO) algorithm with the association rules. PSO refers to an algorithm used to find optimal or near optimal solutions to numerical and qualitative problems. In this research it was implemented for optimization of manufacturing and shipping costs while maximizing the quality level.

2.4 Interestingness Measures

Association algorithms produce large amounts of rules that users cannot interpret. The efficiency of the algorithm is a major issue and support and confidence thresholds are effective in this sense. However, this kind of elimination does not consider the special interest of users or domain knowledge. Hence, developing a strategy to find the "interesting" rules is another issue to be solved (Klemitten, 1994). As a result some interestingness measures have emerged. The user specified constraints are taken into consideration on the kinds of rules generated. These constraints are also considered for defining objective metrics. Then rules are eliminated according to these metrics.

A survey on interestingness measures was conducted by Geng and Hamilton (2006). They defined nine criteria to determine the interestingness. These criteria include conciseness, coverage, reliability, peculiarity, diversity, novelty, surprisingness, utility and actionability. Objective measures are studied in literature to serve the reliability and generality criteria. Geng and Hamilton (2006) give references to Tan et al. (2002), Lenca et al. (2004), Ohsaki et al. (2004) and Lavrac et al. (1999) for the 38 objective measures they review. We will go through some common interesting measures in this section. These measures are usually a function of 2x2 contingency table given in Table 2.1.

Table 2.1. Contingency table for the rule A→B.

	B	not B	
A	n (A∩B)	n(A) - n (A∩B)	n(A)
not A	n(B) - n (A∩B)	N - (n(A) + n(B) - n (A∩B))	N - n(A)
	n (B)	N - n(B)	

Suppose that we have a rule in the form of A→B where A is the antecedent and B is the consequent. Rule support is the probability of consequent and antecedent seen together, i.e.,

Rule Support = P(AB)

Confidence is the probability of the consequent in the tuples where all items in the antecedent are supported:

Confidence = $P(B|A)$

Lift is defined as the ratio of the confidence to its expected confidence. This gives information about the increase in probability of the consequent given the antecedent. Its formula is:

$$\text{Lift} = \frac{P(B|A)}{P(B)}$$

A lift value greater than 1 indicates that antecedent and consequent are more often together than expected. Lift value less than 1 indicates that the antecedent has a negative effect on the occurrence of consequent. When it is close to 1, the antecedent has no significant effect on the occurrence of the consequent.

Lavrac et al. (1999) argues that the interestingness measures should be used relative to some threshold. He proposed a single measure that can be interpreted in five ways: weighted relative accuracy, weighted relative sensitivity, weighted relative precision negative reliability, and weighted relative novelty. Lavrac et al. (1999) show that among these measures weighted relative accuracy dropps sharply after the first few rules. Weighted relative accuracy is

identical to the measure defined by Piatetsky-Shapiro (1991). These two measures are defined as:

$$\text{AddedValue} = P(B|A) - P(B)$$

$$\text{Piatetsky-Shapiro} = P(AB) - P(A) * P(B)$$

Klemitten et al. (1994) use templates to describe the interesting or uninteresting classes of rules. When a pattern is defined by the user as interesting, the rules that match the specified pattern are presented to the user as the interesting rules.

Bayardo and Aggrawal (1999) claim that the best rule according to some metrics must reside along a support/confidence border known as the SC optimality. They show that these metrics include support, confidence, lift, gain, conviction, laplace and Piatetsky-Shapiro's measure. They define a partial order on both support and confidence, and a rule is less interesting if there exists any other rule with higher support and confidence levels.

Tan et al. (2000) states that support is appropriate in rule elimination since it can eliminate mostly uncorrelated or negatively correlated patterns. However, it may not serve as a reliable interestingness measure. This is because rules with high level of support give very general information which is already known in the domain of concern. They also state that confidence may be misleading in many practical situations referencing the work by Brin et al. (1997). Hence, Tan et al. (2000) propose IS measure that both includes support and an explicit measure of variable dependencies. This measure is also used in this thesis and further information is given in Section 3.4.

Tan et al. (2004) describe some key properties for selecting the right measure for a given set of rules and present an algorithm for selecting a set of rules to choose the best measure by ranking this subset.

The literature survey of this thesis shows that the work involved in quality related data can help the improvement of processes especially in achieving the

desired product quality by giving the input parameters covered in the rules. For quality data where there are no measurements, OLAP functions are not sufficient to analyze the data. Hence, data mining has its place in these situations, which is the case we face. Data mining answers the question of the impact of different attributes on the product defects or nonconformities. Hence, it serves as a technique to understand the behavior of production parameters and is the starting point of taking preventive actions.

CHAPTER 3

METHODOLOGY APPLIED

3.1 Data Retrieval and Preprocessing

ASELSAN Inc. has been using SAP since January, 2005. The three plants have been entering the quality notifications into the SAP system since then. There are two main types of notifications. The first type is the notifications entered for the defects occurred in quality inspections during the supply of the raw materials. The second type of notifications entered in the SAP system is for the defects occurred during the production process.

The data for all plants is retrieved from the SAP system via a function module written. This function module retrieves all related data from different database tables, of which further detail is given in Section 4.1.1. Then the data is downloaded for preprocessing.

Data quality is a very important issue in data mining. Data quality depends on three desirable characteristics of data. These are:

- Completeness
- Consistency
- Being noise free (Han and Kimber, 2001)

However, lack of these qualifications is unavoidable in real life data. Missing values or missing attributes prevent the completeness. Some data may be incorrect due to human intervention. There exist contradictions in the values of different attributes resulting in inconsistent data. These problems are present in the ASELSAN Inc. data as well. The notifications are filled in with different

technicians and the problems and causes chosen from the set of defined attribute values may differ from technician to technician. The data retrieved includes the whole data since the first use of the SAP system. During the installation some data is entered for testing the system that causes noises in the data. The attributes used in a notification changed in time for a better use of the system. Hence, all of the data is not complete.

The solution to these problems is achieved by data cleaning (Han and Kimber, 2001). The following steps are applied to the raw data in the following order:

- If the missing values of an attribute can be determined with the help of other attributes, then they are filled with the correct information.
- If the missing information belongs to the consequent or belongs to an attribute that can be a class label, then the tuple is excluded from the data set.
- After the first two steps, if there is still missing information, then these fields are filled in with "#". This is preferred to make the remaining information in these tuples still useful.
- If incorrect entries can be corrected by inference of any data source, then these fields are recoded by their correct values.
- Inconsistency checks are made with the rules defined by the Quality Departments. For example, for a notification type some attributes have to be filled and some attributes are not related with this notification type. If non-related attributes are filled in, there is an inconsistency. These inconsistencies are removed by recoding the related fields or deleting the tuples.

The next step in data preprocessing phase is data discretization. For the attributes that have material quantity values there is the problem of unit of measure. In each tuple the unit of measure in quantity fields may change. For example, even for the same material of a supply type notification some suppliers send the material in boxes of fifty and some suppliers send it in dozens. Then in one tuple the unit of measure will be "box", whereas it will be "dozen" in the other. To handle this problem, the ratios of the defective quantities to the referenced quantities are calculated. For the same example, if the supplier sends five boxes and three boxes are defective, then the defective ratio is 60%. The

new attribute takes the values in range [0, 100]. Since the Apriori algorithm used in constructing the association rules cannot process real variables (ranges), these ratios need to be discretized. Unsupervised discretization is used as the starting point. For one data set, binning with equal intervals (equiwidth) is applied. However, the initial intervals did not yield sufficiently large number of tuples to support the rule discovery. Hence, some initial intervals were merged to expand the ranges. The detail of this work is given in Section 4.1.3.

At the end of the data preprocessing step, we have a complete and consistent data set with no noise. Although there exist some missing data entries, we have filled them with "#" and, technically speaking, the data is complete. We deal with these entries after generating the rules which will be mentioned in Section 3.3.1.

3.2 Apriori Algorithm

For the thesis work, we have chosen Apriori algorithm that best satisfies our needs as defined in Chapter 1. SPSS Clementine 11.1 is a data mining tool that enables users to develop predictive models (SPSS Inc., 2007). A broad range of algorithms in data mining are covered in this tool. As a resource on hand this software is used for Apriori application.

Apriori Algorithm was first introduced by Agrawal and Srikant (1993). It searches for frequent item sets and develops rules from these item sets. An "item" is an attribute-value pair found in a tuple of the data set. An "item set" is the group of items. Each tuple in the data set is an item set. An association rule r_i is denoted by A → B where A is an item set called the antecedent and B is an item called the consequent such that $A \cap B = \varnothing$. The rule support of rule r_i is:

$$Si = P(AB) = \frac{N(AB)}{N}$$

Where N(AB) is the number of tuples that contain the item set ($A \cup B$), and N is the total number of tuples.

Apriori algorithm, presented by Agrawal and Srikant (1993) proceeds in two steps. The first step is finding the frequent item sets and writing them in a table. The second step is generating rules from the table of frequent item sets. A frequent item set is defined as an item set with support greater than or equal to the user specified minimum support threshold S_{min}. The support threshold is set for the antecedent support which is defined as:

$$S(A) = P(A) = \frac{N(A)}{N}$$

where N(A) is the number of tuples that contain the item set (A).

The Apriori algorithm first identifies the item sets with one item (i.e. item sets of length 1) that satisfy the support threshold. Those items supported less than the threshold are discarded. The basic idea of Apriori is removing the infrequent items out of the scope since adding an infrequent item to an item set will always result in an infrequent item set.

Next, Apriori generates larger item sets using the item sets identified in the first step. Every possible pair of item sets from the previous step (frequent item sets) are merged as candidate set. Among the candidate set, the support is calculated for each item set to understand if the candidate item set is frequent or not. Those item sets that have a support level over the threshold are added to the list of frequent item sets. Any infrequent candidate item set is immediately removed from further consideration. The algorithm recursively works by increasing the number of items one by one. This process stops when all the items except one (for the consequent) are included in the item set or when the number of items in the antecedent reached a limit set by the user (Agrawal and Srikant 1994).

When all frequent item sets have been identified, the algorithm extracts rules from the frequent item sets. For each frequent item set, the subsets are constructed. These subsets include one item in its output and all other items in its input. For each subset, the confidence is calculated. If the calculated confidence is over the minimum confidence level set by the user, then this subset is added to the rules formed. The software application used in this study is based upon Borgelt's implementation (2002) for the Apriori implementation.

Minimum confidence level is the confidence threshold. Rules that do not satisfy this level are eliminated. Confidence is the conditional probability of the consequent given the antecedent. It is defined as:

$$C = P(B \mid A) = \frac{P(AB)}{P(A)}$$

Apriori algorithm has four basic parameters to be decided. They are: direction, minimum antecedent support, minimum confidence level and maximum number of items in the antecedent part of the rule.

Each attribute is selected as input, output or both according to the decision of the direction. If the attribute is selected to be in the antecedent, then it should be set as input. If the attribute is selected to be the consequent, then it should be set as output. If the attribute is selected to be at both sides of a rule, then "both" should be set as the direction. For ASELSAN Inc. data the attributes that can be used as class labels are selected as consequents and others are selected for the antecedents.

Minimum antecedent support is the threshold for the support level to find the frequent item sets in the Apriori algorithm.

Association rule models can be built by Apriori algorithm with either tabular or transactional data. An example of both data types are given in Figure 3.1. Transactional data has a separate tuple for each item purchased by a customer. Tabular data has a single tuple for each customer. Items purchased by a customer, appears as attributes in the customer's tuple. The association rules that can be generated from both tables are the same. ASELSAN Inc. data is tabular format.

As stated in Section 2.2, Apriori algorithm generates many rules where most of them are redundant. Different minimum antecedent support and confidence levels are used in each run. Furthermore, some elimination approaches are applied on the resultant rules. The rules generated by Apriori are saved in SQL

database for further analysis. Then, elimination techniques are applied on these rules by our Java code. The results of the different runs are examined to decide a best run policy to gather the final set of valid rules covering a high percentage of the tuples with a minimum acceptable confidence level.

Transactional Data

Customer	Purchase
1	jam
2	milk
3	jam
3	bread
4	jam
4	bread
4	milk

Tabular Data

Customer	Jam	Bread	Milk
1	T	F	F
2	F	F	T
3	T	T	F
4	T	T	T

Figure 3.1. Examples to transactional and tabular data formats.

3.3 Elimination of Rules

The number of rules Apriori generates simply depends on the parameters. The number of rules generated increases as the minimum support or minimum confidence levels decrease or number of the items in the antecedent increases. The number of tuples that support the final rule set also depends on these parameters. To achieve a higher coverage on the tuples, low support and confidence levels are necessary. However, too many rules generated in each case are of little value since some of these rules are redundant. Some of the rules have misleading information due to the "#" sign added as a legitimate item.

Then the next step is preparing a rule set to present to the Quality Department so that they can see the relations of attributes to the problems occurring in the supply or production phases. Then they can take corrective or preventive actions to solve the quality related problems, if any. Since the number of rules generated is too large to present as it is, further process on the generated association rules is required to find the interesting rules. The elimination process

is conducted in three phases. These phases are described in the subsequent sections.

3.3.1 Elimination due to Missing Data

In Section 3.1, it was mentioned that the missing data that cannot be filled in any way is set to the generic "#" sign. Apriori algorithm takes this sign as a legitimate attribute assignment and this item is included in many rules. These rules, in fact, would only have a meaning if the item with the missing data was excluded from the rule statement. However, there may already exist an identical rule among the generated rules (i.e., exactly the same item set except that the "#" item is excluded), unless it is eliminated (due to the minimum confidence level). If there does not exist such a rule, then we really do not need the rule with "#". The idea behind that elimination is explained below with an example.

Consider the case where an Apriori run is made with the specified minimum antecedent support set to S_{min} and minimum confidence level set to C_{min}. Consider the two rules in the rule set R generated:
r_1: Attr1 = a, Attr2 = b, Attr3 = # → Attr4 = d with S_1, C_1
r_2: Attr1 = a, Attr2 = b → Attr4 = d with S_2, C_2
Where S_1 and S_2 are the antecedent supports and C_1 and C_2 are the confidence levels of the rules r_1 and r_2, respectively. From these two rules it can be seen that $S_2 >= S_1$. since items of r_2 are a subset of the items of r_1. Furthermore assume that $S_1 >= S_{min}$. Then,

- r_1 is included in the generated rules: $r_1 \in R$.
- r_2 is in the frequent item sets table as $S_2 >= S_{min}$. However, recall from Section 3.2 that Apriori algorithm generates rules from the frequent item sets only if the confidence of the rule is larger than the specified minimum confidence level. Thus,
    - If $C_2 >= C_{min}$, then $r_2 \in R$ and r_1 is not needed in the generated rules. Hence, it can be eliminated.
    - If $C_2 < C_{min}$, then $r_2 \notin R$. In this case, Apriori eliminates r_2, which is equivalent to r_1 without the "#" item. Hence, r_1 can be eliminated.

Hence, the rules that include this sign in any of its items are eliminated.

Therefore, in the rest of the work, the rules with "#" items in the antecedent are deleted from the generated rules in all runs.

3.3.2 Elimination Using SC Optimality

Bayardo and Agrawal (1999) proposed an approach to find the most interesting rules in 1999. They argue that the best rule according to an evaluation measure such as confidence, support, gain and so on, must reside along a support/confidence border. In this section, we explain the idea of this approach and how it is applied to ASELSAN Inc. data in the elimination process.

Let U be the set of all input items that can appear in the antecedent. The generic problem statement for optimized rule mining process is defined by Bayardo and Agrawal (1999) as finding a set of antecedent items $A_1 \subseteq U$ such that:

1. A_1 satisfies the input constraints. The input constraints are the specified levels of minimum antecedent support and confidence.
2. There exists no item set $A_2 \subseteq U$ such that A_2 satisfies the input constraints and $A_1 < A_2$ where "<" is used for the preference order of item set. (i.e., A_2 is preferred to A_1.

Any rule A → C whose antecedent is a solution to an item C in the consequent of the optimized rule mining problem is said to be Instance-optimal (I-optimal) where instance refers to the consequent i.e., there exist one Instance-Optimal rule for each value of the item in the consequent or just optimal if the value of the item in the consequent is fixed.

Bayardo and Agrawal (1999) replace this optimized rule mining problem with the partial-order optimized rule mining problem. They define a partial ordered relation based on support and confidence using the (anti-)monotonicity property of functions of different measures. (For example, confidence function in terms of rule support and antecedent support is antimonotone in antecedent support when rule support is held fixed.(Bayardo and Agrawal, 1999) The partial order is shown as $\leq_{sc}$. Given rules r_i and r_j, r_j is preferred to r_i, $r_i \leq_{sc} r_j$, if and only if:

- $S_i \leq S_j$ and $C_i < C_j$ or
- $S_i < S_j$ and $C_i \leq C_j$

- $B_i = B_j$

$r_i =_{sc} r_j$ if and only if:

- $S_i = S_j$ and $C_i = C_j$
- $B_i = B_j$

where S_i is the antecedent support, C_i is the confidence and B_i is the consequent item of rule r_i.

Similarly, the partial order defined as $r_i \leq_{s\neg c} r_j$ if and only if

- $S_i \leq S_j$ and $C_i > C_j$ or
- $S_i < S_j$ and $C_i \geq C_j$.

An I-optimal set where this partial order is contained is given in Figure 3.2. Having the consequent fixed, each rule is defined as a dot on the graph according to its support and confidence levels. The rules are ranked according to their support and confidence levels. As can be seen from the figure, the dark points forming the upper border show the SC-optimal rules.

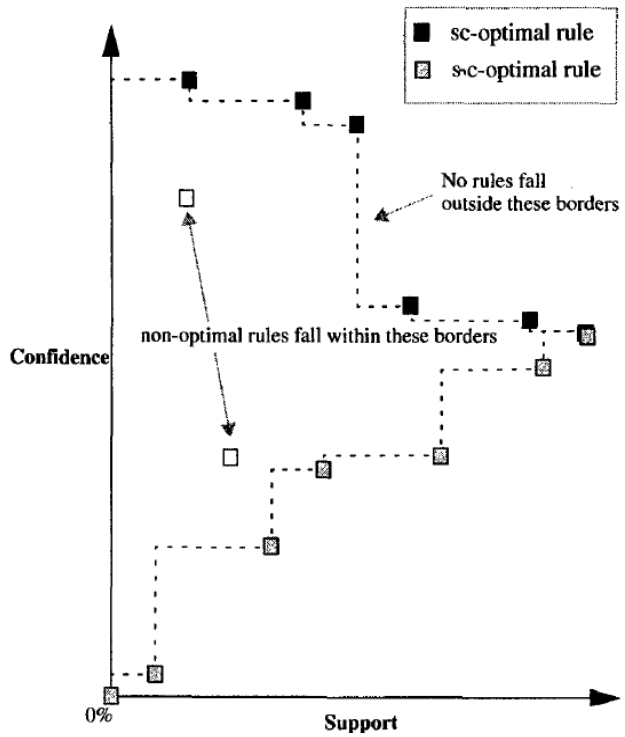


Figure 3.2 Upper and lower support-confidence borders by Bayardo and Agrawal (1999).

The lower border on Figure 3.2 shows the border where no rules satisfying the input constraints can exist outside. The upper border forming the SC optimal rules is of great interest, and the lower border is used to eliminate rules from the outset.

Bayardo and Agrawal (1999) also show that mining the upper support/confidence border identifies the most interesting rules according to different interestingness measures. These measures include support, confidence, conviction, lift, laplace, gain and the measure of Piatetsky-Shapiro. More information on interestingness measures used in this thesis is given in Section 3.4.

This approach is used for the generated rules of ASELSAN Inc. data with some additional restrictions. Bayardo and Agrawal's SC optimality approach decreases the number of rules to a great extent. However, during elimination some information covered in the antecedent is lost. For example, consider the two rules:

r_1: Attr1 = a, Attr2 = b → Attr4 = d with S_1, C_1

r_2: Attr3 = c, Attr5 = e → Attr4 = d with S_2, C_2

Assume that $S_1 > S_2$ and $C_1 > C_2$. Then, according to SC optimality, $r_1 > r_2$. However, the antecedents of these rules convey totally different information. Eliminating r_2 may lead to information loss within the final rule set. Hence, we add a restriction to the SC optimality application. Of the two rules satisfying the SC optimality conditions, we eliminate one rule only if its antecedent is more specific than the other's. We still prefer to call these rules SC optimal rules. For example, consider the rule:

r_3: Attr1 = a, Attr2 = b, Attr5 = e → Attr4 = d with S_3, C_3

Assume again $S_1 > S_3$ and $C_1 > C_3$. Then r_3 is eliminated since their antecedents satisfy $A_1 \subset A_3$. That is, antecedent of r_3 includes the antecedent of r_1 and some additional items. Rule r_1 is more general than r_3 (or r_3 is more specific than r_1) and we regard rule r_3 redundant since every case identified by r_3 is identifiable by r1.

More formally, given the rules r_i and r_j, having $B_i = B_j$, r_i is redundant if:

- r_j is more general than r_i, $A_i \subset A_j$ and

- r_j is preferred to r_i in the partial order of support and confidence: $r_i \leq_{sc} r_j$.

A Java application was coded to eliminate the redundant rules according to our definition of SC optimality and form the non-eliminated rule set. This function reads the rules from the SQL database. For each rule, it compares support and confidence of the rule with all other more general rules that have the same consequent. If both support and confidence of the rule are less than or equal to those of a more general rule, then this rule is eliminated.

For each data set of ASELSAN Inc., the algorithm is applied in each run with different parameter settings for the minimum support and confidence. The experiments were conducted to obtain a rule set that best covers the data set (supported by some portion of data) with as small number of rules as possible.

3.3.3 Elimination Using Metarules

Berrado and Runger (2007) proposed an approach for organizing and grouping the association rules. Their approach is based on finding metarules, rules that express the associations between the generated rules themselves. According to the metarules, the generated rules are later grouped in sets as equivalent rules. In this section, this approach and its application to ASELSAN Inc. data are explained in detail.

Berrado and Runger's approach treats the rule statements as items and consist of finding one way associations between these previously generated rules. These are the rules with one item in the antecedent and one item in the consequent and are called metarules. One way association, rather than multi way is preferred for conciseness in the metarules and to overcome the difficulty in combining antecedents in different rules.

Formally, let I be the set of items (attribute-value pairs) available in the data set, T is the set of n tuples, where each tuple t_j contains an item set from the items in I. R is the rule set generated by Apriori. Let $R' = \{r'_1, r'_2, \ldots, r'_m\}$ represent the set of rules in R which have the same consequent. A rule r'_i from

R′ is supported by tuple t_j or tuple t_j supports rule r'_i if all of the items in the antecedent of rule r'_i are items of t_j. This relationship is shown as $r'_i \subseteq t_j$.

The first step in Berrado and Runger's approach is creating a new set of tuples of rules. This is denoted as $Q = \{q_1, q_2, \ldots, q_l\}$ where $l \leq n$. Every element q_j of Q is a new item set consisting of the generated rules as its items such that
$q_j = \{r'_i \in R' \mid r'_i \subseteq t_j\}$.

The second step is applying the Apriori algorithm to the set Q. The purpose is grouping of the rules by the tuples shared. Since only one way relations between rules are searched, number of items in the antecedent is set to one. The minimum support threshold can be set to zero so as to investigate all the relationship between all the rules. However, the authors advise that confidence levels should be set high. The rules formed from Q are the metarules and they are in the form of $r'_i \rightarrow r'_j$.

Let MR be the set of metarules from the metarules set. The next step is grouping the equivalent rules by analyzing the metarules in set MR. Consider the rule r'_i from R′. Let OUT_i denote the set of rules r'_j from R′ where in the metarules any r'_j is the consequent when r'_i is the antecedent, i.e.
$OUT_i = \{r'_j \in R' \mid r_i \rightarrow r_j \in MR\}$
Let IN_i refer to the set of rules r'_j from R′ where in the metarules any r'_j is the antecedent when r'_i is the consequent:
$IN_i = \{r'_j \in R' \mid r_j \rightarrow r_i \in MR\}$.
Then, consider the two metarules:
MR1: $r_i \rightarrow r_j$ with C_{mri}
MR2: $r_j \rightarrow r_i$ with C_{mrj}
Where $C_{mri.}$ and C_{msrj} denote the confidence of metarules MRi and MRj respectively. Berrado and Runger (2007) declare r_i and r_j equivalent if the following conditions hold:

- Mutuality: $r_i \in OUT_j$ when $r_i \in IN_j$
- Identical spans: $OUT_i \setminus \{r_j\} = OUT_j \setminus \{r_i\}$
- Identical covers: $IN_i \setminus \{r_j\} = IN_j \setminus \{r_i\}$

where $OUT_i \setminus \{r_j\}$ denotes the set of rules in OUT_i excluding rule r_j. Furthermore if $C_{mri} = C_{mrj} = 100\%$ then the two rules r_i and r_j are supported exactly by the

same tuples. Thus, from the metarules, equivalent rule sets can be formed. An equivalent rule set is formed for each group of equivalent rules.

Although metarules approach aims to group and organize the rules, it can also be used for the elimination of the rules (Berrado and Runger, 2007). Regardless of its support or confidence, if a rule in an equivalence set is more specific (or complex) than another rule in the same set, then the more specific rule can directly be eliminated, as the specific rule is automatically implied by the generalization (besides they are equivalent).

For the application of this approach, two Java functions were developed. The first algorithm reads the rules from the SQL Server database, reads the tuples in the data set and creates the set Q. Then, Apriori is run for this data set to generate the metarules. After the metarules are formed by Apriori, the second algorithm is used for grouping the equivalent rules according to the metarules.

For ASELSAN Inc. data, the created data set Q is transformed into the transactional data format and this is called the transactional data set for metarules. The set of tuples Q is created for each run with the specified parameters of Apriori. To identify the effects of the two elimination approaches (SC optimality and metarules), the metarules approach is applied directly to the original rules as well as the rules that pass the SC optimality.

3.4 Validation

As a final step in the thesis work, we test the validity of the approach on some test data. The rule set with the minimum antecedent support of 1% and the minimum confidence level of 50% is chosen for validation. This rule set contains the rules remaining after all of the elimination steps. The approach is validated using the test data retrieved separately from the training data. It is assumed that the tuples are time independent. Hence, rather than randomly selecting the test data from the original data set, new data is retrieved from the SAP system. The retrieval and preprocessing steps described in Section 3.1 for the training data is also applied to the test data. The amount of test data is taken as 25% of

the training data. Test data covers the period that immediately follows the training data period.

Some interestingness measures measured over test data are used for evaluating the rules using the test data. Geng and Hamilton (2006) define the interesting measures as a "broad concept that emphasizes:

- Conciseness
- Generality/Coverage
- Reliability
- Peculiarity
- Diversity
- Novelty
- Surprisingness
- Utility
- Actionability/Applicability".

Geng and Hamilton (2006) also mention that a good interestingness measure should include both generality and reliability. A rule is general if it is supported by a large number of tuples in the data set. Rule support and antecedent support can be used to measure the generality of the rule. A rule is reliable if the association described by the rule (consequent) is valid for a large portion of the tuples that support the antecedent. Confidence or a dependence measure such as lift or added value can be used to measure the reliability of the rule. Consider a rule with many items in its antecedent. If this rule is supported over the minimum support threshold, its confidence will be high and hence reliable, but its support is low and it is not a general rule. When we remove one item from the antecedent the rule becomes a more general (less complex) rule. However, as its support increases the frequency of its consequent decreases resulting in a loss in confidence. Then the rule is less reliable. These two properties can also be observed together. This is the case if the rule has few items, a high support and still a high confidence.

Lift is the ratio of confidence to the consequent support:

$$L = \frac{P(B \mid A)}{P(B)}$$

For example, if there exists a rule A → B with confidence 80% and the support of B is 10%, then the lift is calculated as eight meaning that having the antecedent items in the rule gives a fairly interesting information concerning the consequent. Consider another example with a rule C → D having the confidence 10% and the support of consequent is 10%, then the lift is calculated as unity. This means, having the antecedent items does not make a significant difference in the probability of observing the consequent. Thus, rules with a lift value different from one will be more interesting than the rules with the lift value close to one (SPSS Inc., 2007).

Added value is the difference between confidence and consequent support, i.e.
$AV = P(B|A) - P(B)$.
For the two examples above added value is calculated as 70% for rule A → B. This means that having the antecedent makes a significant contribution to the probability of observing the consequent. For the rule C → D above, the added value is zero meaning that having the antecedent items in the rule does not add any more information. Hence this rule is not interesting according to this measure.

Another measure proposed by Lavrac et al. (1999) discussed by Geng and Hamilton (2006) is weighted relative accuracy, which is defined as:
$WRAcc = P(A)* (P(B|A) - P(B)) = P(A) * AV$.
This measure combines the antecedent support and the added value.

In the survey by Geng and Hamilton (2006), Tan et al. (2000) is given as a reference to an interestingness measure called the IS measure. Tan et al. (2000) state that using support for eliminating rules is appropriate since support eliminates mostly uncorrelated or negatively correlated rules. The IS measure includes both support and an interest factor helping to identify the interesting rules while pruning the uncorrelated rules. IS is defined as:

$$IS = \sqrt{I * P(AB)}$$

Here I is the ratio between the joint probability of the consequent and the antecedent (rule support) to the product of their individual probabilities (support of consequent and support of antecedent). I is given by:

$$I = \frac{P(AB)}{P(A) * P(B)} = \frac{P(B|A)}{P(B)}$$

I is also the ratio of confidence to consequent support (equivalent to the lift). Thus, IS measure includes confidence, consequent support and rule support. This is a measure that combines generality and reliability. The value of IS measure increases as the confidence increases, consequent support decreases and rule support increases.

For the test data, six measures are chosen for analysis concerning the two criteria of interestingness (generality and reliability). Two of them, the antecedent support and the rule support, measure generality as an interestingness criterion. For reliability, confidence and lift measures are chosen. The measures combining both generality and reliability are the IS measure and the weighted relative accuracy.

3.4.1 Comparative Evaluation for Validity

For each of the remaining rules after the elimination steps, these measures are calculated on the test data. The results of these calculations show the performance of the rule in the test data. Hence, internal validity of our approach depends on the measures calculated both for training and test data sets, i.e. for the rules if a measure calculated on training data set is close to its value in the training data set, then our approach is internally validated. For defining the closeness, ranges are formed. For each measure, the number of rules falling into certain ranges or intervals in the training data and test data are compared. The intervals are determined according to the:

- Type of the measure,
- Minimum and maximum value of measure in training and test data
- Distribution of rules to the intervals.

For example, confidence level has an overall range of 0-100%. In general, equal width (10%) intervals are used for confidence as long as the number of rules in some interval is not too low or too high. In such cases intervals are combined or halved.

Let r_i be a rule in the set of rules remaining after the elimination steps for the training data. Let set of tuples T denote the training data and T' denote the test data. We say that r_i is valid if it is also supported by the test data and the value of the interestingness measure calculated using the test data is close to the value found using the training data. These two conditions are defined formally in the following way:

- $S_i(T') >= S_{min}$

Where $S_i(T')$ is the support of rule r_i in the test data and S_{min} is the minimum antecedent support specified to generate the rules from the training data.

- $M_i(T') = M_i(T)$

Where M is the interval of the measure (such as confidence, support, and so on.) that falls in. The subscript denotes the rule id.

The results show that for production about 90% of the generated rules from the training data are supported by the test data. Hence, about 10% of the rules are further eliminated since they are not supported by the test data for the production rules. For the supply rules generated in training data about 70% of the rules are not supported in test data. The results will be further discussed in Section 4.2.3.

For the internal validation of the approach, confidence is used as the performance measure of the rules. It should be noted that this is only internal validity. This validation does not check the validity of the rules. Validity of the rules themselves is external and requires the contribution of Quality Departments.

For the presentation of the rules to the Quality Department, a rule set is classified in three types of consequents: quality problem, discovery location and assigned cause. These were formed covering a high percentage of tuples in both training and test data sets. The rules that are not supported in the test data set are removed from the final rule set since they are not validated. This rule set still includes many rules. Our aim is presenting a rule set with as small numbers of rules as possible with a high support and confidence. For this aim a final ranking is made. IS measure is used for the ranking being a measure that takes into consideration both support and confidence. While generating the rules, we aimed

to present the Quality Department the maximum information that can be extracted from the data set. Low minimum support and confidence levels serve to this aim. Thus, the run results for the minimum support and confidence levels as low as possible were validated by the test data. Then the rules were order in descending order of IS measure. Finally, the rule set presented to the Quality Department included rules with rather low levels of support and confidence to convey all information that can possibly be gathered in descending order of IS measure.

CHAPTER 4

RESULTS

4.1 Data Pre-processing

Data pre-processing is the first phase of this work. This phase covers:

- Retrieval of the data from the SAP system
- Selection of the data fields (attributes) to be used
- Solving the problems related with the data
- Discretization of range fields

The following sections describe these steps in detail.

4.1.1 Selection of Attributes and Retrieval of Data

The data is entered into the database by means of a transaction screen for quality notifications. An example print of this screen is given in Figure 4.1.

At the background, the data is stored in three main tables (QMEL, QMFE, QMUR) and the master data tables where the text and other information about attribute values are stored. In fact, the main tables include a very large amount of redundant data with many fields. Some of the fields are not valid for ASELSAN Inc. and are left blank. These fields are not considered during data retrieval. Thus, an elimination of the fields was made before extracting data from the SAP system.

The entity relationship and active fields used in ASELSAN Inc. for the main tables is given in Figure 4.2. An (S) associated with a data field in the figure indicates that the field is used for supply notifications, and a (P) indicates that it is used for production notifications. The other fields are common to both types of notifications. A detailed description of these selected fields is given in Appendix A.

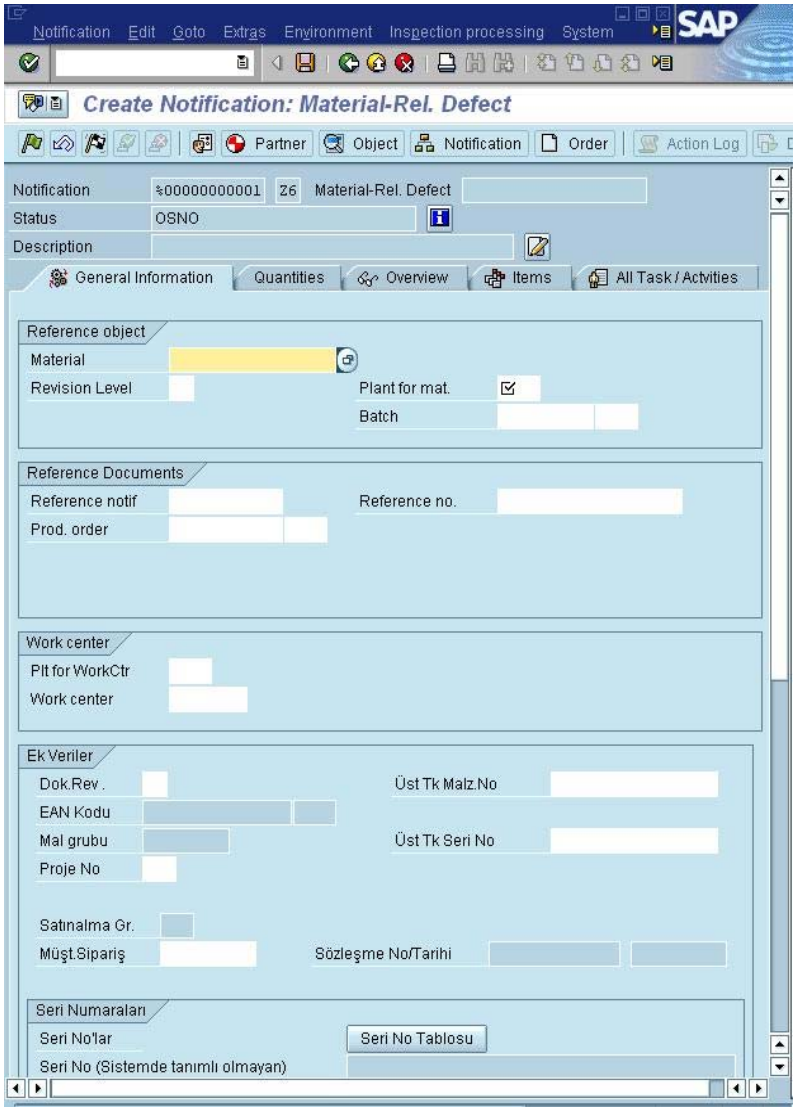


Figure 4.1. Creating a quality notification in SAP.

Of these three tables, QMEL table stores the general data of the notification such as the plant, notification type, material, and so on. This is called the heading information. QMFE table stores the items (one or more problems) of the

notification. The primary key for this table is the notification ID and the item number. These two tables are related with each other with the notification ID field. QMUR table stores the task information and the cause of problems detected. This table also has notification ID and task ID as the primary key. However, QMUR table does not have an entity relationship with QMFE table. It is only related with QMEL table such that different "cause" information can be entered for a single notification. Although the cause may differ for each problem in the same notification, with this entity relationship structure, "cause" cannot be directly related with a "problem" in that notification. Therefore, the Quality Department enters only one cause for each notification even if the notification includes more than one problem. For this reason, it is assumed that the cause entered is valid for all problems in the notification.

QMEL	
QMNUM	***Notification ID***
QMART	Notification Type (S or P)
MATNR	Material Number
MATKL	Material Group
LIFNUM	Vendor (S)
MAWERK	Material Production Plant
HERSTELLER	Manufacturer Number (S)
EKORG	Purchasing Organization (S)
BKGRP	Purchasing Group (S)
FERTAUFNR	Production Order Number (P)
FERTAUFPL	Production Oder Plan Number (P)
PNLKN	Plan Step Number (P)
ARBPL	Object Id for Workcenter (P)
DISPO	Project
UMATN	Assembly Number
MGEIG	Internal Nonconforming Quantity
MGFRD	External Nonconforming Quantity
MGEIN	Unit Of Measure
BZMNG	Reference Quantity
RKMNG	Complaint Quantity
RGMNG	Return Quantity
DENMG	Inspected Quantity
KABMG	Accepted Quantity
HURMG	Scrap Quantity
OGKMG	Used As Is Quantity
YISMG	Rework Quantity
ONRMG	Repaired Quantity

1 — n

QMFE	
QMNUM	***Notification ID***
FENUM	***Notification Item Number***
OTKAT-OTGRP-OTEIL	Location of Nonconformity
FEKAT-FEGRP-FECOD	Problem Code
T_ARBPL	Workcenter (S)
ANZFEHLER	Number of Nonconforming Units

1 — n

QMUR	
QMNUM	***Notification ID***
URKAT-URGRP-URCOD	Cause Code

Figure 4.2. Entity relationships and active fields.

Data is retrieved from the SAP system by using left outer join (using all tuples in QMEL and using the related tuples from QMFE). A sample is given in Figure 4.3. In this figure, for notification ID = 1, there are 3 tuples in QMFE (items) table. Since QMEL (heading) and QMUR (task) tables have one to one relationship, in the final structure we have built, there are three tuples for this notification each with a different problem but the same cause information.

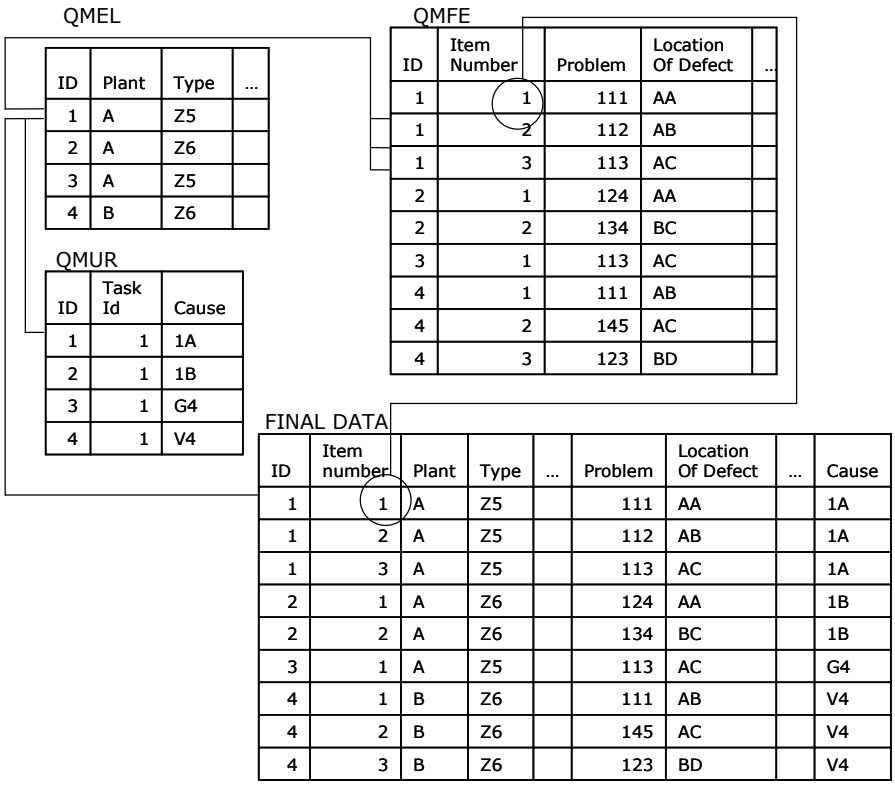


Figure 4.3. Sample tuples from main tables and the structure built to retrieve data.

In the final data structure, the text value information corresponding to the attribute values are retrieved from master data tables. Master data tables store all information belonging to an attribute such as text, catalog number, and so on. When a new attribute value needs to be added to the system, the first place to add the data is its master table. A function was written to retrieve the data set to be mined in the final data structure. The data set formed by this final data structure consists of two types of attributes. The first type is the attributes that

give information on the notification. The second type is the quantity attributes. These attributes are listed in Figure 4.4.

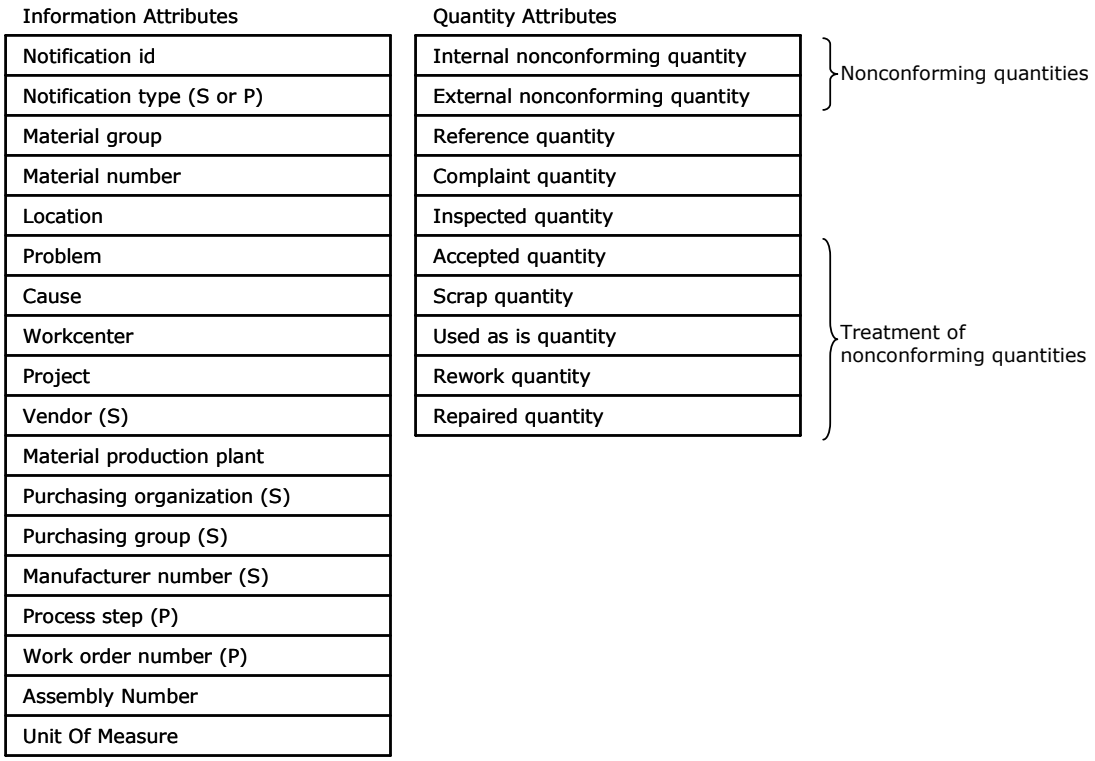


Figure 4.4. Attributes in the data set.

As mentioned in the problem statement given in Chapter 1, ASELSAN Inc. has three plants where mainly two types of notifications are used for supply and production. The selected attribute values differ for each type of plant and notification, i.e. a problem may only belong to one notification type and one plant. All plants have their own master data resulting in the need for dividing the data. Even if we applied data mining to the entire data set, each resulting rule would point to one plant and to one notification type. Thus, we divide the data into six data sets according to three plants and two notification types. Table 4.5 shows the distribution of the data by volume. The data set covers the notifications entered between January 2005 and December 2007.

Table 4.1. The distribution of data according to the plant type and notification type.

Plant	Notification Type	Number of Tuples
Plant A	Supply	4069
	Production	3050
Plant B	Supply	4464
	Production	6346
Plant C	Supply	464
	Production	46
No plant specified		1599
	TOTAL	20038

4.1.2 Solving the Problems Related with the Data

The next step is to examine the data for solving the problems if there are any. The problems are classified in three groups: Missing data, incorrect entries and inconsistent data.

The treatments of these problems differ according to the attribute type as mentioned in Section 3.1. For missing values:

- If there is a way to fill in the values, then the missing fields are filled in.
- Otherwise, these tuples are deleted according to the type of the attribute.

For example, a missing value is unacceptable for material, location, problem or cause since the aim of creating a notification is to enter this information. Therefore, these records are deleted. Some material group values are missing. Fortunately, master data table of material, stores the material group information. Hence, it was possible to fill in these missing values. Entering a value is optional for some attributes. For example the project attribute was left blank in 9162 of 20038 tuples. Although it is not a must field, it is always filled for defined projects and left empty for general purpose projects. 4015 of these 9162 tuples belong to plant A where the production is mainly general purpose and so is the supply. This explains that the missing values are left blank

intentionally and are for general purpose projects. So these missing values are filled with a dummy project code "9999". Some missing values existed for vendor in supply type of notifications. When these data are analyzed further, it is observed that there is a purchase order related with these notifications which means that the mentioned material is moved between plants. Plants in ASELSAN Inc. consider each other as a customer and as a vendor. A plant is treated as if it is another company. However, from these purchase orders it was not possible to understand from which plant the material movement was made. Therefore, for these tuples a new vendor with the code "999999" pointing ASELSAN Inc. itself was introduced and this field was recoded.

The most difficult phase in data pre-processing step is correcting the incorrect entries. Although it is not always possible to detect such entries, some rules could be used to check data consistency. Since we know the history of data, it was known that before October, 2006 there was only one type of notification. That is, in 10066 of 20038 tuples, only one type was entered for all notifications. Hence, rules of entering data were examined to understand the correct notification type. The rules checked are:

- If the values for supply type of attributes (given with (S) in Table YY1) are filled and production type of attribute values are not entered, then this tuple is classified as a supply type of notification.
- If the values for production type of attributes (given with (P) in Table YY1) are filled and supply type of attribute values are not entered, then this tuple is classified as a production type of notification.

The same rules are applied for 1599 tuples missing the plant data.

Finally, inconsistency checks are made on the data. The following items are checked:

- If there are any two fields filled, one attribute of supply type and the other of production type, then there is inconsistency. For example, purchasing group (only valid for supply type) and work order (only valid for production type) should not be filled together. However, there existed such tuples. The Quality Department examined such data and determined which kind of notification it is.

- If the total quantity of nonconforming units treated (scrap, rework and so on) is larger than the total quantity of nonconforming units (see Figure 4.4), then there is inconsistency. In these tuples nonconforming quantities are left blank and these attributes are marked as missing. Also, if all of the attributes concerning treatment of nonconforming units are left blank, then these attributes are marked as missing.
- If the material groups have any inconsistency with the materials' master data, then tuples are changed according to the master data.

These checks were applied and necessary changes were made on the data. A detailed description of the problems and their impact on the distribution of the data are given in Appendix B. A summary of this appendix is given in Table 4.2. In this table, the "+" sign means that since the type of the data set is changed, new tuples are added to that data set. Similarly, "-" sign means that some tuples are excluded from that data set. Tuple counts with no sign means that in these tuples only the attribute values are changed.

Table 4.2. Problems about data and their impact on the data distribution.

Problem Category	Treatment	Number of Tuples							Total Number of Tuples
		Plant A		Plant B		Plant C		No Plant	
		Supply	Production	Supply	Production	Supply	Production		
Initial number of tuples		**4069**	**3050**	**4464**	**6346**	**464**	**46**	**1599**	**20038**
Missing data	Deleted		-3		-51			-4	-58
	Filled in	4026	2955	652	1802	8	5		0
Incorrect entry	Plant type recoded		+137	+3	+1		+5	-146	0
	Notification type recoded	+2817	-2817	+2568	-2568	+3	-3		0
	Plant and notification type recoded	+2		+10	+1430	+2		-1444	0
Inconsistent data	Deleted								0
	Recoded	+5		+1	-1			-5	0
Final number of tuples		**6893**	**367**	**7046**	**5157**	**469**	**48**	**0**	**19980**

As a conclusion, all incorrect entries were either recoded or deleted. Missing data was filled in if it was possible to find the information, and all inconsistencies were corrected. For the remaining missing values which could not be filled, a "#" sign was inserted to mark the field as missing. The use of this sign is explained in methodology chapter. For each plant, two data sets were formed according to

the notification type (supply and production) and the data was made ready for further analysis.

Next, the attribute values of each data set were examined and for each data set histograms were plotted to understand the variety of the attributes' values. Although, the results show that we have a very sparse data with wide range of attribute values, the frequencies were high enough to form rules after mining process. Table 4.3 shows the number of different values an attribute takes for each data set. Although the notifications are based on the information of material, the frequencies of material values are very low. As can be seen from the table 4.3, there are 2866 different kinds of material for 6893 tuples for Plant A's supply type of data corresponding to an average support of 0.034% assuming they are uniformly distributed. Therefore, it would not be surprising that in the association rule set there are no rules including this attribute.

Table 4.3. Total Number of Different Values of Attributes.

	Total Number of Different Values					
	Supply			Production		
Attributes	Plant A	Plant B	Plant C	Plant A	Plant B	Plant C
Material group	198	211	44	55	69	8
Material number	2866	2496	175	268	1364	24
Location	8	33	17	24	36	8
Problem	95	160	15	68	163	21
Cause	19	38	24	31	46	16
Workcenter	7	12	8	23	38	12
Project	1	43	24	35	40	9
Vendor	448	429	25			
Purchasing organization	3	3	3			
Purchasing group	60	63	14			
Manufacturer number	67	429	0			
Process step				0	138	6
Work order number				157	1469	8

4.1.3 Discretization of Quantity Attributes

Quantity attributes are not included in Table 4.3 since the unit of each value may differ for different types of material and in different purchase orders from different vendors. Thus a need to process this data arises. To solve this problem, the ratios of the quantity attributes to the reference quantity are calculated. Following attributes are added to the data set and all original quantity attributes are excluded from the analysis:

- Int. nonconforming Qty. Ratio: The ratio of internal number of non-conformities to the reference quantity.
- Ext. nonconforming Qty. Ratio: The ratio of external number of non-conformities to the reference quantity.
- Return Quantity Ratio: The ratio of returned quantity to the reference quantity.
- Accepted Quantity Ratio: The ratio of accepted quantity to the reference quantity.
- Scrap Quantity Ratio: The ratio of scrap quantity to the reference quantity.
- Rework Quantity Ratio: The ratio of reworked quantity to the reference quantity.
- Used As is Quantity Ratio: The ratio of used as is quantity to the reference quantity.

In the tuples with missing values for reference quantity all attributes above are recoded by "#" as missing. In the tuples where both internal and external nonconforming quantities are left blank, the corresponding ratio fields are recoded by "#" as missing. In the tuples where all quantity attributes on treating the nonconformities are left blank, the corresponding ratio fields are all recoded by "#" as missing.

The newly added ratio fields have values in the range [0, 1]. Since Apriori algorithm cannot propose a range in the rules, these new attributes need to be discretized. Different discretization schemes are analyzed only for one data set and the chosen discretization is used for all other data sets (see Appendix C). Missing values remaining the same, zero values are recoded as "B". In most of the cases, only one kind of treatment (rework, scrap, etc.) or only one kind of

nonconformity (internal or external) exists in the data sets. That is why missing and zero values cover almost 80% of the data on the average. The remaining data is divided into two where C denotes the tuples with values in the interval (0 – 0.25], and D denotes the tuples with values in the interval (0.25 – 1.0]. The main purpose for such discretization is, for each range, to achieve a sufficiently large number of tuples to satisfy the specified support level. Only then C and D can be observed in the association rules formed by Apriori. Table 4.4 shows the distribution of the discretized values for Plant B's production type of notifications. It can be read from the table that, in 43.69% of tuples internal nonconforming quantity ratio was zero.

From the data sets formed, Plant B's production and supply data sets are chosen to be analyzed in the scope of this thesis.

Table 4.4. Discretization of quantity attributes for Plant B's production type of data set.

		Disc. Int. Nonconf. Qty. Ratio	Disc. Ext. Nonconf. Qty. Ratio	Disc. Scrap Qty Ratio	Disc. Accepted Qty Ratio	Disc. Used as is Qty Ratio	Disc. Repaired Qty Ratio	Disc. Rework Qty Ratio	Disc. Return Qty Ratio
#	missing	24.86%	24.86%	37.91%	37.91%	37.91%	37.91%	37.91%	37.91%
B	0	43.69%	30.93%	60.48%	48.28%	61.24%	60.42%	9.73%	57.40%
C	(0 - 25%]	12.90%	19.51%	0.74%	1.45%	0.27%	1.24%	21.64%	2.70%
D	(25%-100%]	18.56%	24.70%	0.87%	12.35%	0.58%	0.43%	30.72%	2.00%

4.2 Generation and Elimination of Rules

After the data preprocessing step, data was made ready for mining operations. As described in Chapter 3, first, rules are generated. Then, three methods are applied for elimination. Rules are generated by SPSS Clementine version 11.1 on a 2.00 GHz laptop with 2 GB RAM and two CPUs. Same hardware is used for the applications of all algorithms for elimination. The functions are written with Java code in NetBeans 5.5.1 IDE.

4.2.1 Decision on the parameters for Apriori Algorithm

Association rule mining is applied for the data sets by Clementine's Apriori Algorithm. The parameters required are decided before application of the algorithm. There is a setting for each attribute if it is going to be input, output or both. This is the setting of the direction. Other settings are the minimum antecedent support, the minimum confidence level and the maximum number of items in the antecedent.

More than one attribute can be chosen as output (consequent) in a single Apriori run. Apriori, then, produces rules for each consequent chosen. However, choosing the consequents in the single run increases the computation time. Therefore, different runs are made regarding each consequent specified. Only Location, Problem and Cause are chosen as the consequents, because, the Quality Department classified these as the most useful attributes. Basically, they want to know why certain problems occur at certain locations and how the causes of these problems emerge.

The algorithm is applied for three different antecedent support and confidence levels. Low support and confidence levels result in more rules; hence more of the interesting rules may be mined. As the support and confidence levels increase, the possibility of eliminating interesting rules also increases. The decision of these two parameters was based on the minimum levels that could be selected with the available hardware resources. With the computer used for the application, the minimum levels were selected as 1% for antecedent support and 50% for confidence in the production data set and 2% for antecedent support and 50% for confidence in the supply data set. To observe the effect of confidence on the number of rules generated, 80% confidence level was also tried. Another run to observe the strongest rules was made with 10% antecedent support and 80% confidence level.

Another parameter is the "number of items in the antecedent". Consider the case where there exists a rule with one attribute as consequent and all other attributes are included in the antecedent. Then the maximum number of the

items in the antecedent is one less than the total number of attributes. However, it was not possible for us to use this value for this parameter due to the exponentially increasing computational time of the algorithm. The number of items in the antecedent part of the association rules increases as we decrease the antecedent support and confidence levels. Because, when the support is high, the possibility of seeing combinations of many attributes in the antecedent is low and vice versa. However, even if the support is very low for a rule, it may still be an interesting rule for a different measure. Therefore, for each data set, support and confidence are set at their lowest values and different runs are made increasing the number of the items in the antecedent. Thus, maximum number of association rules are tried to be achieved. Looking at the results of the run for SC optimal rules,

- If the maximum number of the items in the antecedent in the rules found is equal to the parameter's prespecified value, then another run is made increasing the limit.
- If there are no rules having the prespecified number of the items in the antecedent, then that limit is selected for all runs of that data set.

A sample for the choice of this parameter for Plant B's production data is given in Table 4.5. For three of the consequents, when the number of the items in the antecedent is specified as eight, the maximum number of the items in the antecedent used was seven in the resulting SC optimal rules.

Apriori Algorithm is also used in forming the metarules used in rule elimination. This time the algorithm is run for transactional data. The parameters to be decided for Apriori application have to be chosen again. As mentioned in methodology chapter, this time runs are for transactional data and Rule ID's are chosen both as the consequent and as the antecedent. Number of the items in the antecedent should be set at 1. Since the aim of forming metarules is finding the identical rules which apply to same tuples, choosing a high confidence level is important. Thus, 80% confidence level is chosen. Support, on the other hand, should be chosen as small as possible. Because rather than how frequent a rule is seen in tuples, it is more important that the rules are seen together. However, to see the effect of support, a high (10%) and a low (1% or 2%) level is tried for the runs. Due to the hardware restrictions, sometimes it was not

possible to run Apriori with as low as a level of 1%. In these situations, the minimum realizable antecedent support is used.

After the parameters are decided on, for each data set the procedure mentioned in Chapter 3 is applied. This work is presented in the following sections.

Table 4.5. The distribution of antecedent item counts among the rules with 1% support and 50% confidence level for Plant B's production data.

Consequent	Num. of Ant.	Num. of rules	Distribution	
			Num. of ant.	Num. of Rules
Location	5	962	1	29
			2	179
			3	338
			4	263
			5	153
	6	1017	6	55
	7	1027	7	10
	8	1027	8	0
Problem	5	601	1	4
			2	74
			3	208
			4	224
			5	91
	6	625	6	24
	7	630	7	5
	8	630	8	0
Cause	5	1996	1	25
			2	221
			3	641
			4	695
			5	414
	6	2136	6	140
	7	2157	7	21
	8	2157	8	0

4.2.2 Elimination of Rules

Elimination of the rules is performed in three successive steps. These are explained in further detail in Chapter 3. The first step is eliminating the rules that are formed due to the recoding of missing values (rules having "#" value for any item in their antecedent). This step eliminates the rules that do not make sense due to the presence of "#" assigned to missing values. The second step involves eliminating the rules that are not SC optimal. The final step is searching the metarules to find equivalent rules and eliminating the specific rules in each equivalent rule sets.

The latter two steps, eliminating non-SC optimal rules and eliminating the specific rules from equal rules by using metarules could each be used either alone or one after the other. To see the affect of each version on rule elimination, results are analyzed by

- Using only SC optimality,
- Using only Metarules, and
- Using both methods.

4.2.2.1 Elimination of Rules for Production Data

Summary of results for the runs for Plant B's production data set is given in Table 4.6. The table includes the results for the three steps: Apriori, SC Optimality and Metarules. In the Apriori section, the number of the items in the antecedent is eight for all runs. The column "Number of Rules" shows the count of rules produced by Clementine's Apriori algorithm. The column "Number of rules (I)" gives the count of rules remaining after the first elimination process. For SC optimality, the algorithm given in Section 3.3.2 is applied to the rules remaining after the first elimination process. For example, in Table 4.6, the first row is the application of Apriori algorithm for consequent "Location". With 1% support and 50% confidence level, 79882 rules are formed. Among these rules, 62128 include "#" value in their antecedent items and are eliminated. The remaining number of rules is 17754. Further elimination is done on these 17754 rules. In the SC Optimality section of the table, if only Metarules is used for that run, then "Applied/not applied" field is set as "not applied". In this case,

"Number of Rules (II)" column for that row is set to the "Number of Rules (I)". When SC Optimality is applied, this column gives the number of SC optimal rules. The last section gives the results for application of Metarules. "Number of Tuples" is the amount of data generated in the transactional data set and "Number of Metarules" is the number of rules formed by Apriori transactional data algorithm in Clementine with the specified antecedent support and confidence Levels.

The metarules show the associations between the rules formed previously. From these metarules equivalent rule sets are found. After the metarules are found by Apriori, all equivalent rules are grouped in one set. If a rule has no equivalent, then it is counted in the "Unique Rule" column. "Number of Rules Sets Cover" column counts the rules included in the sets given in "Multiple Rules". In this step, for the rules in the same set, if there exists a rule Ri $\subset$ Rj (Rj is more general), then the specific rule is eliminated. "Number of Rules III" column gives the number of rules remaining after all three elimination steps.

Consider the first row in Table 4.6. "Number of Rules (II)" is 1027 meaning that there are 1027 SC optimal rules with 1% antecedent support and 50% confidence level. The transactional data set is formed with 1027 rules and Apriori run resulted in 41389 metarules. When the sets are created as described in Section 3.3.3, 193 sets were formed. These sets covered 594 rules. 199 of these rules were more specific than another rule in its set. So when these rules were eliminated 1027 – 199 = 828 rules were left.

After the application of Apriori algorithm, the number of rules found with different support values gives an idea about how sparse the production data set is. No rules were found for Plant B's production data when the consequent was selected as "Cause" or "Problem" with 10% antecedent support and 80% confidence level. When antecedent support is decreased to 1%, number of rules increases drastically. This was inevitable and foreseen when the histograms for the data were analyzed. The wide variety in values of attributes resulted in large numbers of rules.

Table 4.6. Runs results for Plant B's production data set.

Consequent	Apriori				SC Optimality		MetaRules							
											Num. of Sets with		Num. of Rules Sets Cover	Num. of Rules (III)
	Ant. Sup. %	Conf. %	Num. of Rules	Num. of Rules (I)	Applied/ Not applied	Num. of Rules (II)	Ant Sup. %	Conf. %	Num. of Tuples	Num. of MetaRules	Unique Rule	Multiple Rules		
Location	1	50	79882	17754	applied	1027	1	80	219771	41389	433	193	594	828
Location	1	50	79882	17754	applied	1027	10	80	219771	15	1025	1	2	1009
Location	1	50	79882	17754	not applied	17754	3	80	1100232	233745	16583	300	1171	16883
Location	1	50	79882	17754	not applied	17754	10	80	1100232	272	17744	4	10	17748
Location	1	80	35072	5657	applied	256	1	80	25157	2213	123	54	133	229
Location	1	80	35072	5657	applied	256	10	80	25157	75	242	6	14	254
Location	1	80	35072	5657	not applied	5657	5	80	436350	135576	5016	171	641	5187
Location	1	80	35072	5657	not applied	5657	10	80	436350	5509	5560	27	97	5587
Location	10	80	6	2	-	2	-	-	-	-	-	-	-	2
Problem	1	50	25560	8054	applied	630	1	80	34362	29991	156	164	474	479
Problem	1	50	25560	8054	applied	630	10	80	34362	1	628	1	2	630
Problem	1	50	25560	8054	not applied	8054	1	80	25896	14758	7717	138	337	7997
Problem	1	50	25560	8054	not applied	8054	10	80	25896	1	8052	1	2	8053
Problem	1	80	1768	155	applied	31	1	80	10206	98	13	7	18	26
Problem	1	80	1768	155	applied	31	10	80	10206	86	16	6	15	26
Problem	1	80	1768	155	not applied	155	1	80	20481	3970	28	36	127	87
Problem	1	80	1768	155	not applied	155	10	80	20481	3752	142	70	13	154
Problem	10	80	0	0	-	0	-	-	-	-	-	-	-	0
Cause	1	50	71653	20481	applied	2157	1	80	152278	169386	609	522	1548	1579
Cause	1	50	71653	20481	applied	2157	10	80	152278	1	2155	1	2	2162
Cause	1	50	71653	20481	not applied	20481	3	80	1200031	280625	19087	272	1394	19642
Cause	1	50	71653	20481	not applied	20481	10	80	1200031	1	20479	1	2	20488
Cause	1	80	15808	3712	applied	364	1	80	23257	9392	92	94	272	274
Cause	1	80	15808	3712	applied	364	10	80	23257	2	362	1	2	363
Cause	1	80	15808	3712	not applied	3712	1	80	219771	36016	3371	68	341	3495
Cause	1	80	15808	3712	not applied	3712	10	80	219771	2	3710	1	2	3711
Cause	10	80	0	0	-	0	-	-	-	-	-	-	-	0

Since many missing fields are treated as having values by recoding, rules produced by Apriori included many meaningless rules. Consider the case in the run where "Problem" is chosen as the consequent with 1% support and 50% confidence level in Table 4.6. Almost 70% of the 25560 rules were eliminated due to such rules with missing values.

Recall from Section 3.3.2 that, for a pair of rules if Ri $\subset$ Rj (Rj is more general than Ri), then Si <= Sj. In this case if Ci <= Cj then Ri is redundant. When these redundant rules are eliminated, SC optimal rules are obtained. This approach also eliminates almost 90% of the remaining rules.

For elimination through metarules approach, transactional data set is formed by the algorithm described in methodology chapter. With this transactional data set, metarules are formed by the Apriori algorithm. Due to the hardware restrictions, the minimum possible support value was larger than 1% for some runs. This mostly occurred in cases where the number of tuples in transactional data was over than one million.

Metarules approach achieved, on the average, 25% reduction in the SC optimal rules. This reduction was achieved in 80% confidence level for metarules. When SC optimality was not applied, metarules alone could only reduce 1% of the rules on the average. However, this resulted in very large final number of rules since the number of rules is already large when SC optimality is not applied. The best results were observed when metarules is applied after SC optimality.

In the next step, the final rules formed in each run are analyzed. Tables 4.7 through 4.9 show the distribution of consequent values in the data set and the rules. Table 4.7 shows the results of the runs where "Location" is chosen as the consequent for Plant B's production data set. The location values are sorted in descending order of the number of tuples in the data set. In 1132 of 5157 tuples, location was BL1 in the data set. When Apriori algorithm is run with 1% antecedent support and 50% confidence level, the SC Optimal rules having the consequent value of BL1 covered 1076 of the 1132 tuples. Since metarules approach is interested only in eliminating from equivalent rules which are seen in the same tuples, the number of tuples covered by the rules does not change

when antecedent support and confidence levels are different for metarules. For BL1, when the final SC Optimal rules are eliminated by metarules application, 1027 rules are formed when antecedent support is 10% and confidence level is 80%. Of the 1027 rules, 539 have BL1 as the consequent.

Table 4.7. Rule statistics for runs where "Location" is the consequent for Plant B production data.

Value of Location	Number of Tuples for Location	Ant. Sup. = 1%, Conf. = 50%			Ant. Sup. = 1%, Conf. = 80%		
		Num. of Tuples Covered by Rules	Metarules		Num. of Tuples Covered by Rules	Metarules	
			A. Sup.= 10% Conf. = 80%	A. Sup.= 1% Conf. = 80%		A. Sup.= 10% Conf. = 80%	A. Sup.= 1% Conf. = 80%
BL1	1132	1076	539/1027	401/828	946	117/254	102/229
BL2	895	782	157/1027	143/828	709	82/254	79/229
BL3	696	381	71/1027	61/828	58	5/254	4/229
BL4	616	404	38/1027	33/828	223	11/254	10/229
BL5	479	382	60/1027	57/828	46	1/254	1/229
BL6	249	140	8/1027	8/828	108	5/254	5/229
BL7	201	133	63/1027	54/828	N/A	N/A	N/A
BL8	164	77	2/1027	2/828	N/A	N/A	N/A
BL9	114	101	89/1027	69/828	94	33/254	28/229
Others	611						
SUM	5157	3476			2184		

Table 4.8. Rule statistics for runs where "Problem" is the consequent for Plant B production data.

Value of Problem	Number of Tuples for Problem	Ant. Sup. = 1%, Conf. = 50%			Ant. Sup. = 1%, Conf. = 80%		
		Num. of Tuples Covered by Rules	Metarules		Num. of Tuples Covered by Rules	Metarules	
			A. Sup.= 10% Conf. = 80%	A. Sup.= 1% Conf. = 80%		A. Sup.= 10% Conf. = 80%	A. Sup.= 1% Conf. = 80%
BP1	583	330	339/630	243/479	55	5/26	5/26
BP2	515	266	96/630	74/479	42	1/26	1/26
BP3	349	291	52/630	48/479	46	4/26	4/26
BP4	299	N/A	N/A	N/A	N/A	N/A	N/A
BP5	269	188	121/630	94/479	81	6/26	6/26
BP6	261	236	11/630	10/479	236	10/26	10/26
BP7	207	37	6/630	6/479	N/A	N/A	N/A
BP8	178	55	5/630	4/479	N/A	N/A	N/A
Others	2496						
SUM	5157						

BL7 and BL8 are not seen when confidence level is 80%. Although the consequent support is high, the antecedent values for these locations are not confident enough to form rules. On the overall 3476 of 5157 tuples are covered by the rules corresponding to 67% coverage when antecedent support is 1% and

confidence level is 50%. Coverage drops to 2184 tuples (42%) when confidence level is increased to 80%.

Tables 4.8 and 4.9 provide the same statistics for the consequents "Problem" and "Cause", respectively.

Table 4.9. Rule statistics for runs where "Cause" is the consequent for Plant B production data.

Value of Cause	Number of Tuples for Cause	Ant. Sup. = 1%, Conf. = 50%			Ant. Sup. = 1%, Conf. = 80%		
		Num. of Tuples Covered by Rules	Metarules		Num. of Tuples Covered by Rules	Metarules	
			A. Sup.= 10% Conf. = 80%	A. Sup.= 1% Conf. = 80%		A. Sup.= 10% Conf. = 80%	A. Sup.= 1% Conf. = 80%
BC1	1039	746	362/2157	262/1579	204	103/363	62/274
BC2	778	694	1266/2157	896/1579	407	141/363	107/274
BC3	654	552	92/2157	72/1579	370	23/363	21/274
BC4	488	370	59/2157	56/1579			
BC5	395	372	342/2157	259/1579	314	87/363	75/274
BC6	335	188	26/2157	24/1579	87	2/363	2/274
BC7	260	236	7/2157	7/1579	236	7/363	7/274
BC8	179	N/A	N/A	N/A	N/A	N/A	N/A
BC9	161	N/A	N/A	N/A	N/A	N/A	N/A
BC10	126	N/A	N/A	N/A	N/A	N/A	N/A
BC11	109	N/A	N/A	N/A	N/A	N/A	N/A
BC12	84	N/A	N/A	N/A	N/A	N/A	N/A
BC13	65	N/A	N/A	N/A	N/A	N/A	N/A
BC14	59	N/A	N/A	N/A	N/A	N/A	N/A
BC15	57	52	3/2157	3/1579	N/A	N/A	N/A
Others	368						
SUM	5157	3210			1618		

For a data set, it is possible to combine the final rules for three different consequents generated with the same antecedent support and confidence level. When these rules are combined, there exist rules such that consequent of the first rule is in the antecedent of the second rule and vice versa. Since our purpose is to discover associations, it does not matter whether an attribute is in the antecedent or consequent, and, there still exist equivalent rules after the combination of these rules. Hence, the run with the highest coverage of a data set and with the lowest number of rules is chosen and metarules elimination method is applied on these combined rules. This is chosen as the run with antecedent support as 1%, and confidence level as 50%. SC optimal rules eliminated by the metarules application are selected. Thus, 828 rules with "Location" as the consequent, 479 rules with "Problem" as the consequent and

1579 rules with "Cause" as the consequent constitute a set of 2886 rules. For these final rules, again metarules are formed with antecedent support chosen as 1% and confidence level chosen as 80% on the transactional data. 496 rule sets formed by the metarules included equivalent rules, and 1323 of the rules were covered in these rule sets. Finally, after the elimination from the equivalent rule sets, the number of remaining rules is 2632.

Next, the distributions are analyzed for "Location", "Problem" and "Cause". Table 4.10 shows the values of "Location" covered in the overall final rules. "No location" is added to the table in order to show that there are rules that do not include location information in their consequent or antecedent. The location attribute values given in the table are included in 4614 tuples in the data set. The actual coverage by rules is 3716. This corresponds to 72% (3716 / 5157) of coverage on the whole data set. This means that in the worst case, the rules produced covers 72% even if other rules with different consequents do not cover any more tuples.

Table 4.10. Distribution of location values among production data set and rules.

Value of Location	Num. of Tuples for Location	Number of Tuples Covered by Rules	Number of Rules
BL2	895	836	322/2632
BL4	616	444	68/2632
BL6	249	140	10/2632
BL1	1132	1091	742/2632
BL5	479	397	82/2632
BL7	201	157	98/2632
BL10	68	35	1/2632
BL8	164	104	6/2632
BL9	114	102	87/2632
BL3	696	410	97/2632
no location			1119/2632
SUM	4614	3716	2632

Tables 4.11 and Table 4.12 show the same statistics for "Problem" and "Cause", respectively. These 2632 rules, in total, cover 4358 of 5157 tuples corresponding to approximately 85% of the data and include all so called interesting rules.

Table 4.11. Distribution of problem values among data set and rules.

Value of Problem	Num. of Tuples for Problem	Number of Tuples Covered by Rules	Number of Rules
BP9	138	134	11/2632
BP10	68	65	2/2632
BP11	72	69	3/2632
BP3	349	291	58/2632
BP4	299	47	2/2632
BP5	269	236	153/2632
BP6	261	236	11/2632
BP12	108	94	15/2632
BP13	54	46	2/2632
BP2	515	408	156/2632
BP14	91	68	17/2632
BP15	103	70	1/2632
BP16	77	26	1/2632
BP17	57	46	1/2632
BP8	178	134	13/2632
BP7	207	164	55/2632
BP1	583	414	536/2632
no problem			1595/2632
SUM	3429	2548	2632

There are some interesting points to be mentioned in these tables. There are cases where only a few rules cover around 80% of related tuples. For example, the problem "BP10" in Table 4.7 is seen in 68 of 5157 tuples. Only two rules cover 65 of 68 tuples corresponding to 95% coverage. This may mean that these two rules are very generic.

There are also cases where the number of generated rules exceeds the number of tuples they are generated from. "Problem" with the value "BP1" is an example of such a situation where there are 536 rules covering 414 of the tuples.

Table 4.12. Distribution of cause values among data set and rules.

Cause	Num. of Tuples for Cause	Number of Tuples Covered by Rules	Number of Rules
BC6	335	218	63/2632
BC7	260	236	11/2632
BC8	179	96	9/2632
BC15	57	56	6/2632
BC12	84	38	3/2632
BC3	654	579	95/2632
BC5	395	379	275/2632
BC4	488	400	87/2632
BC9	161	49	2/2632
BC2	778	700	999/2632
BC1	1039	799	302/2632
no cause			780/2632
SUM	4430	3550	2632

4.2.2.2 Elimination of Rules for Supply Data

The analysis conducted for production data is repeated for supply data. However, different characteristics of the data sets yield different results. The variety in values of attributes and the number of tuples included in the mining process affect the number of rules produced. Supply data set consists of 7046 tuples while there are 5157 tuples in the production data set. It is expected to have more rules for supply data set. This was valid for cause and location. It was not possible to run Apriori with 1% support for the consequent "cause" with the available hardware resources. Hence, 2% support is used for all consequents. This shows that this data is less sparse for these consequents than the production data set. This is because the variety in values of the attributes is less in supply data set.

Table 4.13. Runs results for Plant B's supply data set.

Consequent	Apriori				SC Optimality		Metarules						
	Ant. Sup. %	Conf. %	Num. of Rules	Num. of Rules (I)	Applied/ Not applied	Num. of Rules (II)	Ant Sup. %	Conf. %	Num. of MetaRules	Num. of Sets with		Num. of Rules Sets Cover	Num. of Rules (III)
										Unique Rule	Multiple Rules		
Location	2	50	162814	78621	applied	3835	1	80	322556	1061	659	2774	2333
Location	2	50	162814	78621	applied	3835	10	80	689937	2799	296	1036	3405
Location	2	50	162814	78621	not applied	78621	-	-	-	-	-	-	78621
Location	2	80	68999	35593	applied	661	1	80	71605	138	120	523	414
Location	2	80	68999	35593	applied	661	10	80	12299	504	54	157	617
Location	2	80	68999	35593	not applied	35593	-	-	-	-	-	-	35593
Location	10	80	3346	1838	applied	10	10	80	38	1	4	9	9
Location	10	80	3346	1838	applied	10	1	80	38	1	4	9	9
Location	10	80	3346	1838	not applied	1838	10	80	2532398	175	482	1663	907
Problem	2	50	9816	911	applied	109	1	80	2843	15	26	94	76
Problem	2	50	9816	911	applied	109	10	80	2843	15	26	94	76
Problem	2	50	9816	911	not applied	911	1	80	1288289	818	26	93	879
Problem	2	50	9816	911	not applied	911	10	80	1288289	818	26	93	879
Cause	2	50	190865	91935	applied	3775	1	80	1415999	814	870	2961	1818
Cause	2	50	190865	91935	applied	3775	10	80	526957	2506	233	1269	3003
Cause	2	50	190865	91935	not applied	91935	-	-	-	-	-	-	91935
Cause	2	80	158294	81339	applied	3208	1	80	241007	1002	640	2206	2173
Cause	2	80	158294	81339	applied	3208	10	80	619315	1870	281	1338	2519
Cause	2	80	158294	81339	not applied	81339	-	-	-	-	-	-	81339
Cause	10	80	11953	6274	applied	833	1	80	177621	148	154	685	450
Cause	10	80	11953	6274	applied	833	10	80	177621	148	154	685	450
Cause	10	80	11953	6274	not applied	6274	1	80	-	-	-	-	6274

However, this observation is not valid for "problem". The runs with 2% of support and 80% of confidence did not produce any rules for the consequent "problem" whereas 50% of confidence with the same support produced 9816 rules. This shows that the problem definitions are so general that for many different values of attributes the same problem code is filled in so that the confidence threshold cannot be exceeded. Hence, with the consequent "problem" data is rather sparse in the supply data.

Results for the application of the three phase elimination process to the supply data set are summarized in Table 4.13. It is seen that for each run almost 50% of the rules were eliminated due to the missing data (rules including # sign).

As in the production case, using SC optimality approach in supply data set eliminated over 90% of the redundant rules. However, metarules approach could not be used for the runs where SC optimality was not applied in the supply data set with the available hardware. Because, the number of the rules was very high and the transactional data generated prior to forming the metarules had millions of tuples. The hardware restrictions prevented us from running Apriori for these. Metarules approach with 1% antecedent support and 80% confidence eliminated on the average 30% of the remaining rules after SC optimality was applied.

Table 4.14. Rule statistics for runs where "Location" is the consequent for Plant B supply data.

Value of Location	Number of Tuples for Location	Ant. Sup. = 2%, Conf. = 50%			Ant. Sup. = 2%, Conf. = 80%		
		Num. of Tuples Covered by Rules	Metarules		Num. of Tuples Covered by Rules	Metarules	
			A. Sup.= 10% Conf. = 80%	A. Sup.= 1% Conf. = 80%		A. Sup.= 10% Conf. = 80%	A. Sup.= 1% Conf. = 80%
BL8	2475	2475	1986/3405	1666/2333	2475	146/617	142/414
BL2	870	870	1014/3405	371/2333	870	377/617	197/414
BL6	641	641	168/3405	140/2333	554	62/617	50/414
BL11	497	453	126/3405	84/2333	406	2/617	1/414
BL12	418	256	15/3405	13/2333	256	15/617	13/414
BL13	354	292	96/3405	59/2333	247	15/617	11/414
	1791						
	7046	4987		2333	4808		414

For each consequent the distribution of its values in the resultant rules is given in Tables 4.14 through 4.16. For location in Table 4.14 70% of the tuples in the supply data set is covered in the rules with a minimum support of 2% and

confidence of 50%. The tuples with location data of BL8, BL2 or BL6 are all covered in the rules for this run.

For the consequent "problem" in Table 4.15, number of tuples covered by the rules is very small. This gives a hint about a problem in the definition and use of problem codes again.

The runs for the consequent "cause" show that for all problems four types of causes were observed in Table 4.16. These rules covered %83 of the whole data set for supply.

Table 4.15. Rule statistics for runs where "Problem" is the consequent for Plant B supply data.

Problem	Number of Tuples for Problem	Ant. Sup. = 2%, Conf. = 50%		
		Num. of Tuples Covered by Rules	Metarules	
			A. Sup.= 10% Conf. = 80%	A. Sup.= 1% Conf. = 80%
BP9	1137	104	30/76	30/76
BP4	651	135	9/76	9/76
BP2	448	N/A	N/A	N/A
BP10	378	N/A	N/A	N/A
BP11	248	N/A	N/A	N/A
BP12	205	N/A	N/A	N/A
BP13	204	N/A	N/A	N/A
BP14	193	192	37/76	37/76
Others	3582			
SUM	7046	431		

Table 4.16. Rule statistics for runs where "Cause" is the consequent for Plant B supply data.

Cause	Number of Tuples for Cause	Ant. Sup. = 2%, Conf. = 50%			Ant. Sup. = 2%, Conf. = 80%		
		Num. of Tuples Covered by Rules	Metarules		Num. of Tuples Covered by Rules	Metarules	
			A. Sup.= 10% Conf. = 80%	A. Sup.= 1% Conf. = 80%		A. Sup.= 10% Conf. = 80%	A. Sup.= 1% Conf. = 80%
BC10	338	137	27/3003	19/1818	N/A	N/A	N/A
BC16	178	178	122/3003	84/1818	N/A	N/A	N/A
BC1	5428	5428	2785/3003	1680/1818	5428	2519/2519	2173/2173
BC17	169	168	69/3003	35/1818	N/A	N/A	N/A
Others	933						
	7046	5911					

The best runs that cover a high percentage of the tuples with fewer rules is the run with 2% of minimum antecedent support and 50% of confidence level and then the application of metarules with 1% of minimum support and 80% confidence level. Hence, three of the runs (for each consequent) are chosen for the analysis.

For each rule set, there are rules which are similar to some other rules in the other rule sets. The only difference in these rules is that the consequent of one rule is in the antecedent part of the other and vice versa. More elimination on the rules is possible when these rule sets are combined in a unique set and similar rules are eliminated. Applying the metarules approach on this final set can group these similar rules. Hence, the rules resulting from the runs for each consequent; 2333 rules for location, 76 rules for problem and 2360 rules are combined as the final rule set of 4227 rules for the next process.

Metarules approach is applied on this final set of rules. 713 sets having multiple rules were formed by Apriori covering 1691 of 4227 rules. When the elimination of rules having the same set of items is made, 4012 final rules have been induced. The frequencies of values for the three consequents supported in these rules are given in Tables 4.17 through 4.19 for location, problem and cause, respectively.

Table 4.17. Distribution of location values among supply data set and rules.

Value of Location	Num. of Tuples in Data Set	Num. of Tuples Covered by Rules	Number of Rules
BL8	2475	2475	1749
BL2	870	870	351
BL6	641	641	166
BL11	497	476	91
BL13	354	292	75
BL12	418	385	14
BL15	275	244	12
BL14	259	177	1
BL16	147	119	1
no location			1552
SUM	5936	5679	4012

These frequencies are expected to be higher than the runs having one type of attribute as the consequent. This is because these frequencies include the coverage of the attributes being in either the consequent or the antecedent part of the rules. For example, in Table 4.14 there are six different values of location meaning that location is the consequent in the rules where as in Table 4.17 nine different values exist meaning that these nine types of location is observed either in consequent or in antecedent.

Table 4.18. Distribution of problem values among supply data set and rules.

Value of Problem	Num. of Tuples in Data Set	Num. of Tuples Covered by Rules	Number of Rules
BP9	1137	1081	405
BP2	448	434	90
BP4	651	586	78
BP12	205	194	64
BP14	193	192	35
BP10	378	311	21
BP15	190	186	14
BP16	204	190	7
BP17	153	141	4
BP18	148	143	3
BP19	248	172	2
BP20	185	160	2
BP21	142	128	2
no problem			3285
SUM	4282	3918	4012

Table 4.19. Distribution of cause values among supply data set and rules.

Value of Cause	Num. of Tuples in Data Set	Num. of Tuples Covered by Rules	Number of Rules
BC1	5428	5428	2521
BC16	178	178	84
BC10	338	245	20
BC17	169	168	20
BC8	290	149	11
no cause			1356
SUM	6403	6168	4012

4.2.3 Validation

When the final rules are formed, the final step is validating our approach using some test data. The test data is retrieved from the SAP system for the period that immediately follows the training data period (from January 2008) by the same processes mentioned in Section 4.1. For testing purposes, the number of tuples was decided to be 25% of the number of tuples in the training data. The next four months data was sufficient to form the test data set; 1003 tuples for Plant B's production, 1350 tuples for Plant B's supply data were used.

The approach is verified by comparing interestingness measures calculated in training and test data sets. Recall from Section 3.4 that if rules's interestingness measures found with the test data set have a value close to the ones found with the training data, our approach is validated. Hence, for all 2632 rules of the production data and 4012 rules of the supply data, a comparison is made between their interestingness measures in training data set and test data set. Table 4.20 shows the distribution of rules over confidence ranges in the training data set for the production. Similarly, for each range the distribution of rules over the confidence ranges in test data set is included. Consider the first range, [95-100%] for instance. In the training data set 100 of 2632 rules fall into this range of confidence. For the test data set, the confidence of each rule is calculated. Of the 100 rules, 64 still fall into this range. We see in the histograms given next to the table that, the distribution in the test data tends to be close to the one in the training data for the production rules.

Similar results are reported for supply data in Table 4.21. However, in this case the percentages of rules that are not supported or supported with confidence less than 50% constitute the highest bars. These two correspond to 72% of the supply rules. In the first confidence range, (95,100], 23% of the rules fall into the same confidence range in the training data. Although this percentage is not as high as the production data set, it is still, the highest percentage of the rules in the same confidence range as in the training data set proceeding the no support or low confidence range (0, 50) percentages. The bar with the highest number of rules slides to the left as the confidence ranges decrease.

Table 4.20. The distribution of rules among confidence ranges in Plant B's production training and test data sets.

Training		Test	
Confidence Range %	Number of Rules	Confidence Range %	Number of Rules
(95, 100]	100	(95, 100]	64
		(90 - 95]	12
		(80, 90]	3
		(70, 80]	5
		(60, 70]	1
		[50, 60]	1
		No support	14
(90, 95]	86	(95, 100]	20
		(90 - 95]	10
		(80, 90]	10
		(70, 80]	20
		(60, 70]	10
		[50, 60]	
		[0, 50)	5
		No support	11
(80, 90]	294	(95, 100]	26
		(90 - 95]	16
		(80, 90]	47
		(70, 80]	32
		(60, 70]	40
		[50, 60]	39
		[0, 50)	63
		No support	31
(70, 80]	443	(95, 100]	17
		(90 - 95]	17
		(80, 90]	87
		(70, 80]	43
		(60, 70]	107
		[50, 60]	65
		[0, 50)	93
		No support	14
(60, 70]	600	(95, 100]	24
		(90 - 95]	10
		(80, 90]	44
		(70, 80]	71
		(60, 70]	101
		[50, 60]	83
		[0, 50)	250
		No support	17
[50, 60]	1109	(95, 100]	40
		(90 - 95]	10
		(80, 90]	46
		(70, 80]	102
		(60, 70]	143
		[50, 60]	188
		[0, 50)	552
		No support	28

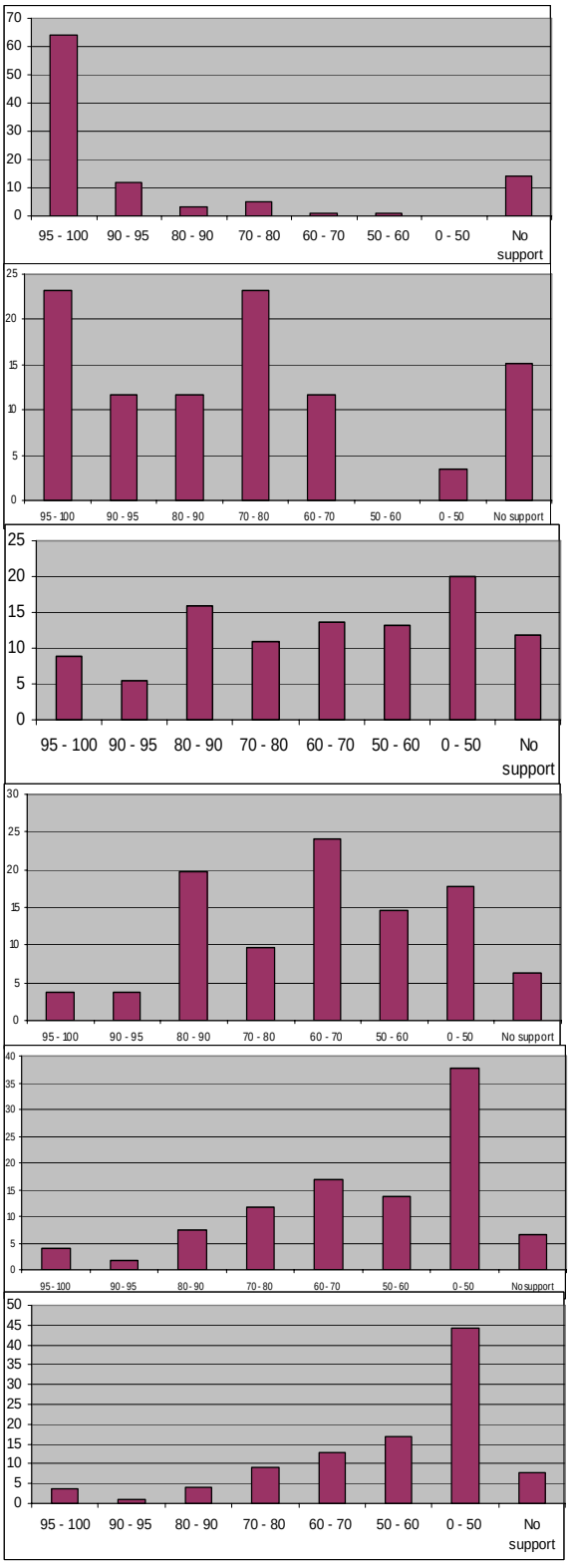
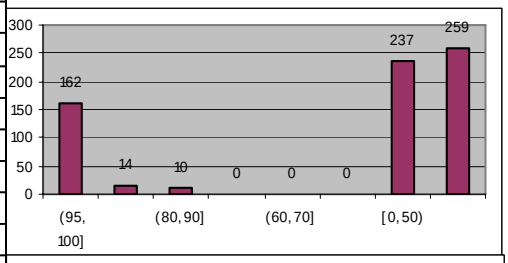
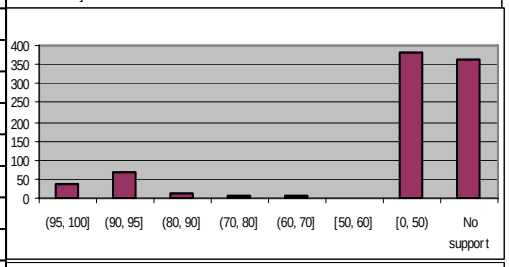
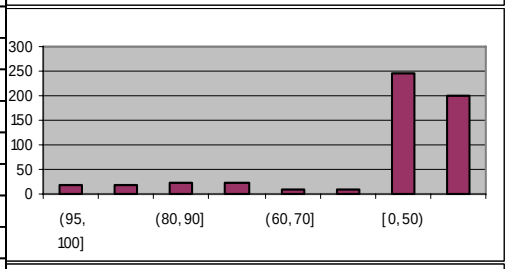


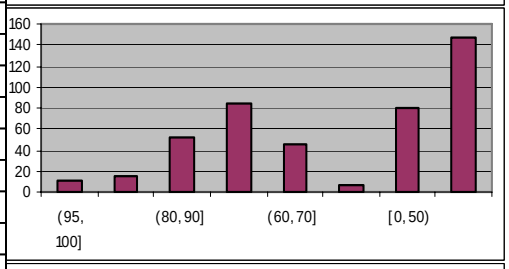
Table 4.21. The distribution of rules among confidence ranges in Plant B's supply training and test data sets.

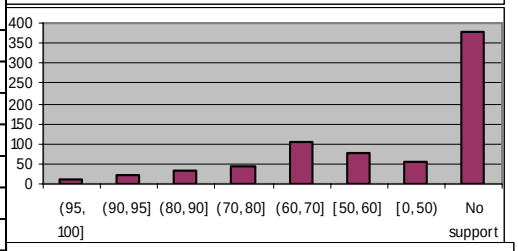
Training		Test	
Confidence Range %	Number of Rules	Confidence Range %	Number of Rules
(95, 100]	682	(95, 100]	162
		(90, 95]	14
		(80, 90]	10
		(70, 80]	-
		(60, 70]	-
		[50, 60]	-
		[0, 50)	237
		No support	259
(90, 95]	875	(95, 100]	37
		(90, 95]	68
		(80, 90]	11
		(70, 80]	6
		(60, 70]	7
		[50, 60]	1
		[0, 50)	379
		No support	366
(80, 90]	547	(95, 100]	16
		(90, 95]	20
		(80, 90]	23
		(70, 80]	23
		(60, 70]	9
		[50, 60]	9
		[0, 50)	245
		No support	202
(70, 80]	442	(95, 100]	11
		(90, 95]	16
		(80, 90]	51
		(70, 80]	84
		(60, 70]	45
		[50, 60]	6
		[0, 50)	81
		No support	148
(60, 70]	725	(95, 100]	13
		(90, 95]	23
		(80, 90]	35
		(70, 80]	45
		(60, 70]	104
		[50, 60]	75
		[0, 50)	53
		No support	377
[50, 60]	741	(95, 100]	6
		(90, 95]	3
		(80, 90]	17
		(70, 80]	28
		(60, 70]	41
		[50, 60]	76
		[0, 50)	145
		No support	425

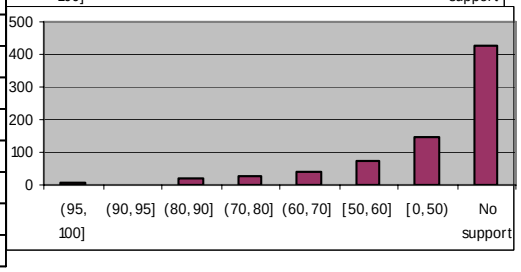












"No support" row in Tables 4.20 and 4.21 shows the number of rules where rule support is zero for test data. While 218 of 2632 (8%) production rules do not win any support in the test data set, 2732 of 4012 (72%) supply rules are not supported in the supply test data. This may be due to the seasonal characteristic of data for supply. As new projects are conducted, new types of materials are supplied from new vendors. Moreover, the supply period is shorter than the production period. All rules generated from training data covers a time period of two years. Within two years, the same projects may continue in production. However, according to the production plan, the types of supplied materials may change.

The same analysis is repeated for all interestingness measures mentioned in Section 3.4 and details are given in Appendix D. The results support the validity of the rules formed in the same way as the confidence level does. Tables 4.22 and 4.23 show the distribution of rules for different ranges of measures used for production and supply data sets, respectively. In the production data set for 218 of rules, rule support is zero. For Lift and IS measure, 103 rules have a value of zero meaning that they have no confidence.

Table 4.22. The distribution of number of rules among different measures in the production test data.

Rule Support %	Number of Rules
[10, 100]	8
[5, 10)	119
[2, 5)	595
[1, 2)	657
(0,1)	1035
0	218
SUM	2632

Antecedent Support %	Number of Rules
[10, 100]	54
[5, 10)	315
[2, 5)	1061
[1, 2)	515
(0,1)	572
0	115
SUM	2632

WRACC	**Number of Rules**
[4,)	79
[2, 4)	331
[1, 2)	701
[0,1)	1202
(..,0)	204
No support	115
SUM	2632

Lift	Number of Rules
[10, 100]	167
[5, 10)	521
[2, 5)	1351
(0, 2)	375
0	103
No support	115
SUM	2632

IS Measure	Numb er Of Rules
[4,)	260
[3, 4)	476
[2, 3)	686
[1, 2)	589
(0, 1)	403
0	103
No support	115
SUM	2632

Table 4.22 shows that all interestingness measures calculated in test data are close to the ranges in the training data for all rules. There is an exception; 218 of them which have no support in production test data set. Table 4.23 supports the idea that a great deal of the rules is not valid for supply data.

Table 4.23. The distribution of number of rules among different measures in the supply test data.

Rule Support %	Number of Rules
[10, 100]	202
[5, 10)	174
[2, 5)	469
[1, 2)	234
(0,1)	201
0	2732
SUM	4012

Antecedent Support %	Number of Rules
[10, 100]	515
[5, 10)	495
[2, 5)	703
[1, 2)	345
(0,1)	177
0	1777
SUM	4012

WRACC	Number of Rules
[4,)	197
[2, 4)	238
[1, 2)	347
[0,1)	427
(..,0)	1026
No support	1777
SUM	4012

Lift	Number of Rules
[10, 100]	142
[5, 10)	45
[2, 5)	561
(0, 2)	532
0	955
No support	1777
SUM	4012

IS Measure	Number of Rules
[4,)	365
[3, 4)	214
[2, 3)	300
[1, 2)	305
(0, 1)	96
0	955
No support	1777
SUM	4012

After all of the elimination processes, there still are many rules. Among these rules supported by the test data, 20 rules are selected for close examination by the Quality Department. The selection includes with one or two rules for each of the consequent values. Ten rules that have a confidence level over 80% and ranked top according to the IS measure and ten rules that have a confidence level higher than 90% and a support lower than 3% are included. These rules are given in Tables 4.24 and 4.25 respectively for production and supply data sets. The consequents of these rules were not disclosed and the Quality Department was asked to fill in the consequent. It was expected that, if there are rules that the Quality Department fails to guess, then these rules are

interesting in the sense that they provide unknown information. Unfortunately, Quality Department successfully discovered all the consequent values.

Finally, considering the values of these measures in test data set, a ranking is made so as to present all of the rules to the Quality Department in a meaningful manner. IS measure is chosen as the appropriate measure since it includes both the antecedent support and added value.

In Tables 4.24 and 4.25, the antecedent parts of the rules include all types of attributes used in the study. Interval ratio attributes stand for the interval that the ratio falls in. "B" stands for 0, "C" stands for the range (0, 25%] and D stands for (25%, 100%].

Mainly, it was observed that some information that was entered in the notification was useless. For example, location and material group attributes had similar meanings. Consider rule number 2 in Table 4.25. The rule is:

If material group is Cable Pack then the location is Wiring.

Both the antecedent and the consequent hold the same data and the rule does not convey any new information. Hence, if any rule having location as the consequent also has the material group in its antecedent, then these rules become redundant. This is because for a certain kind of material group, only one location is true. Hence, the location attribute carries redundant information. For a better result they can enter detailed information about the location of defect on the material. This situation is also observed for supply rules.

Some attributes are considered for revision by the Quality Department of Plant B. For example, the inspection work center holds the information where the nonconformity is observed. Consider the rule number 4 in Table 4.24:

If Problem is insufficient or no solder and the workcenter is 10401 then cause is Board Assembly.

Work center 10401 is the locality of the inspections made by Quality Department. There are three types of inspections the Quality Department executes; electrical tests, mechanical tests and eye inspections. Being included

in the rule does not give any new information because solder check can only be made by eye inspection. During production, the material passes through many work centers. Instead of entering the inspection work center, if the sequence of process work centers where the defect or nonconformity might occur were entered, then the relationships between the problem (or location or cause) and work centers could be established. For the supply of materials phase, the work center is filled in for the type of the inspection process. Hence, only for production notifications work center attribute does not give any new information.

Nevertheless, the rules helped the quality engineers to see the relationships of different attributes which they have never analyzed before. For example, the treatment of nonconforming quantities was not analyzed before. Many rules included these attributes in their antecedents. Consider the rule number 13 in Table 4.24. The rule is:

If material group is Printed Circuit Pack and Workcenter is 10401 (eye inspection) and the ratio of external quantity of nonconformities to the general reference quantity falls into the range (25%, 1] then location is printed circuit pack.

This shows that during production the nonconformities in Printed Circuit Pack exist due to external reasons, which should be analyzed further.

Table 4.24. 20 Rules ordered by IS Measure for Plant B's production data set.

Rule Num	Consequent	Antecedent	Rule Sup. %	Conf. %	IS Measure
1	Problem = Incorrect Measurement	Cause = Purchasing Inspection Unit	2	83.3	7.58
2	Location = Wiring	Material Group = Cable Pack	9.09	95.8	8.59
3	Cause = Purchasing Inspection Unit	Problem = Incorrect Measurement and Project = 14 and Internal Defective Ratio Interval = B	0.4	80	3.65
4	Cause = Board Assembly	Problem = Insufficient or no solder and Workcenter = 10401	0.5	100	1.87
5	Cause = Test/Material Error	Problem = Functional Defect in Material and Location = Printed Circuit Pack	4.4	84.6	4.45
6	Location = Elektromechanical Pack	Cause = EMM - Equipment and Material Group = Elektromechanical Pack	0.6	100	2.37
7	Cause = Test/Material Error	Process Step in Production Order = Electrical Test and Problem = Functional Defect in Material and Workcenter = 10401	5	84.8	4.75
8	Cause = Wiring	Problem = Wrong Assembly and Material Group = Cable Pack and Workcenter = 10401	0.1	100	1.77
9	Location = Printed Circuit Pack	Cause = Board Assembly and Material Group = Printed Circuit Pack	8.69	94.6	5.64
10	Location = Sistem Device Accessory	Problem = No Output Unit and Material Group = Sistem Device	2.8	87.5	3.56
11	Cause = Test/Material Error	Project = 16 and Problem = Functional Defect in Material and Rework Ratio Interval = C and Location = Printed Circuit Pack	1.2	100	2.53
12	Cause = Manufacturer/ Subcontractor	Workcenter = 20802 and Disc. Return Quantity/General Reference = C and Rework Ratio Interval = B and Disc. Scrap Quantity/General Reference = B	0.3	100	1.13
13	Location = Printed Circuit Pack	Material Group = Printed Circuit Pack and Disc. ExtNumber/GeneralReference = D and Workcenter = 10401	1.6	94.1	2.41
14	Location = Printed Circuit Pack	Material = x and Process Step in Production Order = Electrical Test	0.6	100	1.52
15	Location = Wiring	Problem = Wrong Assembly and Material Group = Cable Pack and Disc. Accepted Quantity/General Reference = B	1.1	100	3.05
16	Location = Wiring	Workcenter = 20901 and Disc. ExtNumber/GeneralReference = D	0.1	100	0.92
17	Problem = Incorrect Measurement	Cause = Purchasing Inspection Unit and Project = 14 and Disc. Repaired Quantity/General Reference = B	0.6	100	4.55
18	Problem = Incorrect Measurement	Cause = Purchasing Inspection Unit and Project = 14 and Disc. Return Quantity/General Reference = B	0.3	100	3.22
19	Cause = Test/Material Error	Process Step in Production Order = Electrical Test and Problem = Functional Defect in Material and Material Group = 01 and Location = Printed Circuit Pack	0.8	100	2.06
20	Location = Wiring	Material Group = Cable Pack and Workcenter = 90001 and Rework Ratio Interval = D	2	100	4.12

Table 4.25. 20 Rules ordered by IS Measure for Plant B's supply data set.

Rule Num.	Consequent	Antecedent	Rule Sup. %	Conf. %	IS Measure
1	Problem = Expiration day has passed	Location = Chemicals	16.96	99.13	9.42
2	Location = Mechanical Pack	PurcGroup = 31 and Material Group = B58000000 and Workcenter = 90002	9.78	100.00	7.18
3	Location = Metalic Parts	PurcGroup = EF and Accept Ratio Interval = B and Cause = Manufacturer/Subcontractor and Internal Defective Ratio Interval = B and Rework Ratio Interval = B	2.15	100.00	3.37
4	Location = Metalic Parts	Return Ratio Interval = D and Workcenter = 90002 and Cause = Manufacturer/Subcontractor	3.04	100.00	4.00
5	Location = Wiring	Problem = Design Error and Accept Ratio Interval = D and Used as is Ratio Interval = B and Cause = Manufacturer/Subcontractor	26.81	92.58	5.50
6	Cause = Manufacturer/ Subcontractor	Problem = Dimentional Measurement Error and Location = Metalic Parts and Rework Ratio Interval = B	0.67	100.00	1.46
7	Location = Ametalic Parts	Material Group = B60000000 and Workcenter = 90002 and Rework Ratio Interval = B	1.48	100.00	4.59
8	Location = System Mechanic, Thermo, Hardware	Material Group = 01 and Used as is Ratio Interval = D and Workcenter = 90001	2.81	90.48	1.76
9	Cause = Manufacturer/ Subcontractor	Location = System Mechanic, Thermo, Hardware and External Defective Ratio Interval = D	0.30	100.00	1.25
10	Location = Wiring	Material Group = C07010000 and Cause = Manufacturer/Subcontractor	1.85	96.15	6.67
11	Cause = Manufacturer/ Subcontractor	Material Group = C07010000 and Location = Wiring and Workcenter = 90001 and Internal Defective Ratio Interval = B	0.81	100.00	1.61
12	Cause = Manufacturer/ Subcontractor	Used as is Ratio Interval = D and Location = Metalic Parts and Workcenter = 90002 and External Defective Ratio Interval = D	0.67	100.00	1.46
13	Location = Wiring	Project = 13 and Used as is Ratio Interval = B and Rework Ratio Interval = B	1.19	100.00	5.83
14	Location = Wiring	Material Group = C07010000	1.56	100.00	6.68
15	Location = Metalic Parts	PurcGroup = HI and Used as is Ratio Interval = B and Accept Ratio Interval = B and Cause = Manufacturer/Subcontractor	1.04	100.00	2.34
16	Location = Metalic Parts	Problem = 9/MST-MK/108 and Workcenter = 90002 and External Defective Ratio Interval = D and Cause = Manufacturer/Subcontractor	0.74	100.00	1.98
17	Location = Ametalic Parts	Material Group = B60000000 and Workcenter = 90002 and Rework Ratio Interval = B	1.48	100.00	4.59
18	Location = System Mechanic, Thermo, Hardware	PurcGroup = HI and Material Group = 01 and External Defective Ratio Interval = D	2.52	100.00	1.75
19	Problem = Expiration day has passed	Location = Chemicals and Workcenter = 90002	2.59	100.00	8.63
20	Cause = Manufacturer/ Subcontractor	Problem = Design Error and Location = Wiring	0.37	100.00	0.67

CHAPTER 5

CONCLUSIONS

In this thesis, an implementation for analyzing the data of ASELSAN Inc. using association rules technique of data mining was studied. ASELSAN Inc. uses the SAP system as its enterprise database. The nonconformance data for two years from 2005 to 2007 stored in the SAP as notifications for defective parts were analyzed. These data included nonconformities observed during both supply and the production phases. The aim of the analysis was generating a rule set that extracts the maximum information from the data that covers as much data as possible with an acceptable number of rules produced. Hence, the results could be used for process improvement in supply and production.

Since there was not a way of relating the raw materials supplied and their usage in production such as a serial number or a related work order ID for the supplied raw material, the data sets were analyzed separately. The notifications included data such as the problem, cause of the problem, location of defect, and other information. A data pre-processing phase in our study constituted almost 30% of the effort in this thesis work. At the end of this phase, we gathered data with missing values marked, range values discretized and all other anomalies removed.

Data mining techniques in generating association rules were used to form rules to define the relationships between the attributes that may affect the quality separately in the supply and the production phases. Association rule mining was preferred rather than other rule induction methodologies. This is because the Quality Department was specifically interested in finding associations among attribute values. Furthermore, rule generation could be kept under close control,

and data was mainly categorical by nature. Apriori Algorithm was used to form these rules. This algorithm is a two step process where the first step generates the frequent item sets where the item set is supported by a minimum support threshold set by the user. In the second step, among the frequent item sets, the rules are formed by calculating the confidence of an item in the item set chosen as the consequent. To infer the best set of parameters, different threshold levels for support and confidence were set in running Apriori.

Association rules is a technique known to produce a large number of rules. In our case too, Apriori algorithm generated many rules where some of them were redundant and had to be processed further. Hence, a three phase elimination process was applied. The best run was decided as the one where the percentage of rules eliminated were high and the number of tuples that support the final rule set was also high. This way, we achieved a compact but concise set of rules.

For the aim of extracting the maximum information that the data set could produce, some missing data was coded with a special sign in the data preprocessing phase. Hence, in the rule set formed by Apriori many rules included this sign in their antecedent. Those rules would be meaningful only if this item with the special sign was removed from the antecedent. However, if the support of such a rule was above the minimum support threshold, then it must have been also formed as another rule. Hence, removing these rules with the sign did not cause any information loss. The first step of the elimination was removing these rules from the data set. 43% of the production rules and 78% of the supply rules generated had this sign of missing value in their antecedent and were removed.

The second elimination phase was a restricted version of removing the rules where SC optimality conditions did not hold (Bayardo and Agrawal, 1999). A rule is SC optimal if there are no other rules with the same consequent where the support and confidence levels are higher. Since the items in the antecedent parts may be different, using SC optimality is not suitable for eliminating rules. Hence, we restricted the conditions to eliminate only those rules with the same consequent. We removed the rules for which the antecedents were more specific than some other rules. We still called the remaining rules SC optimal although

these are more specialized than SC optimality requires. We still do not loose information with this process. This is because we only eliminate the specific rules with less support and less confidence. When an item is added to the antecedent part of a rule, support is decreased and if confidence does not increase, adding this item (making it more specific) does not give more information since we still have a more general rule with a higher confidence. This elimination process gave very good results. On average 90% of the rules in the production data set and 94% of the rules in the supply data set were eliminated in this phase.

The final elimination approach was using metarules (Berrado and Runger, 2006). A metarule identifies the relationship between any couple of rules. This means, these two rules are supported by the same tuples if the confidence of this metarule is 100%. From these metarules, equivalent rule sets are formed. Two rules are equivalent if both rules have the same associations with other rules. A rule can be eliminated if there is a more general (or less complex) rule in the same equivalence set. Since two rules in the same set are equivalent, removing the rule does not cause any information loss. To see the effect of support for the metarules, runs are made for two different support levels. Confidence is set to 80% as advised by the originators. When the support is set to low level (1%), the approach gave better results as expected.

This approach resulted in eliminating 10% of the rules for production data set and 25% of the rules elimination in supply data set whether the SC optimality is applied or not. However, when SC optimality is not applied, the number of the rules is very large and most of these rules are redundant. Hence, the best run is decided as the one with a low level of support and confidence where first the SC optimality and then the metarules approaches are applied. For the metarules application, low level of support with a high level of confidence gave the best result.

The application of metarules approach in literature is based on the same consequent. At first the runs were made separately for each consequent as stated by Berrado and Runger (2006). When all the elimination steps were completed a final rule set including the three consequents was formed. Some of these rules had the same items. Where one rule had the item in its antecedent,

the other included it in its consequent and vice versa. Hence, the final rule set formed still included redundant rules having the same meaning with the other rules and they had to be eliminated. We changed the implementation of metarules to include different consequents. Metarules approach was applied once more on this final rule set where different types of consequents are included in the rules. The rules having the same item set (union of the antecedent and the consequent) were grouped in the same equivalence set. In each equivalence set, the rule having the higher level in both support and confidence was preserved and the other rules were eliminated. If one of these levels was lower, then the comparison was made based on the lift.

The approach was validated by a separate test data set that included the subsequent four months' data. The tuples in these data sets were about 25% of the training data sets in size. For the production rules, 92% of the rules generated by the training data were supported by the test data. However, in supply rules this was only 68%. This may be because our approach was not so successful with the supply data or because the characteristics of the supply data change in time. For every new project, the supply period is much shorter than the production phase causing a faster pace of pattern change in the data.

The results were presented to the Quality Department and the Production Management. Quality engineers stated that the rules were meaningful and as expected. The resultant rules did not include any "interesting" rules that revealed unexpected relationships. The main reason lies in the original data. When these rules were analyzed it was understood that some of the attributes had similar information expressed multiple times such as the material group and the location. In ASELSAN Inc. a material group refers to a set of materials which are entered as well in the notifications. There were rules such that whenever one of the two items is in the consequent, the other item carrying the similar information was found in the antecedent. This shows the redundant work during the data entry as one of these attributes is not needed. More information could be derived if location was entered with more specifics than merely the material group, like subcategories on the materials. This situation was also observed for the work centers. During the production, material goes through many work centers for the process steps and quality check is made in the inspection work

centers. The work center in the notification holds the information of the work center where the non-conformity was observed i.e., the inspection work center. If it were entered as the work center where the defect or nonconformity really occurs, then the rules would help see the relations of the problems with the work centers. In general, there is a need to study the attributes, select useful ones, eliminate redundant ones, and even create new ones in some cases.

Although the rule elimination processes applied reduced the number of rules by over 90%, there were still many rules remaining. It was observed in the remaining rules that most of the rules included the same items but different combinations. Some of the rules were not eliminated since the SC optimality conditions were still valid. Grouping these rules having the same items and choosing one of them as a representative of these rules can be studied as a future work. There seems to be a need for more effective methods of eliminating redundant association rules.

Another future work may be applying this methodology on standard data sets. We have shown that the methodology we apply reduces the number of the rules by 90% for SC optimality and 25% for meta rules. For a better understanding of the performance of the approach, the performance on standard data sets may be analyzed.

This methodology can be applied by the Quality Departments to these two kinds of notifications (supply and production) data periodically. It takes approximately two hours to apply Apriori algorithm and all other functions of elimination phases. Although our methodology produces the set of rules presenting all of the information that can be generated from the data set, validating the rules themselves is still an open issue. In literature semantic based measures are used for such analysis (Geng and Hamilton, 2006). A utility function that reflects the goals of the practitioner is the most common type of semantic measures. The resultant rules of our approach can further be mined optimizing the utility function of the Quality Department which can be studied as a future work. Then fewer rules may be presented to the Quality Department and rules may be ranked according to their utility.

The change in the rules as time passes is observed during the thesis work particularly for supply data. This was observed because of different projects being executed in different time periods. The problem of time dependency of the tuples could be solved by mining the data on project basis. In the lifecycle of a project periodical mining can be made for the tuples related to that project. However, this kind of mining may not be able to produce the general rules valid in all projects due to support restriction. This may cause information loss. Data partitioning during Apriori may be another approach for this problem. The data can be partitioned according to the rules observed. If a rule is observed in the first periods and not observed in the following periods, a partition may be defined choosing the time period. This also helps defining the time horizon for the validity of the rules. For example when some rules are observed in the first month and not observed in the following month, a partition may be defined on monthly basis. First month's data can be mined so that time dependent rules may be observed. This will also mean that, these time dependent rules are valid for one month. This would help implement timely rules and not the obsolete ones. An alternative can be to sample training and test data sets from the same time period.

More analysis would be possible if all plants enter the same coding for the same problems, causes, and other comparable attributes. For this aim the codes of these attributes could be standardized in order to remove data duplications, redundancies. Consequently these can achieve sufficient information that would eventually generate interesting rules. Then, data mining including all plants' data can be possible in the future. This is good for especially the supply process. With all plants' data, vendor evaluation may also be possible.

It is also important to capture the relations of supply and manufacturing processes. If any relation between supply and production phases can be formed through a material serial number or work order ID, then, rules that give an idea on the causes of some of the production issues may possibly turn to occur much related with the peculiarities in supply.

SAP offers many functions of business intelligence and knowledge discovery. Using these tools, online data mining activities can be possible including all work

mentioned above. We expect this thesis work to be the starting point of studying the existing information in notifications and also lead to process improvement in supply and production phases. This way, the road to continuous improvement can be made smoother and faster to roll on.

REFERENCES

Agrawal, R., .Imielinski, T., and Swami, A., 1993. Mining Association Rules Between Sets of Items in Large Databases. Proceedings of ACM SIGMOD International Conference on Management of Data, 207-216

Agrawal, R., Srikant, R., 1994. Fast Algorithms for Mining Association Rules. Proceedings of VLDB, 487-499.

Bayardo, R., J., 1998. Efficiently Mining Long Patterns from Databases. Proceedings of 1998 ACM SIGMOD International Conference on Management of Data, 85-93.

Bayardo, R., J., Agrawal, R., 1999. Mining the Most Interesting Rules. Proceedings of the 5th ACM SIGMOD International Conference on Knowledge Discovery and Data Mining, 145-154.

Berrado, A., Runger, G., C., 2007. Using Metarules to Organize and Group Discovered Association Rules by Data Mining Knowledge Discovery. 14:409-431.

Borgelt, C., Kruse, R., 20022. Induction of Association Rules: Apriori Implementation. 15th Conference on Computational Statistics.

Borgelt, C., http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html, last accessed on January 2008.

Brin, S., Motwani, R. and Silverstein C., 1997. Beyond Market Baskets: Generalizing Association Rules to Correlations. Proceedings of ACM SIGMOD International Conference Management of Data.

Chen, W., C., Tseng, S., S., Wang, C., Y., 2004. A Novel Manufacturing Defect Detection Method Using Data Mining Approach. R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp.77-86.

Chien, C., F., Wang, W., C., and Cheng, J., C., 2007. Data Mining for Yield Enhancement in Semiconductor Manufacturing and an Empirical Study. Expert Systems with Applications, Oxford: Elsevier, 33(1), pp. 192-198.

Geng, L., and Hamilton, H., J., 2006. Interestingness Measures for Data Mining: A survey, ACM Computing Surveys, Vol:38, No:3 Article 9.

Han, J., Kamber, M., 2001. Data Mining: Concepts and Techniques, Simon Fraser University, Academic Press, pages 105-112.

Han, J., Pei, J., Yin, Y., 2000. Mining Frequent Patterns without Candidate Generation. Proceedings of ACM SIGMOD International Conference on Knowledge Discovery and Data Mining.

Harding, J., A., Shahbaz, M., Srinivas, Kusiak, A., 2006. Data Mining in Manufacturing: A Review. Journal of Manufacturing Science and Engineering, Vol. 128 / 969, 976.

Hou, T., Liu, W.L., Lin, L., 2003. Intelligent Remote Monitoring and Diagnosis of Manufacturing Processes Using an Integrated Approach of Neural Networks and Rough Sets. Journal of Intelligent Manufacturing 14, 239-253.

Houtsma, M., and Swami, A., 1995. Set-Oriented Mining of Association Rules. Proceedings of 11th International Conference, 25-33

Huang, H., Wu, D., 2005. Product Quality Improvement Analysis Using Data Mining: A Case Study in Ultra-Precision Manufacturing Industry. Lecture Notes in Computer Science, Fuzzy Systems and Knowledge Discovery. Springer Berlin, 577-580.

Klemettinen, M., Manila, H., Ronkainen, P., .Toivonen, H., Verkano, A. I., 1994. Finding Interesting Rules from Large Sets of Discovered Association Rules. The 3rd International Conference on Information and Knowledge Management, 401-407.

Kotsiantis, S., Kanellopoulos, D., 2006. Association Rule Mining: A Recent Overview. GESTS International Transactions on Computer Science and Engineering, Vol.32(1), 71-82.

Kusiak A., Kurasek C., 2001. Data Mining of Printed-Circuit Board Defects. IEEE Transactions on Robotics and Automation. Vol: 17(2):191–197.

Kwak, C., Yih, Y., 2004. Data Mining Approach to Production Control in the Computer-Integrated Testing Cell. IEEE Transactions on Robotics and Automation, Vol. 20, No.1, February.

Lavrac, N., Flach, P., Zupan, B., 1999. Rule Evaluation Measures: A Unifying View. Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP), 174-185.

Pawlak, Z., 1991. Rough Sets: Theoretical Aspects of Reasoning about Data. Dordrecht: Kluwer Academic Publishing.

Piatetsky-Shapiro, G., 1991. Discovery, Analysis and Presentation of Strong Rules. Knowledge Discovery in Databases.

Quality Improvement with DM (QIwDM) Project Group, articles website, http://144.122.98.184/forum/

Rokach, L., and Maimon, O., 2006. Data Mining for Improving the Quality of Manufacturing: A Feature Set Decomposition Approach. Journal of Intelligent Manufacturing, Kluwer Academic Publishers, Dordrecht, Netherlands, 17(3), pp. 285-299.

Sadoyan, H., Zakarian, A., Mohanty, P., 2006. Data Mining Algorithm for Manufacturing Process Control. The international Journal of Advanced Manufacturing Technology Volume 28 Numbers: 3-4, 342-350.

Selvamani, R., B., Khemani, D., 2005. Lecture Notes in Computer Science. Pattern Recognition and Machine Intelligence. Springer, Vol.3776, 786-791.

Shahbaz, M., Srinivas, Harding, J., A., and Turner M., 2006. Product Design and Manufacturing Process Improvement Using Association Rules. Proceedings of Instution of Mechanical Engineers, Part B: J. Engineering Manufacture, London: Professional Engineering Publishing Ltd., 220 (2), pp. 243-254.

Shiue, Y., R., Guh, R., S., 2006. The Optimization of Attribute Selection In Decision Tree Based Production Control Systems. The International Journal of Advanced Manufacturiing Technology, Vol. 28, Num. 7,8, 737-746.

SPSS Inc., 2007, Clementine 11.1 Application Guide

SPSS Inc., 2007, Clementine 11.1 Clementine Algorithms Guide

SPSS Inc., 2007, Clementine 11.1 Node Reference

Tan, P., N., .Kumar, V., Srivastava, J., 2004. Selecting The Right Objective Measure for Association Analysis. Information Systems 29, 293-313.

Tong, K., W., S., Eynard, B., Roucoules, L., Matta, N., 2007. Application of Data Mining in Manufacturing Quality Data. IEEE , 1-4244-1312-5/07.

Tseng, B., Kwon, Y., Ertekin, Y., M., 2005. Feature-Based Rule Induction in Machining Operation Using Rough Set theory for Quality Assurance.

Wang, K., Tong, S., Eynard, B., Roucoules, L., .Matta, N., 2007. Review on Application of Data Mining in Product Design and Manufacturing. Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 0-7695-2874-0/07 2007 IEEE.

Wu, C., Yue, Y., Li, M., .Adjei, O., 2004. The Rough Set Theory and Applications. Journal of Engineering Computations, Vol: 21/5, 488-511.

Wuenscher, T., Feldmann, D., G., and Krause, D., 2006. Product and Process Improvements Based on Data Mining of Shop-Floor Information. International Design Conference, 1319-1326.

Zaki, M., J., Ogihara, M., 1998. Theoretical Foundations of Association Rules. The 3rd ACM SIGMOD Workshop On Research Issues in Data Mining and Knowledge Discovery.

Zaki, M. J., 2000. Generating Non-Redundant Association Rules. Knowledge Discovery and Data Mining, pages: 34-43 (online). Available: http://citeseer.ist.psu.edu/751007.html

Zaki, M., J., Hsiao, C., J., 2002. CHARM: An Efficient Algorithm for Closed Item Set Mining. The 2nd SIAM International Conference on Data Mining.

Zaki, M., J., 2004. Mining Non-Redundant Association Rules. Journal of Data Mining and Knowledge Discovery, Vol. 9 No. 3, 223-248.

Zhai, L.Y., Khoo, L.P., Fok, S.C., (2002). Feature Extraction Using Rough Set Theory and Genetic Algorithms – An Application for the Simplification of Product Quality Evaluation. Computers & Industrial Engineering, 43, 661-676.

APPENDIX A

DATA DICTIONARY

QMNUM: Notification Number:

This field stores the key that identifies the notification.

FENUM: Notification Item Number:

If there is more than one type of problem for same material, then, for the same notification, items are created. This field defines the key that identifies the item of the notification.

QMART: Notification Type:

This field stores the key that enables notifications to be differentiated and grouped according to notification category, notification origin and other criteria. The following notification types are used for mining process:

- Notifications that are created for supply of material. (For problems encountered when the material arrives at ASELSAN Inc.).
- Notifications that are created during production.

MATNR: Material Number:

The key of the material which the problem is encountered is stored in this field. This field identifies only the type of the material. Serial number is the unique key for a specific material. For example:

Material Number: 111-222-111 corresponds to an ASSY PCB MICROCONTROL BOARD (Takım mikrokontrol kartı). All ASSY PCB MICROCONTROL Boards have the same material number. Serial number is different for each ASSY PCB MICROCONTROL BOARD.

MATKL: Material Group:

This field stores the key that is used to group together several materials.

LIFNUM: Vendor Account Number: Valid for Supply Type of Notifications (S).

This field stores the number that explicitly identifies the vendor or creditor. In general for some specific material it is preferred to supply it from the same vendor. However, there may be cases where a material is supplied from a different vendor or produced at ASELSAN.

MAWERK: Material Production Plant:

This field stores the key uniquely identifying which plant of ASELSAN is producing the material subject to the specified notification.

HERSTELLER: Manufacturer Number: Valid for Supply Type of Notifications (S).

If the material is obtained from distributors, this field stores the manufacturer of the goods.

EKORG: Purchasing Organization: Valid for Supply Type of Notifications (S).

This field denotes the purchasing organization in ASELSAN.

BKGRP: Purchasing Group: Valid for Supply Type of Notifications (S).

This field stores the key for a buyer or a group of buyers, who is/are responsible for certain purchasing activities.

FERTAUFNR: Production Order Number: Valid for Production Type of Notifications (P).

This field stores the key that clearly identifies a process order or production order.

FERTAUFPL: Production Order Plan Number: Valid for Production Type of Notifications (P).

This field is used with PNLKN field to find the step of the routing where the problem is encountered.

PNLKN: Plan Step Number: Valid for Production Type of Notifications (P).

This field is used with FERTAUFPL to find the step of the routing where the problem is encountered.

NARBPL: Workcenter: Valid for Production Type of Notifications (P).

This field denotes the workcenter where the problem is observed.

DISPO: MIP Responsible: Valid for Production Type of Notifications (P).

This field corresponds to the MIP responsible of the project, thus, corresponds to the project in our case.

UMATN: Assembly Number: Valid for Production Type of Notifications (P).

This field defines the material number of the assembly.

T_ARBPL: Workcenter: Valid for Supply Type of Notifications (S).

This field gives the workcenter for supply type of notifications.

OTKAT-OTGRP-OTEIL: Location:

These three fields together, give the key for the place where the problem is realized.

FEKAT-FEGRP-FECOD: Problem Code.

FEKAT stores the catalog number, FEGRP stores the group number and FECOD stores the problem code. These three fields together is the key identifying the encountered problem.

URKAT-URGRP-URCOD: Cause Code.

URKAT stores the catalog number, URGRP stores the group number and URCOD stores the cause code. These three fields together give the key for the cause of the damage determined.

MGEIG: Internal Nonconforming quantity.

This field stores the share of the nonconforming quantity that is identified and acknowledged as being caused internally. The entries in this field, together with the nonconforming quantity (external) must not exceed the complaint quantity.

MGFRD: External Nonconforming quantity:

This field stores the share of the quantity reported as nonconforming which is not acknowledged as being caused internally. The values in this field together with the nonconforming quantity (internal) must not exceed the complaint quantity.

MGEIN: Unit of Measure:

This field stores the unit of measure, in which the defective stocks of the material are dealt with in the quality notification.

BZMNG: Reference Quantity.

This field stores the quantity, to which the notification refers. If the notification deals with an internal problem, the reference quantity is the same as the production quantity. In the "Complaint quantity" field, you define how much of the production quantity is subject to complaint.

RKMNG: Complaint Quantity:

This field specifies the complaint quantity in the notification. This has no bearing on the returned quantity. In other words, you can specify a complaint quantity that is greater or less than the quantity you actually return. The total nonconforming quantity (internal + external) must not exceed the complaint quantity.
Nonconforming quantities do not have to exist, however, to be able to specify a complaint quantity.

- Although only part of the delivery is nonconforming, you want to return the entire delivery.

- If you require the goods urgently, you can return less than the complaint quantity or nothing at all (even though you have complained about a part of the delivery).

DENMG: Inspected quantity.

This field stores the quantity inspected from reference quantity due to sampling.

RGMNG: Return Quantity:

This field specifies the quantity returned.

There is no relation to the complaint quantity. In other words, you can return a quantity greater or less than the complaint quantity. For example, the following situations are possible:

- Although only part of the delivery is nonconforming, you want to return the entire delivery.

- If you urgently require the goods, you may want to return less than the complaint quantity or nothing at all, even though you issued a complaint about a part of the delivery.

KABMG: Accepted Quantity.

This field stores the quantity accepted of the reference quantity. This quantity of the material is used.

HURMG: Scrap Quantity.

This field stores the quantity separated as scrap.

OGKMG: Quantity used as it is.

This field stores the quantity when an acceptable nonconformity is observed during inspection. Then the material is used as it is. This type is generally occurs when there is a mismatch in its document etc.

YİSMG: Quantity reworked.

This field stores the quantity of the material that is reworked and applicable in production.

ONRMG: Quantity repaired.

This field specifies the quantity repaired and applicable in supply.

ANZFEHLER: Number of Nonconforming Units.

This field indicates the number of nonconformities observed.

APPENDIX B

PROBLEMS ABOUT DATA

Table B.1. Problems About Data

APPENDIX C

DIFFERENT DISCRETIZATION SCHEMES STUDIED

Table C.1 Discretization with 3 equal widths.

	Disc. Int. Nonconf. Qty. Ratio	Disc. Ext. Nonconf. Qty. Ratio	Disc. Scrap Qty Ratio	Disc. Accepted Qty Ratio	Disc. Used as is Qty Ratio	Disc. Repaired Qty Ratio	Disc. Rework Qty Ratio	Disc. Return Qty Ratio
missing	0.14%	0.14%	0.14%	0.14%	0.14%	0.14%	0.14%	0.14%
[0%, 33%]	82.26%	77.12%	99.17%	87.78%	99.30%	99.44%	71.48%	98.04%
(33%, 66%]	3.16%	4.65%	0.12%	2.40%	0.08%	0.02%	5.47%	0.33%
(66%, 100%]	14.45%	18.09%	0.58%	9.68%	0.48%	0.41%	22.92%	1.49%

Table C.2 Discretization with 5 equal widths.

	Disc. Int. Nonconf. Qty. Ratio	Disc. Ext. Nonconf. Qty. Ratio	Disc. Scrap Qty Ratio	Disc. Accepted Qty Ratio	Disc. Used as is Qty Ratio	Disc. Repaired Qty Ratio	Disc. Rework Qty Ratio	Disc. Return Qty Ratio
missing	0.14%	0.14%	0.14%	0.14%	0.14%	0.14%	0.14%	0.14%
[0%, 20%]	80.65%	73.96%	98.91%	87.34%	99.28%	99.44%	67.68%	97.79%
(20%, 40%]	2.48%	4.48%	0.27%	0.70%	0.02%	0.00%	5.51%	0.37%
(40%, 60%]	2.11%	2.93%	0.06%	1.20%	0.06%	0.02%	3.37%	0.19%
(60%, 80%]	0.62%	0.81%	0.06%	2.52%	0.02%	0.00%	0.99%	0.02%
(80%, 100%]	14.00%	17.68%	0.56%	8.11%	0.48%	0.41%	22.32%	1.49%

Table C.3 Discretization with 3 different widths.

	Disc. Int. Nonconf. Qty. Ratio	Disc. Ext. Nonconf. Qty. Ratio	Disc. Scrap Qty Ratio	Disc. Accepted Qty Ratio	Disc. Used as is Qty Ratio	Disc. Repaired Qty Ratio	Disc. Rework Qty Ratio	Disc. Return Qty Ratio
missing	0.14%	0.14%	0.14%	0.14%	0.14%	0.14%	0.14%	0.14%
[0%, 15%]	79.14%	69.36%	98.82%	87.03%	99.28%	99.34%	64.98%	97.36%
(15%, 85%]	6.90%	13.19%	0.48%	6.30%	0.12%	0.12%	13.01%	1.03%
(85%, 100%]	13.83%	17.32%	0.56%	6.53%	0.47%	0.41%	21.87%	1.47%

APPENDIX D

VALIDITY OF THE RULES

Table D.1. Validity of the rules by rule support for production data.

Rule Support Levels %	Number of Rules in Training	Rule Support Levels %	Number of Rules in Test
[10, 100]	2	[10, 100]	1
		[5, 10]	1
[5, 10)	16	[10, 100]	5
		[5, 10]	6
		[2, 5)	4
		[0, 2)	1
[2, 5)	305	[10, 100]	2
		[5, 10]	62
		[2, 5)	99
		[0, 2)	139
		0	3
[0, 2)	2309	[5, 10]	50
		[2, 5)	492
		[0, 2)	1552
		0	215

Table D.2. Validity of the rules by rule support for support data.

Rule Support Levels %	Number of Rules in Training	Rule Support Levels %	Number of Rules in Test
[10, 100]	738	[10, 100]	177
		[5, 10)	5
		0	556
[5, 10)	899	[10, 100]	20
		[5, 10)	154
		[2, 5)	29
		0	650
[2, 5)	1588	[10, 100]	5
		[5, 10)	15
		[2, 5)	346
		[1, 2)	117
		(0,1)	72
		0	1033
[1, 2)	787	[2, 5)	48
		[1, 2)	117
		(0,1)	129
		0	493

Table D.3. Validity of the rules by antecedent support for production data.

Antecedent Support Levels %	Number of Rules in Training	Antecedent Support Levels %	Number of Rules in Test
[10, 100]	7	[10, 100]	5
		[5, 10]	2
[5, 10)	105	[10, 100]	27
		[5, 10]	66
		[2, 5)	11
		[0, 2)	1
[2, 5)	764	[10, 100]	22
		[5, 10]	192
		[2, 5)	401
		[0, 2)	142
		0	7
[0, 2)	1756	[5, 10]	55
		[2, 5)	649
		[0, 2)	944
		0	108

Table D.4. Validity of the rules by antecedent support for supply data.

Antecedent Support %	Number of Rules in Training	Antecedent Support %	Number of Rules in Test
[10, 100]	979	[10, 100]	466
		[5, 10)	59
		0	454
[5, 10)	1081	[10, 100]	42
		[5, 10)	380
		[2, 5)	151
		[1, 2)	6
		0	502
[2, 5)	1952	[10, 100]	7
		[5, 10)	56
		[2, 5)	552
		[1, 2)	339
		(0,1)	177
		0	821

Table D.5 Validation of the rules by added value for production data.

Added Value Levels %	Number of Rules in Training	Added Value Levels %	Number of Rules in Test
[90, 100]	4	[80, 90)	1
		[70, 80)	1
		[60, 70)	1
		[40, 50)	1
[80, 90)	85	[90, 100]	6
		[80, 90)	27
		[70, 80)	12
		[60, 70)	9
		[50, 60)	5
		<50	12
		No Support	14
[70, 80)	260	[90, 100]	3
		[80, 90)	38
		[70, 80)	49
		[60, 70)	29
		[50, 60)	21
		[40, 50)	20
		[0, 40)	53
		No Support	42
		(.., 0)	5
[60, 70)	384	[90, 100]	9
		[80, 90)	9
		[70, 80)	37
		[60, 70)	65
		[50, 60)	47
		[40, 50)	81
		[0, 40)	111
		No Support	13
		(.., 0)	12

Table D.5. (Continued) Validation of the rules by added value for production data.

Added Value Levels %	Number of Rules in Training	Added Value Levels %	Number of Rules in Test
[50, 60)	554	[90, 100]	5
		[80, 90)	12
		[70, 80)	22
		[60, 70)	74
		[50, 60)	73
		[40, 50)	120
		[0, 40)	203
		No Support	16
		(.., 0)	29
[40, 50)	820	[90, 100]	1
		[80, 90)	18
		[70, 80)	31
		[60, 70)	50
		[50, 60)	90
		[40, 50)	140
		[0, 40)	355
		No Support	25
		(.., 0)	110
[0, 40)	525	[80, 90)	2
		[70, 80)	25
		[60, 70)	24
		[50, 60)	41
		[40, 50)	77
		[0, 40)	303
		No Support	5
		(.., 0)	48

Table D.6. Validation of the rules by added value for supply data.

Added Value Levels %	Number of Rules in Training	Added Value Levels %	Number of Rules in Test
[90, 100]	5	[90, 100]	3
		[80, 90)	1
		[0, 40)	1
[80, 90)	102	[90, 100]	13
		[80, 90)	29
		[70, 80)	5
		[0, 40)	3
		No Support	52
[70, 80)	64	[90, 100]	14
		[80, 90)	9
		[70, 80)	1
		[50, 60)	1
		[40, 50)	1
		[0, 40)	14
		No Support	24
[60, 70)	150	[90, 100]	6
		[80, 90)	14
		[70, 80)	6
		[60, 70)	51
		[50, 60)	2
		[40, 50)	3
		[0, 40)	15
		No Support	44
		(.., 0)	9

Table D.7. (Continued) Validation of the rules by added value for supply data.

Added Value Levels %	Number of Rules in Training	Added Value Levels %	Number of Rules in Test
[50, 60)	241	[90, 100]	12
		[80, 90)	15
		[70, 80)	19
		[60, 70)	3
		[50, 60)	7
		[40, 50)	15
		[0, 40)	42
		No Support	81
		(.., 0)	47
[40, 50)	286	[90, 100]	4
		[80, 90)	5
		[70, 80)	4
		[60, 70)	18
		[50, 60)	14
		[40, 50)	28
		[0, 40)	63
		No Support	121
		(.., 0)	29